中山大學
SUN YAT-SEN UNIVERSITY

计算机图形学

作业六：光照效果

数据科学与计算机学院　17大数据与人工智能

17341015　陈鸿峥

一、 实验原理

二、 实验结果

双击teapot.exe即可运行，实验结果如图1所示。

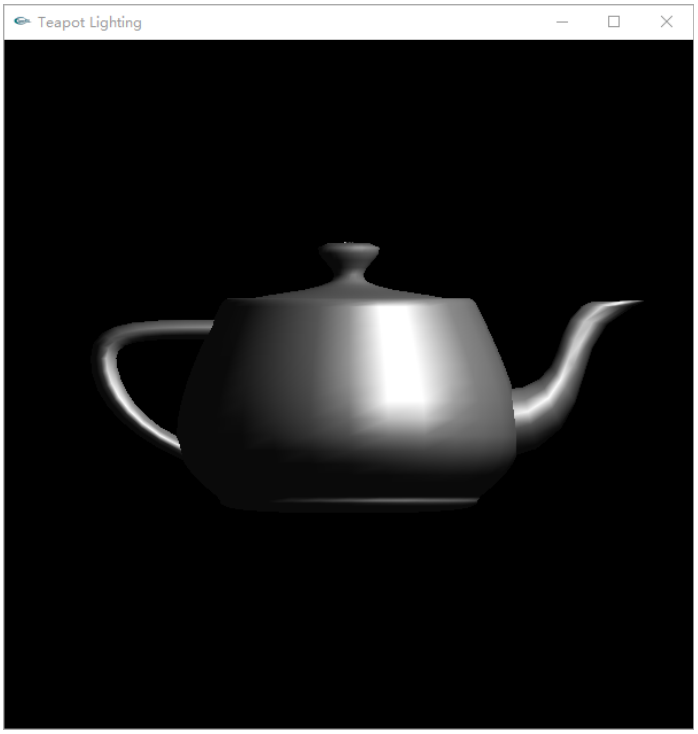

图 1: Teapot光照结果

附录 A. 源代码

```
#include <windows.h> // must be the first one to be included!
#include <stdlib.h>
#include <GL/glut.h>

GLuint teapotList;

void init(void)
{
    GLfloat ambient[] = {0.0, 0.0, 0.0, 1.0};
    GLfloat diffuse[] = {1.0, 1.0, 1.0, 1.0};
    GLfloat specular[] = {1.0, 1.0, 1.0, 1.0};
    GLfloat position[] = {4.5, 4.5, 3, 1.0}; // fix position by model view matrix

    GLfloat lmodel_ambient[] = {0.2, 0.2, 0.2, 1.0};
    GLfloat local_view[] = {0.0};

    // initialize lighting model
    glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, specular);
    glLightfv(GL_LIGHT0, GL_POSITION, position);
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, lmodel_ambient);
    glLightModelfv(GL_LIGHT_MODEL_LOCAL_VIEWER, local_view);

    glFrontFace(GL_CW);
    glEnable(GL_LIGHTING); // global
    glEnable(GL_LIGHT0); // each lighting
    glEnable(GL_AUTO_NORMAL);
    glEnable(GL_NORMALIZE);
    glEnable(GL_DEPTH_TEST); // depth buffer
    teapotList = glGenLists(1); // make teapot display list
    glNewList(teapotList, GL_COMPILE);
    glutSolidTeapot(1.0);
    glEndList();
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    GLfloat mat[4];
    glPushMatrix();
    glTranslatef(2.0, 2.0, 0.0); // x, y, z

    /*
     * material properties
     * constants reference from
     * https://www.opengl.org/archives/resources/code/samples/redbook/teapots.c
     */
    mat[0] = 0.19225; mat[1] = 0.19225; mat[2] = 0.19225; mat[3] = 1.0; // rgb
    glMaterialfv(GL_FRONT, GL_AMBIENT, mat);
    mat[0] = 0.50754; mat[1] = 0.50754; mat[2] = 0.50754;
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat);
    // mat[0] = 0.508273; mat[1] = 0.508273; mat[2] = 0.508273;
    mat[0] = 1; mat[1] = 1; mat[2] = 1; // reflect white lights
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat);
    glMaterialf(GL_FRONT, GL_SHININESS, 0.2 * 128.0); // shine
    glCallList(teapotList);

    glPopMatrix();
    glFlush();
}

void reshape(int w, int h)
{
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    // void glOrtho(GLdouble left, GLdouble right,
    //    GLdouble bottom, GLdouble top,
    //    GLdouble nearVal, GLdouble farVal);
    if (w <= h)
        glOrtho(0.0, 4.0, 0.0, 4.0*(GLfloat)h/(GLfloat)w, -10.0, 10.0);
    else
        glOrtho(0.0, 4.0*(GLfloat)w/(GLfloat)h, 0.0, 4.0, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(600,600);
    glutInitWindowPosition(50,50);
    glutCreateWindow("Teapot Lighting");
    init();
    glutReshapeFunc(reshape);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

// gcc teapot.c -lglu32 -lglut32 -lopengl32 -o teapot.exe
```

编译指令如下：

```
gcc -I.\include -L.\lib teapot.c -lglu32 -lglut32 -lopengl32 -o teapot.exe
```