《三维点集最大空心球及可视化》实验报告

李瑞龙　　计研 163　2016211003

董欣　　计研 163　2016211011

欧阳雯琪　计研 163　2016211002

一. 问题描述

给定一组三维散乱点集，求点集的最大空心球问题本质上即求该点集的三维 Delaunay 三角剖分，在求得三维点集的 Delaunay 三角剖分后，通过遍历 Delaunay 三角剖分得到的四面体的外接球，即可求出最大空心球。

Voronoi 图是计算几何里一种基于距离的平面划分方法，它可以扩展到更高的维度。三维 Voronoi 图是根据三维空间中 n 个不重合的种子点，将三维空间划分为 n 个多面体区域，使得每个多面体区域内的点到它所在区域的种子点的距离比到其它区域种子点的距离近。其中，Delaunay 三角剖分与 Voronoi 图为对偶关系。

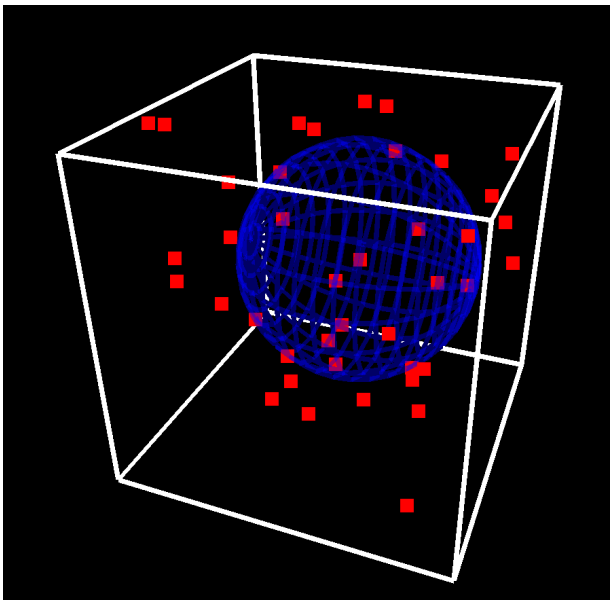

图 1　问题描述

二. 算法介绍

给定一组三维散乱点集，求该点集的最大空心球主要分为以下几个步骤，下面分别进行介绍。

1. 求三维点集的 Delaunay 三角剖分

由于目标是求解三维点集的最大空心球，而 Delaunay 三角形剖分与 Voronoi 图互为对偶图，以二维 Voronoi 图与 Delaunay 三角剖分为例，两者的关系如图 2 所示。其中红色节点为点集 S 中的节点，灰色细线为三角剖分结果，黑色实线为 Voronoi 图结果。

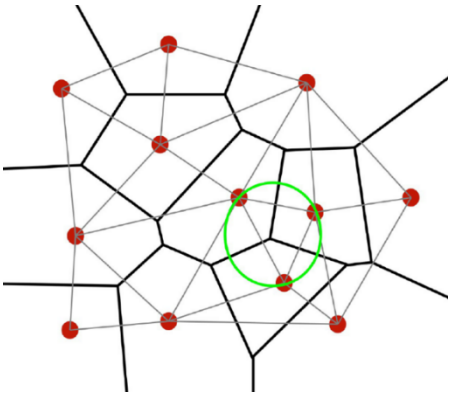

图 2 二维 Voronoi 图与 Delaunay 三角剖分对偶关系图

在介绍具体算法之前，首先分析三维 Voronoi 图与 Delaunay 三角形剖分的特征，理解三维 Voronoi 图与 Delaunay 三角形剖分直接的关系。

三维 Voronoi 图的特征：

①. 三维 Voronoi 图结果是唯一的

②. Voronoi(S)把三维空间分为 n 个多面体区域，每个区域包含且仅包含三维点集 S 中的一个点 P，每个多面体区域中的点到点 P 的距离都比到 S 中其他点的距离近

②. Voronoi(S)的每个面垂直于 S 中一对点的中垂线

③. Voronoi(S)的每个多面体区域的顶点到与该顶点相邻的多面体区域包围的 S 中的点 P 的距离相等，故以 Voronoi 点为球心，Voronoi 点到临近点 P 的距离为半径的球内不包含三维点集中的任何点

