实验七 直线反走样实验

时间：2022年4月27日

地点：信息学院机房2202

1、实验内容

使用opengl，完成教材P458中的鼠标函数

使用opengl，实现任一反走样技术。

2、实验目的

熟悉鼠标函数，调用函数完成直线反走样实验。

3、实验代码

|  |
| --- |
| #include<windows.h>  #include <GL/glut.h>  void InitEnvironment()  {  glClearColor(0.6, 0.6, 0.6, 0);  glClear(GL\_COLOR\_BUFFER\_BIT);  glPointSize(6);  glMatrixMode(GL\_MODELVIEW);  glLoadIdentity();  gluOrtho2D(0, 400, 0, 400);  }  void myDisplay(void)  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glFlush();  }  void OnMouse(int button, int state, int x, int y)  {  if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)  {  glColor3f(1, 0, 0);  glBegin(GL\_POINTS);  glVertex2f(x, 400 - y);  glEnd();  glFlush();  }  if (button == GLUT\_MIDDLE\_BUTTON && state == GLUT\_DOWN)  {  glColor3f(0, 1, 0);  glBegin(GL\_POINTS);  glVertex2f(x, 400 - y);  glEnd();  glFlush();  }  if (button == GLUT\_RIGHT\_BUTTON && state == GLUT\_DOWN)  {  glColor3f(0, 0, 1);  glBegin(GL\_POINTS);  glVertex2f(x, 400 - y);  glEnd();  glFlush();  }  }  int main(int argc, char\* argv[])  {  glutInit(&argc, argv); //初始化GLUT  glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);  glutInitWindowPosition(500, 200);  glutInitWindowSize(400, 400);  glutCreateWindow("OpenGL");  InitEnvironment(); //初始化  glutMouseFunc(&OnMouse); //注册鼠标事件  glutDisplayFunc(&myDisplay); //回调函数  glutMainLoop(); //持续显示，当窗口改变会重新绘制图形  return 0;  } |

反走样实现：

|  |
| --- |
| #include<windows.h>  #include <GL/glut.h>  #include <stdio.h>  //线的反走样  static float rotAngle = 0.;  void init(void)  {  GLfloat values[2];  glGetFloatv(GL\_LINE\_WIDTH\_GRANULARITY, values);  printf("GL\_LINE\_WIDTH\_GRANULARITY value is %3.1f\n", values[0]);  glGetFloatv(GL\_LINE\_WIDTH\_RANGE, values);  printf("GL\_LINE\_WIDTH\_RANGE values are %3.1f %3.1f\n", values[0], values[1]);  glEnable(GL\_LINE\_SMOOTH);  glEnable(GL\_BLEND);  glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);  glHint(GL\_LINE\_SMOOTH\_HINT, GL\_DONT\_CARE);  glLineWidth(1.5);  glClearColor(0.0, 0.0, 0.0, 0.0);  }  void display(void)  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glColor3f(0.0, 1.0, 0.0);  glPushMatrix();  glRotatef(-rotAngle, 0.0, 0.0, 0.1);  glBegin(GL\_LINES);  glVertex2f(-0.5, 0.5);  glVertex2f(0.5, -0.5);  glEnd();  glPopMatrix();  glColor3f(0.0, 0.0, 1.0);  glPushMatrix();  glRotatef(rotAngle, 0.0, 0.0, 0.1);  glBegin(GL\_LINES);  glVertex2f(0.5, 0.5);  glVertex2f(-0.5, -0.5);  glEnd();  glPopMatrix();  glFlush();  }  void reshape(int w, int h)  {  glViewport(0, 0, w, h);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  if (w <= h)  gluOrtho2D (-1.0, 1.0, -1.0\*(GLfloat)h/(GLfloat)w, 1.0\*(GLfloat)h/(GLfloat)w);  else  gluOrtho2D (-1.0\*(GLfloat)w/(GLfloat)h, 1.0\*(GLfloat)w/(GLfloat)h, -1.0, 1.0);  glMatrixMode(GL\_MODELVIEW);  glLoadIdentity();  }  void keyboard(unsigned char key, int x, int y)  {  switch (key)  {  case 'r': case 'R':  rotAngle += 20.;  if (rotAngle >= 360.0) rotAngle = 0.0;  glutPostRedisplay();  break;  case 27:  exit(0);  break;  default:  break;  }  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(200, 200);  glutCreateWindow(argv[0]);  init();  glutReshapeFunc(reshape);  glutKeyboardFunc(keyboard);  glutDisplayFunc(display);  glutMainLoop();  return 0;  } |

5、实验总结

在对图形进行操作时，和用户的交互也是十分重要的，因此，鼠标控制函数有着十分重要的作用。而反走样技术指的是在硬件计算时输出高分辨率的图像，并通过增加采样点或者图像处理等手段去除图像中的锯齿等走样现象的技术，也十分重要