分类号：TP311　　单位代码：10422

密　级：　　学　　号：201834876

山东大学

# 硕士学位论文

（专业学位）

论文题目：　三维 Apollonius 图的快速计算

作者姓名　王培辉

学院名称　计算机科学与技术学院

专业学位名称　计算机技术

指导教师　辛士庆

合作导师

年　　月　　日

# 原创性声明和关于论文使用授权的说明

## 原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

论文作者签名：_____ 日期：_____

## 关于学位论文使用授权的声明

本人完全了解山东大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权山东大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。（保密论文在解密后应遵守此规定）

论文作者签名：_____导师签名:_____日期：_____

# 目　录

# CONTENTS

# 摘　要

Apollonius 图又名加权的 Voronoi 图，是传统的 Voronoi 图的一种扩展衍生结构。不同于传统的 Voronoi 图采用欧式距离来度量距离长度，Voronoi 图采用欧式距离减去每个种子点本身的权重这样一个附带权值的距离度量。Voronoi 图及其众多的衍生机构（例如 power 图，附加权值的 Voronoi 图，质心 Voronoi 图，质心 power 图等）在运动规划，材料科学，分子生物学，晶体学和计算机图形学等众多领域具有理论和实际意义。Apollonius 图与传统的 Voronoi 图和 Power 图相比，有一个显著的不同点，传统的 Voronoi 图和 Power 图的区域分界线都是直线，而 Apollonius 图的区域分界线是双曲线（或曲面）。虽然二维的 Apollonius 图的计算已经具有了比较鲁棒的算法，但是由于三维 Apollonius 图其复杂的拓扑结构以及其单元间分割界限为曲面的特性，三维的 Apollonius 图的计算并没有一个真正的可用于实际应用的鲁棒算法。

在这篇文章中，我们系统的分析了三维 Apollonius 图的拓扑结构，并且我们给出了一个鲁棒的快速计算三维 Apollonius 的算法。我们的算法主要包括顶点的定位，边的追踪，以及面的构建三部分，其中最为关键的一步便是寻找顶点的准确位置，我们动态的将正方体包围盒分割成一系列小的包围盒，来保证每个包围盒最多包含一个 Apollonius 图的顶点，以此来确定顶点的初步位置和候选的种子点，最终通过解方程的方式计算出 Apollonius 顶点精确的位置。最后，我们使用质心 Voronoi 图将 Apollonius 图的曲面离散为高质量三角形网格结构。

我们通过广泛的指标评估和实验来验证算法的有效性和鲁棒性，并且我们给出了一个有趣的应用：计算质心 Apollonius 图。我们这篇文章的主要贡献有：

（1）对三维 Apollonius 图的拓扑和几何特征进行系统分析，并讨论退化案例；

（2）提出一种用于计算三维 Apollonius 图的快速算法。关键思想是将一个初始的正方体包围盒自适应地细分为一组足够小的正方体包围盒，这样每个包围盒最多包含一个 Apollonius 顶点；

（3）提出了一种使用高质量的细分的三角形网格表示三维 Apollonius 图的

计算工具，可以为计算几何中的其他相关工作提供便利。

**关键字：** Apollonius 图，Voronoi 图，加权的 Voronoi 图，质心 Voronoi 图

# ABSTRACT

Apollonius diagrams, also known as additively weighted Voronoi diagrams, are an extension of Voronoi diagrams, where the weighted distance is defined by the Euclidean distance minus the weight. Voronoi diagrams and the variants of Voronoi diagrams which include power diagram, additively weighted Voronoi diagrams, Centroidal Voronoi Diagrams (CVT) and Centroidal Power Diagrams (CPD), have theoretical and practical applications in many fields, such as motion planning, material science, molecular biology, crystallography and computer graphics. The bisectors of Apollonius diagrams have a hyperbolic form, which is fundamentally different from traditional Voronoi diagrams and Power diagrams. Though robust solvers are available for computing 2D Apollonius diagrams, there is no robust solver for the 3D version since the latter is more complicated - the structure of a 3D Apollonius diagram has a higher combinatorial complexity and each bisector is a curved hyperbolic surface.

In this paper, we systematically analyze the structural features of 3D Apollonius diagrams, and then develop a fast algorithm for robustly computing Apollonius diagrams in 3D. Our algorithm consists of vertex location, edge tracing and face extraction, among which the key step is to adaptively subdivide the initial large box into a set of sufficiently small boxes such that each box contains at most one Apollonius vertex. Finally, we use centroidal Voronoi tessellation (CVT) to discretize the curved bisectors with well-tessellated triangle meshes.

We validate the effectiveness and robustness of our algorithm through extensive evaluation and experiments. We also demonstrate an application on computing centroidal Apollonius diagram. This paper makes the following contributions:

1. A systematical analysis of the topological and geometric features of 3D Apollonius diagrams and a full discussion on degenerate cases.

2. A fast algorithm for computing 3D Apollonius diagrams. The key idea is

to adaptively subdivide an initial large box into a set of sufficiently small boxes such that each box contains at most one Apollonius vertex.

3. A computational tool for representing 3D Apollonius diagrams using well-tessellated triangle meshes that can benefit the downstream applications in digital geometry processing.

**Keywords:** Apollonius diagram; power diagram; the additively weighted Voronoi diagrams; centroidal Voronoi tessellation (CVT)

# 第 1 章 绪论

## 1.1 课题的研究背景

Apollonius 图又名加权的 Voronoi 图，是传统的 Voronoi 图的一种扩展结构。Voronoi 图以俄罗斯数学家格奥尔吉·沃罗诺伊[12]的名字命名，是一种经典的空间分割算法，也被称作狄利克雷镶嵌或泰森多边形。Voronoi 图及其衍生图在许多领域具有重要的理论和实际意义，例如运动规划[3][4][5][31]，材料科学[29]，分子生物学[60][61]，晶体学[17]和计算机图形学[9][50]等。Voronoi 图通常是由 $n$ 个种子点（$\boldsymbol{x_i}$，$i = 1,2,\ldots,n$）根据点到种子点距离最近原则对所处空间进行区域划分的一种空间分割算法，传统的 Voronoi 图中,种子点通常是一个只有空间位置信息的普通空间内的点。举一个简单的例子，对于一个由 n 个二维种子点构成的二维的 Voronoi 图，它将一个二维的平面区域分割成 n 个互不联通的平面区域，每个种子点分别控制一个区域，对于某个种子点所控制的区域里的任意一点，到该种子点的欧氏距离小于或等于到其他种子点的欧式距离。我们将划分好的每个区域叫做 Voronoi 单元（Voronoi cell）。相邻的两个区域之间由一根直线段作为分割线，我们将这个直线段称为 Voronoi 边，Voronoi 边上的任意一点到最近的相邻两个种子点的欧式距离是相等的。Voronoi 边的交点我们称为 Voronoi 顶点。每个 Voronoi 顶点周围有三个及以上的 Voronoi 单元，每个 Voronoi 顶点到周围最近的相邻种子点（一般是三个，退化情况下会出现三个以上）的距离是相等的。

传统的 Voronoi 图有很多的变体扩展结构。首先，Voronoi 图的种子点可以是点，线段，曲线，圆，球或者其他的凸形状。其次，空间中任意一点到种子点之间的距离度量具有多种形式。通常使用的距离度量包括：传统的 Voronoi 图中的欧式距离$||x - x_i||_2$，$L_p$-Voronoi 图[47]中的闵可夫斯基距离$||x - x_i||_p$，Power 图[15]中的加权平方距离$||x - x_i||_2^2 - w_i$，以及 Apollonius 图[26]中的加权距离$||x - x_i||_2 - w_i$。事实上，如果函数在$x_i$处可以取得最小值，这些距离函数可以是关于欧式距离的任意凸函数[51]。最后，同样可以通过使用测地距离来代替欧式距离[46]，把传统的 Voronoi 图从欧式空间推广到曲面空间（例如，二维的黎曼表

面）。

总之，Voronoi 图及其相关的衍生推广结构具有许多的实际应用。例如，在地理信息系统（GIS），研究人员通过 Voronoi 图对空间进行分析，以助于地图创建和目标定位[4][5]。在计算几何中，质心 Voronoi 图（CVT）被使用在网格的优化上，一方面可以使种子点分布更加均匀，另一方面可以使网格质量得以提高。在最优传输中，Power 图可以表示最优传输的最终结果。在电力系统中[1]，可以利用 Apollonius 图设计变电站的位置和功率，来尽可能的使每个变电站覆盖其最大的区域，提高变电站的利用率，降低成本。

我们在本文中主要研究附加权值的种子点 $(x_i, w_i), i = 1,2,\dots,n$ 在三维欧式空间中使用加权距离$||x - x_i||_2 - w_i$生成的 Voronoi 图，也就是 Apollonius 图（又称为加权的 Voronoi 图）。通常情况下 Apollonius 的种子点可以被看作是圆（在二维的欧式空间）或者球（在三维的欧式空间中）。因此，许多学者又把 Apollonius 图称为是圆（球）的 Voronoi 图。图 1-1 我们展示了 Apollonius 的二维和三维结构。

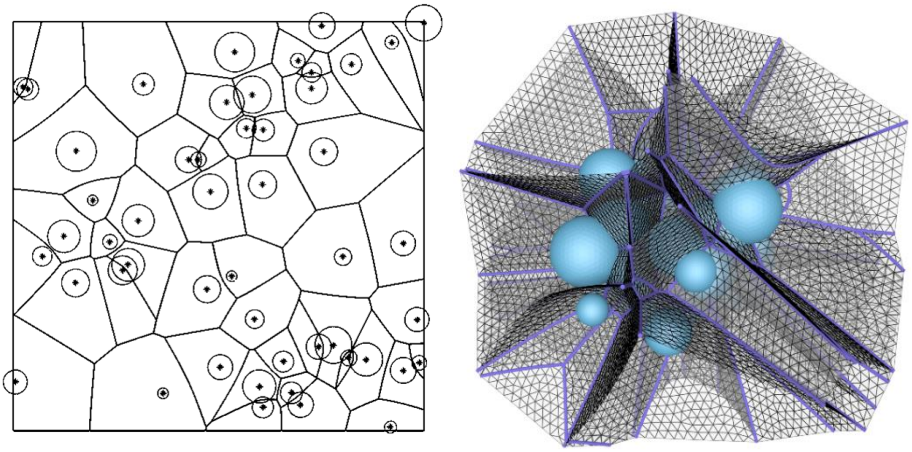


图 1-1 二维 Apollo 图（左）和三维 Apollonius 图（右）

## 1.2 课题研究意义和解决的主要问题

三维 Apollonius 图在流体，固体[58]和蛋白质[44][60]等大型结构的空间分析和可视化方面有着重要的应用。因此一个可以实际应用的 Apollonius 图的求解算法是迫切需要的。虽然此前关于传统 Voronoi 图的研究已经有了相当长的历史，并且成熟的算法和相关的文献也是数不胜数，但是关于 Apollonius 图的研究工作却并不多，并且在为数不多的 Apollonius 图的研究工作中关于三维的 Apollonius 图

的文献和算法更是少之又少。

在已有的 Apollonius 图的研究工作中，二维的 Apollonius 的计算已经拥有了比较实用和鲁棒的算法，被收录在计算几何算法库 CGAL[65]中。但是由于三维 Apollonius 图其复杂的拓扑结构以及其单元间分割界限为曲面的特性，三维的 Apollonius 图的计算并没有一个真正的可用于实际应用的鲁棒算法。1999 年，Will[61]研究了三维 Apollonius 图（文章中是三维球的 Voronoi 图），他发现三维 Apollonius 的每个空间单元都有$O(n^2)$的组合复杂度，并且他使用较低级的包络算法来提取单个 Apollonius 图的空间单元，其提取单个空间单元的算法的时间复杂度为$O(n^2)logn$。Will 提出的算法显然并不能满足实际的 Apollonius 的计算需求。2002 年，Boissonnat 和 Karavelas[19]在其研究工作中介绍了在高维空间中，Apollonius 图，power 图以及球形的凸包结构之间的关系。他们通过求解球的凸包结构来推测 Apollonius 图的结构。随后，自 2004 年以来，Kim 以及他的合作者[38][39][40][41][42][43][44]在三维 Apollonius 图的计算和应用问题上做了大量的研究工作，据我们所了解的知识，Kim 等人提出的区域扩充算法[42]是目前为止最为前沿且在理论中目前运行最快的算法。但是由于其并没有公布算法源码，我们也无法得知算法的可靠性。最后，除了以上提到的各种顺序算法以外，基于 CPU 加速的并行算法[36][48]也越来越受到研究人员的关注。

从上述的研究工作可以看出，少量研究人员已经为三维 Apollonius 图的计算做了前期的研究，也给我了我们一些可以借鉴的经验。在这篇文章中我们在前人的理论基础上我们希望可以解决以下问题：

（1）归纳总结已有的关于三维 Apollonius 图的相关工作；

（2）系统的分析三维 Apollonius 图的空间拓扑结构以及其独有的特性

（3）提出一个可用于实际的鲁棒算法。

## 1.3 本文的主要工作

在本篇文章中，首先我们系统的整理了三维 Apollonius 图的相关工作研究，包括 Voronoi 图，power 图，三维 Apollonius 图，质心 Voronoi 图以及质心 power 图。然后我们重点论证了三维 Apollonius 的一些良好特性并系统的分析了三维 Apollonius 图的拓扑结构总结了三维 Apollonius 的顶点，边，面和单元空间（cell）

