DOI:10.13203/j.whugis20120702　　文章编号:1671-8860(2014)03-0346-07

含内孔多面体的约束 Delaunay 四面体剖分算法

李昌领[1]　张　虹[1]　朱良峰[2]

1　中国矿业大学环境与测绘学院,江苏 徐州,221008

2　华东师范大学地理信息科学教育部重点实验室,上海,200062

摘　要:针对四面体网格生长算法数据量大和效率低的问题,引入分离面的概念,建立了分离面定理、线段与平面不相交判定定理、三角面与平面不相交判定定理,把线段与三角面的不相交检测问题转化为较为容易计算的分离面与三角面的不相交检测问题。在此基础上,给出了一个完整的基于多面体内外边界面的三维约束 Delaunay 四面体网格直接生长算法。实验表明,算法运行稳定,剖分结果正确,较少用户干预,具有很高的自动化水平。

关键词:多面体剖分;约束 Delaunay 四面体;顶点对可见性;分离面;分离面定理

中图法分类号:P208　　**文献标志码**:A

三维任意域的约束 Delaunay 四面体化算法大致分成两类[1,2],一类是边界一致的 Delaunay 四面体化算法[3-5],另一类为约束 Delaunay 四面体化算法[6-9]。前者会增加许多内部或边界额外顶点,原来的一个约束面会分解成多个约束子面;后者在不增加额外顶点或增加少量必要额外点(由于出现了 Schönhardt Polyhedron[10] 情况,导致该类多面体不能直接被四面体化)条件下完成空间任意域的约束 Delaunay 四面体化。在约束边界恢复上,Lewis 等[11]、杨钦等[9]采用"穿透法"恢复边界边、边界面;Barry Joe[12]、Chen 等[13]先通过 4 种基本的变换($T23$、$T32$、$T22$、$T44$)恢复约束边和约束面,对一些没有恢复的约束边,通过插点分解相交的四面体;Ghadyani 等[1]先用待恢复的约束面形成空洞,然后调用 LAST RESORT 算法把空洞剖分成四面体,直至所有约束面恢复;Shewchuk[7] 提出 gift-wrapping 算法,以四面体种子生长为基础生成约束 Delaunay 四面体网格。本文在三维任意域约束 Delaunay 四面体化算法的基础上,提出一种高效的基于多面体内外表面(约束面)的约束 Delaunay 四面体网格直接生长算法。

1　算法描述

1.1　算法思想

利用分离面快速检测多面体中全部顶点对的可见性,从多面体外表面中任选一个三角面,计算三角面的 3 个顶点的公共可见顶点集合,排除集合中位于三角面反面的顶点,把其中每个顶点作为第 4 点分别与三角面组成多个假想四面体。若某四面体空间中包含其他三角面,则该四面体的第 4 顶点应从集合中删除。分别计算集合中各顶点与三角面组成四面体的外心、外心与三角面的有符号距离,其中与三角面最小距离的外心对应的顶点称为最佳第 4 点,它与当前三角面组成第一个约束 Delaunay 四面体。

分别以第一个约束 Delaunay 四面体新形成的 3 个面为种子,向四面体外部寻找最佳第 4 点,分别生成新的约束 Delaunay 四面体。如此重复,直到多面体内部空间全部剖分成四面体。

若多面体中还有少量空间无法四面体化,则用边界一致的 Delaunay 四面体化算法,通过加入中间点和中间面,把剩下的空间剖分成四面体。

1.2　顶点与顶点相互可见性快速检测

顶点与顶点可见性检测是整个剖分算法的关键所在。值得注意的是,算法中要检测所有顶点对的可见性,且检测每对顶点可见性时必须与几乎所有多面体的面进行求交测试,当多面体顶点数、面数很多时,工作量会很大。

