周美军, 罗添榆. 复杂多面体生成算法及其应用[J]. 智能计算机与应用,2026,16(4):138-146. DOI:10.20169/j.issn.2095-2163.24061202

复杂多面体生成算法及其应用

周美军, 罗添榆
(广东工业大学 机电工程学院, 广州 510006)

摘 要: 多面体在计算几何中有着非常重要的地位,并在最小包围盒、碰撞检测、分子动力学、三维点云重建和不规则形体建模等领域得到广泛应用。相关算法包括 gift wrapping 算法、分治算法和增量式算法等。对于存在大量内部点的情况而言,分治算法最为有效,但是三维的分治算法编程困难,不如增量式算法普及。为解决此问题,通过对分治算法进行改进并与随机增量算法相结合,得到了一种新的凸多面体生成算法,可以应用于分子动力学领域当中由大量原子点组成的模型的几何形状的提取。并且在此凸多面体生成算法的基础上进一步拓展了凹多面体生成算法。实验表明,当存在足够多的内部点时,新的凸多面体生成算法较随机增量式算法效率更高、耗时更短,而基于凸多面体重新拓扑而成的凹多面体也能够达到较好的预期效果,这对于复杂模型的建模有着重要的意义。

关键词: 多面体; 分治算法; 增量式算法; 凸多面体; 凹多面体
中图分类号: TP391.41 **文献标志码**: A **文章编号**: 2095-2163(2026)04-0138-09

Algorithm for generating complex polyhedron and its application

ZHOU Meijun, LUO Tianyu
(School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Polyhedra play a very important role in computational geometry and have been widely used in fields such as minimum bounding box, collision detection, molecular dynamics, 3D point cloud reconstruction, and irregular shape modeling. The related algorithms include gift wrapping algorithm, divide and conquer algorithm, and incremental algorithm. For situations with a large number of internal points, divide and conquer algorithms are the most effective, but three-dimensional divide and conquer algorithms are difficult to program and are not as popular as incremental algorithms. To solve this problem, a new convex polyhedron generation algorithm was developed by improving the divide and conquer algorithm and combining it with the random increment algorithm. This algorithm can be applied to extract the geometric shape of models composed of a large number of atomic points in fields such as molecular dynamics. And based on this convex polyhedron generation algorithm, the concave polyhedron generation algorithm has been further expanded. Experiments have shown that when there are enough internal points, the new convex polyhedron generation algorithm is more efficient and less time-consuming than the random incremental algorithm, and concave polyhedra reconstructed based on convex polyhedra can also achieve good expected results, which is of great significance for the modeling of complex models.

Key words: polyhedron; divide-and-conquer algorithm; incremental algorithm; convex polyhedron; concave polyhedron

0 引 言

多面体包括凸多面体(凸包)和凹多面体(凹包),是计算几何中非常重要的几何结构,在诸多领域都有应用,也是被广泛研究的问题之一[1-2]。

凸多面体的应用最为广泛,其在最小包围盒、碰撞检测、颗粒建模、分子动力学等领域都有广泛的应用价值[3-5]。最早的凸多面体生成算法是 Chand 和 Kapuer 的 gift wrapping 算法[6],时间复杂度为 $O(n^2)$。后来 Preparata 和 Hong 提出了二维的分治算法[7],文献[8-9]提出了三维的分治算法,该算法通过将小凸多面体合并形成大凸多面体,是一个时间复杂度 $O(\text{nlogn})$ 的算法,但该算法实现难度较高,应用并不广泛。如今,构建凸多面体常用的是增

作者简介: 周美军(1997—),男,硕士研究生,主要研究方向:计算机图形学,分子动力学。Email:zhoumeijun117@163.com; 罗添榆(1999—),男,硕士研究生,主要研究方向:计算机图形学,分子动力学。

收稿日期: 2024-06-12

量式算法，其中包括 Clarkson 和 Shor 提出的随机增量算法[10-11]，Barber 等[12]的 QuikHull 算法，其时间复杂度均为 $O(\text{nlogn})$。由于分子动力学模型通常由许多点组成，其中存在大量内部点，难以观测其几何形状。本文在吸收分治算法和增量式算法的思想后，提出了新的凸多面体生成算法，用于更高效的提取几何形状，减少对冗余点的计算。

相对而言，凹多面体则更多应用于游戏建模、三维点云重建和不规则形体建模等领域。Edelsbrunner[13-14]首先提出了 Alpha-shape 算法，用于网格生成、医学图像分析和可视化地震数据领域等，现在也用于三维点云的表面重构[15-16]，李庆等[17]利用 Alpha-shape 算法对树冠三维激光扫描后的点云数据进行模型构建。付昱兴等[18]利用 Alpha-shape 算法构建枣树点云的三维模型。Bernardini F 等[19]提出 Ball Pivoting 算法，该算法让一个球在点云数据上滚动生成外表面。胡丝兰等[20]改进了 Ball Pivoting 算法，让算法适用于处理不均匀的点云数据而不产生洞。这些方法都是对点云数据进行收束成面，但仍然需要调制参数来获得理想外表面，得到的表面存在误差。为解决这一问题，在凸多面体的基础上，利用四面体互补面替换的方法修饰多面体，得到了一种能够更加准确地形成目标凹多面体的方法。