之间的关系。最后，我们给出了一个鲁棒的快速计算三维 Apollonius 的算法。并且通过广泛的评估和实验来验证算法的有效性和鲁棒性，为了体现我们算法的实际意义，我们在设计了一个有趣的应用：计算质心 Apollonius 图。

文章的算法主要包括顶点的定位，边的追踪，以及面的构建三部分，其中最为关键的一步便是寻找三维 Apollonius 图顶点的准确位置，我们通过动态的正方体包围盒分割技术和类 Dijkstra 扫描技术来寻找候选种子点点集达到确定顶点的初步位置的目的，通过解方程组计算出顶点精确的位置。然后通过边追踪的方式联通所有的 Apollonius 顶点形成 Apollonius 边。最后，我们使用质心 Voronoi 图（CVT）来生成一个高质量的三角网格结构，使用该三角网格结构来表示 Apollonius 图的曲面。

本篇文章我们的主要贡献有：（1）对三维 Apollonius 图的拓扑和几何特征进行系统分析，并讨论退化案例；（2）提出一种用于计算三维 Apollonius 图的快速算法。关键思想是将一个初始的正方体包围盒自适应地细分为一组足够小的正方体包围盒，这样每个包围盒至多包含一个 Apollonius 顶点；（3）提出了一种使用细分的三角形网格表示三维 Apollonius 图的计算工具，可以为计算几何中的其他相关工作提供便利。

## 1.4 本文的组织结构

第 1 章绪论，主要阐述关于三维 Apollonius 图的研究背景，并介绍了当前有关该工作国内外的研究现状，使读者了解我们研究该工作的意义，同时介绍了本文要解决的实际问题以及我们完成的主要工作。

第 2 章相关工作。此章节我们系统的总结了与我们课题的相关工作，包括 Voronoi 图，Power 图，三维的 Apollonius 图以及质心 Voronoi 图和质心 Power 图。在此章节中我们对后续用到的专有章节做了系统的定义。

第 3 章定理性质。本章主要详细介绍三维 Apollonius 图的一下良好性质，我们通过定理和证明的方式给了详细的介绍。

第 4 章 Apollonius 图的拓扑结构。本章主要详细的介绍了 Apollonius 的拓扑结构，从顶点，边，面层层递进，由下而上系统的分析了 Apollonius 的拓扑结构。让读者对 Apollonius 图的顶点，边，面直接的关系有一个直观的理解。

第 5 章算法流程与实现。本章主要介绍针对解决三维 Apollonius 图的计算问题提出的算法。首先介绍了整个算法的流程框架。然后，按照顶点，边，面的顺序，依次介绍算法在每个部分的实现细节和流程。

第 6 章实验结果与对比。本章首先展示了我们算法的实验结果，然后分析了本文的算法的时间代价。最后通过可扩展性和网格的质量展现了本文算法独有的优势。

第 7 章应用部分-质心 Apollonius 图。为了体现本文算法的实际效果，在本章将算法用于实际的应用之中，将 Apollonius 图推广到质心 Apollonius 图。

第 8 章对本文工作做了全面总结，总结了本文算法的优势与不足，以及对后续工作的展望。

# 第 2 章 相关工作

Apollonius 图是传统 Voronoi 图的推广衍生结构。传统的 Voronoi 图的推广衍生结构还包括 Power 图，质心 Voronoi 图和质心 Power 图等。这些结构之间即存在着共同点又有各自的独特性质。关于传统的 Voronoi 图及其衍生结构，在过去的时间中，研究人员做了大量的相关工作。基于我们前期对本文相关工作的详细调研，在本章节中系统的总结了国内外的相关工作。

## 2.1 Voronoi 图

在实际的生活或者研究工作中，有时我们需要对空间进行区域划分来完成实际的任务。Voronoi 图是一种经典的空间区域划分算法。如同 2-1 所示，通过给定的 n 个二维的点，利用距离最近的原则，将所在的空间划分成互不联通的空间区域。我们将分割该区域的点叫做种子点。将划分好的空间区域称为 Voronoi 单元区域（cell）。相邻的两个 Voronoi 单元之间的边界被成为 Voronoi 边。所有 Voronoi 边的交点称其为 Voronoi 顶点。一个完整的 Voronoi 图是被种子点分割而，由顶点，边，单元区域组成的空间结构。

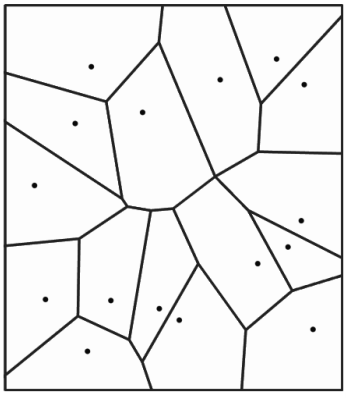


图 2-1 Voronoi 图

通常 Voronoi 图具有许多良好的特性，使其在实际应用中备受欢迎，例如，每个 Voronoi 单元区域都是一个凸区域，每个单元区域内的任意点到某个特定的种子点会存在比到其他种子点更近的欧式距离，我们称该种子点控制了该 Voronoi 单元区域。每个 Voronoi 单元区域只能被一个种子点控制，每个种子点也只能控制一个 Voronoi 单元区域，他们之间的关系是以一一对应的；相邻两个

Voronoi 单元区域之间的边界（也就是 Voronoi 边）是一条直线段或者射线。直线上任意一点到控制两侧 Voronoi 单元区域的种子点的欧式距离是相等的，因此我们也称两个种子点可以控制一条 Voronoi 边（二维空间中）；每个 Voronoi 顶点周围延伸三条 Voronoi 边和三个 Voronoi 单元区域，Voronoi 顶点到其周围单元区域内种子点的欧式距离相等。每个 Voronoi 单元区域可表达为式 2-1。

$$\Omega_i = \{x \in \Omega |\ ||x - x_i|| \leq ||x - x_j||, i \neq j\}, \quad (2\text{-}1)$$

这里$\Omega_i$指的是种子点$x_i$所控制的区域。

由于 Voronoi 图具有良好的几何特性，它已被广泛应用于许多科学和工程领域[16][55]，包括计算机图形学，图像处理，机器人导航，计算化学，材料科学，气候学等。Voronoi 图的计算有着成熟的算法，总结起来大概可以分为四类：分治法，增量构造，扫描线算法和间接方法。Shamos 和 Hoey[56]通过递归的思想实现了一种简单的算法分治法，我们称这类算法为分治法，虽然其思路简单，但是很难动态更新。1978 年 Green 和 Sibson[34]通过不断的动态的向已经生成的 Voronoi 图增加站点来实现动态的生成 Voronoi 图，此类算法我们称为增量构造法，该方法可以动态增加删减种子点，在实际使用过程中更加方便灵活。1987 年，Fortune[27]提出了经典的扫描线算法，成为教科书级别的算法。最后一种是间接法。间接法的主要思想便是不直接计算 Voronoi 图，而是计算其对偶结构—Delaunay 三角剖分[22]. Delaunay 三角剖分在计算几何中也是一个非常重要的课题。当然 Delaunay 三角剖分也可以通过扩展方法[28],因此也能达到动态更新 Voronoi 图的目的。随着人们对计算速率要求的提高，并行化算法越来越受到广大研究人员的关注，包括随机化，分治法[20]和基于图形硬件的实现[2][35][52][63]。

## 2.2 Power 图

类似于 Voronoi 图，power 图作为 Voronoi 图的一个典型的扩展衍生结构也是一种非常重要的空间剖分工具。一个完整的 power 图同样是由种子点，power 顶点，power 边和 power 单元区域构成。同 Voronoi 图一样，每个 power 单元区域都是凸区域，每个 power 边也是直线段或者射线。任意位于 power 单元区域，power 边和 power 顶点上的点与种子点之间的距离关系也与 Voronoi 图相似。但不同于 Voronoi 图是，首先 power 图的种子点在记录点的空间位置的同时，每个

种子点附带一个权重，虽然种子点与 power 单元区域的关系依旧是一一对应的，但是每个种子点并不一定在其所控制的 power 单元区域内部。在 power 图中距离度量发生了改变，从传统的欧式距离引申为加权平方距离$||x - x_i||_2^2 - w_i$。一个 power 图的数学表示形式如式 2-2 所示。

$$\Omega_i = \{x \in \Omega | \, ||x - x_i||^2 \, - w_i \leq \, ||x - x_j||^2 \, - w_j \, , i \, \neq j\}, \quad (2\text{-}2)$$

这里$\Omega_i$指的是种子点$x_i$所控制的区域，$w_i$ 是种子点$x_i$权重的平方值。

当所有的种子点的权重相等的时候，此时的 Power 图就会重新退化为传统的 Voronoi 图[57]。Power 图的研究历史可以追溯到二十世纪八十年代[15][37]。1996 年，Gavrilova[32]等人根据平面上的 Voronoi 图重构出平面上的 Power 图。并且后来证实，Power 图有助于解决半离散的最优传输问题[14]。目前构建 Power 图比较流行的算法依然是通过它的对偶图正则三角化来构建 Power 图[30]。由于度量空间的不同，计算网格曲面上的 Power 图要困难得多。中科院的严冬明博士提出了限制在曲面上的 Power 图，然而它并不是真正的测地 Power 图，在站点较少或者曲面细节较多的情况下，它与测地 Power 图相距甚远。再者，窦志扬[25]等人提出与用内侧球构建的 Power 可以重建模型的原始表面。

## 2.3 三维 Apollonius 图

Apollonius 图，又名加权的 Voronoi 图，同样是 Voronoi 图的一种衍生。不同于 Voronoi 图和 Power 图，Apollonius 图具有更复杂的拓扑结构，并且其相邻两单元区域的边界不再是平面，而是双曲面的一支，并且可能包含不连通的区域。由于其边界由双曲线和曲面构成，其每个单元区域不具有严格的凸区域的特征。每个单元区域虽然可以被一个种子点控制，但是可能存在了有的种子点并不能控制一个单元区域的情况，这类种子点，我们称其为隐藏点。隐藏点出现的原因主要是在 Apollonius 图中的距离度量发生了变化，是传统的欧式距离减去种子点自身的权重。当所有的种子点的权重为正值时，我们可以将所有的种子点看作是空间中的一些小球。假如我们将种子点的位置信息当作球心，将种子点的权重当作球的半径，空间中任意一点到种子点的距离，就是到球的表面的距离。隐藏点就相当于一个被其他球包裹在里面的球，永远无法为空间中任意一点提供最近的带权距离。在 Apollonius 图中，每个单元区域内的任意一点到控制该单元区域的种

子点的带权距离小于等于到其他种子点的带权距离，如式 3-3 所示，

$$\Omega_i = \{x \in \Omega | ||x - x_i|| - w_i \leq ||x - x_j|| - w_j, i \neq j\},（3-3）$$

这里$\Omega_i$指的是种子点$x_i$所控制的单元区域，$w_i$ 是种子点$x_i$的权重。

二维 Apollonius 图的边可以通过投影一组扎根的圆锥体的下部包络得到[13][60][61]。2006 年，Emiris 和 Karavelas[26]讨论了计算二维 Apollonius 图的必要前提。并且在 CGAL 算法库中已经收纳了鲁棒的计算二维 Apollonius 的算法。2018 年，王文嵩[59]等人通过使用二维的 Apollonius 图技术追踪出高质量的测地等高线。相比于二维的 Apollonius 图，我们对三维的 Apollonius 图的计算更加感兴趣。对于三维的 Apollonius 图，只有很少的研究工作，在这其中，Kim 以及他的合作者在三维 Apollonius 图的计算问题上做了大量的研究工作[38][39][40][41][42][43][44]，据我们了解其中 Kim 等人提出的区域扩充算法[42]是目前为止最为前沿以及在实际应用中目前运行最快的算法。计算三维 Apollonius 图的主要挑战是两方面：（1）尽管我们可以通过检查在$n^4$个备用种子点找到真正的 Apollonius 图顶点，但是许多种子点在计算的过程中都一一翻了四倍，在计算上消耗的代价太大。如何设计一个接近线性的时间复杂度的算法才是重中之重;（2）Apollonius 图的等分线具有双曲线的形式，这对一般的数字几何处理任务并不友好。在本文中我们的动机便是克服这两个困难，设计一个鲁棒的解决三维 Apollonius 的算法。

## 2.4 质心 Voronoi 图和质心 Power 图

质心 Voronoi 图(CVT)[23]是一种特殊的 Voronoi 图，它的每一个站点都与对应 Voronoi 单元区域的质心重合。相对于给定的密度函数，质心 Voronoi 图生成非常均匀的点分布，是计算最佳空间剖分的关键技术[18][49][64]因此被广泛应用于图像处理、数据压缩、数值差分和网格化[11]等问题中在数学上，质心 Voronoi 图可以通过最小化式（3-4）所示的凸函数求得：

$$E(X) = \sum_{i=1}^{n} \int_{\Omega_i} ||x - x_i||^2 \rho(x) d\sigma,（3-4）$$

其中$\rho(x)$是定义在区域上的密度函数。在过去的研究中，普遍的计算质心 Voronoi 图的方法主要有两种，由于几何上的直观性，Lloyd[21]算法一度是求解质