本文对待检测的每对顶点及其连成的线段,用分离面相关技术,高效排除大部分甚至绝大部

收稿日期:2013-09-11

项目来源:国家自然科学基金资助项目(40902093)。

第一作者:李昌领,博士生,讲师,研究方向为计算机图形学、3D 仿真、3D GIS, 3D GMS。E-mail: lcl07@263.net

通信作者:张虹,教授。E-mail:hongzh@cumt.edu.cn

分不可能与线段相交的三角面，大大降低了顶点对可见性检测的时间复杂度。

定理 1　给定不共面空间线段 P_1P_2 和平面 Ω，P_1P_2 不在平面 Ω 内。令 S_1、S_2 分别为 P_1、P_2 到平面 Ω 的有符号距离，若 S_1 和 S_2 不为 0 且为同号，则线段 P_1P_2 必不与平面 Ω 相交。

证明　① 假设 P_1P_2 与平面 Ω 相交，且交于某个端点，如 P_2，则 S_2 必为 0，与题设矛盾；② 假设 P_1P_2 与平面 Ω 相交，且交于 P_1P_2 线段内部一点，则 P_1、P_2 必位于平面 Ω 的两侧，那么 S_1、S_2 中必有一个大于 0，另外一个小于 0，与题设矛盾。定理得证。

定理 2　给定不共面空间三角面 ABC 和平面 Ω，令 S_1、S_2、S_3 分别为 A、B、C 到平面 Ω 的有符号距离，若 S_1、S_2、S_3 不为 0 且为同号，则空间三角面 ABC 必不与平面 Ω 相交。

证明　① 假设三角面 ABC 与平面 Ω 相交，且交于三角面的某个端点（如 A）或两个端点（如 A、B），则 $S_1=0$ 或 $S_1=0$ 且 $S_2=0$，与题设矛盾；② 假设三角面 ABC 与平面 Ω 相交，且交于三角面 ABC 的非端点处，则 P_1、P_2、P_3 中必有一端点位于平面 Ω 的一侧，其余两端点位于平面 Ω 的另一侧，则 S_1、S_2、S_3 中至少有一个小于 0，一个大于 0，与题设矛盾。定理得证。

定理 3（分离面定理）　给定不共面的空间线段 P_1P_2 和空间三角面 ABC 且 P_1、P_2 位于空间三角面 ABC 所在平面 Ω 的异侧，构造过 P_1P_2 且分别平行于三角面 ABC 的边 AB、BC、AC 的分离面 Ω_1、Ω_2、Ω_3（若 P_1P_2 与某边平行，则取过 P_1P_2 且垂直于面 ABC 的分离面），若存在任一 $\Omega_i\in\{\Omega_1、\Omega_2、\Omega_3\}(1\leqslant i\leqslant 3)$，使得三角面 ABC 的三个顶点至平面 Ω_i 的有符号距离不为 0 且为同号，则空间线段 P_1P_2 和空间三角面 ABC 必不相交。如图 1 所示。

证明　根据定理 2 知，空间三角面 ABC 必不与平面 Ω_i 相交，所以空间三角面 ABC 与 Ω_i 内部任何一点都不相交。由于线段 P_1P_2 完全位于平面 Ω_i 内，于是空间三角面 ABC 不与线段 P_1P_2 上任一点相交，也就是说，空间线段 P_1P_2 和空间三角面 ABC 必不相交。

基于分离面的顶点 P_i、P_j 可见性快速检测算法思想描述如下。

第一步，借助于分离面，快速排除那些不可能与线段 P_iP_j 相交的面（一次点积、一次乘法、一次加法）。构造一个过 P_iP_j 且平行于某个坐标轴的分离面 Ω（P_iP_j 在该轴上投影长度最小），如图 2

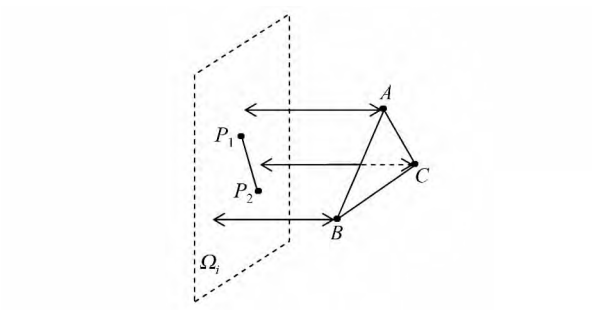


图 1　分离面定理示意图

Fig. 1　Sketch of Separating-plane Theorem

所示。利用定理 2 把分离面 Ω 与多面体上所有三角面进行快速相交测试，那些与分离面不相交的面（也不会与 P_iP_j 相交）均被排除，与分离面相交的面称为粘连面。图 2 中 Ω 与 Ω_1、Ω_2 内部相交，与 Ω_3 交于一个顶点，与 Ω_4 交于一条边，Ω_1、Ω_2、Ω_3、Ω_4 为粘连面，$\Omega_5\sim\Omega_{10}$ 为非粘连面。若多面体的面很多且大小比较均衡，绝大部分三角面都将被排除。