三维 Delaunay 三角形剖分的特征：

①. Delaunay 三角形剖分结果是唯一的

②. Delaunay 三角形剖分后每个四面体的外接球内均不含 S 中的任意点

③. Delaunay 三角形剖分外边界三角面片构成了点集 S 的多边形凸包

Delaunay 三角形剖分算法步骤：逐点插入法

(a). 概述

关于三维 Delaunay 剖分，主要有逐点插入算法、分治算法、三角网生成

法等，由于逐点插入算法实现简单，运行占用的空间较小，在点数量较少的情况下时间效率比较高，故本实验的 Delaunay 三角形剖分采用逐点插入法[1]实现。

(b). 伪代码

输入：有一系列三维坐标构成的点集 S
输出：以 S 中点为顶点的四面体集合，即 Delaunay 三角形剖分结果
Teras.clear()
构造一个超级四面体，包含 S 中所有待剖分的点，将该四面体加入到 Teras 中
While !S.empty
　　将一个新点 P 插入到目前的剖分中来
　　遍历当前的四面体列表 Teras 找到外接球包含点 P 的所有四面体构成集合 Tmp
　　将 P 与集合 Tmp 中的每个四面体的任意三个点相邻，构成四个新的四面体，从 Teras 列表中删除 Tmp 中的四面体
　　遍历 Tmp 中的四面体，如果有两个四面体相同则删除这两个四面体
　　将 Tmp 中剩余的四面体加入到 Teras 列表中
对超级四面体的四个顶点依次进行删除，每删除一个点则更新一下四面体列表，具体删除过程见第四节遇到的问题与解决方法
返回 Teras 列表

(c). 算法过程图示

以 6 个三维点为例，首先需要构造超级正四面体，使该四面体包含所有的点：

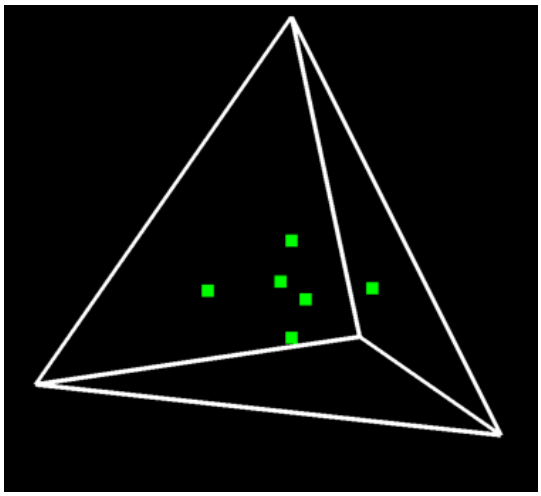

图 3-1

对点集 S 中的点，逐点进行处理：

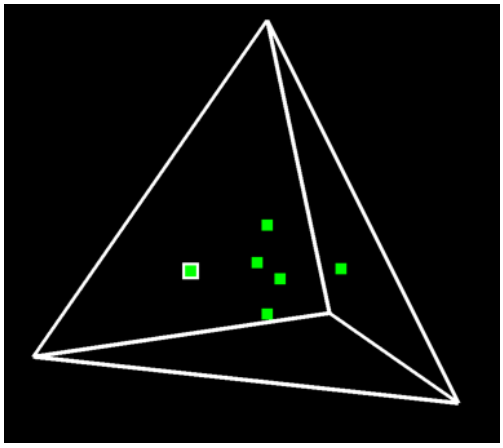

图 3-2

遍历四面体集合 Teras，找出外接球中包含该点的所有四面体 Tmp，对 Tmp 中的每个四面体，在 Teras 中去掉该四面体，并将该点与去掉四面体的四个点相连得到四个新的四面体 NewList：