心 Voronoi 图的主流算法，2009 年，微软亚洲研究院的刘洋博士等基于拟牛顿法[50]提出了求解质心 Voronoi 图的超线性收敛算法，并成功应用于网格质量优化这一重要问题中。中科院的严冬明博士等[66]提出了限制在曲面表面的质心 Voronoi 图，进一步提高了网格质量优化的效率。此后，质心 Voronoi 图在数字几何领域被广泛研究。法国 INRIA 的 Levy 教授和微软研究院的刘洋博士[67]提出了基于 p 范数的质心 Voronoi 图并展示了它在四边形网格化中的应用。山东大学的吕琳教授等[68]提出基于蒙特卡罗和模拟退火算法来进一步优化质心 Voronoi 图。新加坡南洋理工大学的王晓宁博士等[69]提出了计算测地质心 Voronoi 图的快速算法。最近，清华大学的叶畋宇博士等[7]提出了基于局部优化的思想计算质心 Voronoi 图的方法。

当进一步施加质量约束时，质心 Voronoi 图可以推广到质心 Power 图(CPD)。与质心 Voronoi 图相比，质心 Power 图的每个站点不但在其控制的区域的质心位置，而且每个站点的重量必须满足给定的质量约束。由于质心 Power 图有能够精确控制容量的优点，被应用于高质量蓝噪声采样，信息可视化，机器人路径规划等许多重要问题。该问题采用不同的数学描述方法，人们提出了各种各样的算法。Balzer 等[70]提出在 Voronoi 图剖分中添加面积约束，通过极小化当前面积与给定面积的差值优化站点的位置。然而，他们提出的算法时间复杂度高而且收敛很慢。厦门大学的陈中贵副教授等[6]从能量的观点改写了目标函数，给出了相应的梯度表达，提高了计算速度，并应用于蓝噪声合成。合肥工业大学的郑利平教授等[8]通过为每个站点调整其邻居站点的权值，优化该站点通知区域的质心位置，并按相同比例缩放它统治的区域，最终优化出所需的质心 Power 图。加州理工学院的 de Geos 等[24]发现带有面积约束的采样问题可通过最优传输理论进行诠释，通过交替优化站点的位置和权值，给出了一阶收敛的优化算法并应用于蓝噪声的生成。辛士庆及合作者[62] 基于包络定理提出了超线性收敛的优化算法，并在更多的应用场合中展示了质心 Power 图的有用性。合肥工业大学的郑利平教授等[10]在 de Goes 算法的基础上，提出一种针对所有变量一体优化的策略，以达到加速质心 Power 图生成的目的。在已有的文献中研究人员只把 作为核函数。在本文中我们使用$||x - x_i||$代替$||x - x_i||^2$作为新的核函数，以此将质心 Voronoi 图推广到质心 Apollonius 图。

## 2.5 本章小结

本章系统的介绍了关于 Voronoi 图及其衍生图的相关定义，以及国内外学者已经做过的相关研究工作。围绕着 Voronoi 图，我们分析了其衍生结构与传统的 Voronoi 图的异同，并且重点介绍了关于 Apollonius 图的研究现状，以及我们研究 Apollonius 图的动机。了解完 Voronoi 图及其衍生图的相关基础知识，我们将在下面的章节详细的介绍我们本文的重点 Apollonius 图。

# 第 3 章 定理性质

在本章中，我们将系统的讨论了三维 Apollonius 图的一些基本性质。这些基本性质为进一步了解 Apollonius 图顶点，Apollonius 图边和 Apollonius 图单元区域的组合结构奠定了基础。在此章节中我们部分讨论的性质是在某些已有的文献的基础上做出的延伸[33][43]。

## 3.1 基本性质

我们给定 n 个三维的种子点$\{x_i\}, i = 1,2,3,\dots,n$，对于每一个种子点我们给予一个对应的权值$w_i$。接下来我们就在此基础上详细的讨论一般 Apollonius 图的基本性质。

如图 3-1 所示，种子点$x_i$在种子点$x_j$的影响下成为了隐藏点，因为种子点$x_i$无法为空间内的任意一点$x$提供最近的带权距离：

$$||x_i - x|| - w_i \geq ||x_j - x|| - w_j, \text{(3-1)}$$

以此可见，对于空间中的任意一点种子点$x_j$都可以提供比种子点$x_i$更近的加权距离，因此种子点 $x_i$无法控制一块属于自己的区域，在整 Apollonius 图的构建过程中没有做出任何贡献，我们称其为隐藏点。在定理 1 中，我们详细的论证了隐藏点出现的充分和必要条件。

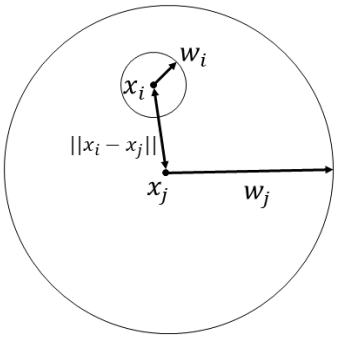



图 3-1 隐藏点

【定理 1】一个 Apollonius 图结构不会出现隐藏点当且仅当对于任意一对种子点$x_i$和$x_j$，他们都必须满足：

$$||x_i - x_j|| > |w_i - w_j|。(3\text{-}2)$$

【证明 1】假设对于任意一对种子点$x_i$和$x_j$（$i \neq j$），满足$||x_i - x_j|| > |w_i - w_j|$。那么我们就可以推断种子点$x_i$至少可以控制其本身（当$x = x_i$的时候），

$$||x_i - x_i|| - w_i < \min\{||x_j - x_i|| - w_j,\ i \neq j\}。(3\text{-}3)$$

因此，对于种子点$x_i$，存在一个非空点集（至少包括$x_i$本身），其内部的点到种子点$x_i$的带权距离小于到其他种子点的距离。这就证明了$x_i$不是隐藏点。反之亦然。

在接下来的其他性质的证明中，为了方便起见，我们假设我们提供的这一组种子点中不存在隐藏点（尽管如此，我们的算法是支持隐藏点的计算的）。进一步，我们排除掉空间中两类种子点的偶然排列情况：

（1） 对于任意三个种子点$(x_i, x_j, x_k)$并不排列在一条直线上；

（2） 对于任意四个种子点$(x_i, x_j, x_k, x_l)$并不分布在同一平面上。

首先，对于每一个不是隐藏点的种子点，其所控制的区域都是一个星状结构。通俗的讲，就是每个非隐藏点的种子点在其所控制的区域的内部。下面我们对这个定理进行详细的证明。

【定理 2】给定 n 个三维的种子点$\{x_i\}, i = 1,2,3,\dots,n$，对于每一个种子点给予一个对应的权值$w_i$。那么每个种子点所控制的区域是对于该种子点是星状结构。

【证明 2】假设点$x$是种子点$x_i$所控制区域内的任意一点，那么我们可以得到

$$||x_i - x|| - w_i < \min\{||x_j - x|| - w_j,\ i \neq j\}。(3\text{-}4)$$

对于任意的点$p \in \overline{xx_i}$，$||x_i - p|| - w_i = ||x_i - x|| - w_i - ||\overline{xp}||$。考虑到对于任意 $i \neq j$，$||x_j - p|| - w_i \geq ||x_j - x|| - w_j - ||\overline{xp}||$，我们可以得到

$$||x_i - p|| - w_i < \min\{||x_j - p|| - w_j, i \neq j\}，(3\text{-}5)$$

由此我们可以看到每个种子点所控制的区域是对于该种子点是星状结构。

【定理 3】给定两个种子点$x_1$和$x_2$，分别赋予权重$w_i$和$w_j$，并且满足条件$0 < w_i - w_j < ||x_1 - x_2||$。到种子点$x_1$和$x_2$具有相同带权距离的点集形成的是一个双曲面$\mathcal{H}$。假设我们把Π记作$x_1$和$x_2$的中垂面。如果我们把点$h \in \mathcal{H}$投影到中垂面$p \in \Pi$。 那么我们可以得到一个一一对应的投影关系。点$h \in \mathcal{H}$和投影点$p \in \Pi$的周围的无穷小的面积比为$\frac{|N\cdot(x_2-x_1)|}{||N||\cdot||x_2-x_1||}$， 这里 N 指的是点 h 在双曲面上的法向量$N = \frac{h-x_1}{||h-x_1||} - \frac{h-x_2}{||h-x_2||}$。图示见图 3-2。

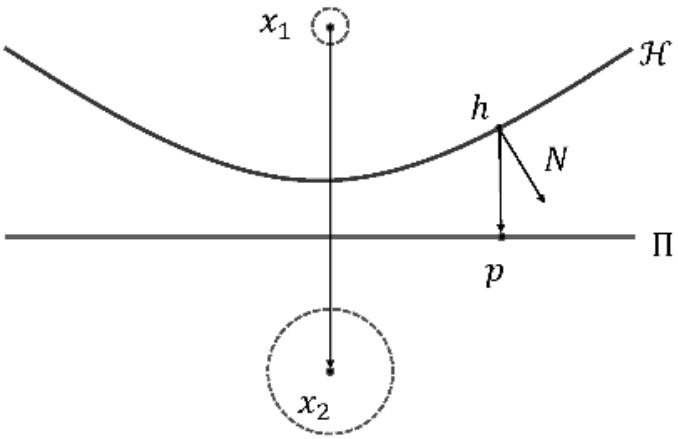



图 3-2 定理 3 图示

【证明 3】对于任意的点$h \in \mathcal{H}$，都满足$||x_1 - h|| - w_1 = ||x_2 - h|| - w_2$。其在中垂面上的投影点$p \in \Pi$可以被表示为

$$p = h + \lambda \cdot (x_2 - x_1)。(3\text{-}6)$$

因为$p$同样也位于$x_1$和$x_2$的中垂面上，因此我们可以得到

$$\left(p - \frac{x_2+x_1}{2}\right) \cdot (x_2 - x_1) = 0。(3\text{-}7)$$

将两个方程联立，我们可以得到

$$\lambda = \frac{\left(\frac{x_2+x_1}{2}-h\right)\cdot(x_2-x_1)}{||x_2-x_1||^2}。(3\text{-}8)$$

从中我们可以发现，对于曲面上的任意一点$h \in \mathcal{H}$，我们都可以在中垂面上找到它的投影点$p \in \Pi$。换言之，每个不同的$h$都会对应不同的投影点$p$。

另一方面，当我们将 $p$ 记作中垂面上的一点。那么 $h = p + \mu \cdot (x_1 - x_2)$ 便是双曲面上的一点，该点的位置是两个种子点 $x_1$ ，$x_2$ 的连线与其形成的双曲面的交点。$\mu$ 可以通过求解二次方程的根来得到，恰好只有其中一个有意义。

最后，可以从兰伯特的余弦定律快速获得面积比。

【定理 4】给定 n 个三维的种子点$\{x_i\}, i = 1,2,3,\dots,n$，对于每一个种子点给予一个对应的权值$w_i$。那么任意四个种子点确定的 Apollonius 图顶点的数量可能是 0,1,2。

【证明 4】假设点$(x,y,z)$是由四个种子点$x_1$，$x_2$，$x_3$，$x_4$确定 Apollonius 图的顶点，那么我们可知

$$\begin{cases}\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}-w_1=r\\ \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}-w_2=r\\ \sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}-w_3=r\\ \sqrt{(x-x_4)^2+(y-y_4)^2+(z-z_4)^2}-w_4=r\end{cases}, \quad (3\text{-}9)$$

我们解方程可得

$$\begin{cases}x=\alpha_1 r+\beta_1\\ y=\alpha_2 r+\beta_2\\ z=\alpha_3 r+\beta_3\end{cases}, \quad (3\text{-}10)$$

其中$\alpha_i$，$\beta_i$ 都是常数。我们将解带回方程组，我们可以得到一个关于$r$ 的一元二次方程，求解可得两个不同的解 $r_1$， $r_2$。如果将$r_1$， $r_2$带入原方程组可得两个解，并且满足到该四个种子点的距离小于到其他种子点的距离，那么就能同时得到两个 Apollonius 图的顶点；如果将$r_1$， $r_2$带入原方程组可得两个解，只有一个满足到该四个种子点的距离小于到其他种子点的距离，那么就能得到一个 Apollonius 图的顶点；否则无法得到 Apollonius 图的顶点。我们在图 3-3 种展示了 Apollonius 图顶点可能存在的情况。

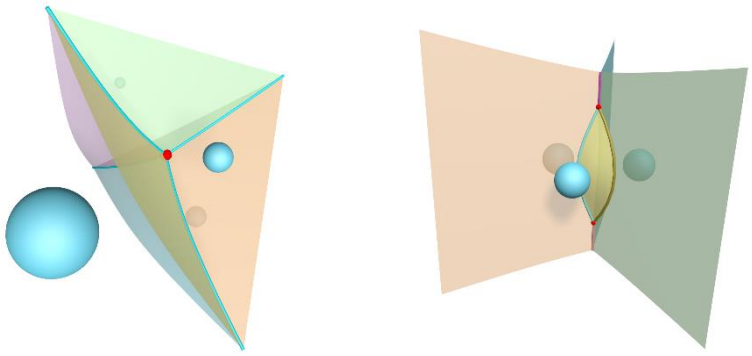


图 3-3 Apollonius 图的顶点结构

【定理 5】如图 3-4 所示，Apollonius 图的边存在两种类型，一种是两端无限延伸的曲线结构，一种是分别的环状结构。并且 Apollonius 图边上的任意一点$h$的切向量为$\left(\frac{h-x_1}{||h-x_1||}-\frac{h-x_2}{||h-x_2||}\right)\times\left(\frac{h-x_2}{||h-x_2||}-\frac{h-x_3}{||h-x_3||}\right)$.

