3-D 点集 Delaunay 三角剖分及 Voronoi 图算法实现

侯聪　　韦奕多　　王鹏

摘要：本实验对 3-D 点集的 Delaunay 三角剖分及 Voronoi 图的生成算法进行了实现。主要思想是首先利用增量算法生成 3-D 点集的三角剖分，然后利用局部转换算法生成 Delaunay 三角剖分，最后根据对偶图性质由 Delaunay 三角剖分得到 3-D 点集的 Voronoi 图，实验结果显示了非退化 3-D 点集的 Delaunay 三角剖分及 Voronoi 图。

一、背景及简介

Voronoi 图是一种广泛通用的几何结构，其应用领域包括物理学、天文学、机器人学以及计算机图形学等等众多领域. Delaunay 三角剖分是 Voronoi 图的直线对偶。对于 2-D 的情况二者的生成算法研究已经很成熟。所以我们的大实验主要目的是对 3-D 点集情况下二者生成算法进行探索和实现。主要依据的文献是[1]、[2]、[3]。

二、基本定义和主要概念

为了更好的描述本实验的思路和算法，这个部分将把基本的定义、主要的概念做详细介绍。

（1）三维点集的 Voronoi 图：这个点集的一个空间划分，划分后点集中的每个点（Voronoi 站点）对应一个多面体形状 Voronoi 区域（Voronoi cell），这个区域是距离该点最近的一个区域，并且各个区域无重叠无缝隙地拼 接在一起布满点集确定的整个空间。

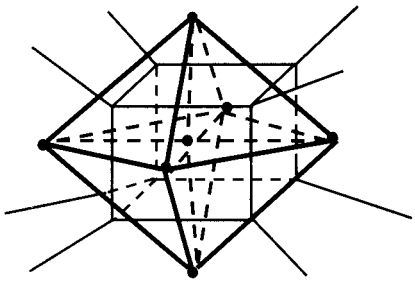
（2）Delaunay 三角剖分：Voronoi 图的直线对偶图,满足以下性质

①三维点集 S 的三角剖分是覆盖 S 凸包的互不重叠四面体的集合。

②记三维点集 S = { s1 , s2 , …, sn} , 其凸包为 CH (S) , 三角剖分后所生成的四面体集合 T = { T1 , T2 , …, Tn} 满足:(A)T 中所有四面体顶点均属于 S .(B)T 中任意两个四面体的交集为或有一个顶点重合 , 或有一条边重合 , 或有一个面重合, 或为空集。

③ Delaunay 三角剖分满足**空球性质**, 即三角剖分中任一四面体的外接球内不含点集中除四面体顶点以外任一点。

④如果 n 个点中任意五点不共球, 则其 Delaunay 三角剖分唯一。



（图中细线为 Voronoi 图，粗线是 Delaunay 三角剖分，每个 Delaunay 单元为四面体）

（3）Delaunay 四面体的形状度量因子 μ：前面刚介绍了 3-D 点集的 Delaunay

三角剖分基本单元是四面体，而且满足空球性质。为了判断是否是这种或接近这种最优的 Delaunay 性质四面体，文献[1]提出了形状度量因子的判断标准。这种因子 μ应该是对平移、旋转、反射和缩放都不变的量。而且当四面体是满足正则性质的四面体时此因子取到最大值;当是很不合法的接近退化的四面体时因子接近于零。显然这同我们在二维情况时的“角度最优”判断三角剖分是本质思想一致的，只不过这里用了三维的判断量不再是角度。在我们的实验中选取的形状因子是半径率 ρ，计算公式为：

$$\rho = \frac{216v^2}{\sum_{i=1}^{4} s_i \sqrt{(p_1+p_2+p_3)(p_1+p_2-p_3)(p_1+p_3-p_2)(p_2+p_3-p_1)}}$$

