

Algorithm

郑华

2016 年 5 月 2 日

1 Delaunay 三角网与 Voronoi 图

1.1 Voronoi 图

- 1- 由一组两邻点直线的垂直平分线组成的连续多边形组成
- 2-N 个在平面上有区别的点，按照最邻近原则划分平面
- 3-每个点与它的最近邻区域相关联
- 4-Delaunay 三角形是由与相邻 Voronoi 多边形共享一条边的相关点连接而成的三角形
- 5-Delaunay 三角形的外接圆圆心是与三角形相关的 Voronoi 多边形的一个顶点

1.2 Delaunay 三角网

- 1- Delaunay 三角网是唯一的
- 2- 三角网的外边界构成了点集 P 的凸多边形“外壳”
- 3- 没有任何点在三角形的外接圆内部
- 4-如果将三角网中的每个三角形的最小角进行升序排列，则 Delaunay 三角网的排列得到的数值最大

Meanwhile Delaunay 三角形网的特征又可以表达为以下特性：

- 1- 在 Delaunay 三角形网中任一三角形的外接圆范围内不会有其它点存在并与其通视，即空圆特性
- 2- 在构网时，总是选择最邻近的点形成三角形并且不与约束线段相交
- 3- 形成的三角形网总是具有最优的形状特征，任意两个相邻三角形形成的凸四边形的对角线如果可以互换的话，那么两个三角形 6 个内角中最小的角度不会变大
- 4- 不论从区域何处开始构网，最终都将得到一致的结果，即构网具有唯一性。

1.3 伪代码

<http://www.tuicool.com/articles/EvUVRfV>

input: 顶点列表 (vertices) //vertices 为外部生成的随机或乱序顶点列表

output: 已确定的三角形列表 (triangles)

初始化顶点列表

创建索引列表 (indices = new Array(vertices.length)) //indices 数组中的值为 0,1,2,3,.....,vertices.length-1

基于 vertices 中的顶点 x 坐标对 indices 进行 sort //sort 后的 indices 值顺序为顶点坐标 x 从小到大排序（也可对 y 坐标，本例中针对 x 坐标）

确定超级三角形

将超级三角形保存至未确定三角形列表 (temp triangles)

将超级三角形 push 到 triangles 列表

遍历基于 indices 顺序的 vertices 中每一个点 //基于 indices 后，则顶点则是由 x 从小到大出现

初始化边缓存数组 (edge buffer)

遍历 temp triangles 中的每一个三角形

计算该三角形的圆心和半径

如果该点在外接圆的右侧

则该三角形为 Delaunay 三角形，保存到 triangles

并在 temp 里去除掉

跳过

如果该点在外接圆外（即也不是外接圆右侧）

则该三角形为不确定 //后面会在问题中讨论

跳过如果该点在外接圆内则该三角形不为 Delaunay 三角形

将三边保存至 edge buffer

在 temp 中去除掉该三角形

对 edge buffer 进行去重
将 edge buffer 中的边与当前的点进行组合成若干三角形并保存至 temp triangles 中
将 triangles 与 temp triangles 进行合并
除去与超级三角形有关的三角形
end