1　算法思想

1.1　凸多面体生成算法思想

凸多面体是指能够以最小的表面积包裹所有给定点的多面体，本文通过改进分治算法并将其与随机增量算法相结合，提出了一种新型凸多面体生成算法。下面将分别介绍这两种算法的基本思想。

1.1.1　分治算法

分治算法多应用于二维场景，在三维领域的应用则相对较少，主要是由于分治算法需要克服许多拓扑障碍。根据文献[8]和文献[9]，点集合中任意 4 个数据点构成一个四面体，随后通过两两合并的方式，最后形成一个囊括所有给定点的最小凸多面体。假定存在两个凸多面体，以一个凸多面体上所有的顶点为观察点向另一个凸多面体观察，所有观察到的外表面定义为可见面，若被观察的表面的邻面无法被观察到，则该表面被定义为弱可见面，随后以每个弱可见面的一条或多条边界为基础，构建切边界环。外表面的可见性可以通过下式的正负符号判断，P_4 相对于三角形面 $P_1P_2P_3$，下式结果为正则表示可见，结果为负则表示不可见。

$$F(P_1,P_2,P_3,P_4)=\begin{vmatrix}1 & x_1 & y_1 & z_1\\ 1 & x_2 & y_2 & z_2\\ 1 & x_3 & y_3 & z_3\\ 1 & x_4 & y_4 & z_4\end{vmatrix} \tag{1}$$

如图 1 所示，以 C_L 上所有顶点为观测点，观察 C_R 所有的面，除了顶面与底面不可见外，其余均为可见面，其中，$\triangle bed$ 和 $\triangle bef$ 为弱可见面，于是 $\triangle bed$ 贡献了切边界线 $b\to e\to d$，$\triangle bef$ 贡献了切边界线 $f\to e\to b$。所有的弱可见面贡献的边界线构成一个切边界环 $b\to e\to d\to f\to e\to b\to c\to a\to b$。由于切边界环的复杂情况比较多，导致分治算法需要克服许多拓扑障碍。

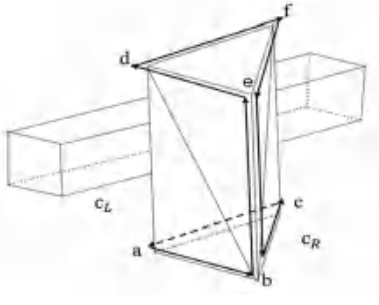


图 1　凸多面体的观测示意图

Fig. 1　Observation diagram of convex polyhedron

障碍 1　切边界环不能两次及两次以上从同一方向穿过同一条边。

障碍 2　切边界环不能跨越其自身。

障碍 3　左右切边界环的总顶点数应至少为 4。

障碍 4　右切边界线的左侧与左切边界环的右侧应形成一个简单连通区域。

障碍 5　切边界环上存在公共点的桥接边不能相互平行。

1.1.2　增量式算法

根据文献[10]和文献[11]，增量式算法的原理为：先初始化一个四面体，随后不断通过添加外部顶点来扩充凸多面体。如图 2 所示，向四面体 $P_1P_2P_3P_4$ 中添加 P_5，P_5 仅与 $\triangle P_1P_3P_4$ 呈外部关系，据此可构建出一个六面体。但该算法不可避免的会对内部顶点进行无差别运算。当数据点足够多时，将造成大量冗余运算，这是大规模运算应该避免的。如果能和分治算法相结合就能消除冗余的运算量，将大幅提升增量式算法的运算速度。

1.2　凹多面体生成算法思想

凹多面体无法通过算法实现完全唯一的表示，需借助凸多面体作为唯一表示的中间状态完成过渡。凸多面体表面均为三角形面，由于 3 个顶点可

唯一确定一个面,且一条边连接两个三角形面,因此共享一条边的两个三角形面能够唯一确定一个四面体。通过选择一条边,当与之相连的两个三角形面开口朝着多面体内部时,则可从几何意义上删去一个四面体,以此达到凹陷的几何效果。若当开口朝着多面体外部时,则可从几何意义上填充一个四面体,便达到了凸起的效果。

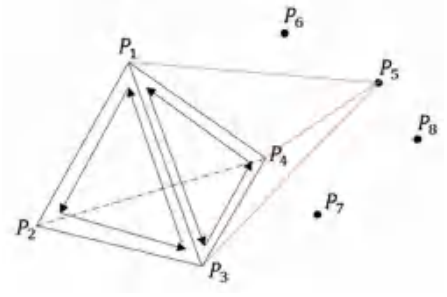


图 2 增量法原理示意图

Fig. 2 Schematic diagram of incremental method

2 凸多面体生成算法内容

前文在凸多面体的算法思想当中介绍了分治算法与增量算法,这两种算法均存在自身的局限性,因此无法达到最优运算效果。为此,将二者优点相结合,弥补各自的不足,可能会得到更好的效果。在分治算法的思想上构建点集合的二叉树,每个节点采用数据结构表示,通过删除完全处于内部的节点,从而达到分治剪枝的目的,减少冗余的计算。每个节点下面都有一个多面体,通过合并子节点到父节点,子节点下的多面体两两合并,并传递给父节点。如果某些叶子节点下的数据点共面,无法构建多面体时,则采用随机增量算法进行补充。

