《计算机图形学》

实验指导书

目 录

目 录 1
概 述 2
实验 1 熟悉实验环境 3
1.1 实验要求和目的 3
1.2 实验课时 3
实验 2 直线的生成 4
2.1 实验要求和目的 4
2.2 实验课时 4
2.3 实验环境 4
2.4 实验平台简介： 4
2.5 思考题（选做） 5
实验 3 多边形扫描转换算法 7
3.1 实验要求和目的 7
3.2 实验课时 7
3.3 实验环境 7
3.4 实验平台简介： 7
实验 4 BSpline 曲线绘制 10
4.1 实验要求和目的 10
4.2 实验课时 10
4.3 实验环境 10
4.4 实验平台介绍 10
实验 5 光照模型(Illumination Model) 13
5.1 实验要求和目的 13
5.2 思考题 13
5.3 实验课时 13
5.4 实验环境 13
5.5 实验平台介绍 13
附录 A：实验报告 17

概 述

(1) 实验概述

运用某种程序设计语言设计并实现计算机图形学的直线、曲线、简单多面体（四面体）等基本图形元素的表示和绘制，以检验和巩固计算机图形学中的基本知识、加深对本课程原理、方法和技术的理解，锻炼和培养学生实际操作技能和解决实际问题的能力，使学生熟悉解决实际问题的过程。

(2) 实验目的和要求

实验目的是检验和巩固所学知识与方法，通过实现基本图形元的表示和绘制过程，理解并掌握计算机图形学的原理、方法和技术，并灵活运用它们解决实际问题。

要求理解各实验相关的原理和实现方法，通过这些实验的训练，加深对课程中原理、方法和技术的理解，验证和巩固计算机图形学中的基本知识，锻炼和培养学生熟悉图形编程环境，理解课程中基本问题的求解算法和性能改进方法，并对结果进行充分测试。

(3) 主要原理与概念

一般来说，计算机图形学的基本内容包括图形的表示（如三维形体的表示，曲线、曲面的表示等）、图形变换和观察、图形生成（基本图形生成，消隐、真实感绘制等）三个方面，涉及大量数据结构、算法。本试验大纲主要涉及到多面体表示法、直线扫描生成、多边形填充、平行投影变换、消隐算法等概念和算法。

(4) 实验环境

Visual C++ 6.0

(5) 实验内容

- 实验一：实验环境介绍
- 实验二：直线生成算法
- 实验三：多边形扫描转换
- 实验四：自由曲线绘制
- 实验五：光照模型(Illumination)

实验1　熟悉实验环境

1.1 实验要求和目的

熟悉实验环境：熟悉 Visual C++的图形编程环境；掌握 Visual C++中 GDI（图形设备接口）、画笔、画刷等基本概念；掌握基本绘图函数；

1.2 实验课时

1 学时

实验2　直线的生成

2.1 实验要求和目的

理解直线生成的原理；掌握典型直线生成算法；掌握步处理、分析实验数据的能力；

编程实现 DDA 算法、Bresenham 中点算法；对于给定起点和终点的直线，分别调用 DDA 算法和 Bresenham 中点算法进行批量绘制，并记录两种算法的绘制时间；利用 excel 等数据分析软件，将试验结果编制成表格，并绘制折线图比较两种算法的性能。

2.2 实验课时

3 学时

2.3 实验环境

- 开发环境：Visual C++ 6.0
- 实验平台：Experiment_Frame_One（自制平台）

2.4 实验平台简介：

本实验提供名为 Experiment_Frame_One 的平台，该平台提供基本绘制、设置、输入功能,学生在此基础上实现 DDA 算法和 Mid_Bresenham 算法，并进行分析。

- 平台界面：如图 2-1 所示
- 设　　置：通过 view->setting 菜单进入，如图 2-2 所示
- 输　　入：通过 view->input…菜单进入.如图 2-3 所示
- 实现算法：
  - DDA 算法：void CExperiment_Frame_OneView::DDA(int X0, int Y0, int X1, int Y1)
  - Mid_Bresenham 算法：void