（其中 v是四面体体积、s_i是四个面的面积，p_i是四面体相对边的长度的乘积。）

（4）3-D 点集 $\mathcal{V}$ 的最优三角剖分：设 $\mathcal{T}$ 和 $\mathcal{T}'$ 是 $\mathcal{V}$ 的不同的三角剖分，$\mathcal{T}$ 的形状因子序列定义为：$\mathbf{u} = (\mu_1, \mu_2, \ldots, \mu_p)$，这里有 $\mu_1 \le \mu_2 \le \cdots \le \mu_p$，是 $\mathcal{T}$ 的各个四面体形状因子。同样对于 $\mathcal{T}'$ 有 $\mathbf{u}' = (\mu'_1, \mu'_2, \ldots, \mu'_q)$，如果字典序上 $\mathbf{u}$ 大于 $\mathbf{u}'$，则我们说 $\mathcal{T}$ 是更优的三角剖分，$\mathcal{V}$ 的最优三角剖分即具有最大字典序的 $\mathbf{u}$ 形状因子序列的三角剖分，可以通过局部转换来逐步得到。可见这同二维情况是本质上相似的。

三、原理及算法实现：

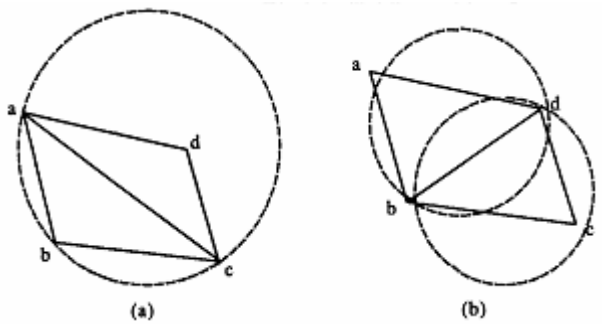
（1）算法本质可以表述为：“增量算法”+“局部转换”+“对偶变换”。

基本思路是这样的:1.S 是三维空间中 n 个点组成的集合,假定 S 中前 i－1 个点的 Delaunay 三角剖分已经生成,记为 D(i － 1) ,加入 S 中第 i 个点 pi 到 D(i － 1) 中 ,通过一系列局部交换使之保持 Delaunay 性质,结果为 D(i) 。重复此过程直至 i = n。如果实现正确,这个策略总是能生成给定点集 S 的 Delaunay 三角剖分。这里的 Delaunay 性质就是前面提到的空球准则.2.根据 Voronoi 图是 Delaunay 三角剖分的对偶图得到 Voronoi 图。

在刚开始时我们以为第一步应该很容易，而第二步会困难些，但随着实现的深入，我们发现第一步是非常主要的。

（2）在具体描述算法之前，我们必须来对算法核心的“局部转换”环节进行进一步的深入分析：

首先来回忆一下二维局部交换，如下图所示：



对三维点集来说,局部交换主要是指局部改变四面体对的拓扑连接关系。局部交换过程基于五个互异的不共面三维点 a, b, c, d, e。设 CH 是这五个点的凸包。对这五点的三角剖分有以下五种可能的结果：

A. 当五点中任意四点不共面时,有两种可能的三角剖分,如图 2(1) 、(2) 所示。①(11) , (12) 中凸包 CH 由所有五点组成。此时 CH 是一个凸六面体。可用(11) 、(12) 两种方式剖分。若线段 de 穿过面 abc 的内部 ,则 (11) 的三角剖分由两个四面体组成 , Tabce 和 Tabcd。(12) 的剖分有三个四面体 ,分别为 Tabde , Tbcde , Tacde。②CH 的边界只含五点中的四点。如(2) ,点 e 位于 CH 的内部。则此时的三角剖分只有一种 , 由四个四面体组成, Tabde, Tbcde , Tacde , Tabce。

B. 当五点中恰有四点共面, 有三种可能的情况, 如图 3 的(3) 、(4) 、(5) 。③abde 组成一个严格的凸四边形 , 有两种可能的三角剖分, 如图 3(31) 、(32) 。(31) 中三角剖分由四面体 Tabcd , Tabce 组成。(32) 剖分中的四面体为 Tacde 和 Tbcde。④四边形 退化成一个三角形,此时只存在唯一的三角剖分,由两个四面体组成,分别为 Tabcd 和 Tabce。⑤abde 组成一个非凸的四边形,其唯一可能的三角剖分由 3 个四面体组成,包括 Tacde , Tabcd , Tabce。