2.1 数据结构

如图 3 所示,为了算法的实现,需要采用自定义结构体作为数据结构,对最基本的几何数据进行封装,主要包括点(Point)、面(Face)、二叉树节点(TreeNode)三类,各结构的属性定义如下:

(1) Point:有 3 个属性, x、y 和 z 为坐标值。

(2) Face:有 4 个属性, a、b 和 c 是顶点 Point 在顶点数组当中的下标,且 3 个下标按次序逆时针绕向;flag 为面的观测标志位,用于标志该面是否被观察到。

(3) TreeNode:包含 12 个属性,其中 start 和 end 标识顶点数组的起始下标和末尾下标;isused 表示节点是否被使用,用于做二叉树的剪枝处理;polygon 为凸多面体对象; $x_{\min}$、$x_{\max}$、$y_{\min}$、$y_{\max}$、$z_{\min}$ 和 $z_{\max}$ 为对应节点的空间边界;left 和 right 是二叉树的左指针和右指针。

Point	x	y	z	
Face	a	b	c	flag
TreeNode	Start	end	isused	polygon
	$x$min	$x$max	$y$min	$y$max
	$z$min	$z$max	left	right

图 3 基本几何数据结构

Fig. 3 Basic geometric data structure

2.2 分治剪枝

若 $S=P_1,P_2,P_3,\cdots,P_n$ 是由 n 个数据点组成的集合,用一个顶点数组 Points 维护,并在载入的过程中获得这 n 个点的边界值 $x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$, $z_{\min}$, $z_{\max}$, 其对应 S 所在六面体空间在 3 个维度上的边界参数。

采用前序递归的方式构建分治二叉树,视对应节点 TreeNode 的顶点数组的空间为 D_j,在生成节点的同时,节点对应顶点数组会随递归深度重复 x,y,z 方向排序,不断二分数组,如图 4 所示。

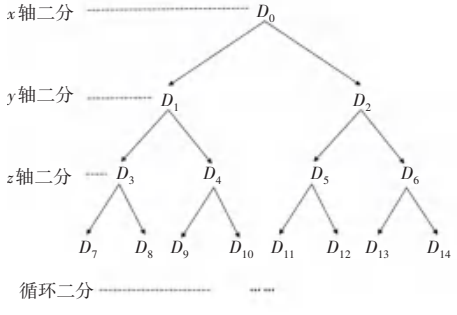


图 4 分治二叉树示意图

Fig. 4 Schematic diagram of divide-and-conquer binary tree

顶点的数量按照下面公式进行:

左边:

$$L = \left\lfloor \frac{\left\lfloor \frac{n}{4} \right\rfloor + 1}{2} \right\rfloor \times 4 \tag{2}$$

右边:

$$R = n - L \tag{3}$$

分割点:

$$\text{mid} = \text{start} + L - 1 \tag{4}$$

该公式中[ ]为向下取整,通过该公式取点有三大优势:左右节点对应空间顶点尽可能均分;左侧节点的顶点数永远为 4 的倍数;靠近左侧的叶子节点的空间中都是 4 个数据点。

如果任何节点对应的六面体空间的边界参数中

不包含根节点的边界参数中任何一个边界参数，那么可以认为该节点对应的空间为冗余空间，对应顶点为冗余顶点，如图 5 所示。包含内部空间的节点不参与凸多面体的相关运算，从而达到分治剪枝的目的，有利于后续对凸多面体的合并。

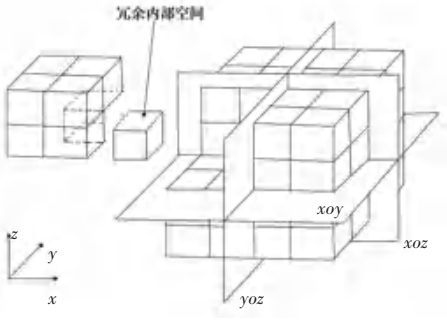


图 5　剪枝冗余空间的示意图

Fig. 5　Schematic diagram of pruning redundant space

2.3　合并凸多面体

基于分治剪枝操作，对冗余节点 TreeNode 及其下属的数据点进行删除，减少了运算量，同时将各个节点 TreeNode 下的凸多面体置于独立空间，以便实现凸多面体的两两合并。合并凸多面体的过程采用对 TreeNode 根节点进行后序遍历，这时节点两两合并，凸多面体 polygon 也两两合并。这里合并凸多面体有 3 个步骤：克服分治障碍；获取切边界环；桥接切边界环。三个步骤是承前启后的关系，其中桥接切边界环为核心步骤，在两个切边界环之间构建密闭有效的三角形面，使两个凸多面体沿着切边界环进行融合，然后合并成一个凸多面体。