CExperiment_Frame_OneView::Mid_Bresenham(int X0, int Y0, int X1, int Y1)

2.5 思考题（选做）

如何测试比较算法的性能？

- 提示 1：因为绘制 1 条直线时间很短，所以需要绘制大量直线才能比较它们之间的性能；
- 提示 2: drawpixel 需要耗费时间，但它的时间性能和直线绘制算法无关，因此在比较不同算法性能时，应该屏蔽它的影响，如何屏蔽？

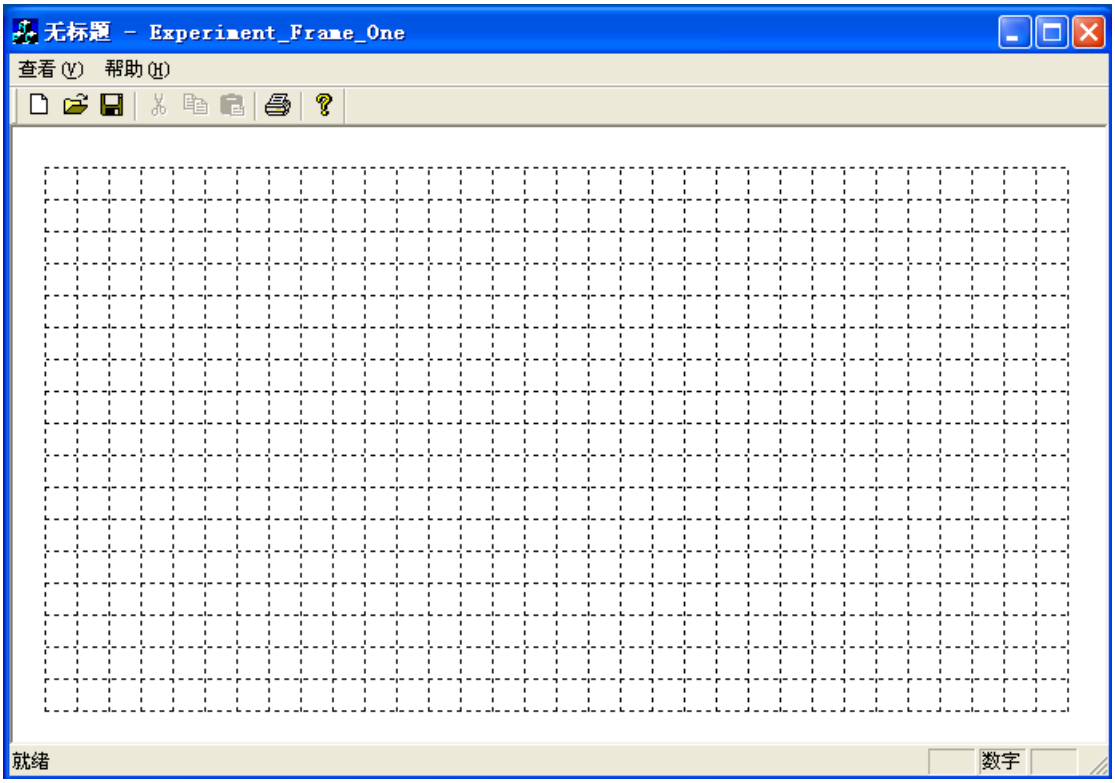

图 2-1 总界面

图 2-2 设置界面

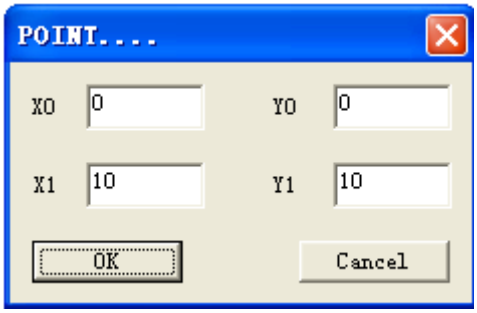

图 2-3 输入界面

实验3　多边形扫描转换算法

3.1 实验要求和目的

理解多边形扫描转换的原理；掌握典型多边形扫描转换算法；掌握步处理、分析实验数据的能力；

- 编程实现基本 X-扫描线转换算法（必做）；
- 编程实现有效边表转换算法（选做）

3.2 实验课时

4 学时

3.3 实验环境

本试验提供自带实验平台

- 开发环境：Visual C++ 6.0
- 实验平台：Polygon_Conversion（自制平台）

3.4 实验平台简介：

本实验提供名为 Polygon_Conversion 的平台，该平台提供基本绘制、设置、输入功能，学生在此基础上实现 X-扫描线算法和有效边表转换算法。

- 平台界面：如图 3-1 所示
- 多边形输入，界面如图 3-2 所示：
  - 用户按【功能】→【输入多边形……】菜单开始输入多边形；
  - 单击鼠标左键输入多边形顶点；
  - 点击鼠标右键结束多边形输入，并将最后一个顶点和第一个顶点进行连接；
- 参数设置：界面如图 3-3 所示

- 用户按“【功能】→【设置……】”启动设置对话框
- 设置内容：
  - 填充色
  - 是否填充多边形
  - 选择转换算法
- 实现扫描转换算法
  - X-扫描线转换算法：

    ```
    void  CPolygon_ConversionView::X_Scan_Line_Conersion
    (int Vertices[][2], int VertexNum)
    ```

  - 有效边表转换算法：

    ```
    void  CPolygon_ConversionView::Active_Edge_Table_Conersion
    (int Vertices[][2], int VertexNum)
    ```