【证明 5】假设点$(x,y,z)$是由三个种子点$x_1$，$x_2$，$x_3$确定 Apollonius 图的边上的一点，那么我们可知

$$\begin{cases}\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}-w_1=r\\ \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}-w_2=r\\ \sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}-w_3=r\end{cases}, \quad (3\text{-}11)$$

通过解方程可得

$$\begin{pmatrix}x\\y\\z\end{pmatrix}=r\begin{pmatrix}\alpha_1\\\alpha_2\\\alpha_3\end{pmatrix}+\begin{pmatrix}\beta_1\\\beta_2\\\beta_3\end{pmatrix}+t\gamma, \quad (3\text{-}12)$$

其中$\alpha_i$，$\beta_i$ 都是常数，$t$ 是一个从负无穷到正无穷的参数。

## 3.2 本章小结

本章节我们介绍了 Apollonius 图的隐藏点的概念，分析了 Apollonius 单元区域的形状特征，验证了 Apollonius 图的顶点的个数以及 Apollonius 图边的形状特征。本章中总结的 Apollonius 图的基本性质为下面我们算法的设计提供了理论的保证。

# 第 4 章 Apollonius 图的拓扑结构

在本章节中，我们详细讨论 Apollonius 图的拓扑结构。以下所有的内容我们设定 Apollonius 图没有发生退化情况，我们将依次按照 Apollonius 顶点，Apollonius 图的边以及 Apollonius 图的面来介绍整个 Apollonius 图的结构关系。

## 4.1 Apollonius 顶点

正如上文提及，每一个 Apollonius 图的顶点都是由四个不同的具有权重的种子点（$x_1$，$w_1$），（$x_2$，$w_2$），（$x_3$，$w_3$），（$x_4$，$w_5$）所形成。如图 4-1 所示，从一个 Apollonius 图的顶点出发，存在着六个 Apollonius 图的面，四个 Apollonius 图的边 $e$。例如，$x_1, x_2, x_3$ 可以决定一条 Apollonius 图的边。基于定理 5，Apollonius 图的边 $e$ 在顶点 $v$ 处的切线方向如式 4-1 所示，

$$T = \left(\frac{v - x_1}{||v - x_1||} - \frac{v - x_2}{||v - x_2||}\right) \times \left(\frac{v - x_2}{||v - x_2||} - \frac{v - x_3}{||v - x_3||}\right). \quad (4\text{-}1)$$

但是接下来面临的问题便是如何判断切线的方向与 Apollonius 的边 $e$ 的走向是相同的还是相反的。为了解决这个问题，我们构建了一个能量方程式 3-13，

$$F(t) = ||x_4 - (v + t * T)|| - w_4 - ||x_1 - (v + t * T)|| + w_1, \quad (4\text{-}2)$$

函数满足$F(0) = 0$，其一阶导函数如式 4-3 所示，

$$F'(t) = \frac{(v + t*T - x_4)\cdot T}{||x_4 - (v + t*T)||} - \frac{(v + t*T - x_1)\cdot T}{||x_1 - (v + t*T)||}, \quad (4\text{-}3)$$

根据其一阶导函数我们可以得到式 4-4，

$$F'(0) = \left(\frac{v - x_4}{||v - x_4||} - \frac{v - x_1}{||v - x_1||}\right) \cdot \left(\left(\frac{v - x_1}{||v - x_1||} - \frac{v - x_2}{||v - x_2||}\right) \times \left(\frac{v - x_2}{||v - x_2||} - \frac{v - x_3}{||v - x_3||}\right)\right) \neq 0, \quad (4\text{-}4)$$

因为这三个向量每一个都是曲面上点$v$ 的法向量，因此他们三个是不可能共面的（除非当形成这个 Apollonius 图顶点的四个种子点是在同一个平面上的）。当$F'(0) > 0$ 时，Apollonius 边 $e$ 的延伸方向一定与切向量 $T$ 是相同的。当 $F'(0) < 0$ 时，我们将根据切向量 $T$ 的反方向来追踪 Apollonius 图边 $e$ 的延伸法向。在图 4–1 中，我们标注了点 $v$ 处的四个切方向向量。

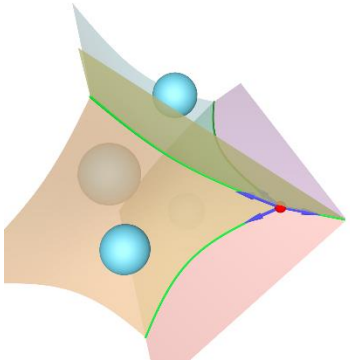


图 4-1 典型的 4 岔口的 Apollonius 的顶点

## 4.2 Apollonius 边

事实上一共有四种类型的 Apollonius 图的边结构包括：（1）两边无限延伸的双曲线的一支;（2）没有任何 Apollonius 图顶点的环状结构;（3）一端是 Apollonius 图的顶点，一边无限延伸的顶点；（4）两端都是 Apollonius 图顶点的线段。

在图 4-2 中我们展示了前两种类型的边结构，后两种类型的结构可以通过裁剪拦截第一种类型的结构获得。对于第四种结构，我们假设 Apollonius 图的边连接着两个 Apollonius 图的顶点 $v_1$ 和 $v_2$。假设 Apollonius 顶点 $v_1$ 由种子点（$x_1^{(v_1)}, x_2^{(v_1)}, x_3^{(v_1)}, x_4^{(v_1)}$）决定形成，顶点 $v_2$ 由种子点（$x_1^{(v_2)}, x_2^{(v_2)}, x_3^{(v_2)}, x_4^{(v_2)}$）决定形成。从中我们可以看到，形成两个顶点 $v_1$ 和 $v_2$ 的两组种子点是相同的或者他们共享三个相同的种子点。当他们共享三个及以上的种子点的时候，那么就达到了生成 Apollonius 边的条件。

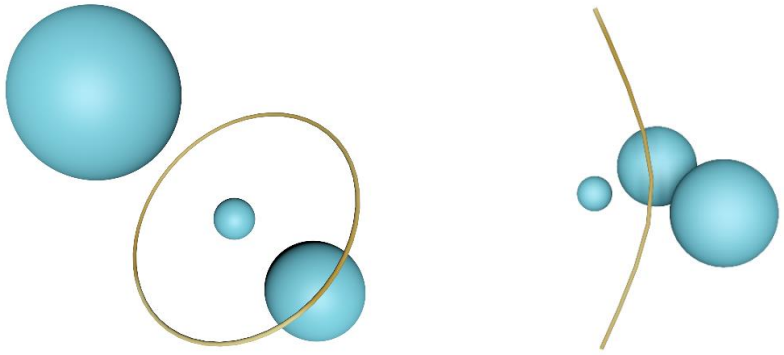


图 4-2 Apollonius 图的边结构

## 4.3 Apollonius 面

Apollonius 面最简单的类型是一个完整且无界的双曲面结构。但是在实际的应用情况下，每个 Apollonius 图的面基本都会存在着边界，这个边界是两个相邻的 Apollonius 图的面相交而成。但是他不同于传统的 Voronoi 图的是，每一个 Apollonius 图的面可能并不是简单的联通。可能在每个 Apollonius 图的面或者边上会存在着鼓包。具体的样例参考图 4-3.

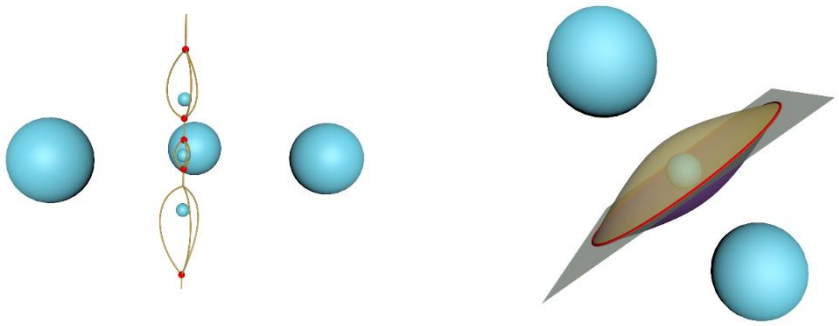


图 4-3 面或边上鼓包的案例

在传统的 Voronoi 图及其衍生的 Power 图中，两个顶点之间的连接边是唯一的。但是 Apollonius 图的边结构有些不同，下面的两个案例就可以很好的解释 Apollonius 图的边结构的特殊性：（1）两个 Apollonius 图的顶点可能有三条 Apollonius 图的边连接；（2）可能存在着一个封闭的 Apollonius 图的环边，没有任何 Apollonius 顶点的存在。

这两种案例的出现导致了 Apollonius 图面上鼓包和边上鼓包这两种特殊情况的发生。进一步这些可能会导致组合鼓包情况的发生的案例。如图 4-4 所示，从左到右分别展示了开放的 Apollonius 图的边上边的鼓包，封闭的 Apollonius 图的边上边的鼓包以及 Apollonius 图的面上面的鼓包现象。

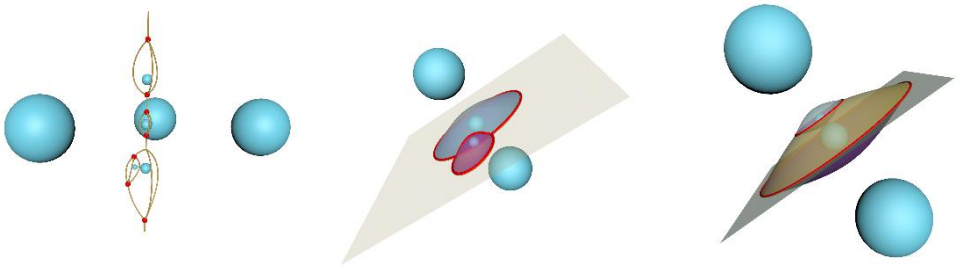


图 4-4 Apollonius 图的面或边鼓包的组合案例

## 4.4 本章小结

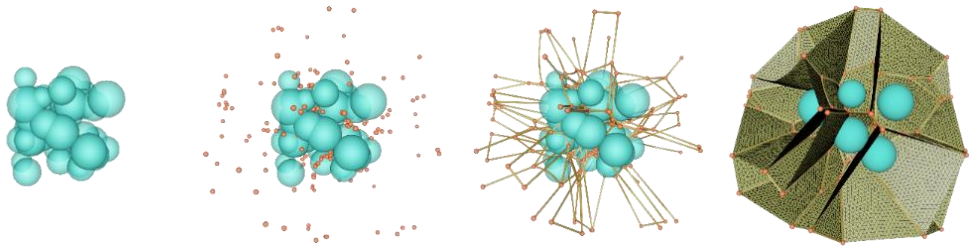
本章主要详细的介绍了 Apollonius 的拓扑结构，从顶点，边，面层层递进，由下而上系统的分析了 Apollonius 的拓扑结构。让读者对 Apollonius 图的顶点，边，面直接的关系有一个直观的理解。并且结合第三章 Apollonius 图的基本性质特征为我们第五章算法的设计提供了理论保证。

# 第 5 章 算法流程与实现

在本章中主要介绍我们的整个算法的实现流程以及实现细节，我们依次从算法的整个流程框架，Apollonius 图顶点的实现，Apollonius 图边的实现以及 Apollonius 图面的实现四个方面系统的介绍我们算法的主要步骤。

## 5.1 算法框架

给定 N 个带权重的种子点，我们的任务是通过加权距离来建立所有种子点之间的相似度。如图 5-1 所示，我们的算法依次从顶点定位，边的追踪，面的构建自下而上一步步完成整个 Apollonius 图的构建。首先，我们通过快速扫描技术实现 Apollonius 图的定位。然后，我们通过边的追踪技术将 Apollonius 图的顶点连接成 Apollonius 的边。最后，我们通过质心 Voronoi 图(CVT)的技术构建一个质量良好的三维网格结构来表示我们的 Apollonius 图的面。



（a）种子点　(b)顶点定位　(c)边追踪　(d)面构建

图 5-1 算法流程

## 5.2 Apollonius 顶点的实现

实现 Apollonius 顶点的计算的关键点在于找到可以形成 Apollonius 图顶点的种子点的点集。最简单的方法可以通过排列组合的方式依次排查 N 个种子点中每四个种子点的组合，将不符合形成 Apollonius 图定点的额种子点的组合排除。但是排列组合的种子点集可高达$O（n^4）$。其计算耗时非常高昂。因此如何快速的定位可以形成 Apollonius 图的顶点的点集是计算 Apollonius 图顶点的关键步骤。我们的策略是动态的将种子点所在的区域分割成小的包围盒，每个包围盒里