2.3.1　克服分治障碍

分治算法需要克服前面所述的 5 个障碍才能获得有应用价值的切边界环，这是合并凸多面体的基础。

对于障碍 1，有向的切边界环可能存在共起点的情况，如图 6(a)所示，两个弱可见面的公共边则形成孤立边，不可见面形成孤岛。多个孤岛和孤立边的存在使得切边界环的获取比较困难，本文通过仅获取一个孤岛的边界环，无论按照任何的顺序获取边界线都可以获得一个完整的有序边界环，以此克服障碍 1。

对于障碍 2，如图 6(b)所示，存在孔洞的情况，通过分治剪枝，共父节点的左右节点都是相邻空间，不存在碰撞干涉，也就不会形成空洞，以此克服障碍 2。

对于障碍 3，本文对共面一侧的数据点采用随机增量的方法对另一侧凸多面体进行更新，以此克服障碍 3。

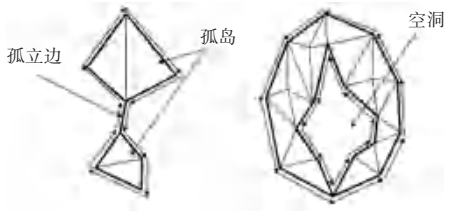


图 6　孤岛、孤立边和空洞的示意图

Fig. 6　Schematic diagram of isolated islands, isolated edges and voids

对于障碍 4，由于共父节点的左右节点空间相邻且不干涉，那么左右两侧的孤岛存在连通空间，以此克服障碍 4。

对于障碍 5，本文采用了全新的桥接切边界环的方法，会对每个桥接三角形面进行有效性检验，有公共点且平行的桥接边在算法中无法构成有效的三角形面，避免了有公共点的桥接边平行的情况产生，以此克服障碍 5。

2.3.2　获取切边界环

切边界环是两个凸多面体相互观察下最外层边界，在 1.1 小节介绍了获取切边界环的方法，基于该方法需要执行以下步骤：如图 7(a)所示，以 $C(D_L)$ 为对象，$C(D_R)$ 上的所有顶点与 $C(D_L)$ 上的三角形面使用公式 1 进行符号判断。此时 $C(D_L)$ 上剩余的面可能有一个或多个孤岛面，随机选择剩余面中的一个面进行共边索引(两个相邻面共用一个边)跨越边界寻找邻面，如果邻面无法跨越(即邻面已经被删除)，此时记录边界，可以获得逆时针绕向的左切边界环。然后以 $C(D_L)$ 的左切边界环上的顶点对 $C(D_R)$ 做相同操作，得到顺时针绕向的右切边界环，如图 7(b)所示。

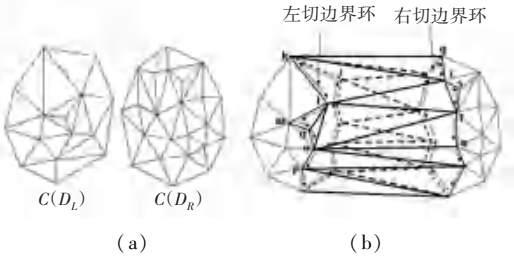


图 7　合并凸多面体原理图

Fig. 7　Schematic diagram of merging convex polyhedron

2.3.3　桥接切边界环

为了使两个多面体合并，需要对切边界环进行桥接操作，使两个凸多面体进行融合。桥接切边界环是一个切边界环上的边找另一个切边界环上的点去构建三角形面的过程，左边的边与右边的点构建

的面称为正桥接三角形面,右边的边与左边的点构建的面称为副桥接三角形面,左边的边与左边的点构建的面称为正补三角形面,右边的边和右边的点构建的面称为副补三角形面。桥接的过程需满足 4 个条件:第一个是正点条件,左切边界环上的边寻到的右切边界环上的点两侧临近顶点均在桥接三角形面的内部,称为正点;第二个是副点条件,相邻两个正点不重合,那么两个正点对应相邻的正桥接三角形的交点称为副点,落在左切边界环上;第三个是起始边条件,第一个正桥接三角形在左切边界环上对应边的两个临近点均在桥接三角形内部;第四个是正副边条件,左右切边界环上对应边的下一个临近点在桥接三角形内部,左切边界环上的边称为正边,右切边界环上的边称为副边。

桥接需确定第一个正桥接三角形的位置,假定起始时以 $C(D_L)$ 上的 kl 为边,寻到 $C(D_R)$ 上的顶点 r,构成 $\triangle klr$,假设满足正点条件,还满足起始边条件,那么 r 称为正点,$\triangle klr$ 称为正桥接三角形面。

左切边界线跳转到边 lm,lm 从正点 r 开始寻点,假设不满足正点条件,直到跳转到顶点 t,lm 与 t 构成 $\triangle lmt$ 面,假设满足正点条件,那么可以确定 t 为正点,假设不满足正副边条件,那么 lm 与 n 构成正补三角形面。然后以 ln 为边界与 t 构建 $\triangle \ln t$ 面,假设判断不满足正副边条件,那么 ln 与 o 构成正补三角形面。那么以 lo 为边构建 $\triangle lot$ 面,假设判断满足正副边条件,可以确定 $\triangle lot$ 为第二个正桥接三角形面。假设满足副点条件,这两个正桥接三角形面的交点 l 为副点。此刻有了两个不同的正点,然后从右切边界环正点 r 开始起边与副点 l 构建副桥接三角形面,以边 rs 与副点 l 构建 $\triangle rsl$,假设判断不满足正副边条件,那么 rs 与 t 构建副补三角形面,因为 t 是下一个正点,那么直接以 rt 为边与 l 构建副桥接三角形面,记录正点 t。同理,按照同样的方法进行桥接,直到形成一个完整的凸多面体。