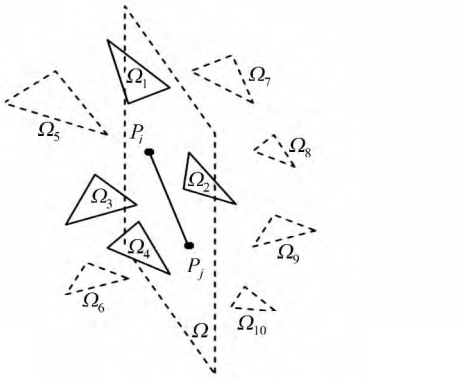


图 2　使用分离面快速排除不相交面示意图

Fig. 2　Sketch of Quickly Eliminating Non-intersection Faces by Separating-plane Ω

第二步，进一步检测线段 P_iP_j 与所有粘连面是否存在相交情况。按照线段与三角面的空间位置关系可分成共面和非共面两种情况。

先讨论线段 P_iP_j 与三角面共面的情形，此时 P_i 和 P_j 可见时只有三种情况，如图 3 所示。其他空间位置关系均表示 P_i、P_j 不可见。

快速检测方法是：判断 P_iP_j 是否为三角面的一条边（二个端点重合），若是，P_i、P_j 可见，如图 3（c）；否则，构造过 P_i、P_j 且垂直于三角面的分离面，检测线段的一个端点是否与三角面某个顶点重合且另一端点不在三角面内部。若是，如图 3（b），P_i、P_j 可见；否则，用分离面定理判断是否属于图 3（a）的情况。若是，P_i、P_j 可见，否则 P_i、P_j 相对于三角面 ABC 不可见。不可见的所有情况下，P_iP_j 与三角面的空间关系无需作更进一步的

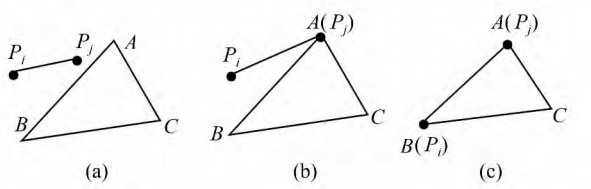


图3 多面体顶点P_i、P_j连线(线段)与三角面共面且P_i、P_j相互可见的情况

Fig. 3 Cases of Visibility Between P_i and P_j when P_iP_j is Coplanar with Triangle ABC

判断。

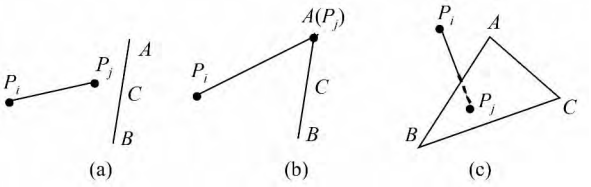


图4 多面体顶点P_i、P_j连线(线段)与三角面不共面且P_i、P_j相互可见的情况

Fig. 4 Cases of Visibility Between P_i and P_j When P_iP_j Is not Coplanar with Triangle ABC

不共面情况下顶点P_i、P_j可见的快速检测方法是:首先快速检查是否有图4(b)情况,若有,P_i、P_j相对于三角面ABC可见;否则,用定理1检测是否存在图4(a)情况。若存在,P_i、P_j可见;否则,用分离面定理检测图4(c)的情况。若满足分离面定理的条件,P_i、P_j可见;否则,P_i、P_j相对于三角面ABC相互不可见。