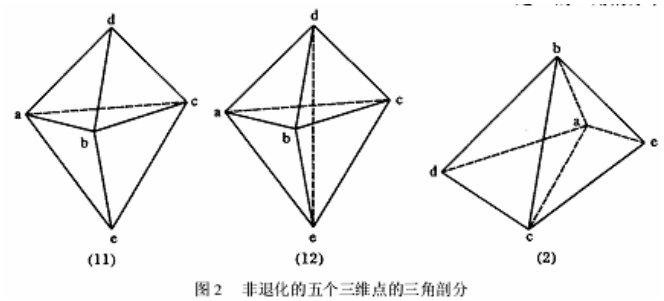


图 2　非退化的五个三维点的三角剖分

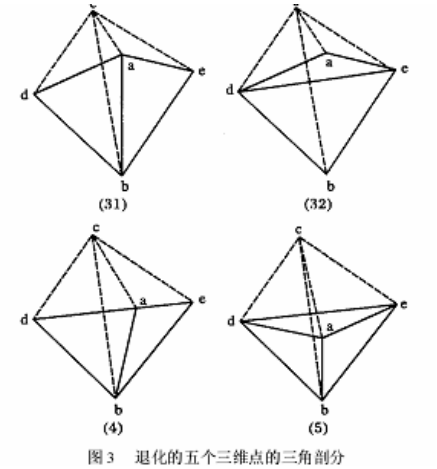


图 3　退化的五个三维点的三角剖分

下面对算法中用到的局部交换做进一步讨论。设 F(abc)是 S 的三角剖分中的一个内部面。和 abc 相连的两个四面体为 Tabcd 和 Tabce。我们把面 F(abc)的类型定义为 Trs 或 Nrs。T 表示可交换，N 表示不可交换。则可能的类型有 T23，T32，N32，T22，T44，N44，N40，N30 和 N20。除 r = s = 4 外，r 是 abcde 交换前三角剖分中含四面体的个数。s 为 0 表示只有一种可能的三角剖分。其余表示 s 是交换后 abcde 三角剖分中含四面体的个数。T44、N44 类型实际涉及六个顶点,需要同时使用一对局部交换,交换前后都是四个四面体。

T23：图 2(11)中面的类型为 T23。T32：在图 2(12)中，设 Tabde，Tacde 为剖分 T 中两个四面体,若四面体 Tdebc 也在剖分 T 中，则面 ade 的类型是 T32。

N32：在图 2(12)中，设 Tabde，Tacde 为剖分 T 中两个四面体,若四面体 Tdebc 不在剖分 T 中，则面 ade 的类型是 N32。

T22：如果 abd，abe 在剖分 T 边界上，则图 3(31)中面 abc 的类型是 T22。

T44：在图 3(31)中，设面 abd 和面 abe 是 T 的内部面。设 Tabdf 和 Tabdg 是和 abd，abe 相连的其它四面体（f 和 c 互异，g 和 c 互异）。若 f = g 则 abc 的类型是 T44，即同时使用两个局部转换(用 de 边替换 ab 边)。如果只执行两个置换之一不能生成有效的三角剖分。

N44：条件同 T44 所述，如果 f 和 g 不等，则面 abc 的类型为 N44，内部边 ab 至少和五个面相连，至少有一个面 abf 或 abg 被删除,面 abc 才能成为 T44 类型的。

N40：图 2(2)中，面 abc 的类型为 N40。

N30：图 3(5)中面 abc 的类型为 N30。

N20：图 3(4)中面 abc 的类型为 N20。

总之,面 abc 类型为 T25，T32，T22 或 T44 时 abc 是可交换的。面 abc 是不可交换的可能是 a，b，c，d，e 只有一种可能的三角剖分,面 abc 类型为 N40，N30 和 N20。也可能是由于和一边相连的多余面的存在(类型 N32 或 N44)。