3 凹多面体生成算法内容

基于凹多面体生成算法的思想,通过对选中边的邻接面的开口朝向进行判断,然后利用四面体互补面替换的方法实现凹多面体的生成。

3.1 开口朝向判断

对于邻接面开口的朝向可以通过非共边顶点连线的中点来判定。设有两个共 AB 边的面,$\triangle ABC$ 和 $\triangle ADB$,二者均是逆时针绕向朝外的,$\triangle ABC$ 是 $A \rightarrow B \rightarrow C \rightarrow A$,$\triangle ADB$ 是 $A \rightarrow D \rightarrow B \rightarrow A$,可以很容易获得 C 和 D 连线的中点 P,通过判断中点 P 相对共 AB 边的 $\triangle ABC$ 和 $\triangle ADB$ 的内外关系判断,具体运算可以按照下式进行判断:

$$F(A,B,C,P)=\begin{cases}>0, & \text{开口朝外}\\ =0, & \text{共面}\\ <0, & \text{开口朝内}\end{cases} \tag{5}$$

3.2 四面体互补面替换

凸多面体的任何一条边都可以唯一确定两个相邻的三角形面,称为原始邻三角形面。如果原始邻三角形面不共面,那么由两个原始邻三角形面的 4 个顶点可唯一确定一个四面体,称为邻面四面体。通过选择对应边,删除对应边相连的原始邻三角形面,并生成由对应边确定的互补三角形面,称为邻面互补三角形面,从而完成替换。如果原始邻三角形面的开口朝内,采用四面体互补面将造成塌陷,如果开口朝外,那么将造成膨胀。其原理如图 8 所示,可以局部修整凹多面体。

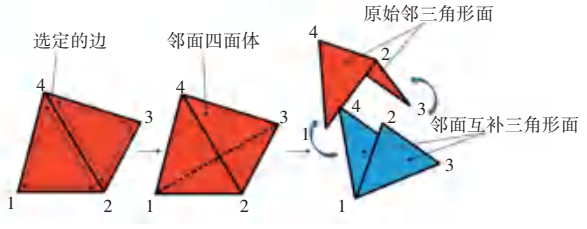


图 8 四面体互补面替换原理图

Fig. 8 Schematic diagram of tetrahedral complementary surface replacement

4 算法步骤与流程

为了详细描述整个算法,对凸多面体和凹多面体的生成将采用一个统一的算法步骤进行描述,如图 9 所示。

Step 1 针对需要建模的凸多面体物体,提供所有顶点的点集合,用一个顶点数组维护点集合中的顶点,并获取所有顶点的空间边界参数;

Step 2 对于顶点数组进行分治剪枝,得到一棵剪去冗余空间的分治二叉树;

Step 3 通过后序遍历分治二叉树,合并左右节点,并合并左右节点下对应的凸多面体对象,如果遍历的当前节点的左右节点有且只有一个存在,转到 Step 4;如果左右节点都存在,左右节点对应的凸多面体对象有且只有一个存在,那么就转到 Step 5,如果凸多面体对象均不存在,那么转到 Step 6,如果凸多面体对象均存在就转到 Step 8;