1.3 四面体最佳第4顶点的选择

已知一个三角面及其法线,在其法线正向一侧寻找第4个顶点来生成新四面体。

首先,据§1.2方法分别取得三角面每个顶点的可见点集合V_1、V_2、V_3,令$V=V_1\cap V_2\cap V_3$,为3个顶点的公共可见顶点集合;其次,排除V中不在法线一侧的点;再次,对V中的每个顶点,与三角面构成假想四面体,借助于定理1、定理2、分离面定理和分离轴定理[15],判断有没有其他面相交于四面体内部,若有,从V中删除该顶点。

接下来从V中选择构成四面体的最佳第4点。求出V中各候选点与三角面构成四面体的外心[16],分别计算各外心与三角面的有符号距离,其中与三角面距离最小的外心对应的顶点就是最佳第4顶点。寻找第4点的形象描述如图5所示,在法线N的正向一侧为三角面ABC寻找第4点,P_1、P_2、…为候选点,S_p为ABC的外接球,开始时球心位于N负方向无穷远处。球心沿N方向运动,球面S_p向N方向动态膨胀,若S_p碰到点(如P_1)即停止运动,该点(P_1)即为最佳第4点。

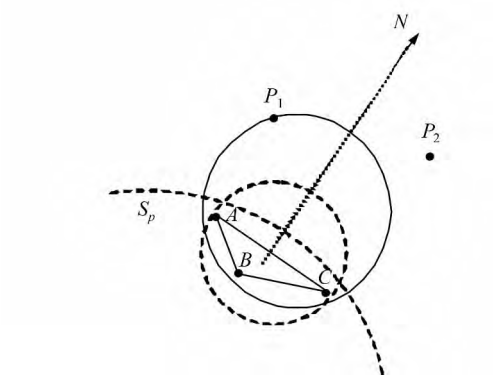


图5 基于三角面ABC外接球寻找四面体第4点模型图

Fig. 5 Model of Finding the Fourth Vertex of a Tetrahedron Based on Circumspheres Passing Through Triangle ABC

1.4 数据结构

参考三维边界表示法模型(B-Rep模型),设计了满足算法特殊性的数据结构。

顶点表。每个顶点除了空间坐标信息外,增加了顶点与平面距离是否存在标志及距离存储变量,避免顶点与面距离的重复计算。

面表。用来存储多面体的外表面、内表面和四面体生长过程中的中间面。

顶点的可见点链表。保存了与该顶点相互可见的顶点集合,链表的头指针存放于顶点表中。

顶点的邻接点链表和邻接面链表。每个顶点一个,其头指针存储于顶点表中。

生长面队列。动态存储待生长的三角面。

1.5 算法设计

1.5.1 基于多面体内外表面(约束面)的约束Delaunay四面体网格直接生长算法

1)初始化顶点表、面表、点的邻接点链表、点的邻接面链表和四面体剖分结果集T;

2)生成所有顶点的可见点链表;

3)初始化生长面队列Queue;

4)从多面体外环面中任选一个三角面△,面△进入队列Queue;

5)从队头取出一个三角面△,其顶点记为A、B、C,据顶点的可见点链表得到A、B、C的公共可见点集V,删除V中不在△法线正向一侧的点;把V中每一点分别与三角面△形成假想四面体,若该四面体包含全部或部分其他面,应从V中删除该点;

6)从点集V中选择最佳第4点P,增加3个面APB、BPC、CPA(若已存在,便无需新增),与三角面△组成一个新四面体Γ;另外,再增加

APB、BPC、CPA 的 3 个反面 PAB、PBC、PCA，以便据此向外生长（若 APB、BPC、CPA 中有的面已存在，则无需增加相应的反面）；新增的反面顺序加入队列 Queue，把面△、APB、BPC、CPA 打上完成标记，四面体 Γ 放入结果集 T；