(3) 具体算法描述：

我们在实验中采取的算法可以大致描述如下：

基本数据结构：采用自然的几何结构：:四面体 t,表面 s 和点 p。相关联的元素都有指针相联系。

算法简述：

```
Step1:输入四个初始点生成初始四面体。
      If(退化情况:两个点重复，三点共线，四点共面)
        则进入异常循环或不做处理。
      end
Step2: while(当插入一个新点)
       首先查找插入点的位置（定位）
       If (插入点位于已生成的三角剖分中)
          找到包含插入点的四面体;
           Do
          Case1:(插入点位于一个四面体内部)
           把该四面体分割成四个四面体;
          Case2:(插入点位于一条边上)
           把该边分成两半，再分割和该边相连的四面体;
```

Case3(插入点和一个已存在的点相同)

不做任何事情。

caseend

把所有新生成的面保存到 fliplist,以待以后局部交换时用到。

Elseif(插入点位于已生成三角剖分的外部)

查找从插入点 insertpoint 观察可见的四面体边界面。(可见性可用 insertpoint 和一个三角面组成的四面体体积的正负号来判断)保存的到 fliplist 中。同时生成由 insertpoint 和所有可见面构成的四面体。

ifEnd

whileEnd

Step3:(把 fliplist 中所有非 Delaunay 面变成 Delaunay 的)

#Input: Arbitrary triangulation $\mathcal{T}$ of $\mathcal{V}$.
#Output: μ locally optimal triangulation.
Put all interior faces of $\mathcal{T}$ in a queue and set *front* and *back*
to be pointers to the first and last faces of queue
flipmu($\mathcal{T}$, *front*,*back*)

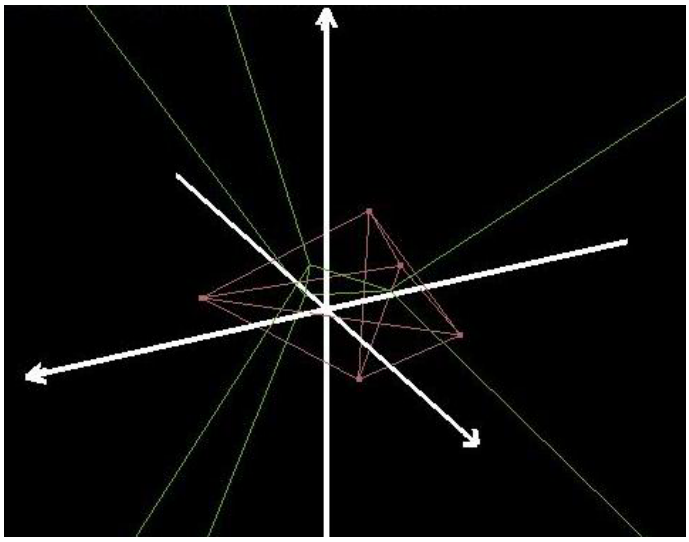
procedure flipmu($\mathcal{T}$, *front*,*back*)
#Input and output: Triangulation $\mathcal{T}$, queue pointers *front* and *back*.
while *front* $\neq$ null do
abc := front face of queue
front := pointer to next face of queue
if *abc* is still in triangulation then
if *abc* has type T23, T32, T22, or T44 then
if *abc* is not μ locally optimal then
Apply the appropriate T*rs* flip to remove *abc*; and
add all interior faces of $\mathcal{F}$(*abc*), which are not yet
in queue, to end of queue and update queue pointers
endif
endif
endif
endwhile
endproc

