# 图形学实验一报告

3140102384 郑驭聪

## 问题描述

构建一个正方形，并三角网格化内部区域后，将其形变为圆形，要求网格形变均匀。

## 求解思路

我将这个问题分为三个部分，首先是对一个正方形进行三角化，然后进行一个正方形边缘到圆形边缘的形变，最后调整内部三角形网格，使之形变均匀。所谓形变均匀，我的理解是，网格拓扑结构不变，长宽变化和外轮廓变化比例一致。

对于三角化，可以采用Delauney算法。Delauney算法可以对平面上的一个点集，构建一个满足空圆特性、最大化最小角特性、凸包特性的三角网格。注意到本身正方形内部并没有点，因此可以在正方形边上均匀撒上一些边缘点，在内部随机撒上一些内部点，再进行Delauney算法。另外，内部随机撒点可能会造成三角形网格不均匀，因此，这里也会采用和第三步一致的调整网格使形变均匀的算法。

对于形变，我的选择是从正方形边映射到圆形边。由于之前已经在正方形边缘上撒了一些点，现在将这些点均匀地分布在圆上即可。在具体变换时，每次都朝着目标点前进若干分之一（比如1/10），这些在动态显示时，就能让变换更加平滑。

对于形变均匀，我的办法是二维拉普拉斯平滑。在三维网格中，一种常用的平滑方式是将每个顶点都移动到相邻顶点的平均位置，即采用所谓伞状算子。这种方法就是拉普拉斯平滑。我在二维上模拟了这样的算法，每次迭代时，都将点移动到其一环领域的中心。理论上，无数次的平滑操作后，所有的点将均匀分布。在外形发生变化时，它也能使形变非常均匀。

### 具体实现

Delauney算法我采用的是比较高效的Bowyer-Watson算法，它只需遍历每个点一次，即可产生三角剖分，而不需遍历每种可能的三角形。其算法流程如下：

1、构造一个超级三角形，包含所有散点，放入三角形链表。

2、将点集中的散点依次插入，在三角形链表中找出外接圆包含插入点的三角形（称为该点的影响三角形），删除影响三角形的公共边，将插入点同影响三角形的全部顶点连接起来，完成一个点在Delaunay三角形链表中的插入。

3、根据优化准则对局部新形成的三角形优化。将形成的三角形放入Delaunay三角形链表。

4、循环执行上述第2步，直到所有散点插入完毕。

形变过程非常容易，其中正方形的四个顶点和四条边的中点映射到圆在东南西北和东南、东北、西南、西北的位置，其他点按比例均匀插入其中即可。

二维拉普拉斯算法本质在于得到伞状算子的矩阵，这个矩阵的数学表达式如下：



以后每次只要多次乘上这个矩阵就可以使顶点分布均匀了，这也有点类似图像处理中的滤波算法。

## 实验结果

|  |  |
| --- | --- |
|  |  |
| 正方形 | 过渡 |
|  |  |
| 圆形 |  |