7）调整顶点的可见点链表，把因新面 PAB、PBC、PCA 的加入导致的不可见点删除；

8）调整顶点的邻接面链表、邻接点链表，把因新面 PAB、PBC、PCA 的加入而增加的邻接面、邻接点信息分别插入相应顶点的邻接面链表、邻接点链表；

9）检查队列 Queue 是否为空，为空则转步骤 10），否则转步骤 5）；

10）检查多面体中是否还有未被剖分的空间，若有，用边界一致的 Delaunay 四面体化算法把该空间剖分成四面体，直至多面体空间全部剖分成四面体。

1.5.2　所有顶点对可见性检测算法

多面体中所有顶点对可见性检测算法为：

```
voidGenerateVisiblePointList()
{
for （第一个顶点索引 i=0；i<顶点总数；i++）
  for（第二个顶点索引 j = 0；j < i；j++）  {
   if（顶点 vj 为 vi 邻接点）{
   vj 与 vi 相互可见，顶点 vj 插入到 vi 的可见点链表，
   顶点 vi 插入到 vj 的可见点链表；
   }
   else if（（vi、vj 为外环面上的顶点）or（vi、vj 均
        为同一个内环面 k 上的顶点））{
        取得 vi 的所有邻接面并分别求 vj 到这些邻接
        面的有符号距离；
        if（若距离中有一个小于 0）  // vi、vj 连成
          的线段完全不在或不全在四面体中，vj 与
        vi 相互不可见；
        else {
          取得 vj 的所有邻接面并分别求 vi 到这些邻
          接面的有符号距离；
          if（若距离中有一个小于 0）// vi、vj 连成
        的线段不在或不全在四面体中，vj 与 vi 相互
        不可见。
        else if  (IsVisibleEachOther（vi，vj））
          //进一步判断 vi、vj 是否可见。vj 与 vi 相互
          可见，vj 插入到 vi 的可见链表，vi 插入到 vj 的
          可见链表；
  }
}
else if（IsVisibleEachOther（vi，vj））  {
     vj 与 vi 相互可见，顶点 vj 插入到 vi 的可见链表，
     顶点 vi 插入到 vj 的可见链表；
   }
  }
}
```

1.5.3　单对顶点的可见性检测算法

判断两个顶点是否可见是借助分离面快速计算并排除大部分与 v_i、v_j 不相交的面，再依据定理 1、2 及分离面定理确定剩下的少量粘连面是否与 v_i、v_j 相交。

```
BOOL IsVisibleEachOther（vi，vj）
{
  S0：外环面上三角面的集合；
  S1～Sk：内环面 1 - k 上三角面的集合；
S：粘连面集合，开始时为空；
BOX1～ BOXk：内环面 S1～Sk 的 AABB 包围盒；
vivj：顶点 vi、vj 连成的线段；
//(1)快速排除大部分不可能与 vivj 相交的多面体面
构造过 vivj 且平行于某个坐标轴的分离面△（vivj 在
该轴上的投影长度最小）；
for （每个三角面 f ∈外环面 S0）{
  计算面 f 的 3 个顶点与面△距离 d1、d2、d3；
  if（d1、d2、d3 不为 0 且同号），面 f 不会与 vivj 相
  交，排除面 f；
  else  可能相交，面 f 为粘连面，加入 S；
}
for （对每一个内环面 Si 中，i∈(1，k)）{
  线段 vivj 与 BOXi 相交测试；
  if（不相交）vi、vj 相对于内环面 Si 中所有面均可
见，Si 中所有面均被排除；
else {
  for （每个三角面 f ∈内环面 Si）{
   计算面 f 的 3 个顶点与分离面△的有符号距离
d1、d2、d3；
   if（d1、d2、d3 不为 0 且同号），面 f 不会与 vivj 相
交，面 f 排除；
   else  可能相交，面 f 为粘连面，加入 S；
  }
 }
}
//(2)进一步确定 S 中的粘连面是否与 vivj 相交；
for （每个三角面 f ∈S）{
  visible = IsVisibleOnOneFace(vi，vj，f)；
  if(！visible)  // vi、vj 相对于 f 不可见，不必再
对其他面测试；
    return FALSE；
  }
  return TRUE；
}
```