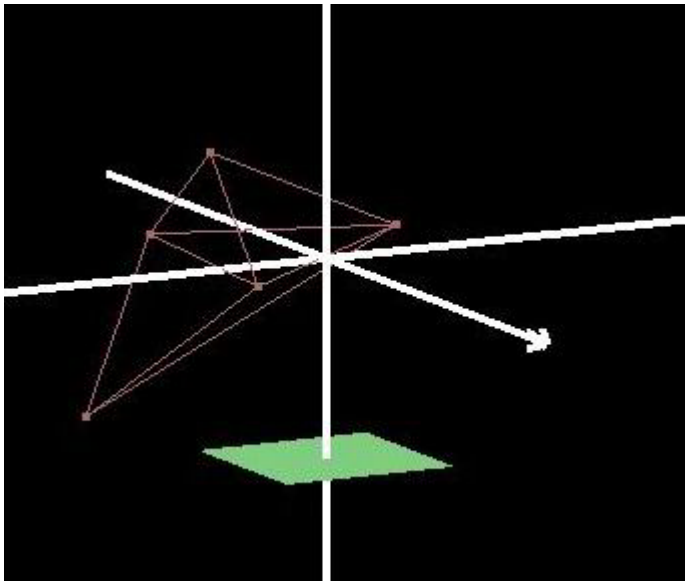
Step4:利用对偶图性质，遍历已经得到的 Delaunay 三角剖分的每个面，做出其 Voronoi 图（不再细述其过程）。

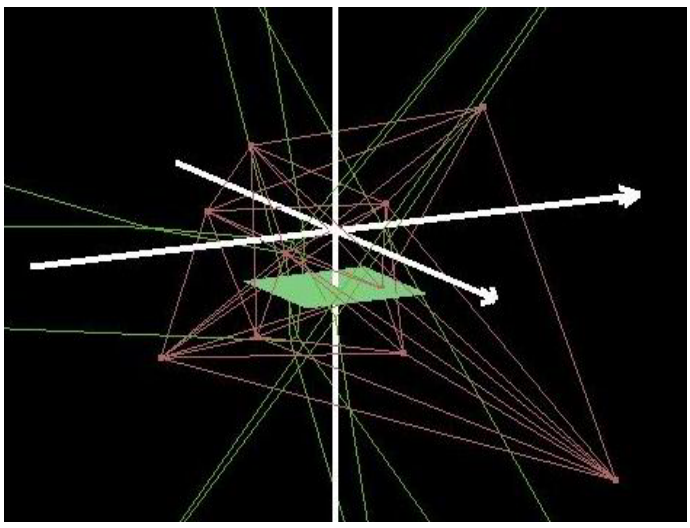
四、实验结果、问题及不足

本实验主要采用了 VC 做为开发工具，特别用 OPEN GL 来进行界面显示。

实验结果较好的显示了 3-D 点集的 Delaunay 三角剖分和 Voronoi 图。如下图所示：







在本实验的进行过程中，发现了原文的一个问题，就是[1]中提出的最优三角剖分的形状因子判断法其实不是 Delaunay 三角剖分的充要条件，有例外表明完全有的最优四面体不满足空球性质，只是在大多数情况下满足。因此在实现时后来我们实际上在一些地方加入了空球判断条件，这使得算法效率变低，组织上比较不归整。

本实验算法和方法有不少需要提高的地方。正如邓老师所说那样，我们采用表示几何的数据结构不是高效的通用结构；而且算法效率也不理想；以及没有对算法性能和大规模输入点集进行过测试保存，界面显示也有待于改进。由于时间所限，这些都有待于继续研究，但是我们一直都在努力来完成这个实验，而且也都有很大的收获。

五、致谢

最后要感谢邓老师在课堂上以及实验进程中给予我们的指导、关心和指正，对我们有很大帮助。在计算几何这门课中，我们收获了很多，这将成为自己美好的回忆。

参考文献

[1] Joe B. Construction of three - dimensional Delaunay triangulations usinglocal transformations[J]. Computer Aided Geometric Design ,1991 ,8 :123 - 142.

[2] 刘爽等，用随机增量局部转换算法实现三维点集的 **Delaunay** 三角剖分 计算机应用 **JuneV o,2l.0 2033**

[3] 刘雪娜 三维点集 **Voronoi** 图的算法实现 计算机辅助工程 **2006** 年 **3** 月，第 **15** 卷，第 **1** 期。