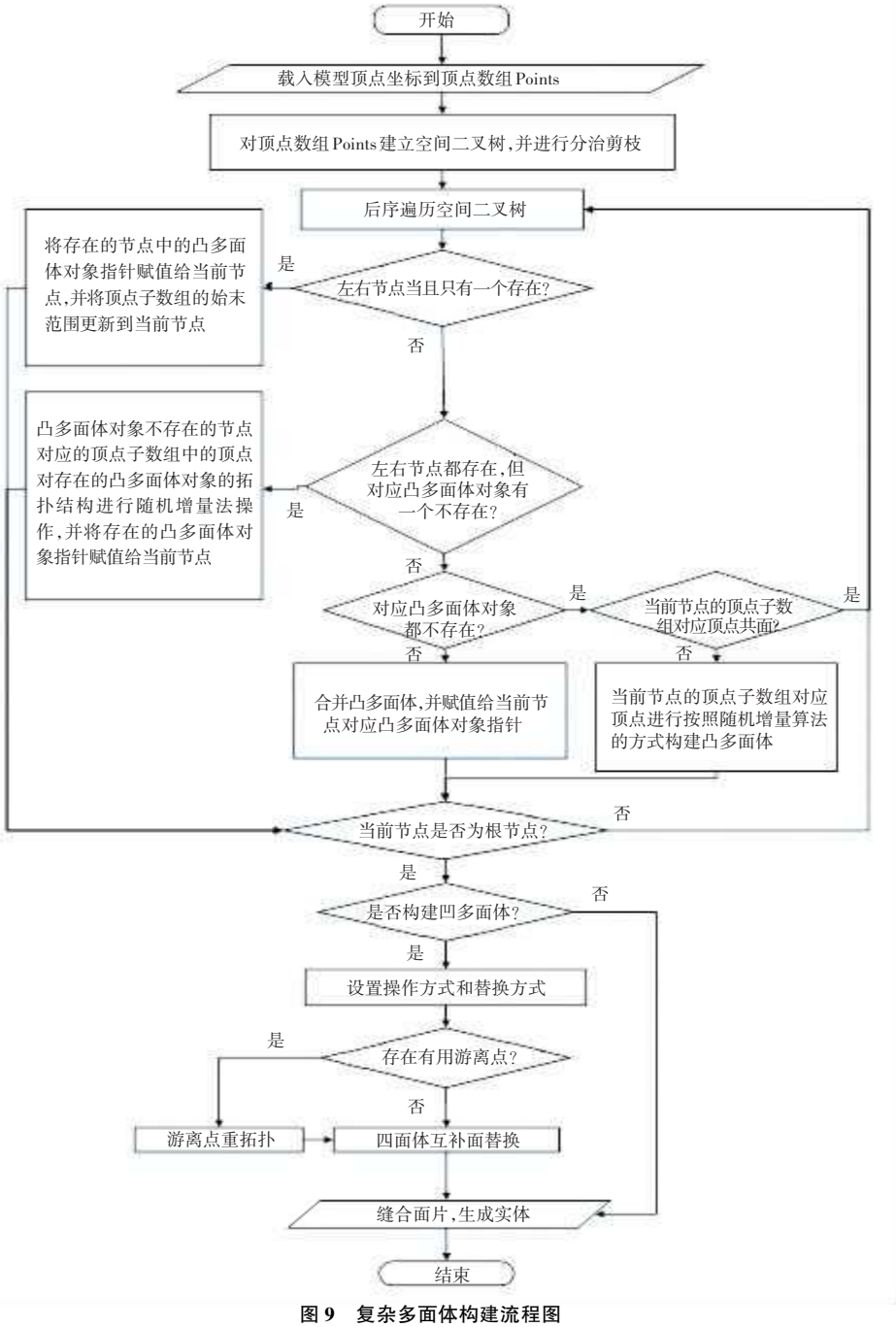


图 9　复杂多面体构建流程图

Fig. 9　Flow chart of complex polyhedron construction

Step 4　当左右节点有且只有一个存在时，将存在的节点的凸多面体对象指针赋值给当前节点的凸多面体对象指针，并将该节点对应顶点子数组的始末范围赋值更新到当前节点；

Step 5　凸多面体对象不存在的节点对应顶点子数组范围内的顶点对凸多面体对象存在的节点的凸多面体对象的拓扑结构采用 Step 7 扩充凸多面体，然后将凸多面体对象存在的节点的凸多面体对象指针赋值给当前节点凸多面体对象指针；

Step 6　若左右节点的凸多面体对象均不存在，那么随机选择当前节点对应顶点数组范围内的 4 个顶点，判断是否共面，如果不共面，为当前节点的凸多面体对象开辟内存，构建初始四面体，然后剩余的点按照 Step 7 的方式扩充凸多面体；如果共面，那么就更换顶点子数组的顶点，直到不共面为止，再按照 Step 7 扩充凸多面体；

Step 7　采用随机增量法将孤立点添加进凸多面体对象中；其中，顶点在没有建立拓扑关系前均认为是孤立点；

Step 8　通过后序遍历，合并左右节点的凸多

面体对象的拓扑结构;

Step 9 通过后序遍历,重复 Step 8 操作,直到将合并的凸多面体传递给根节点,该凸多面体对象的拓扑结构即为载入点集的凸多面体,得到最终建立的凸多面体模型;

Step 10 如果目标复杂多面体存在凹陷,转到 Step 11,如果不存在就转到 Step 17;

Step 11 设置替换方式,塌陷替换或者膨胀替换,默认为塌陷替换;

Step 12 设置操作方式,操作方式有两种,第一种是通过选中点、边或面的方式进行替换,点和面的原理与边的方式一样,其中一个点连接多条边,一个面有三条边,然后按照鼠标选中边的方式依次按照选定的替换方式进行局部替换;第二种是通过规则的几何体(球或者正方体)去选中多面体上的边,凡接触到的线都被选中,然后按照鼠标选中边的方式依次按照选定的替换方式进行局部替换;

Step 13 凹多面体的构造在凸多面体的基础上进行,有两种情况,第一种是无游离点的情况,按照 Step 14 进行,第二种是有游离点的情况,按照 Step 16 进行;

Step 14 按照替换方式和操作方式选中边,然后转到 Step 15。若替换方式为塌陷替换,那么只对共边面朝向向内的边进行操作,若替换方式为膨胀替换,那么只对共边面朝向向外的边进行操作;

Step 15 采用四面体互补面替换方法构建多面体,然后转到 Step 17;

Step 16 对游离的点重新建立拓扑关系,然后转到 Step 14;