1.5.4　单对顶点相对于某三角面的可见性检测算法

判断两个顶点 v_i、v_j 相对于某个三角面 f 的

可见性算法 IsVisibleOnOneFace 伪码如下：

```
BOOLIsVisibleOnOneFace( v_i , v_j , f )
{
  A、B、C：面 f 的3个顶点；
  d_1 = v_i 到面 f 的有符号距离(若 v_i 是面 f 的一个顶点，则 d_1 为0，无需计算)；
  d_2 = v_j 到面 f 的有符号距离(若 v_j 是面 f 的一个顶点，则 d_2 为0，无需计算)；
  if(线段 v_iv_j 为面 f 的一条边)  return TRUE;  //可见；
  if ( 线段 v_iv_j 与面 f 不共面(d_1、d_2 不全为0)){
    if( (d_1、d_2 中有一个为0)  或  (d_1、d_2 不为0且同号) ) {
      v_i、v_j 相对于面 f 可见，不再进行任何测试；return TRUE; }
  //经过上面的判断，d_1、d_2 异号，即 v_i、v_j 位于三角面 f 所在平面的两侧；
  //构造分离面并应用分离面定理判断面 f 与分离面是否相交。构造过 v_iv_j 且平行于边 AB 的分离面△_AB(若 v_iv_j 与 AB 平行，取过 v_iv_j 且与面△垂直的分离面)；
  if (A、B、C 分别至平面△_AB 的有符号距离不为0且为同号) return TRUE;
  构造过 v_iv_j 且平行于边 BC 的分离面△_BC(若 v_iv_j 与 BC 平行，分离面取法同上)；
  if (A、B、C 分别至平面△_BC 的有符号距离不为0且为同号)  return TRUE;
  构造过 v_iv_j 且平行于边 AC 的分离面△_AC(若 v_iv_j 与 AC 平行，分离面取法同上)；
  if (A、B、C 分别至平面△_AC 的有符号距离不为0且为同号)  return TRUE;
  return FALSE; // 相交，v_i、v_j 相对于面 f 不可见
  }
  else ( 线段 v_iv_j 与面 f 共面){
  if  (端点 v_i、v_j 与面 f 的顶点 A、B、C 有一个重合) {
  构造过 v_iv_j 且垂直于面 f 的分离面 f_s；
  求重合顶点对边的两个端点到分离面 f_s 的距离 d_1、d_2；
  if ( d_1、d_2 同号且不为0 ) {
    v_i、v_j 相对于面 f 可见，return TRUE; }
  }
  构造过 v_iv_j 且垂直于面 f 的分离面△；
  if (A、B、C 分别至平面△的有符号距离不为0且为同号)  return TRUE;
  构造过 AB 且垂直于面 f 的分离面△；
  if (v_i、v_j、C 分别至平面△的有符号距离不为0且为同号)  return TRUE;
  构造过 BC 且垂直于面 f 的分离面△；
  if(v_i、v_j、A 分别至平面△的有符号距离不为0且为同号)  return TRUE;
  构造过 AC 且垂直于面 f 的分离面△；
  if (v_i、v_j、B 分别至平面△的有符号距离不为0且为同号)  return TRUE;
  return FALSE; // 其他情况下，均认为 f 阻碍了 v_i、v_j 的可见性；
  }
  return FALSE;
}
```

2 算法时间复杂度分析

本算法时间主要耗费在两个方面，一是顶点对的可见性判定，二是生成四面体及由此产生的新面对顶点对可见性影响的调整，总的算法时间复杂度为 $O(n_v^2 \cdot \sqrt{n_f} + n_v \cdot n_t)$，其中 n_v 为多面体顶点数(包含内孔上的顶点)，n_f 为多面体内、外表面上的约束三角面的总数，n_t 为结果四面体的总数。

Lewis 普通 CDT 算法的时间复杂度为 $O(n_v \cdot n_{t1} + n_f \cdot n_{t2} + n_f \cdot n_{t3})$，其中 n_{t1} 为逐点插入法形成的标准四面体网格中四面体的个数(时间主要花在点的定位、局部优化上)，n_{t2} 为边恢复后的四面体个数(边恢复要分裂原四面体)，n_{t3} 为面恢复后的四面体个数。

Shewchuk 的 gift-wrapping 算法的时间复杂度是 $O(n_v n_f n_t)$，Ghadyani 的 LAST RESORT 算法的时间复杂度与 gift-wrapping 算法的类似。