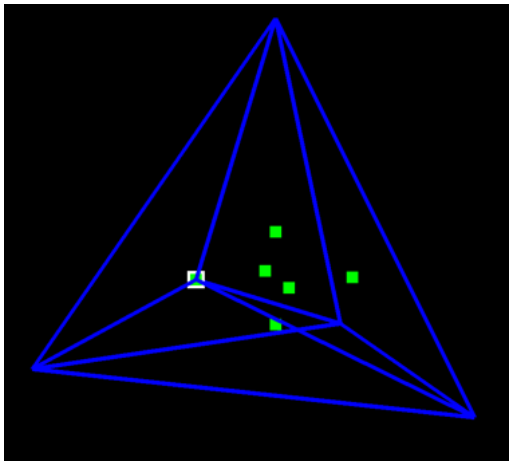

图 3-3

如果 NewList 中有任意两个四面体相同，则说明该剖分是错误的，去除这两个相同的四面体，并将剩下的四面体加入到 Teras 中：

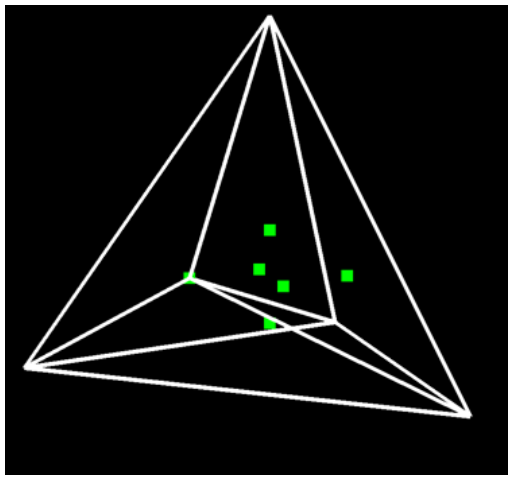

图 3-4

重复以上过程直到处理完 S 中的所有点：

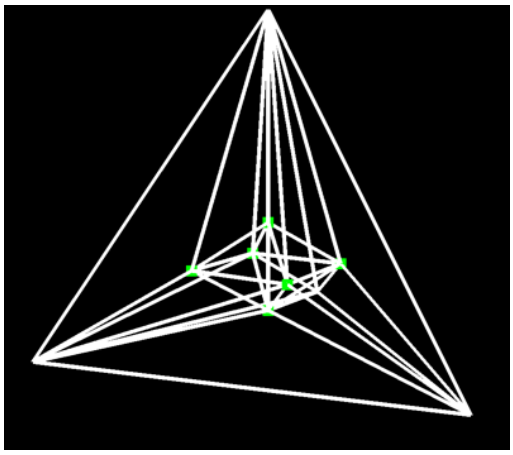

图 3-5

去掉最初添加的超级四面体，得到 S 中点集的四面体剖分结果：

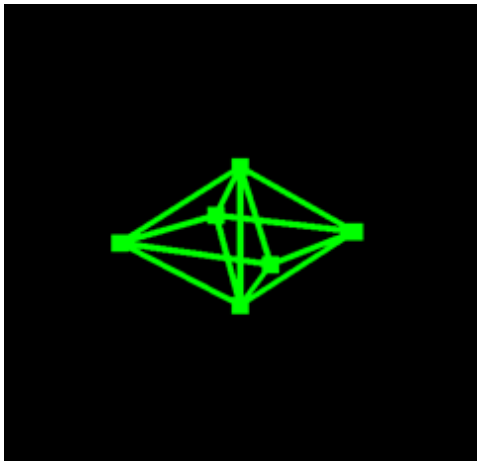

图 3-6

2. 遍历 Voronoi 图的节点求最大空心球

遍历所有 Delaunay 三角剖分四面体的外接球，找到外接球半径最大的，即为所求的最大空心球。可以得到圆心与半径。

3. 作为附加，利用 Delaunay 三角形剖分求三维 Voronoi 图

在三维空间中，由 delaunay 四面体剖分得到的四面体的外心，即是构成这个四面体的四个顶点的一个 voronoi vertex。而对 delaunay 图中的每一个顶点，取其与周围顶点相连的每一条边，作其垂直平分面，这组垂直平分面相交得到的凸包即是该点的晶胞，也即空间中距离该点最近的点集合。

由于点集中，位于最外部，也即凸包上的点的晶胞是无界的。因此，为了能

够很好地显示三维 voronoi 图，应对 voronoi 图设置一个显示视窗，只显示位于视窗内的 voronoi 图。为此，我们需要求解出位于最外部的晶胞，与视窗交界所产生出来的点。

其思路如下：

能够与视窗相交出现相交点，必有这几种情况：

(1).位于最外部的三角面片的外心所发出来的沿着法向量的直线，与视窗的交点。

(2).位于最外部的三角面片的三条边的垂直平分面与视窗的棱的交点

(3).处在最外部的晶胞内部的视窗上的顶点

求解步骤:

(a).计算每个四面体的外心，得到一个 voronoi vertex，并分别加到四面体的四个顶点的 vertices 集中。

(b).遍历所有四面体，对于其所有面，计算其每一侧是否有点，如果有一侧没有点，则认为该面是凸包上的面，讲其按照该侧为正方向的顺序存好。

(c).设置视窗为正方体

(d).遍历每个凸包集中的四面体，计算其位于凸包上的三角面片，由面片外心发出沿着法向量的射线，求其与视窗的交点，并存入构成构三角面片的每一个顶点的 vertices 集中。