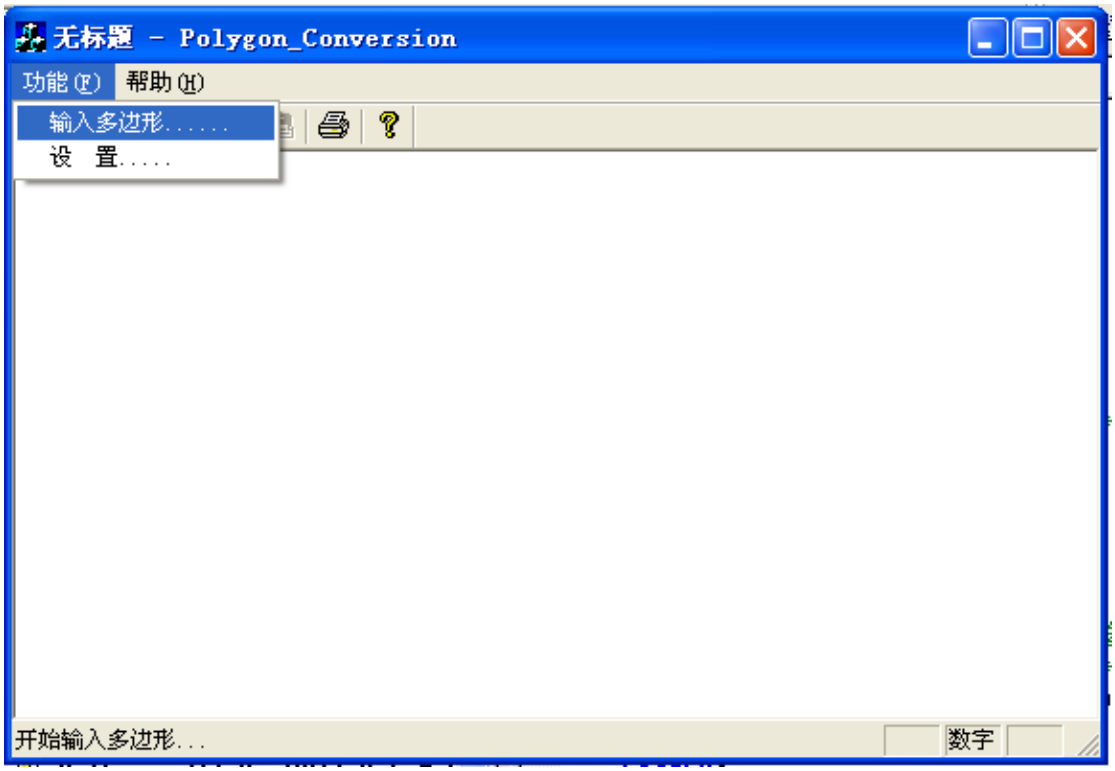

图 3-1 实验平台界面

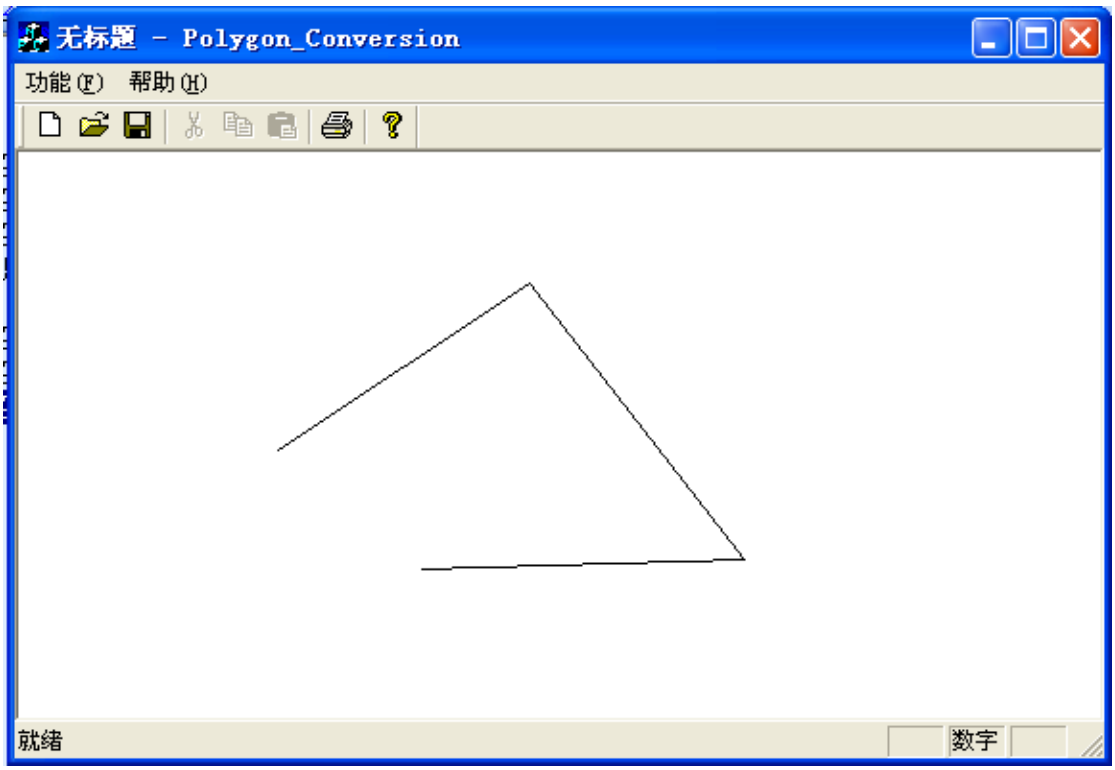

图 3-2 输入多边型

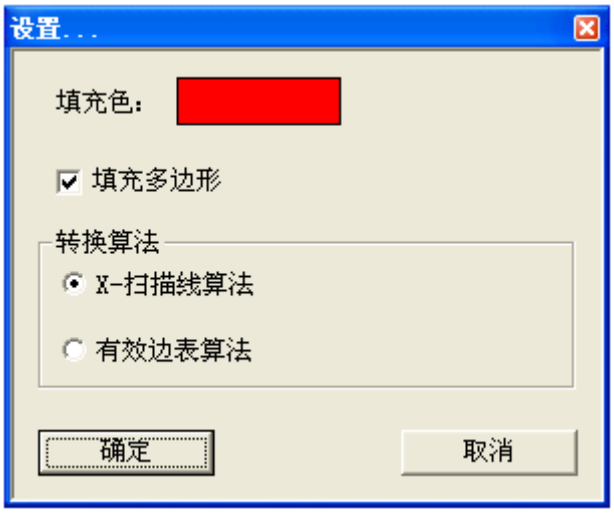

图 3-3 设置对话框

实验4　BSpline 曲线绘制

4.1 实验要求和目的

理解掌握自由曲线生成的基本原理和方法；编程实现三次 B 样条曲线：

- 均匀周期性 B 样条曲线
- 开放均匀 B 样条曲线

4.2 实验课时

4 学时

4.3 实验环境

本试验提供自带实验平台

- 开发环境：Visual C++ 6.0
- 实验平台：Free_Curve（自制平台）

4.4 实验平台介绍

本实验提供名为 Free_Curve 的平台，该平台提供基本绘制、设置、输入功能，学生在此基础上实现：
[1] 编码实现 BSpline 曲线基函数
[2] 编码实现不同参数条件下的节点矢量的生成

- 平台界面：如图 4-1 所示
- 多边形输入，界面如图 4-1 所示：
  - 用户按【功能】→【输入……】菜单开始输入控制多边形；
  - 单击鼠标左键输入多边形顶点；
  - 点击鼠标右键结束控制多边形输入
- 参数设置：界面如图 4-2 所示

- 用户按“【功能】→【设置……】”启动设置对话框
- 设置内容：
  - 控制点设置(包括颜色、是否显示、控制点大小)
  - 控制多边形设置(包括控制多边形的颜色，是否显示)
  - BSpline 曲线设置
- 实现下列函数
  - 实现 BSpline 曲线的基函数；

    float BKM(float t, int k, int m, float nodes[])

    参数含义参考代码注解；
  - 节点矢量的计算：

```
bool Create_Nodes_Vector(int      n,
                         int      m,
                         int      SplineType,
                         float    nodes[])
```