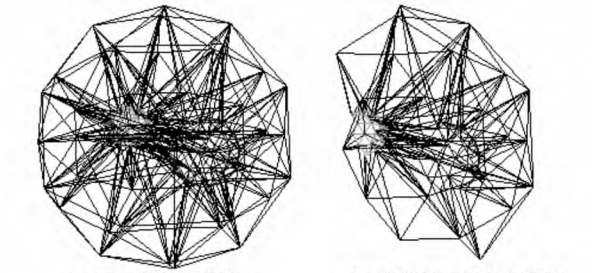
从上述分析看出，与几种典型的同类算法相比，本算法有较大优势。

3 实验结果

构造了两套多面体模型用于算法的测试。一套多面体模型 M1 外表面为球状，内部含两个球状的孔洞，顶点数和面数相对较少；另一套多面体模型 M2 参照了开滦精煤股份有限公司某矿业分公司的煤层数据，设计了一个地层模型，内部模拟了两个不规则的透镜体孔洞，顶点数和边界面数较多。使用本文的算法分别对两套模型进行了约束 Delaunay 四面体剖分，如图6、图7所示。

另外分别模拟了 Lewis 普通 CDT 算法和 Shewchuk 的 gift-wrapping 算法，在开发环境为 VS2005 VC++/OpenGL，运行环境为 IBM Thinkpad R60i T2450 双核处理器上，分别对上述模型进行了四面体剖分，运行时间如表1所示。

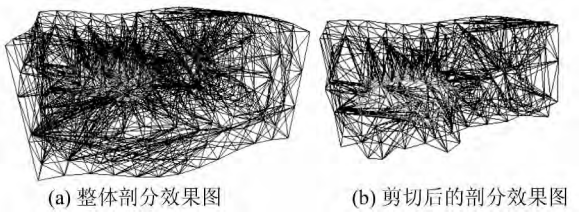
实验表明，所有顶点和边界三角面均参与了四面体的剖分，原多面体空间全部剖分成了约束 Delaunay 四面体，结果正确并符合预期要求。



(a) 整体剖分效果图　(b) 剪切后的剖分效果图

图 6　含两个内孔的简单多面体的约束 Delaunay 四面体网格剖分结果示意图

Fig. 6　Sketch of Tetrahedralization on a Simple Polyhedron with Two Holes



(a) 整体剖分效果图　(b) 剪切后的剖分效果图

图 7　含两个透镜体地层的约束 Delaunay 四面体剖分结果示意图

Fig. 7　Sketch of Tetrahedralization on a Stratum Model Including Two Lenticular Bodies

表 1　3 种不同算法的四面体剖分时间比较

Tab. 1　Comparisons of Running Time Among Three Algorithms

多面体	顶点数	边界面数	本文算法结果四面体数	本文算法执行时间	Lewis 普通 CDT 算法时间	Shewchuk 的 gift-wrapping 算法时间
多面体 M1	126	240	683	5 s	16 s	11 s
多面体 M2	518	1 024	2 595	4 min 41 s	1 h 17 min 8 s	10 min 23 s

参　考　文　献

[1] Ghadyani H, Sullivan J, Wu Ziji. Boundary Recovery for Delaunay Tetrahedral Meshes Using Local Topological Transformations[J]. *Finite Elem Anal Des*, 2010, 46(1-2): 74-83

[2] Du Qing, Wang Desheng. Constrained Boundary Recovery for Three Dimensional Delaunay Triangulations[J]. *International Journal for Numerical Methods in Engineering*, 2004, 61:1 471-1 500

[3] Murphy M, Mount D M, Gable C W. A Point-Placement Strategy for Conforming Delaunay Tetrahedralization[C]. The Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, 2000

[4] Cohen-Steiner D, Verdiere E C, Yvinec M. Conforming Delaunay Triangulations in 3D[J]. *Computational Geometry-Theory and Applications*, 2004, 28(23): 217-233

[5] George P L, Hecht F, Saltel E. Automatic Mesh Generator with Specified Boundary[J]. *Computer Methods in Applied Mechanics and Engineering*, 1991, 92(3): 269-288

[6] Shewchuk J R. Sweep Algorithms for Constructing Higher-Dimensional Constrained Delaunay Triangulations[C]. The Sixteenth Annual Symposium on Computational Geometry, 2000