Step 17 缝合面片,生成实体,结束。

5 实验与分析

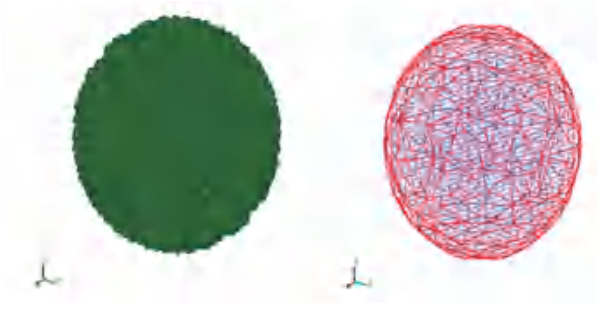
5.1 实验平台与环境

实验设备采用个人电脑,处理器为 Intel(R) Core(TM) i7-7500U CPU,主频为 2.70 GHz,机带 ARM 为 8 GB。实验环境采用 Visual Studio 2015 为编译器,使用 C++作为编程语言构建复杂多面体生成算法,OpenCASCADE 7.3 作为图形渲染工具对复杂多面体进行展示和交互。

5.2 凸多面体

5.2.1 应用实例

由于分子动力学中的晶体模型具有大量规则排列的原子点,并不能直观的辨识其几何结构。可以通过算法就能将分子动力学的晶体模型转换为凸多面体几何模型。如图 10(a),为金属钨的球状晶体模型,钨是体心立方晶格,晶格常数 5.046 nm,总共 4 279 个原子,通过本文算法生成图 10(b)球状凸多面体,顶点数 349 个,三角面片 694 个。



(a) (b)

图 10 分子动力学模型转换成几何模型

Fig. 10 Conversion of molecular dynamics model to geometric model

5.2.2 算法比较

本文为验证凸多面体生成算法效率,与随机增量算法做了对比试验,在一个 200×200×200 的长方体空间中生成指定数量随机点,每组实验反复做 30 次,运行耗时取 30 次的平均值,效果如图 11 所示,对比结果见表 1。

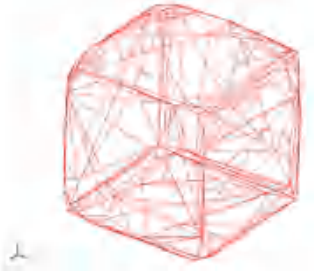


图 11 实验中的凸多面体

Fig. 11 Convex polyhedron in the experiment

表 1 随机增量算法实验结果对比

Table 1 Comparison of experimental results of random increment algorithm

顶点总数/个	外顶点数/个	随机增量算法/ms	本文算法/ms	提升/%
2 000	92	72.9	97.6	-33.9
4 000	106	150.0	173.5	-15.7
6 000	102	230.5	237.5	-3.0
8 000	96	330.0	299.3	9.3
10 000	125	474.2	353.5	25.6
12 000	115	639.5	402.8	37.0
14 000	138	743.7	456.5	38.6
16 000	134	832.3	485.8	41.6
18 000	150	1 018.8	558.0	45.2
20 000	161	1 118.7	607.0	45.7

结合表 1 的数据绘制图 12，从图 12 中可以观察到：本文算法的性能在顶点数少于 6 000 的时候并没有明显的提升，甚至略低，而随着顶点数的继续增加，本算法的耗时增量趋于平缓，而随机增量算法的耗时增量急剧增加。这既体现了本算法的高效性，也体现了本算法对内部冗余点的抗干扰性。

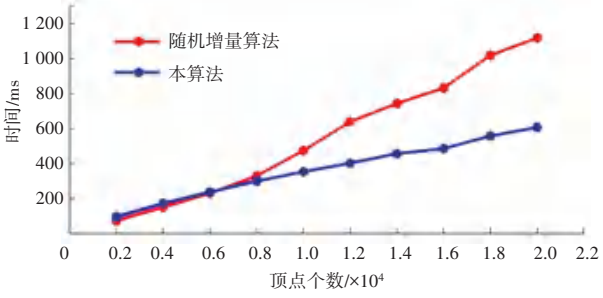


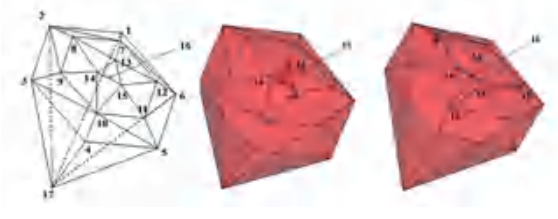
图 12　两种算法的耗时比较

Fig. 12　Time consumption comparison of two algorithms

5.3　凹多面体

凹多面体的构建有助于复杂模型的构建。为进一步验证本文的凹多面体生成算法，构建一个嵌入 Y 形槽的金刚石模型进行验证。

通过凸多面体生成算法生成如图 13(a)所示金刚石模型。其中有一个内部游离顶点 16，通过鼠标交互选中游离顶点 16 和突破三角形面 $\triangle P_{13}P_{14}P_{15}$ 进行游离点重拓扑，重新构建点、线和面的关系，如图 13(b)所示。为了生成 Y 型槽，需要对多面体进行四面体互补面塌陷替换，通过鼠标交互选中边 14-15、边 13-14 和边 13-15，生成嵌入 Y 形槽的金刚石模型最终效果如图 13(c)所示。最终的模型中，该模型由 17 个顶点和 24 个三角形面片组成。