    参数含义参考代码注解；

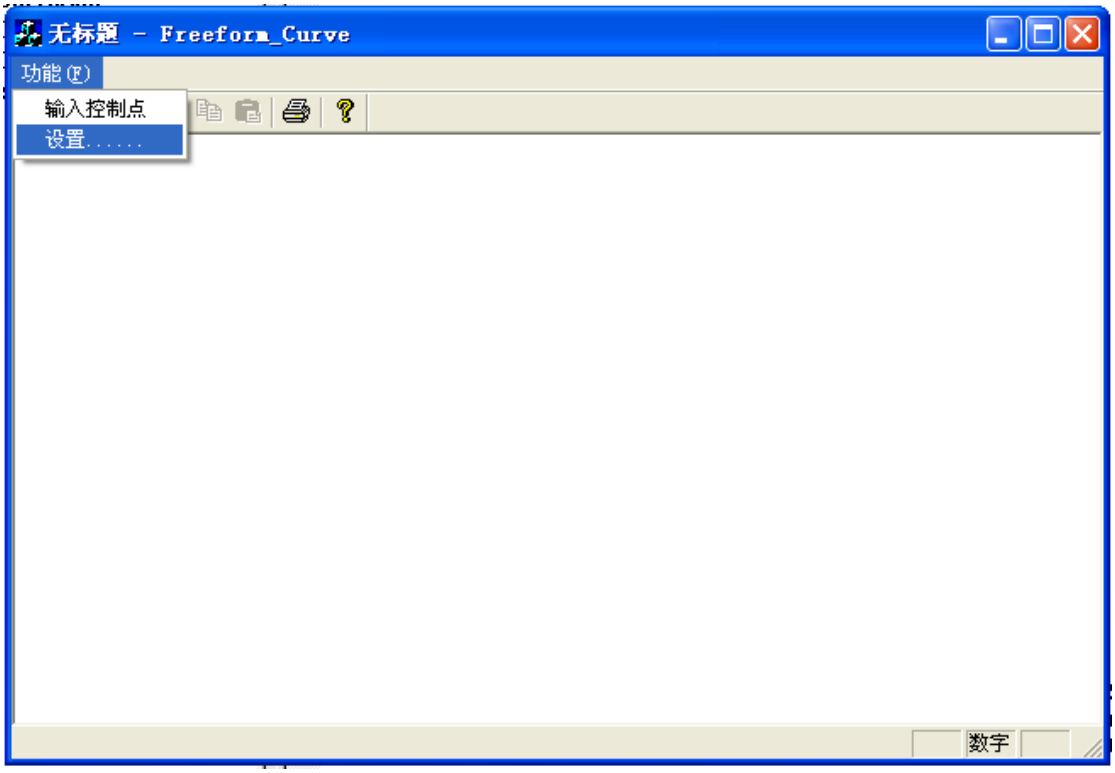

图 4-1 平台界面

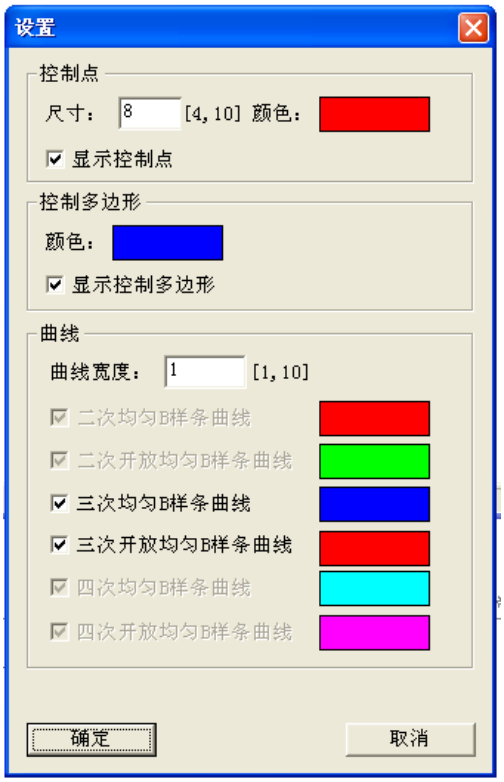

图 4-2 设置界面

实验5　光照模型(Illumination Model)

5.1 实验要求和目的

理解和掌握简单光照模型的基本原理和方法；并编程实现两种常用的明暗处理方法：

- Gouraud 明暗处理方法
- Phong 明暗处理方法

说明：本平台目前仅考虑环境光(Ambient Light)、漫反射光(Diffuse light)，暂不考虑镜面反射光(Specular Light)，而且不考虑光强衰减；

5.2 思考题

- 如何实现光强衰减效果
- 如何实现镜面反射光效果

5.3 实验课时

4 学时

5.4 实验环境

本试验提供自带实验平台

- 开发环境：Visual C++ 6.0
- 实验平台：Illumination（自制平台）

5.5 实验平台介绍

本实验提供名为 Illuminatin 的平台，该平台提供以下功能：

[1] 提供半球面的四边形网格生成功能；
[2] 网格线绘制功能
[3] 恒定光强的多边形绘制功能
[4] 设置功能（见后面的描述）

[5] 提供完整的多边形绘制框架；
[6] 提供二次线性插值运算功能

在此基础上，学生编程实现：
[1] 编码实现 Gouraud 明暗处理方法
[2] 编码实现 Phong 明暗处理方法

- 平台界面：如图 5-1 所示
- 多边形输入，界面如图 5-1 所示：
  - 用户按【功能】→【输入……】菜单开始输入控制多边形；
  - 单击鼠标左键输入多边形顶点；
  - 点击鼠标右键结束控制多边形输入
- 参数设置：界面如图 5-2 所示
  - 用户按“【功能】→【设置……】”启动设置对话框
  - 设置内容：
    - 几何模型参数
    - 网格线设置
    - 环境光设置
    - 漫反射设置
    - 显示控制
  - 实现下列函数
    - 实现 Gouraud 明暗处理方法；

```
double CIlluminationView::Gouraud
      (int      Pt[3],
       int      Vertices[4][3],
       double Normals[4][3],
       int      lightPos[3],
       double Ip,
```

double Kd)