[7] Shewchuk J R. Constrained Delaunay Tetrahedralizations and Provably Good Boundary Recovery[C]. 11th International Meshing Roundtabl, 2002

[8] Hazlewood C. Approximating Constrained Tetrahedrizations[J]. *Computer Aided Geometric Design*, 1993,10: 67-87

[9] Yang Qing, Xu Yongan, Chen Qiming, et al. Research on 3D Constrained Delaunay Triangulation[J]. *Journal of Computer Aided Design and Computer Graphics*, 2000,12(8):590-594(**杨钦,徐永安,陈其明,等. 三维约束 Delaunay 三角化的研究**[J]. **计算机辅助设计与图形学学报**,2000,12(8): 590-594)

[10] Verbree E. Piecewise Linear Complex Representation Through Conforming Delaunay Tetrahedronization[C]. 4th International Conference on Geographic Information Science,2006

[11] Lewis R W, Zheng Yao, Getthin D T. Three-dimensional Unstructural Mesh Generation: Part 3[J]. *Computer Methods in Applied Mechanics and Engineering*,1996,134(3-4): 285-310

[12] Barry J. Construction of Three-dimensional Constrained Triangulations[EB/OL]. http:// members. shaw. ca, 2008-10-01

[13] Chen Jianjun, Zheng Yao. Redesign of a Conformal Boundary Recovery Algorithm for 3D Delaunay Triangulation[J]. *Journal of Zhejiang University SCIENCE A*, 2006, 7(12): 2 031-2 042

[14] Wu Jiangbin, Zhu Hehua. Delaunay Tetrahedralization in an Arbitrary Domain[J]. *Journal of Image and Graphics*, 2007, 12(11): 2 109-2 113(**吴江斌, 朱合华. 3 维任意域内点集的 Delaunay 四面体化研究**[J]. **中国图像图形学报**,2007,12(11): 2 109-2 113)

[15] Akenine-Möller T, Haines E. Real-time Rendering

[M]. Beijing: Peking University Press, 2004(Akenine-Möller T, Haines E. 实时计算机图形学[M]. 北京: 北京大学出版社,2004)

[16] Zhong Zheng, Fan Qibin, Zhang Yeting. Algorithm for Fast Constructing Tetrahedral Network from Three Dimensional Dispersed Data[J]. *Journal of Institute of Surveying and Mapping*, 2004, 21(4): 286-291(钟正, 樊启斌,张叶挺. 3维离散数据四面体快速生成算法研究[J]. 测绘学院学报, 2004,21(4): 286-291)

Algorithm for Dividing a Polyhedron with Holes into Constrained Delaunay Tetrahedrons

LI Changling[1] *ZHANG Hong*[1] *ZHU Liangfeng*[2]

1 School of Environmental Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221008, China

2 Key Laboratory of GISciences for the Ministry of Education, East China Normal University, Shanghai 200062, China

Abstract: To solve the problem of poor calculating efficiency caused by mass data existed in tetrahedral growth algorithm, the concept of a separating-plane is introduced while a separating-plane theorem, and theorems for segment-plane and triangle-plane disjoint tests are established. By transforming the segment-triangle disjoint test into the easier disjoint test between a separating-plane and triangle, a large amount of triangles to be intersected with a segment are eliminated efficiently, greatly shortening testing time. On the basis of the above theorems, a complete algorithm for a direct constrained-Delaunay tetrahedralization based on the boundaries of a polyhedron is presented. Experimental results show that the algorithm runs stably and correctly, has a higher level of automation because of less artificial intervention, and possesses higher efficiency compared with other similar algorithms.

Key words: polyhedron dividing; constrained Delaunay tetrahedron; visibility between a pair of vertexes; separating-plane; separating-plane theorem

First author: LI Changling , PhD candidate, lecturer, specializes in computer graphics, 3D emulation, 3D GIS, 3D GMS. E-mail: lcl07@263.net

Corresponding author: ZHANG Hong, professor. E-mail: hongzh@cumt.edu.cn

Foundation support: The National Natural Science Foundation of China, No. 40902093.