

三维 CAD 建模实验报告

学号：21721059 姓名：林莹莹

1. 实验环境

操作系统：macOS Sierra 10.12.6

编译器：Xcode Version 9.0 (9A235)

引用库：GLUT.framework、OpenGL.framework 以及 GLFW

编程语言：c++

2. 程序结构

本实验程序主要包括四个文件

- StrucDefine.h 文件是对边界表示法里的基本结构进行表示，其中包括 Solid、Face、Loop、Edege、Halfedge、Vertex。
- EulerOperation.h 文件一部分是对欧拉操作的定义，包括 mvfs、mev、mef、kemr、krmrh；还有扫成函数 sweep 和其他生成实体的相关函数。sweepwithhandle
- EulerOperation.cpp
欧拉操作、扫成操作的以及相关函数的实现。
- main.cpp 除了包括一些 OpenGL 的显示和渲染，还有以下三个与实验相关的重要绘制函数：

```
void DrawWithoutSweep();
```

```
void DrawWithSweep();
```

```
void DrawWithHandle();
```

这三个函数分别实现单纯欧拉操作的实体建模、扫成的实体建模、有通孔的实体建模。

3. 操作方法

3.1 键盘控制

q: 退出

l: 改变渲染模式（线框和实体）

a、d、w、s: 上下左右

z: 视野变近

c: 视野变远

空格: 旋转

3.2 实现方式变化

通过改变 main.cpp 中两个地方的函数调用，实现多种实体建模。

第一个地方是 main.cpp 里的 void redraw()函数。通过选择下列三个函数实现单纯欧拉操作的实体建模、扫成的实体建模、有通孔的实体建模。

```
void redraw()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glClearColor(0.9, 0.9, 0.9, 1);
    glLoadIdentity(); // Reset The Current Modelview Matrix

    gluLookAt(eye[0], eye[1], eye[2], // gluLookAt is deprecated: first deprecated in macOS 10.9 - "Use GLKMatrix4MakeLookAt"
             center[0], center[1], center[2],
             0, 1, 0);

    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    GLfloat gray[] = { 0.4, 0.4, 0.4, 1.0 };
    GLfloat pink[] = {0.9, 0.5, 0.8, 1.0};
    GLfloat light_pos[] = {10, 10, 10, 1};
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT,pink);
    glLightfv(GL_LIGHT0, GL_POSITION, light_pos);
    glLightfv(GL_LIGHT0, GL_AMBIENT, gray);
    glEnable(GL_LIGHT0);

    if (bAnim) fRotate += 0.5f;
    glRotatef(fRotate, 1.0f, 0.0f, 0); // Rotate around Y axis
    glScalef(0.4, 0.4, 0.4);

    if(isLineMode){
        glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    }
    else{
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    }

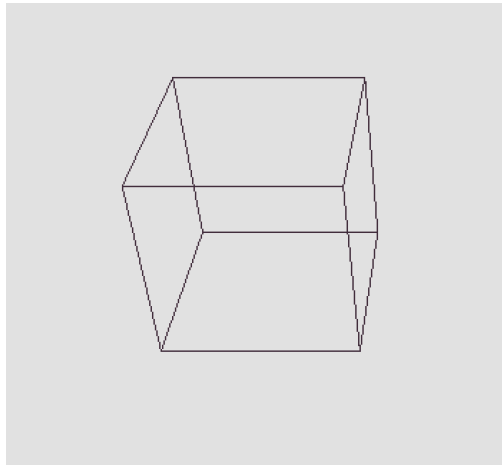
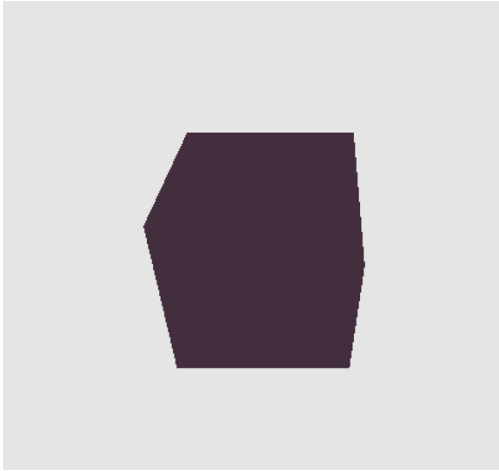
    //DrawWithSweep();
    //DrawWithoutSweep();
    DrawWithHandle();

    glutSwapBuffers(); // glutSwapBuffers is deprecated: first deprecated in macOS 10.9
}
```

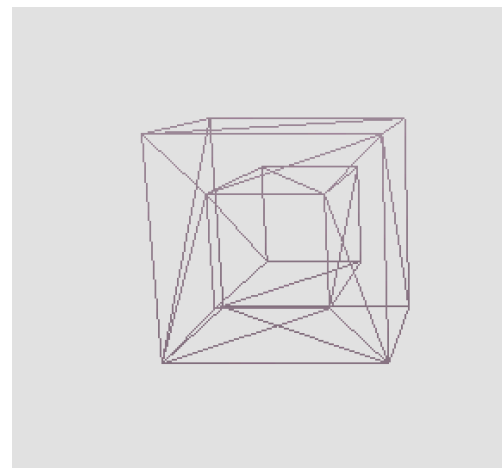
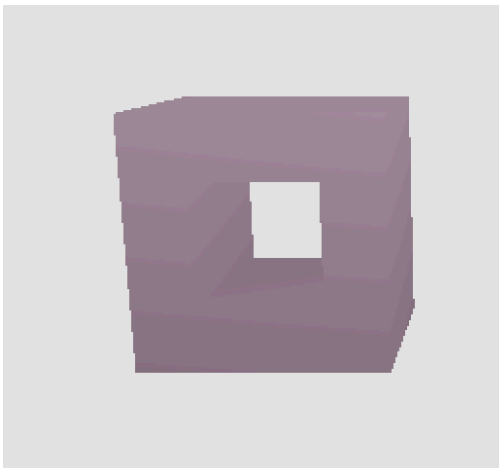
第二个地方是在 main.cpp 的 void DrawWithHandle()函数里。同个对生成 Solid *s 三种不同方法的选择，选择生成通孔的方式和个数。

4. 实现结果：

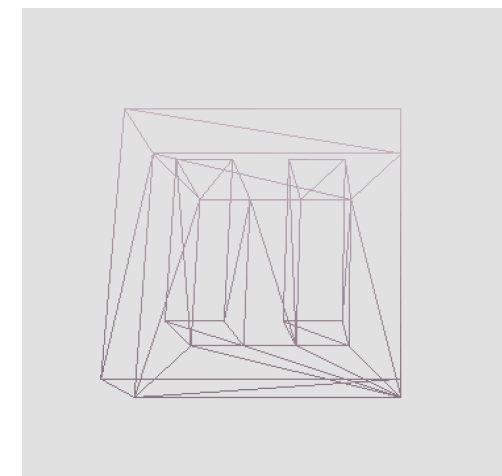
4.1 使用欧拉操作生成的立方体



4.2 使用扫成生成的单通孔



4.3 使用扫成实现的多通孔



5. 其他

若想生成多于两个通孔的模型，只需要对

```
Solid *EulerOperation::SolidWithOneHandle()
```

或者

```
Solid *EulerOperation::SolidWithTwoHandle()
```

的输入数据进行修改即可。