面存储着到该包围盒加权距离最近的种子点的集合，最后保证每个剩下的每个包围盒中，其存贮的种子点信息只能形成一个 Apollonius 图顶点。在本小节中我们将着重介绍我们如何利用包围盒寻找候选种子点点集，以及我们删除无用包围盒的规则，最后通过方程组求最终得顶点的位置。

### 5.2.1 寻找候选种子点集

我们通过动态分割包围盒和类 Dijkstra 扫描的方法快速排出无用包围盒，确定出候选种子点点集。具体操作如下：

首先，我们将整个包围盒分割成一系列大小相等的初始立方体包围盒。初始立方体包围盒的多少可以由用户指定。由于我们需要多轮分割，因此初始包围盒尽量多，但是每次分割过程中会产生大量无用的包围盒，因此我们又希望初始的包围盒尽量不要太小。因此经过两方面的权衡以及实验中的经验，我们一般将初始包围盒的数量设定在 10*10*10 大小。

其次，当我们初始包围盒已经准备完毕，接下来我们需要将所有的种子点根据加权距离的远近将其存储到最近的包围盒之中。整个分配过程类似于 Dijkstra 算法。对于每一个包围盒，我们希望我们记录的种子点是可以控制我们该包围盒部分区域的（具体可见章节 5.2.2）。接下来我们将所有的包围盒存放如一个优先队列之中。每次当从优先队列取出一个包围盒时，我们根据该包围盒中种子点的数量和过滤规则（该规则在章节 5.2.2 中详细论述）来判断该包围盒中是否可能存在着 Apollonius 图的顶点。（1）当该包围盒中种子点的数量少于 4，我们认为该包围盒中一定不存在 Apollonius 图的顶点。此时我们会删除掉该包围盒，不允许去参加下一轮的细分。（2）当包围盒中种子点的数量正好等于 4 时，我们认为该包围盒中可能存在着 Apollonius 图的顶点。我们通过解方程的方式求得到该四个种子点距离相等的点的信息。若求得的点正好位于该包围盒区域之内，则此点为 Apollonius 的顶点，若该点位于包围盒的区域之外，则该点可能不是 Apollonius 图的顶点。我们会删除掉该包围盒。（3）当该包围盒的种子点的数量大于 4 时，我们将该包围盒一分为 8.每个细分之后的子包围盒保留着之前父包围盒的种子点信息。并且我们将细分后的子包围盒放进存储包围盒的有限队列之中，当优先队列为空时，我们该算法结束。算法详细过程见图 5-2.

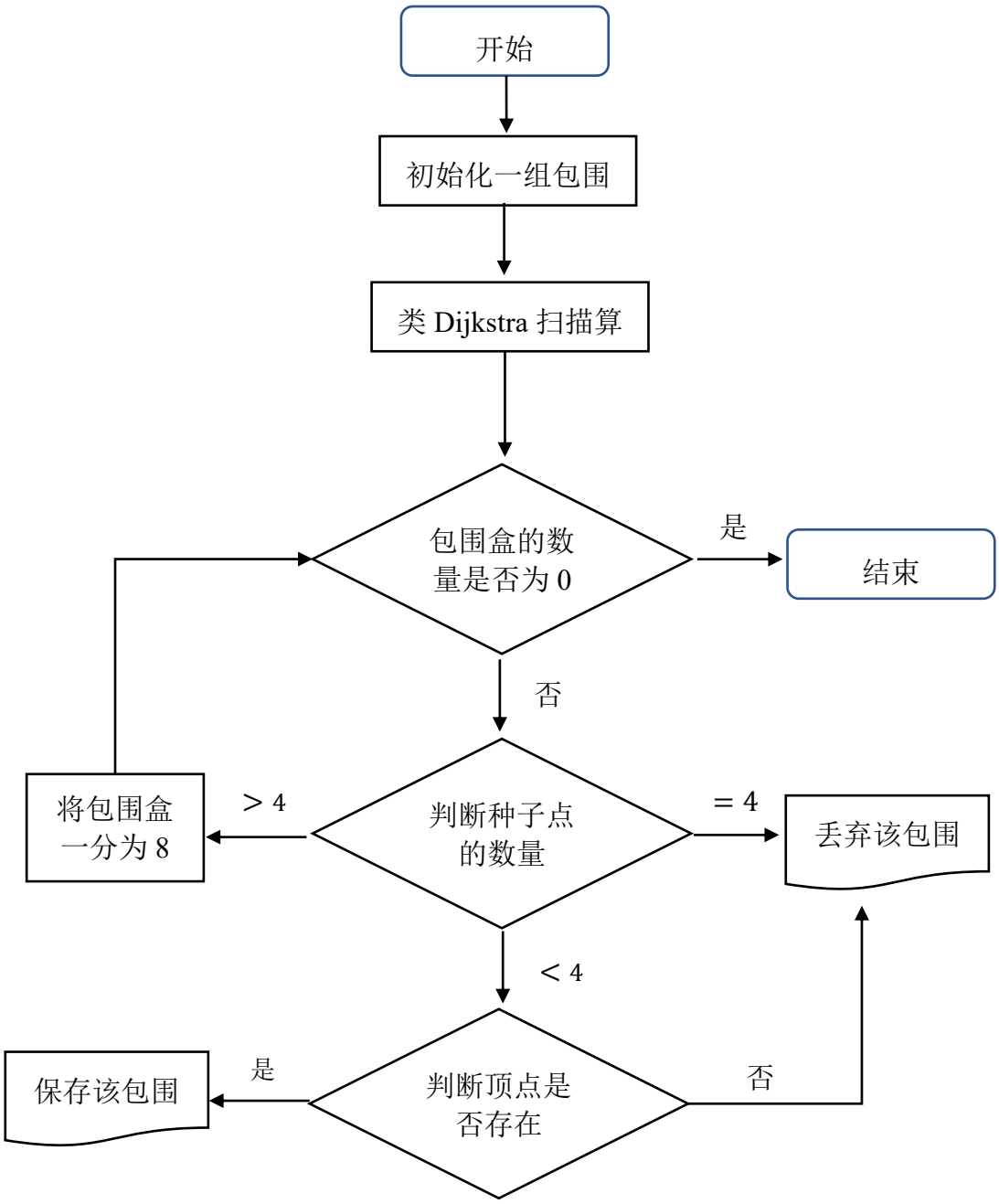



图 5-2 寻找顶点的算法流程图

在算法的具体实现过程中，涉及到一些我们自定义的数据结构，最基础的数据结构是用来表达包围盒：

```
Struct Cube
{
    int      cube_id;     //包围盒 ID
    double cube_edge_length; //立方体包围盒边长
```

```
    vector candidate_sites;   //记录的候选种子点
    point   center;              //包围盒的中心点坐标
    double minimun_weighted_distance_to_center;//记录的种子点中到中心加权距离的最小值。
};
```

在类 Dijkstra 扫描过程中，我们需要重复生成距离传播事件，并使用优先级队列进行处理。我们用以下数据结构记录我们我的距离传播事件：

```
Struct Event
{
   int   source_site;//种子点 ID
   int   cube_id;//包围盒 ID;
  double   weight_distance_to_center;//种子点到包围盒中心的距离;
};
```

在距离传播结束后，我们每个包围盒里面都可以记录一些种子点的信息。这些种子点基本都会控制该包围盒内的一部分区域。

### 5.2.2 筛选规则

之前提到我们采用筛选规则来排除点不存在 Apollonius 顶点的正方体包围盒。接下来我们将通过三个定理以及证明来详细的介绍我们的过滤规则的可靠性与正确性。

首先我们需要确定一个正方体包围盒里面一定不会存在 Apollonius 图顶点的情况。

【定理 6】我们将$l$ 记作是正方体包围盒的边长，$c$ 是正方体包围盒的中心。种子点$x_i$ 到该正方体包围盒的加权距离范围是$\left[||x_i - c|| - w_i - \frac{\sqrt{3}}{2}l,\ ||x_i - c|| - w_i + \frac{\sqrt{3}}{2}l\right]$。如果满足$\left\{x_i \,|||x_i - c|| - w_i - \frac{\sqrt{3}}{2}l < \min\{||x_j - c|| - w_j + \frac{\sqrt{3}}{2}l\}\right\}$的种子点的数量少于 4，那么在该正方体包围盒中一定没有 Apollonius 图的顶点。

【证明 6】假设有一个 Apollonius 图的顶点 $v$ 位于一个半径$r = \frac{\sqrt{3}}{2}l$的圆内，该圆恰好是包围盒$\mathcal{C}$的外接圆。那么一定存在四个种子点$x_1$，$x_2$，$x_3$，$x_4$满足

$$||x_i - v|| - w_i = const < ||x_j - v|| - w_j,\ i = 1,2,3,4, j > 4 。(5\text{-}1)$$

将式（5-2）两个不等式带入式（5-1）左边，

$$||x_i - c|| - ||v - c|| \le ||x_i - v|| ,\ ||v - c|| \le r, (5\text{-}2)$$

我们可以得到式 5-3，

$$||x_i - c|| - w_i - r \le ||x_i - v|| - w_i 。(5\text{-}3)$$

同理根据三角不等式我们可得式 5-4，

$$||x_j - v|| - w_j \le ||x_j - c|| - w_j + r 。(5\text{-}4)$$

通过联立式 5-1，式 5-3，式 5-4 几个不等式我们可以证得式 5-5，

$$||x_i - c|| - w_i - r \le ||x_j - c|| - w_j + r 。(5\text{-}5)$$

以上证明的示意图可见图 5-3。

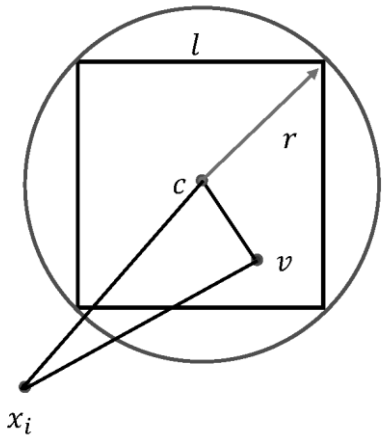



图 5-3 定理 6 示意图

上面的证明可以轻松的使我们快速过滤掉这些相距较远的种子点。但是当连个种子点相距很近，使用上面的定理将很难排除无法为 Apollonius 图顶点的形成提供支持的种子点，这就会导致我们细分包围盒的次数增加，这无疑增加了我们算法的时间复杂度。因此为了提高我们算法的计算效率，我们介绍更多的理论来提高我们的过滤效率。

【理论 7】我们将$l$ 记作是正方体包围盒的边长，$c$ 是正方体包围盒的中心。存在两个带权重的种子点$x_1$，$x_2$，并且我们把包围盒的中心$c$ 在$x_1$，$x_2$的中分曲面上的投影记作$c'$ 。那么当$||c - c' \ge \frac{\sqrt{3}}{2} l||$ 以及正方体包围盒的八个顶点和他的中点都位于$x_1$，$x_2$的中分曲面的一侧时，他们对该正方体包围盒内的 Apollonius 顶点没有任何贡献。

【证明】由于我们假设正方体包围盒的八个顶点都在双曲面的一侧，并且双曲面将空间分为凸的一侧和凹的一侧，我们根据实际情况分别对着两张清空详细分析。

（1）当正方体包围盒的八个顶点都在凸的一侧时，那么整个立方体包围盒都在双曲面的一侧，此时种子点$x_1$，$x_2$的中分曲面就不会穿过该立方体包围盒，说明种子点$x_1$，$x_2$ 也不会为该立方体包围盒内可能存在的 Apollonius 图顶点做出任何贡献。

（2）当正方体包围盒的八个顶点位于凹的一侧时。我们在投影点$c'$做一个双曲面上的切平面，此时整个切平面都位于凹的一侧区域。当$||c - c' \geq \frac{\sqrt{3}}{2} l||$时，整个正方体包围盒都位于切平面的一侧，那么正方体包围盒与$x_1$，$x_2$的中分曲面也不会有任何交集。说明种子点$x_1$，$x_2$ 也不会为该立方体包围盒内可能存在的 Apollonius 图顶点做出任何贡献。

总结以上两种情况，说明当$||c - c' \geq \frac{\sqrt{3}}{2} l||$ 以及正方体包围盒的八个顶点和他的中点都位于$x_1$，$x_2$的中分曲面的一侧时，两个种子点$x_1$，$x_2$不会为该正方体包围盒内可能存在的 Apollonius 顶点做任何贡献。

进一步说，下面的定理也就不难理解。

【定理 8】如图 5-4 所示，我们将$l$ 记作是正方体包围盒的边长，$c$ 是正方体包围盒的中心。存在两个带权重的种子点$x_1$，$x_2$，且种子点$x_1$的权重小于种子点$x_2$的权重。我们把包围盒的中心$c$ 在$x_1$，$x_2$的中分曲面上的投影记作$c'$ 。正方体包围盒的八个顶点分别为$y_1$，$y_2$，… ，$y_8$. $\pi$ 表示投影点 $c'$ 在$x_1$，$x_2$的中分曲面上的切平面。如果$||y_i - x_1|| - w_1 < ||y_i - x_2|| - w_2$，$\mathrm{i} = 1,2,3,\ldots,8$。那么种子点$x_1$可以为正方体包围盒提供比$x_2$更近的加权距离。考虑到种子点$x_1$，$x_2$都位于切平面$\pi$的两侧，如果八个顶点都位于$x_2$一侧，那我种子点$x_2$可以为正方体包围盒提供比$x_1$更近的加权距离。

