实验十五 消隐实验

时间：2022年6月15日

地点：信息学院机房2202

1、实验内容

完成消隐实验，采用Z-buffer算法完成消隐

2、实验目的

熟悉Z-buffer 、画家算法等相关内容

3、实验代码

#define \_STDCALL\_SUPPORTED

#include <GL/glut.h>

#include <math.h>

#include <stdlib.h>

#include <windows.h>

#include <stdio.h>

#include <fstream>

#include <iostream>

using namespace std;

GLfloat pos[]={-2,4,5,1},

amb[]={0.3,0.3,0.3,1.0};

GLfloat front\_amb\_diff[]={0.7,0.5,1.1,1.0};

GLfloat back\_amb\_diff[]={0.4,0.7,1,1.0};

GLfloat spe[]={0.25,0.25,0.25,1.0};

GLfloat theta=0,dt=1.5,axes[3][3]={{1,0,0},{0,1,0},{0,0,1}};

int axis=0;

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glPushMatrix();

if(axis<3)

glRotated(theta,axes[axis][0],axes[axis][1],axes[axis][2]);

else

{

glPushMatrix();

glRotated(theta,0,0,1);

glLightfv(GL\_LIGHT0,GL\_POSITION,pos);

glPopMatrix();

}

glutSolidTorus(.4,1,48,96); // 图片调用函数可以设置图片的属性

glPopMatrix(); // glutWireTeapot(); 茶壶

glutSwapBuffers(); // glutSolidCone(); 圆锥

}

void idle(void)

{

if(theta>=360)

axis=(axis+1)%4;

theta=(theta<360)?theta+dt:dt;

glutPostRedisplay();

}

int main(int argc,char\*\* argv){

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB|GLUT\_DEPTH);

glutInitWindowSize(500,500);

glutInitWindowPosition(200,100);

glutCreateWindow("GLUT Object");

glClearColor(1,1,0,0);//背景颜色黄色

glEnable(GL\_DEPTH\_TEST);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45,1.0,2,8);

glMaterialfv(GL\_FRONT,GL\_AMBIENT\_AND\_DIFFUSE,front\_amb\_diff);

glMaterialfv(GL\_BACK,GL\_AMBIENT\_AND\_DIFFUSE,back\_amb\_diff);

glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_SPECULAR,spe);

glMaterialf(GL\_FRONT\_AND\_BACK,GL\_SHININESS,75);

glLightfv(GL\_LIGHT0,GL\_AMBIENT,amb);

glLightModeli(GL\_LIGHT\_MODEL\_TWO\_SIDE,GL\_TRUE);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glTranslated(0,0,-5);

glLightfv(GL\_LIGHT0,GL\_POSITION,pos);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glutDisplayFunc(display);

glutIdleFunc(idle);

glutMainLoop();

}4、实验结果

图片包含 形状

描述已自动生成

5、实验总结

算法思想是先将Z缓冲器中各单元的初始值置为最小值，当要改变某个象素的颜色值时，首先检查当前多边形的深度值是否大于该象素原来的深度值（保存在该象素所对应的Z缓冲器的单元中），如果大于原来的Z值，说明当前多边形更靠近观察点，用它的颜色替换原象素的颜色。假设xoy面为投影面，z轴为观察方向，过屏幕上任意象素点(x,y)做平行于z轴的射线R，与物体表面相交于p1,p2点，p1和p2点的z值称为该点的深度值，z-buffer算法比较p1和p2的z值，将最大的z值存入z缓冲器中，显然屏幕上(x,y)显示p1点的颜色。