(a)　　(b)　　(c)

图 13　凹多面体的构建示意图

Fig. 13　Construction diagram of concave polyhedron

如图 14 所示，将该凹多面体生成算法应用到分子动力学中，对由凹多面体生成算法生成的嵌入 Y 形槽的金刚石模型进行分子动力学晶体模型转化，可以实现具有复杂形状的分子动力学晶体模型建模。推广到一般场景，众多需要构建凹多面体或者借助凹多面体构建复杂模型的应用场景都可以通过该方法实现，具有较大的应用价值。

(a)　　(b)

图 14　嵌入 Y 形槽的金刚石模型及其晶体模型

Fig. 14　Diamond model embedded in Y-shaped groove and its crystal model

6　结束语

通过改进分治算法，并与随机增量算法结合，形成了新的凸多面体生成算法，新的凸多面体生成算法比传统的随机增量算法在数据点足够多的情况下取得了运算速度的显著提升。并在此基础上进一步提出新的凹多面体生成算法，交互式的凹多面体生成算法可以准确有效的构建目标多面体，可以应对各种复杂形体的建模，从而构成了完整的复杂多面体生成算法。复杂多面体生成算法的实现有利于解决分子动力学领域和其他领域复杂模型建模困难的问题，有着较大的应用价值和研究意义。

参考文献

[1] 邵宁，张德珍. 基于 GPU 的凸包并行设计与研究[J]. 计算机与数字工程，2019，47(10)：2607-2612.

[2] 李志，李儒琼. 一种改进的快速三维凸包生成算法及实现[J]. 计算机工程与科学，2011，33(2)：129-132.

[3] 贾明坤，王伟，张斌，等. 复杂凸多面体随机紧密堆积组构性能的数值研究：形状参数的影响[J]. 计算力学学报，2022，39(3)：273-282.

[4] 张崇峰，陈敏花，侯月阳，等. 冗余空间机械臂试探性搜索障碍规避策略[J]. 上海航天(中英文)，2022，39(2)：1-7.

[5] 陈晶晶，杨旭，赵婷，等. 基于全原子模拟碳化硅纳米切削的材料去除行为分析[J]. 材料导报，2024，38(15)：23030129.

[6] CHAND D R，KAPUR S S. An algorithm for convex polytopes [J]. Journal of the ACM，1970，17(1)：78-86.

[7] PREPARATA F P. An optimal real-time algorithm for planar convex hulls[J]. Communications of the ACM，1979，22(7)：402-405.

[8] MINAKAWA T，SUGIHARA K. Topology-oriented construction of three-dimensional convex hulls[J]. Optimization Methods and Software，1998，10(2)：357-371.

[9] 黄诠，刘浩，梁平元. Delaunay 三角网的并行构网算法[J]. 测绘学，2017，42(6)：171-177.

[10] CLARKSON K L，SHOR P W. Algorithms for diametral pairs and convex hulls that are optimal，randomized，and incremental [C]//Proceedings of the Fourth Annual Symposium on Computational Geometry. New York：ACM，1988：12-17.

[11] 徐志，许宏丽. 一种基于凸包近似的快速体积计算方法[J]. 计算机工程与应用，2013，49(21)：177-179.

[12]BARBER C B, DOBKIN D P, HUHDANPAA H. The quickhull algorithm for convex hulls [J]. ACM Transactions on Mathematical Software (TOMS), 1996, 22(4): 469-483.
[13]EDELSBRUNNER H, KIRKPATRICK D, SEIDEL R. On the shape of a set of points in the plane[J]. IEEE Transactions on Information Theory, 1983, 29(4): 551-559.
[14]EDELSBRUNNER H. Smooth surfaces for multi-scale shape representation[C]//Proceedings of International Conference on Foundations of Software Technology and Theoretical Computer Science. Cham:Springer,1995: 391-412.
[15]翟羽行. 汽车座椅点云数据的网格与曲面重构处理[D]. 长春:吉林大学,2013.
[16]李云帆,谭德宝, 高广, 等. 双阈值 Alpha Shapes 算法提取点云建筑物轮廓研究[J]. 长江科学院院报,2016,33(11):1-4.
[17]李庆, 高祥伟, 费鲜芸, 等. 利用 Alpha-shape 算法进行树冠三维模型构建[J]. 测绘通报,2018,57(12):91-95.
[18]付昱兴,李承明,朱江,等. Alpha-shape 算法构建枣树点云三维模型[J]. 农业工程学报,2020,36(22):214-221.
[19]BERNARDINI F, MITTLEMAN J, RUSHMEIER H, et al. The ball-pivoting algorithm for surface reconstruction[J]. IEEE Transactions on Visualization and Computer Graphics, 1999, 5(4): 349-359.
[20]胡丝兰, 周明全, 税午阳, 等. 一种改进 Ball Pivoting 的散乱点云数据重建算法[J]. 系统仿真学报, 2015, 27(10): 2446-2452.