**实验名称：**绕任意轴旋转

1. **具体内容**

3维空间系统包含两个共享一条边E1的长方体A和B，如下图所示，A绕着过（20,20,20）点且方向为（1,1,1）的轴旋转，无论A处于什么姿态，B总能绕与A的共享边E1（随着A的位置姿态改变而改变）旋转。A转动时B跟着转，B转动时A不动，即A和B存在主次关系

1. **设计思路**

**A转动时，B和A一起转动；B可以单独转动**

1. **主要函数及其简要说明**

//绕任意轴旋转 x,y,z 方向a,b,c

void my\_axis\_rotate\_homogeneous(struct my\_v\_homogeneous\* polygon, int vertex\_count, float x, float y, float z, double a, double b, double c, float angle)

{

float angle1 = atan2(b, a) \* 180 / pi;

float angle2 = atan2(c, sqrt(a \* a + c \* c)) \* 180 / pi;

my\_translate\_homogeneous(polygon, vertex\_count, -x, -y, -z);

my\_rotate\_homogeneous(polygon, vertex\_count, 'z', -angle1);

my\_rotate\_homogeneous(polygon, vertex\_count, 'y', angle2);

my\_rotate\_homogeneous(polygon, vertex\_count, 'x', angle);

my\_rotate\_homogeneous(polygon, vertex\_count, 'y', -angle2);

my\_rotate\_homogeneous(polygon, vertex\_count, 'z', angle1);

my\_translate\_homogeneous(polygon, vertex\_count, x, y, z);

}

void keyboard(unsigned char key, int x, int y)

{

switch (key)

{

case'1':

{

my\_axis\_rotate\_homogeneous(box[0], 8, 20, 20, 20, 1, 1, 1, 30);

my\_axis\_rotate\_homogeneous(box[1], 8, 20, 20, 20, 1, 1, 1, 30);

glutPostRedisplay();

break;

}

case'2':

{

my\_axis\_rotate\_homogeneous(box[1], 8, box[1][0].x, box[1][0].y, box[1][0].z,

box[1][3].x - box[1][0].x, box[1][3].y - box[1][0].y, box[1][3].z - box[1][0].z, 30);

glutPostRedisplay();

break;

}

1. **遇到的主要问题及解决思路：**

**无**

1. **程序运行结果**