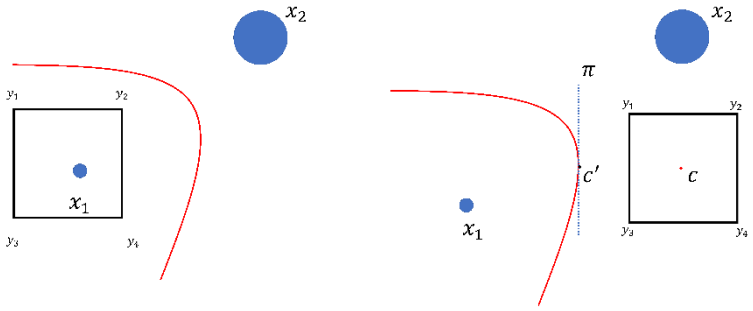



图 5-4 定理 7 图示

在我们的具体实现过程中 Apollonius 顶点的数据结构表示为：

```
Struct ApolloniusVertex
{
    int idOfSite1, idOfSite2, idOfSite3, idOfSite4;
    vector tangent1, tangent2, tangent3, tangent4;
    point vertex;
}
```

## 5.3 Apollonius 边的实现

每个 Apollonius 图的顶点由四个不同的种子点$x_1$，$x_2$，$x_3$，$x_4$决定形成，这四个种子点中的任意三个都可以形成一条开放或者封闭的曲线，该曲线上任意一点到三个种子点的加权距离相等。我们将这条曲线叫做原始的 Apollonius 图边。需要注意的是每条原始的 Apollonius 图的边可能包含多条真正的 Apollonius 图边（例如图 4-3（a））。在一个拥有$m$个 Apollonius 图顶点的 Apollonius 图中，至少存在着$4m$个原始的 Apollonius 图边。我们需要跟随下面的步骤完成 Apollonius 图边的追踪过程

Step1： 通过检查所有的 Apollonius 顶点列表找到所有的原始 Apollonius 图边；

Step2：将每个 Apollonius 图的顶点$v$ 与经过它的四条原始 Apollonius 边做匹配；

Step3：对于所有在原始 Apollonius 边上的 Apollonius 顶点我们标记为“空”，

我们从一个已经标记为“空”的 Apollonius 图顶点开始，沿着其切线方向直到追踪到另一个“空”的顶点。如图 5-5 所示，我们将他们用离散的点和线段连接起来。直到所有的 Apollonius 图的边被寻找到。

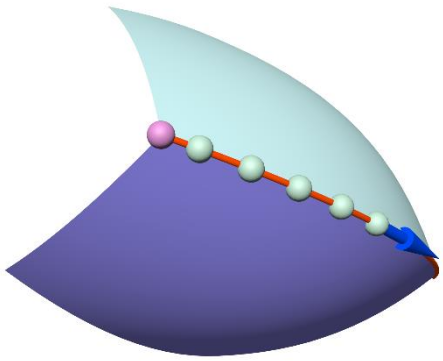


图 5-5 离散的表示一条 Apollonius 图的边

## 5.4 Apollonius 面的实现

经过上面两步的计算，我们现在已经得到了 Apollonius 图的顶点和 Apollonius 的边列表。我们接下来的工作便是寻找那些种子点对可以形成一个 Apollonius 图的面。回顾上一届我们知道每个原始的 Apollonius 边由三个种子点决定，那么其中任意两个种子点可以定一个双曲面。例如，种子点$x_1$ 和 $x_2$ 可以定义的中分曲面如式 5-6 所示：

$$||x_1 - x|| - w_1 = ||x_2 - x|| - w_2, \quad (5\text{-}6)$$

我们称其为原始的 Apollonius 面。当我们得到所有的原始的 Apollonius 图面，我们将 Apollonius 图的边与原始的 Apollonius 图面相匹配。只有将 Apollonius 图的边作为边界的原始的 Apollonius 面才是我们最后真正的 Apollonius 图的面。当我们得到一个原始的 Apollonius 图面以后，为了使其能实际应用，我们需要用一个质量良好的三角网格曲面来表示他们。在此我们用到了二维的质心 Voronoi 图技术。但是我们的 Apollonius 图面是一个曲面，我们使用二维的 CVT 显然无法真正的表示它。我们发现我们可以先将曲面铺展成一个平面，在平面上计算一个带密度的 CVT，然后通过一一对应方式将平面上的点重新投影回曲面。这样我们就可以利用二维的 CVT 技术得到一个质量良好的三维网格结构来表示曲面。整个过程如图 5-6 所示。

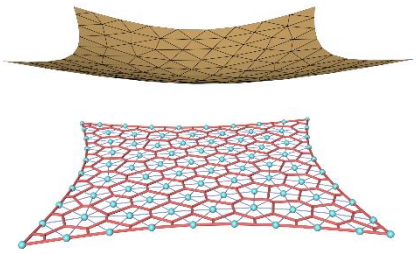


图 5-6 将二维的 CVT 投影到曲面

## 5.5 本章小结

在该章节中我们首先介绍了我们算法的整体流程框架，然后依次按照 Apollonius 顶点的定位，Apollonius 边的追踪，Apollonius 面的构建的流程顺序层次的介绍了我们算法的详细实现过程。其中我们重点介绍了 Apollonius 图顶点的定位过程，在此过程中我们提出了动态包围盒分割技术，类 Dijkstra 扫描技术以及包围盒筛检规则等。

# 第 6 章 实验结果与对比

本章首先展示了我们算法的实验结果，然后分析了本文的算法的时间代价。最后通过可扩展性和网格的质量展现了本文算法独有的优势。我们实验所用的编程语言为 C++，整个工作在配备了 3.07 GHz Intel(R) Core(TM) i5 CPU 和 16G 的内存的 64 位计算机上完成。

## 6.1 实验结果

如图 6-1 所示，我们在[0,1]的区间内随机生成 100,200，…,1000 个三维种子点，并在(0,1)的区间内随机为所有种子点配置一个随机的权重，通过我们的算法我们可以得到一组由高质量的三角网格构成的三维 Apollonius 图结构。通过这些实验结果我们可以发现，在随机生成种子点的情况下我们的算法可以精确的计算出每一种情况下的 Apollonius 结构，体现了我们算法的鲁棒性。

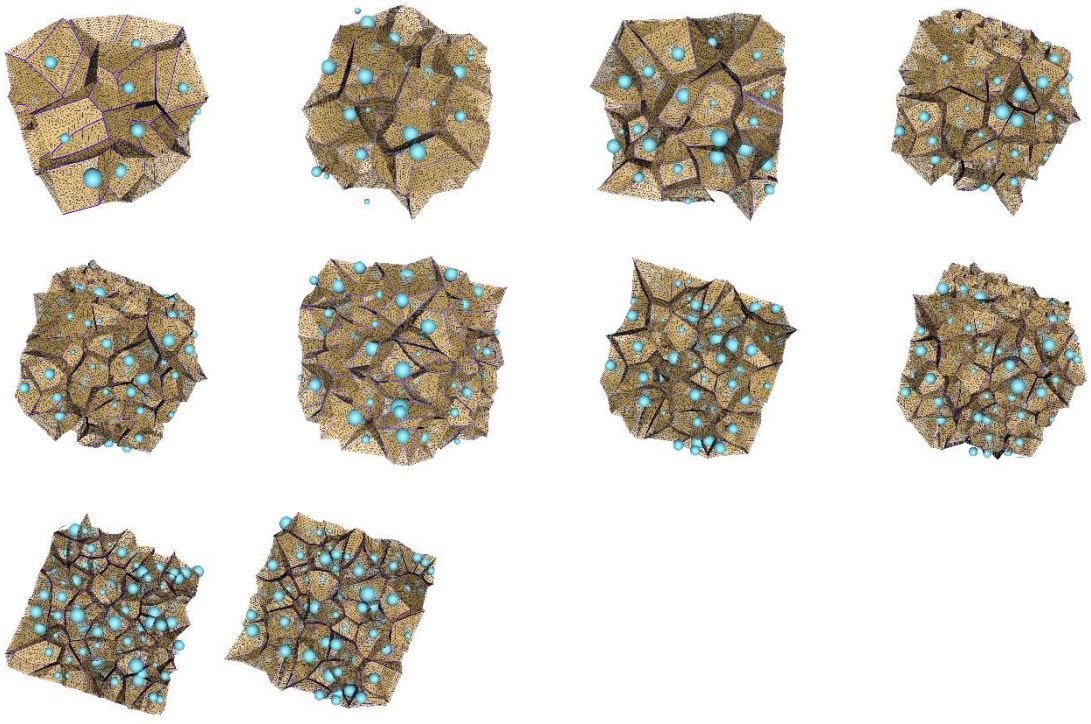


图 6-1 100 至 1000 个随机生成的种子点生成的 Apollonius 图。

## 6.2 时间性能对比

我们在蛋白质库中(https://www.rcsb.org/)挑选了几个蛋白质分子来测试我们的算法，包括：“1all”(具有 92 个原子)，“1crn”（具有 327 个原子），“1hb8”(具有 1074 个原子)，“1isn”（具有 2617 个原子）和“2hhb”（具有 4384 个原子）。由于不同的原子都具有不同的半径，并且对于特定的蛋白质，其原子位置是固定的。因此我们将蛋白子的原子作为我们的种子点，原子的位置便是我们种子点的位置信息，原子的半径作为我们每个种子点的权重信息。我们的算法的时间主要包括三方面：（1）顶点的定位，（2）边的追踪，以及（3）面的构建（生成质量良好的三角形网格曲面）。在图 6-2 中，我们展示了我们的算法的时间折线图。从中可以看出，我们的算法的时间随着种子点的扩充基本呈现线性增长。

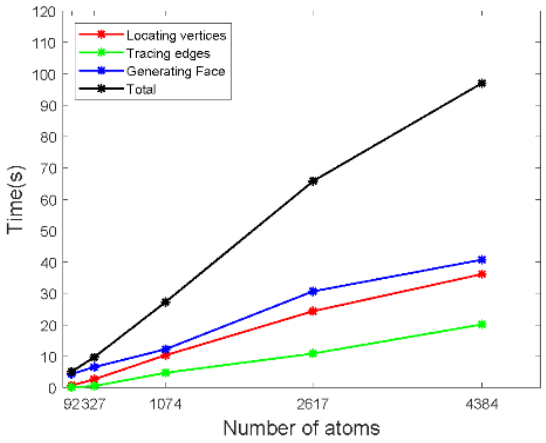


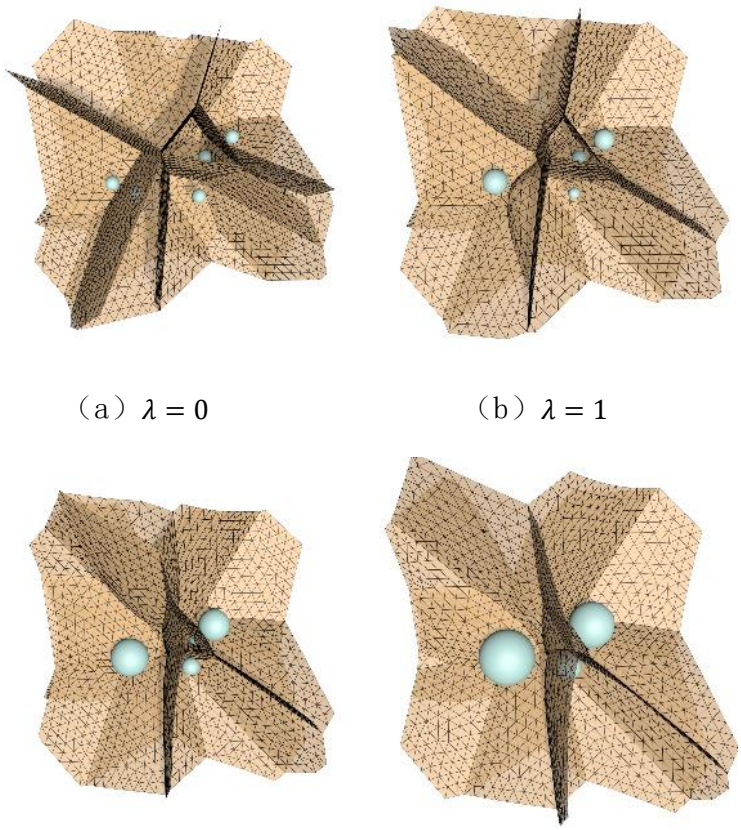

图 6-2 算法时间

此外，进过每轮细分之后，由于有些立方体包围盒里面记录的种子点数量少于 4，我们会移除掉这部分包围盒。有一部分立方体包围盒将被固定住，因为其记录的种子点的数量正好等于 4，另外有些记录种子点数量大于 4 的立方体包围盒，它们将会参与下一轮的包围盒细分过程。通过我们的观察，一般经过四轮细分以后，包含四个或者更多种子点的立方体包围盒就会变得很稀少，这种实时剔除非必要包围盒的操作无疑提高了寻找 Apollonius 图顶点的效率，这也是我们的算法接近线性的主要原因。