由三角面片的每条边，计算垂直平分面与视窗所有棱的交点，并存入构成该边的顶点的 vertices 集中。

(e).遍历视窗的每个顶点，计算其距离所有顶点中最近的那个点（如果有相等情况，记录下所有这样的点），并存入该点（或点集）的 vertices 集中。

(f).遍历每个顶点，由该点的 vertices 集计算凸包，得到构成凸包的三角面片。

```
For tetra in 所有四面体：
    For v in 四面体的所有顶点：
        将四面外心加入 v.vertex
For tetra in 所有四面体：
    For triangle in 所有三角面片：
        If triangle 一侧有点 {
            将三角面片翻转
            If triangle 一侧有点
                cintinue
        }
        构建由三角面片外心发出的沿着法向量的射线 ray
        求 ray 与视窗 cube 的交点 p
        For v in 三角面片的所有顶点：
            将 p 加入 v.vertex
        For segment in triangle 的所有边：
            构造其垂直平分面
            计算其与 cube 的棱的交点 I
            将 I 加入 segment 的顶点.vertex
For v in cube 的所有顶点：
    For s in 空间点集
        计算 v 与 s 的距离，计算最小值对应的点 P
    将 v 加入 P.vertex
For s in 空间点集：
    输入 s.vertex
    构造 convhull
    得到三角面片 s.convhull
返回所有 s 的 cinvhull
```

伪代码：

三. 可视化展示

可视化框架基于 Qt 与 OpenGL 实现，分别展示 Delaunay 三角形剖分，QuickHull 求三维凸包，三维 Voronoi 图生成，最大空心球展示。