参数含义参考代码注解；

◆ 实现 Phong 明暗处理方法：

```
bool Create_Nodes_Vector(int      n,
                         int      m,
                         int      SplineType,
                         float    nodes[])
```

参数含义参考代码注解；

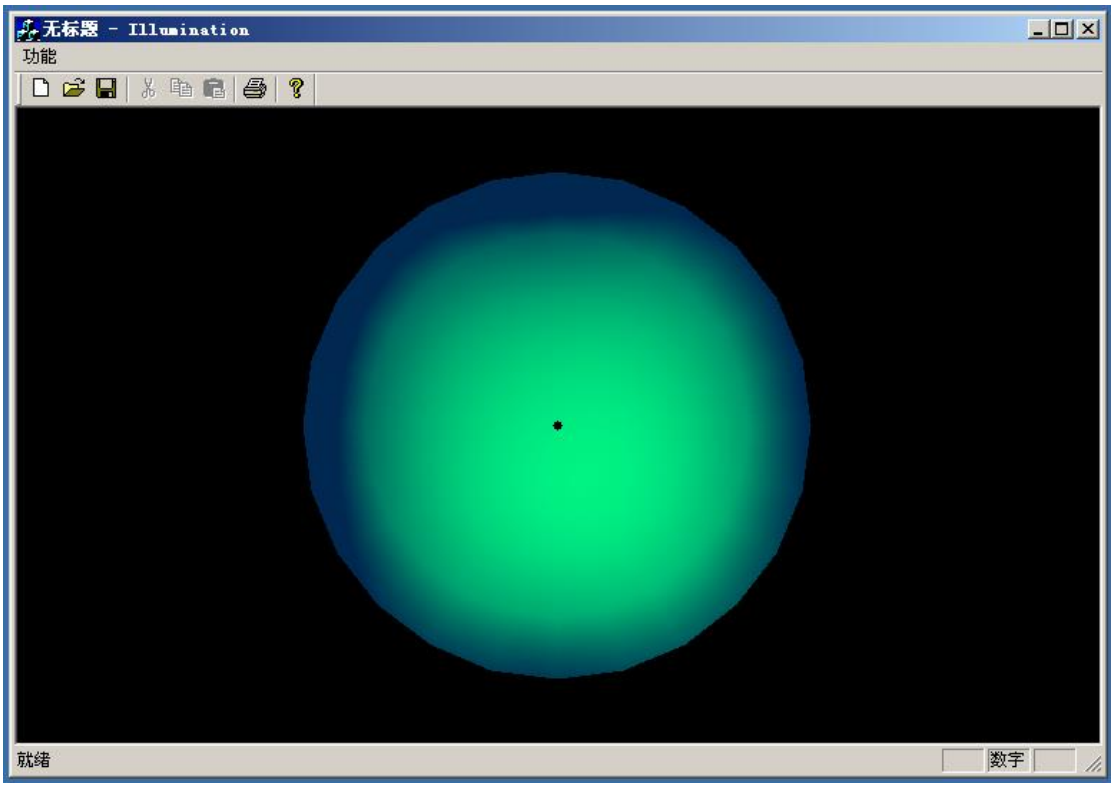

图 5-1 总体界面

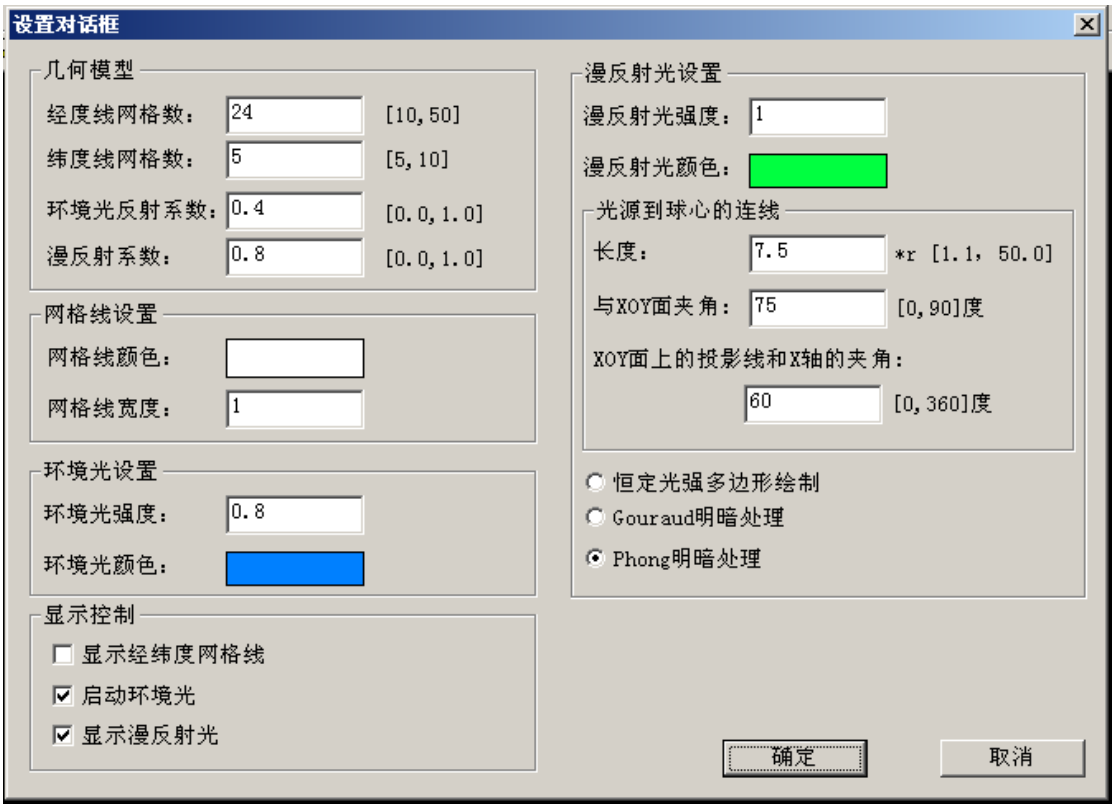
设置对话框
几何模型
经度线网格数：
24
[10,50]
纬度线网格数：
5
[5,10]
环境光反射系数：
0.4
[0.0,1.0]
漫反射系数：
0.8
[0.0,1.0]
网格线设置
网格线颜色：
网格线宽度：
1
环境光设置
环境光强度：
0.8
环境光颜色：
显示控制
显示经纬度网格线
启动环境光
显示漫反射光
漫反射光设置
漫反射光强度：
1
漫反射光颜色：
光源到球心的连线
长度：
7.5
*r [1.1, 50.0]
与XOY面夹角：
75
[0,90]度
XOY面上的投影线和X轴的夹角：
60
[0,360]度
恒定光强多边形绘制
Gouraud明暗处理
Phong明暗处理
确定
取消

图 5-2 设置对话框

附录 A：实验报告

实验 X：实验名称

姓　　名：_____

学　　号：_____

班　　级：_____

实验地点：_____

实验时间：_____

1 实验目的和要求

2 实验环境和工具

3 实验结果

3.1 程序流程图

3.2 程序代码

3.3 运行结果

3.4 运行结果分析

4 思考题（可选）

5 实验心得