## 6.3 可扩展性

与 Voronoi 图不同，Apollonius 图可能包含更加复杂的结构。假设我们为所

有种子点的权重乘以一个共同的系数 $\lambda$。当 $\lambda = 0$ 时，Apollonius 图会退化为一般的 Voronoi 图。当$\lambda > 1$ 时，因为某些小的腔体会被隐藏，因此整个 Apollonius 图的结构会得到简化。当$\lambda$趋近于无穷大时，整个空间会被权重最大的种子点控制。在图 6-3 中，我们展示了这个有趣的现象。



（a）$\lambda = 0$　　（b）$\lambda = 1$

（c）$\lambda = 0$　　（d）$\lambda = 1$

图 6-3 从左到右依次是$\lambda = 0,1,2,3$。我们可以发现随着$\lambda$的增加，整个 Apollonius 的结构在简化。

## 6.4 网格质量

我们使用五个指标[53]来检验我们的网格的质量，他们分别是：

$Q_{\alpha_{min}} = \frac{3\alpha_{min}}{\pi}$，其中$\alpha_{min}$指的是三角形最小的内角；

$Q_{Ll} = \frac{l_{min}}{l_{max}}$，其中$l_{min}$ 和 $l_{max}$分别指的是三角形的最短和最长的边；

$Q_{ALS} = \frac{4\sqrt{3}\,S}{l_1^2+l_2^2+l_3^2}$，其中 S 指的是三角形的面积，$l_1$，$l_2$，$l_3$指的是三角形的三条边的长度；

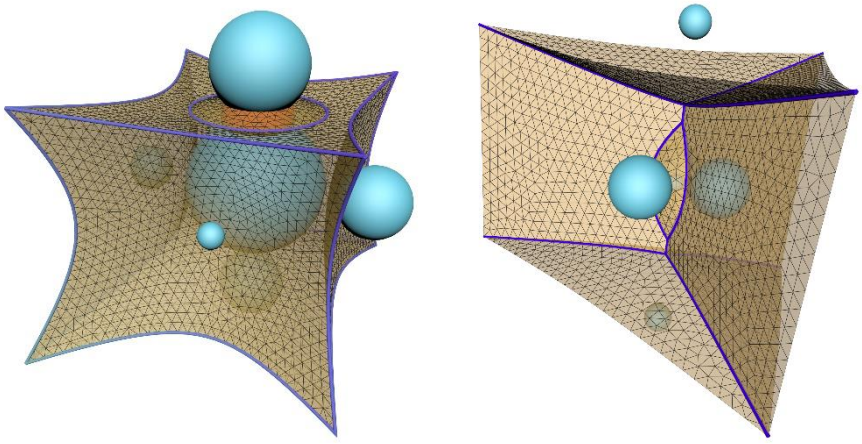
$Q_{Rr} = \frac{2r}{R}$, 其中 r 指的是三角形的内切圆半径， R 指的是三角形的外切圆半径。

$Q_{lr} = \frac{2\sqrt{3}\,r}{l_{min}}$,其中 r 指的是三角形的内切圆半径，$l_{min}$指的是三角形的最短边。

这五个指标的共同特征是，当他们趋近 1 时，三角形越趋近于正三角形（我们认为正三角形的质量是最好的）。如图 6-4 所示，我们两组不同的种子点生成两个 Apollonius 图，我们用高质量的三角网格表示整个 Apollonius 图结构，我们用上述提及的五个指标来衡量这两个 Apollonius 图的网格质量。从表 6-1 的统计数据可以看出，我们使用质心 Voronoi 图来离散曲面的方法最终得到了一个网格质量很高的三角形网格曲面。三角形网格的顶点的个数我们采用了动态采样的方式。在边追踪的过程中，我们把整个包围盒斜边长度的 2%作为我们的边追踪的步长，我们将步长记为$t$.这样我们在边上其实就拥有了一些离散的点，为了使面上的离散的点与边上离散的点对应，达到边界处三角形质量也较高的目的，我们将面内部采样点的数量设置为$\frac{S}{\frac{\sqrt{3}}{4}t^2}$ ，其中 S 指的是面的总面积。

表 6-1 五个指标测量网格质量

| | $Q_{\alpha_{min}}$ | $Q_{Ll}$ | $Q_{ALS}$ | $Q_{Rr}$ | $Q_{lr}$ |
|---|---|---|---|---|---|
| （a） | 0.890776 | 0.875221 | 0.946824 | 0.973214 | 0.955525 |
| (b) | 0.903214 | 0.934154 | 0.913413 | 0.905613 | 0.891646 |



(a) (b)

图 6-4 用于展现高质量网格的两个 Apollonius 图结构

## 6.5 本章小结

本章我们展示了我们的算法在随机生成种子点的情况下的结果图，验证了我们的算法的鲁棒性。并且我们通过五个学术上常用的网格质量检测指标来验证了我们的结果是达到国际学术圈大家的认可标准的。我们通过广泛的指标评估和实验来验证算法的有效性和鲁棒性。

# 第 7 章 应用—质心 Apollonius 图

由 N 个种子点构成的质心 Voronoi 图（CVT）是一种特殊的 Voronoi 图。在质心 Voronoi 图中，每个种子点在其所控制的区域的质心位置。在数学上，质心 Voronoi 图可以通过最小化下面的凸函数求得：

$$E_1(X) = \sum_{i=1}^{n} \int_{\Omega_i} ||x - x_i||^2 \rho(x) d\sigma, \quad (7\text{-}1)$$

其中$\rho(x)$是定义在区域上的密度函数。$\Omega_i$指的每个种子点$x_i$所控制的 Voronoi 图区域。

如果种子点的数量是确定的的，但是我们要给种子点添加一个质量约束，即每个种子点$x_i$对应一个质量约束$M_i$。那么用 2-Wasserstein 度量表示具有最小传输成本的区域划分可以 Power 图表示。因此在数学上直线 Power 图可以最小化下面的函数：

$$E_2(W) = \sum_{i=1}^{n} \int_{\mathrm{PW}_i} ||x - x_i||^2 \rho(x) d\sigma - w_i(m_i - M_i)。(7\text{-}2)$$

其中$\mathrm{PW}_i$指的是 N 个以给定质量约束为权重的的种子点形成的 Power 图，$m_i$指的是在优化过程中每次迭代过程中的真实容积。

在这篇文章中，我们通过把 2-Wasserstein 度量改为 1-Wasserstein 度量将能量函数做了推广

$$E_3(W) = \sum_{i=1}^{n} \int_{\mathrm{AW}_i} ||x - x_i|| \rho(x) d\sigma - w_i(m_i - M_i), \quad (7\text{-}3)$$

其中$\mathrm{AW}_i$指的是种子点$x_i$所控制的 Apollonius 图中的区域。$m_i = \int_{\mathrm{AW}_i} d\sigma$指的是每块区域$\mathrm{AW}_i$真正的容积，$M_i$是由使用者指定的每块区域的容积大小。在优化过程中，为了找到方程（7-3）的稳定的解，我们可以通过继承方案来代替迭代方案。由于目标函数的不同，我们重新定义了三个关键动作：

初始化$\{M_i\}$，$i = 1,2,3,\ldots,n$。（由用户指定初始化）；

While 梯度模长大于给定的误差时 do

计算 Apollonius 图。 //动作 1

计算伪质心：$x_i^* = argmin_{x_i} \int_{\mathrm{AW}_i} ||x - x_i|| d\sigma$;

将种子点$x_i$移动到伪质心$x_i^*$。//动作 2

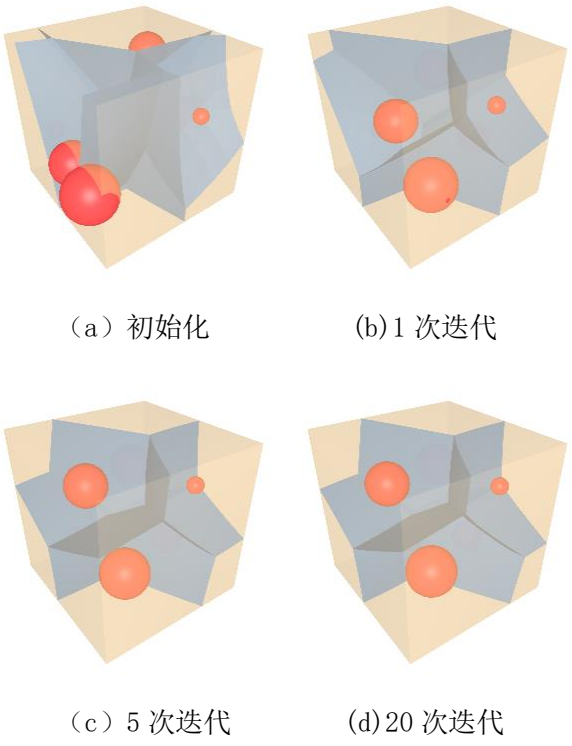
计算$m_i = \int_{\mathrm{AW}_i} d\sigma$；

计算梯度$\nabla_{\mathrm{W}_i} E_3 = M_i - m_i$；

通过梯度下降法更新$\{W_i\}$，$i = 1,2,3,\ldots,n$；//动作 3

End while

在图 7-1 中我们展示了由六个种子点生成质心 Apollonius 图的过程。我们给定六个种子点的容积分别是$\frac{M}{12}$，$\frac{M}{12}$，$\frac{2M}{12}$，$\frac{2M}{12}$，$\frac{3M}{12}$，$\frac{3M}{12}$，$M$指的是总的容积。在计算过程中我们的算法一共进行了 20 次迭代。并且在 3 秒以内结束了整个算法。



（a）初始化　　(b)1 次迭代

（c）5 次迭代　　(d)20 次迭代

图 7—1 从左到右分别是质心 Apollonius 图的初始化，已经迭代 1 次，5 次和 20 次（最终）的结果。

# 第 8 章 总结

在计算几何中，计算三维 Apollonius 图是一个具有挑战性的课题。据我们当前的知识了解，并没有一个实际的解决方案可以解决这个问题。在这篇文章中，我们系统的分析了三维 Apollonius 图单元区域，面，边以及顶点的组合结构，并且我们给出了包含顶点定位，边追踪和面构建的三段式算法。算法的主要关键点是动态的将初始包围盒分割成许多小的包围盒，保证每个小的包围盒最多只能包含一个 Apollonius 图的顶点。最后我们使用二维的质心 Voronoi 图技术来将 Apollonius 图的曲面离散为一个质量良好的三角网格结构。并且我们通过实验对比展现了我们算法的优越性，以及将其推广到了质心 Apollonius 图。

本文的工作我们也存在着一些客观的问题。首先，算法无法实现动态的更新，一个实用的 Apollonius 的算法应该支持种子点的动态增加与删除，显然我们的算法并不能满足这方面的要求，这给实际使用造成了很大的不便；其次，我们在本文的工作中并没有给出足够的实际应用。Apollonius 图作为一个实用的剖分工具，我相信随着大家对其了解程度的加深，肯定会将其用到实际的研究问题之上。

在以后的工作中，我们会继续完善这方面的工作，关于本工作的相关研究接下来大概会在以下几方面：

（1）动态增量计算。我们可以在一个计算完成的 Apollonius 图的基础上动态的增加和删除种子点。

（2）并行计算。Apollonius 图作为一个实用的工具，其计算效率永远是研究人员最为关注的一个客观因素。虽然我们的算法可以接近线性收敛，但是在实际的使用过程中我们肯定还是希望有更加快速的方法。像 Voronoi 图一样，并行计算肯定会成为接下来的研究方向。

（3）三维 Apollonius 图的实际应用。我们相信其曲面的特性在三维表面重构或者物体模拟等方面可能会有更良好的效果。

# 参考文献

[1] 葛少云,李慧,刘洪.基于加权 Voronoi 图的变电站优化规划[J].电力系统自动化, 2007, 31(3):29-34.

[2] 严冬明,刘晓寒,郭建伟.三维裁剪 Voronoi 图的多线程并行计算方法,系统,中国，CN201911255557.9，2020.

[3] 赵文婷,彭俊毅.基于 VORONOI 图的无人机航迹规划[J].系统仿真学报, 2006,018(0z2):159-162,165.

[4] 陈军,赵仁亮,乔朝飞.基于 Voronoi 图的 GIS 空间分析研究[J].武汉大学学报(信息科学版), 2003(S1):32-37.

[5] 郭帅,马书根,李斌,等.基于 Voronoi 地图表示方法的同步定位与地图创建[J]. 自动化学报, 2011.

[6] 刘红伟,曹娟,陈中贵等.基于容积约束 Power 图的图像分片逼近[C]，中国计算机学会;中国自动化学会;中国图学学会;中国图像图形学会;中国系统仿真学会, 2016.

[7] 叶畋宇,王逸群,严冬明等.基于局部优化的重心 Voronoi 图计算[J]. 系统仿真学报, 2019, xx(002):218-226.