1. 三维点集与 Delaunay 三角形剖分可视化

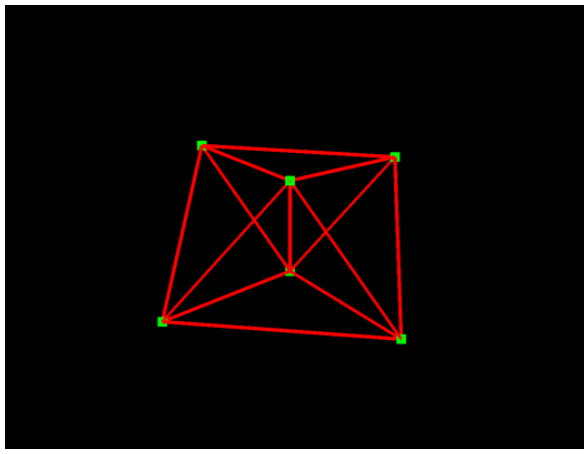

图 4 Delaunay 三角剖分

2. 三维点集最大空心球可视化

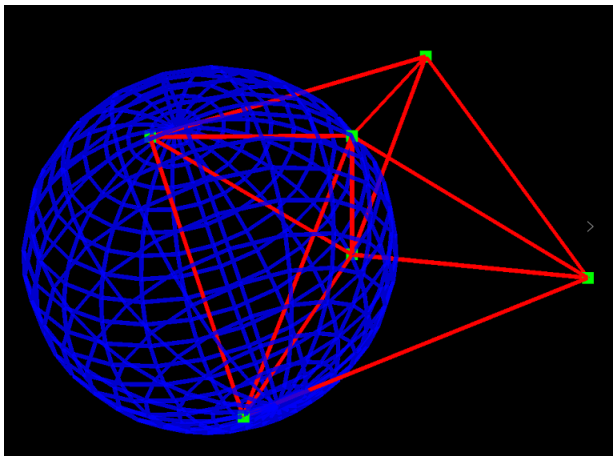

图 5 求最大空心球

3. 附加：利用 Delaunay 三角形剖分得到三维 Voronoi 图

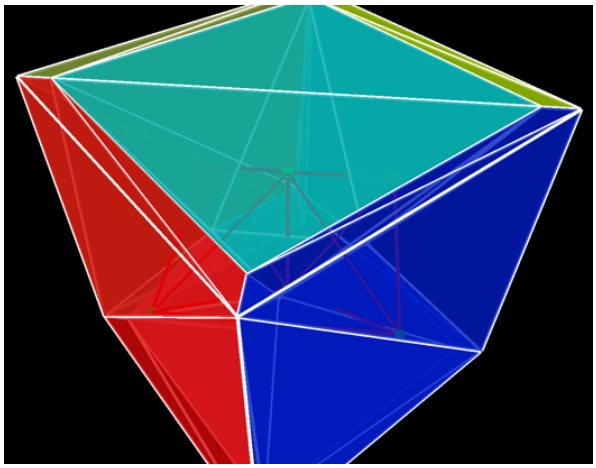

图 6-1 利用 Delaunay 三角剖分求 Voronoi 图

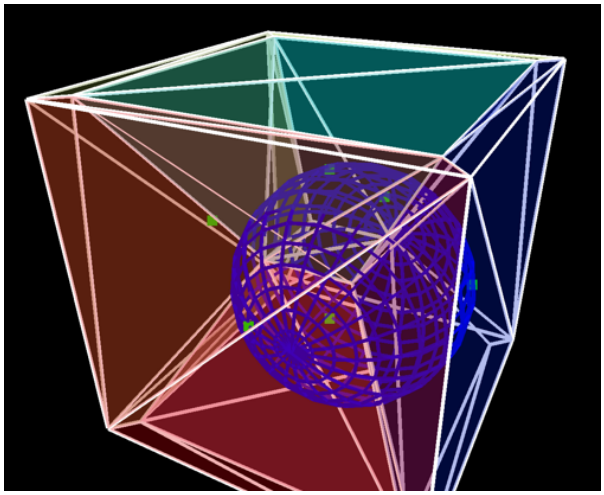

图 6-2 在 Voronoi 图中画出最大空心球

四. 实验中遇到的问题与解决对策

1. **问题：**文献[1]中在删去超级四面体的时候直接从四面体结果列表中将包含超级四面体中顶点的四面体删除，导致最后得到的三角剖分可能会出现“凹陷”。

解决方法：修改论文中删去超级四面体的部分代码，实现 Delaunay 三角剖分中删除一个节点的操作，原理是在删除一个节点时，将所有待删除的四面体保存在一个列表中，删除这些四面体，并得到除要删除节点的其他节点的列表，从列表中枚举四个顶点得到一个四面体待添加列表，遍历该待添加列表，判断是否有 S 中的点出现在该四面体的外接圆中，如果有则丢弃该四面体，如果没有，则说明该四面体也是一个四面体剖分，将该四面体加入到结果列表中。依次处理超级四面体的四个节点，最后得到 Delaunay 剖分结果，所有最外层的三角面片构成了点集 S 的一个凸包。以 100 个点为例，得到的凸包结果如下所示：

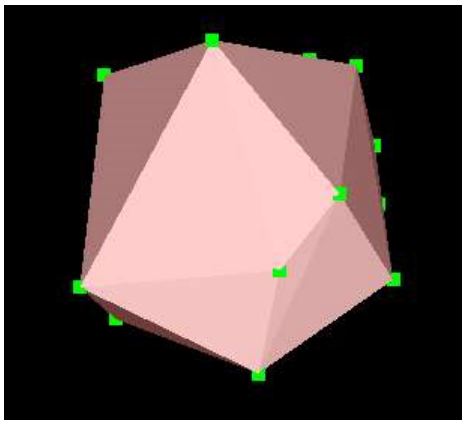

图 7 利用 Delaunay 三角剖分得到凸包

2. **问题：**在计算最外部的三角面片时，原本的计算思路是遍历所有的四面体，

寻找其是否有三个顶点位于凸包上。如果是，则该面片是外部的面片；如果不是，则没有外部的面片。也即该四面体不是最外部的四面体。然而，却存在四面体，其有三个顶点位于凸包上，而这三个顶点构成的面却不是外部的面。

解决办法：我们换了一种思路，遍历所有四面体，对于其所有面，计算其每一侧是否有点，如果有一侧没有点，则认为该面是凸包上的面，讲其按照该侧为正方向的顺序存好。

五.没有解决的问题

1. 求三维空间 Delaunay 三角剖分算法在点集比较大的情况下运行速率较慢，还有较大的改进空间。

2. Delaunay 三角剖分的退化情况可能会导致存在多余的四面体。

3. Voronoi 图的构建时如果 Voronoi 的某个 vertice 在立方体的外面，会导致该 vertice 会在立方体外面显示。

六.参考文献

[1] Bourke, P. "An algorithm for interpolating irregularly-spaced data with applications in terrain modelling." Pan Pacific Computer Conference, Beijing, China. Vol. 6. 1989.

[2] Hiroshi Imai, Masao Iri and Kazuo Murota. "Voronoi Diagram in the Laguerre Geometry and Its Applications." SIAM J. COMPUT. Vol. 14, No. 1, February 1985.

[3] Rasmus Fonseca, Pawel Winter, Kevin Karplus, "Protein Packing Quality Using Delaunay Complexes", Voronoi Diagrams in Science and Engineering (ISVD) 2011 Eighth International Symposium on, pp. 117-122, 2011.

[4] Ronald Boisvert National Institute of Standards and Technology, Guithersburg, MD. "The quickhull algorithm for convex hulls." ACM Transactions on Mathematical Software (TOMS). Volume 22 Issue 4, Dec. 1996.