[8] 郑利平,郜文灿,李尚林等.定点容量限制质心 Power 图生成[J].中国图象图形学报, 2016, 21(9):1229-1237.

[9] 孙德超,辛士庆,周亚训等.重要性驱动的中轴线[J].计算机辅助设计与图形学学报,2016,28(12):2107-2113.

[10] 郑利平,路畅,蔡瑞文,等.质心容量限制 Power 图一体化生成算法[J].系统仿真学报, 2018, 030(007):2489-2496.

[11] 严冬明,全卫泽,郭建伟,等.基于重心 Voronoi 图的非钝角的重新网格化方法, 中国，CN201510680027.4，2018.

[12]Voronoi, Georges. "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites." , Journal für die reine und angewandte Mathematik, 1908, 133 : 97-178.

[13]Anton F, Boissonnat J-D, Mioc D, Yvinec M. An exact predicate for the optimal construction of the additively weighted Voronoi diagram. In Europ. 2002

[14]Aurenhammer F, Hoffmann F, Aronov B. Minkowski-type theorems and least-squares clustering. Algorithmica 20, 1998, 1:61–76

[15]Aurenhammer F. Power diagrams: properties, algorithms and applications. SIAM Journal on Computing 16, 1987, 1, 78–96

[16]Aurenhammer F. Voronoi diagrams—a survey of a fundamental geometric data structure. ACM Computing Surveys (CSUR) 23, 1991, 3, 345–405

[17]Bernauer J, Bahadur R P, Rodier F, Janin J, Poupon A. DiMoVo: a Voronoi tessellation-based method for discriminating crystallographic and biological protein–protein interactions. Bioinformatics 24, 2008, 5, 652–658

[18]Balzer M, Heck D. Capacity-constrained Voronoi diagrams in finite spaces. In Voronoi Diagrams in Science and Engineering, 2008

[19]Boissonnat J-D, Karavelas M I. On the combinatorial complexity of Euclidean Voronoi cells and convex hulls of d-dimensional spheres. Tech. Rep. RR-4504, INRIA, July 2002

[20]Cole R, Goodrich M T, Dúnlaing C Ó. Merging free trees in parallel for efficient Voronoi diagram construction. In International Colloquium on Automata, Languages, and Programming, 1990, Springer, pp. 432–445

[21]Du Q, Emelianenko M, Ju L. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. SIAM journal on numerical analysis 44, 2006, 1, 102–119

[22]Devillers O. Improved incremental randomized Delaunay triangulation. In Proceedings of the fourteenth annual symposium on Computational geometry, 1998, pp. 106–115

[23]Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: Applications and algorithms. SIAM review 41, 1999, 4, 637–676

[24]De Goes F, Breeden K, Ostromoukhov V, Desbrun M. Blue noise through optimal transport. ACM Transactions on Graphics (TOG) 31, 2012, 6, 171

[25]Dou Z, Xin S, Xu R, Xu J, Zhou Y, Chen S, Wang W, Zhao X, Tu C. Top-down

shape abstraction based on greedy pole selection. In Computational Visual Media Conference, 2019

[26] Emiris I Z, Karavelas M I. The predicates of the Apollonius diagram: algorithmic analysis and implementation. Computational Geometry 33, 2006, 1-2, 18–57

[27] Fortune S. A sweepline algorithm for Voronoi diagrams. Algorithmica 2, 1987, 1-4, 153

[28] Fortune S. Voronoi diagrams and Delaunay triangulations. In Computing in Euclidean geometry. World Scientific, 1995, pp. 225–265

[29] Fan Z, Wu Y, Zhao X, Lu Y. Simulation of polycrystalline structure with Voronoi diagram in Laguerre geometry based on random closed packing of spheres. Computational materials science 29, 2004, 3, 301–308

[30] Glickenstein D. Geometric triangulations and discrete laplacians on manifolds. arXiv preprint math/0508188, 2005

[31] Garrido S, Moreno L, Abderrahim M, Martin F. Path planning for mobile robot navigation using Voronoi diagram and fast marching. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, IEEE, pp. 2376–2381

[32] Gavrilova M, Rokne J. An efficient algorithm for construction of the Power diagram from the Voronoi diagram in the plane. International journal of computer mathematics 61, 1996, 1-2, 49–61

[33] Gavrilova M L, Rokne J G. Updating the topology of the dynamic Voronoi diagram for spheres in euclidean d-dimensional space. Comput. Aided Geom. Des. 20, 2003, 231–242

[34] Green P J, Sibson R. Computing Dirichlet tessellations in the plane. The computer journal 21, 1978, 2, 168–173

[35] Hoff III K E, Keyser J, Lin M, Manocha D, Culver T. Fast computation of generalized Voronoi diagrams using graphics hardware. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, 1999, ACM Press/Addison-Wesley Publishing Co., pp. 277–286

[36] Hu Z, Li X, Krishnamurthy A, Hanniel I, Mcmains S. Voronoi cells of non-

general position spheres using the gpu. Computer-Aided Design and Applications 14, 2017, 5, 572–581

[37]Imai H, Iri M, Murota K. Voronoi diagram in the Laguerre geometry and its applications. SIAM Journal on Computing 14, 1985, 1, 93–105

[38]Kim D, Cho Y. Lim D-S. Region expansion by flipping edges for euclidean Voronoi diagrams of 3d spheres based on a radial data structure. In Computational Science and Its Applications – ICCSA 2005 (Berlin, Heidelberg, 2005), Gervasi O, Gavrilova M L, Kumar V, Laganà A, Lee H P, Mun Y, Taniar D, Tan C J K, (Eds.), Springer Berlin Heidelberg, pp. 716–725

[39]Kim D-S, Cho Y, Kim D. Euclidean Voronoi diagram of 3D balls and its computation via tracing edges. Computer-Aided Design 37, 2005, 13, 1412–1424

[40]Kim D-S, Cho Y, Kim D, Kim S, Bhak J, Lee S. Euclidean Voronoi diagrams of 3d spheres and applications to protein structure analysis. Japan Journal of Industrial and Applied Mathematics 22, 2005, 251–265

[41]Kim D-S, Cho Y, Kim D, Cho C-H. Protein structure analysis using euclidean Voronoi diagram of atoms. Proc. International Workshop on Biometric Technologies (BT 2004), 01 2004, 125–129

[42]Kim D, Kim D-S. Region-expansion for the Voronoi diagram of 3d spheres. Computer-Aided Design 38, 2006, 5, 417 – 430

[43]Kim D-S, Kim D, Cho Y. Euclidean Voronoi diagrams of 3d spheres: Their construction and related problems from biochemistry. In Mathematics of Surfaces XI (Berlin, Heidelberg, 2005), Springer Berlin Heidelberg, pp. 255–271

[44]Kim D-S, Kim D, Sugihara K. Unifying method for computing the circumcircles of three circles. International journal of CAD/CAM 2, 2002, 1, 45–54

[45]Karavelas M I, Yvinec M. Dynamic additively weighted Voronoi diagrams in 2d. In European Symposium on Algorithms, 2002

[46]Liu Y-J, Chen Z, Tang K. Construction of iso-contours, bisectors, and Voronoi diagrams on triangulated surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 2010, 8, 1502–1517

[47]Lee D T. Two-dimensional Voronoi diagrams in the lp-metric. J. ACM 27, Oct.

1980, 4, 604–618

[48]Li X, Krishnamurthy A, Hanniel I, Mcmains S. Edge topology construction of Voronoi diagrams of spheres in nongeneral position. Computers and Graphics 82, 2019, 332 – 342

[49]Lu L, Sharf A, Zhao H, Wei Y, Fan Q, Chen X, Savoye Y, Tu C, Cohen-Or D, Chen B. Build-to-last: strength to weight 3D printed objects. ACM Transactions on Graphics (TOG) 33, 2014, 4, 97

[50]Liu Y, Wang W, Lévy B, Sun F, Yan D-M, Lu L, Yang C. On centroidal Voronoi tessellation—energy smoothness and fast computation. ACM Transactions on Graphics (TOG) 28, 2009, 4, 101

[51]Ma L. Bisectors and Voronoi diagrams for convex distance functions. PhD thesis, Fernuniv., Fachbereich Informatik, 2000

[52]Rong G, Liu Y, Wang W, Yin X, Gu D, Guo X. GPU-assisted computation of centroidal Voronoi tessellation. IEEE transactions on visualization and computer graphics 17, 2010, 3, 345–356

[53]Ramos J, Palau J, Huerta A. Numerical representation of the quality measures of triangles and triangular meshes. Communications in Numerical Methods in Engineering 19, 07 2003

[54]Rajasekaran S, Ramaswami S. Optimal parallel randomized algorithms for the Voronoi diagram of line segments in the plane and related problems. In Proceedings of the tenth annual symposium on Computational geometry, 1994, ACM, pp. 57–66

[55]Senechal M. Spatial tessellations: Concepts and applications of Voronoi diagrams. Science 260, 1993, 5111, 1170–1173

[56]Shamos M I, Hoey D. Closest-point problems. In 16th Annual Symposium on Foundations of Computer Science (sfcs 1975), 1975, IEEE, pp. 151–162

[57]Sack J, Urrutia G. Voronoi diagrams. handbook of computational geometry. Ottawa: Elsevier Science 290, 2000

[58]Voloshin V, Beaufils S, Medvedev N. Void space analysis of the structure of liquids. Journal of molecular liquids 96, 2002, 101–112

[59]Wang W, Fang Z, Xin S, He Y, Zhou Y, Chen S. Tracing high-quality isolines for

discrete geodesic distance fields. In Computational Visual Media Conference, 2018

[60]Will H-M. Fast and efficient computation of additively weighted Voronoi cells for applications in molecular biology. In Scandinavian Workshop on Algorithm Theory, 1998, pp. 310–321

[61]Will H-M. Computation of additively weighted Voronoi cells for applications in molecular biology. PhD thesis, ETH Zurich, 1999

[62]Xin S-Q, Lévy B, Chen Z, Chu L, Yu Y, Tu C, Wang W. Centroidal Power diagrams with capacity constraints: Computation, applications, and extension. ACM Transactions on Graphics (TOG) 35, 2016, 6, 244

[63]Xin S-Q, Wang X, Xia J, Mueller-Wittig W, Wang G-J, He Y. Parallel computing 2D Voronoi diagrams using untransformed sweepcircles. Computer-Aided Design 45, 2016, 2, 483–493

[64]Zhou Y, Ju L, Wang S. Multiscale superpixels and supervoxels based on hierarchical edge-weighted centroidal Voronoi tessellation. IEEE Transactions on Image Processing 24, 2015, 11, 3834–3845。

[65]The Computational Geometry Algorithms Library（CGAL）, https://www.cgal.org/.

[66]Yan D M , Bruno Lévy, Liu Y , et al. Isotropic Remeshing with Fast and Exact Computation of Restricted Voronoi Diagram[J]. Computer Graphics Forum, 2009.

[67]Bruno Lévy, Liu Y . L-p Centroidal Voronoi Tessellation and its Applications[J]. ACM Transactions on Graphics, 2010, 29(4).

[68]Lu L , Bruno Lévy, Wang W . Centroidal Voronoi Tesselation of Line Segments and Graphs[J]. Computer Graphics Forum, 2012, 31(2pt4):775-784.

[69]A X W , A X Y , B Y J L , et al. Intrinsic computation of centroidal Voronoi tessellation (CVT) on meshes - ScienceDirect[J]. Computer-Aided Design, 2015, 58(C):51-61.

[70]M. Balzer, "Capacity-Constrained Voronoi Diagrams in Continuous Spaces," 2009 Sixth International Symposium on Voronoi Diagrams, Copenhagen, Denmark, 2009, pp. 79-88.

# 致 谢

时光荏苒，岁月如梭，转眼间三年研究生生涯就要接近尾声。回想当初刚收到研究生录取通知书时的喜悦场景似乎还在眼前。三年的研究生生活使我成长了很多。在此怀着感恩的心感谢在研究生期间帮助过我的导师以及融洽相处的同学们。

首先感谢我的导师辛士庆教授。在科研上，辛老师一直活跃在整个科研一线，不断给我们分享他的新观点。当我在科研中遇到困难时，辛老师往往将我们叫入办公室，大家畅所欲言，各抒己见。最后给我们的观点客观的评价，给我们遇到的难点有效的解决方案。促膝长谈至深夜是家常便饭。在生活中更是关心我们的饮食起居。我们的关系亦师亦友。

其次还要感谢给予我指导的其他老师。这些老师中有些是直接指导了我的学术工作，有的是他们的讲座给予我帮助。论文选题的开始到论文的截止是我的导师全程指导。但是其他的老师一次次的分享他们的科研结晶也不断的拓宽我的学术视野和科研思路，在我的科研过程中也起着重大的帮助。

最后，感谢实验室一起学习生活的兄弟姐们。科研上大家互相鼓励，互相支持。每周的讨论班大家无私分享，手中的横向课题大家精诚合作，营造了一个良好的科研氛围。在生活中，大家互相帮助，互相关心。一次次户外活动让我们成为一个融洽的集体，大家虽来自五湖四海却始终是一家人。