

基于线性方程的三维地理围栏算法研究



刘美, 高佳宁, 孙芊, 杨富钧, 廖雪花*

四川师范大学计算机科学学院, 四川成都 610101

摘要: 三维空间地理围栏是基于位置服务 (LBS, Location Based Services) 的一种新应用, 它可以给围栏关联者提供实时的、基于立体位置变化的相关服务, 其核心是判断目标点与围栏区域的位置关系, 可抽象为几何数学中点与空间图形位置关系判别, 即点包含问题。目前, 主流的射线法在边界判断存在奇异性问题, 虽有诸多改进算法被提出, 但多数算法受限于二维形式, 对于复杂的空间立体环境, 无法满足高精度位置判别需求。因此本文从求解线性方程组的思想出发, 将点与多面体围栏位置关系的判断转化为点与平面位置关系的判断, 提出一种基于线性方程的三维地理围栏新算法, 可以快速、准确完成目标点与围栏位置关系判断。首先, 结合凸剖分思想和 BSP 树技术对复杂多面体围栏进行预处理并形成二叉树; 其次, 递归查询目标点位于二叉树的位置, 并获取子凸多面体围栏数据; 最后, 使用线性方程算法对目标点与子凸多面体进行点包含判断。实验结果表明, 与改进后的射线法相比, 线性方程新算法在凸多面体围栏位置判别上效率提升 40%-48.49%, 在简单非凸多面体围栏上提升 12.5%-20%。

关键词: 地理围栏; 点包含算法; 线性方程; 凸剖分; BSP 树

DOI: [10.57237/j.cst.2023.02.004](https://doi.org/10.57237/j.cst.2023.02.004)

Research on 3D Geographic Fence Algorithm Based on Linear Equation

Liu Mei, Gao Jianing, Sun Qian, Yang Fujun, Liao Xuehua*

School of Computer Science, Sichuan Normal University, Chengdu 610101, China

Abstract: Three-dimensional spatial geo-fencing is a new application based on location-based services (LBS, Location Based Services), which can provide real-time, three-dimensional location change-based related services to fence associates, and its core is to judge the location relationship between target points and fence area, which can be abstracted as point and space figure location relationship discrimination in geometry and mathematics, i.e., point inclusion problem. At present, the mainstream ray method has singularity problem in boundary judgment, and although many improved algorithms have been proposed, most of them are limited to two-dimensional form, which cannot meet the demand of high-precision position discrimination for complex spatial three-dimensional environment. Therefore, this paper starts from the idea of solving a system of linear equations, transforms the judgment of the position relationship between points and polyhedral fences into the judgment of the position relationship between points and planes, and proposes a new algorithm of 3D geo-fencing based on linear equations, which can quickly and accurately complete the judgment of the position relationship between target points and fences. Firstly, the complex polyhedral fence is pre-processed and formed

基金项目: 国家社会科学基金一般项目 (20BMZ092).

*通信作者: 廖雪花, liao_xuehua@163.com

收稿日期: 2023-02-23; 接受日期: 2023-04-10; 在线出版日期: 2023-04-25

<http://www.computscitech.com>

into a binary tree by combining the idea of convex dissection and BSP tree technology; secondly, the target point is recursively queried to be located in the binary tree and the sub-convex polyhedral fence data is obtained; finally, the linear equation algorithm is used to judge the point inclusion between the target point and the sub-convex polyhedron. Experimental results show that the new linear equation algorithm is 40%-48.49% more efficient in convex polyhedral fence position discrimination and 12.5%-20% more efficient in simple nonconvex polyhedral fences compared with the improved ray method.

Keywords: Geographic Fence; Point Inclusion Algorithm; Linear Equation; Convex Subdivision; BSP Tree

1 引言

移动设备的日益普及与定位技术的不断发展,使人们在生活中获取、使用的多数信息均与位置服务有关。基于位置的服务(LBS, Location Based Services)成为与人们日常生活息息相关的基本服务,是移动互联网产业发展的重要支撑[1-2]。

地理围栏(Geo-fencing)是基于LBS的一种新应用,其相关技术成为业界近年来的研究热点之一[3-4]。目前,二维地理围栏技术发展不断成熟,其研究成果应用于各行业,包括零售业[2]、畜牧业、交通运输[5]、智能家居[6-7]和旅游业等,凡是持带定位功能移动设备的用户均可通过地理位置信息获得相关服务。但是,在一些领域仅通过二维的平面围栏区域判断位置是不精确的,立体地理围栏技术也产生大量应用需求,如无人机探测[8-9]、地理信息采集[10]、高空设备巡检和居家健康监测等。

立体地理围栏技术的核心点在于如何判断围栏与终端的位置关系,可以将其抽象为数学中的点与多面体(终端视为点,地理围栏视为多面体)关系判别问题(也称点包含问题,研究点包含问题的算法称为点包含算法[11-12]),是地理围栏技术的基础算法。近年来业界对点包含算法展开大量研究工作,其中,射线法的研究热度最高。文献[13]提出三维点包含转换为与二维点包含一致的算法,通过射线法的奇偶性判断三维点包含,降低了算法复杂度。文献[14]中提出虚交点概念,将射线经过多边形或多面体的某些顶点(一部分或者全部)视为虚交点,不计入内外点判断,减少奇异性判断。文献[15-16]中对传统射线法存在的奇异性问题进行了重新划分,虽然解决了临界处判断失误情况,但产生额外时间开销。此外,点包含算法中角度法和四面体法是通过计算对待测点与多面体顶点连线的夹角之和或体积之和判断点与多面体围栏关系,但对精度要求极高。

基于上述背景,本文针对三维空间地理围栏技术研究现状,提出基于一种基于线性方程的三维地理围栏新算法,能够快速、准确的判定围栏关联者与围栏的位置关系。

2 线性方程算法原理分析

本章详细介绍了点与凸多面体位置关系判别的具体设计和完整的实现流程,包括点与平面位置关系、点与凸多面体地理围栏位置关系。

2.1 点与平面位置关系

判断点与平面关系的基本思想是:若某平面的方程为 $ax+by+cz+d=0$,则过空间内任意一点 $p(x_0, y_0, z_0)$ 且与该平面平行的平面方程为 $ax_0+by_0+cz_0+d_0=0$,若 d_0 满足

$$d_0 \begin{cases} < d & \text{点} p \text{位于该平面右(上)方} \\ = d & \text{点} p \text{位于该平面上} \\ > d & \text{点} p \text{位于该平面左(下)方} \end{cases} \quad (1)$$

依据此基本思想,平面与坐标轴的关系可以分为以下几种情况进行讨论:

(1) 平行 XOY 面或 XOZ 面或 YOZ 面

假设空间中有两平面 $z=a$ 、 $z=b$ (a, b 为任意常数,且 $a < b$)与 XOY 面平行(同理 XOZ 面/YOZ 面),如图 1 所示,

围成的空间区域 M 与 XOY 面平行,则位于该区域中的任意一点 $p(x_0, y_0, z_0)$ 满足条件:

$$\begin{cases} z_0 > a & \text{点} p \text{位于平面} z=a \text{的上方} \\ z_0 < b & \text{点} p \text{位于平面} z=b \text{的下方} \end{cases} \quad (2)$$

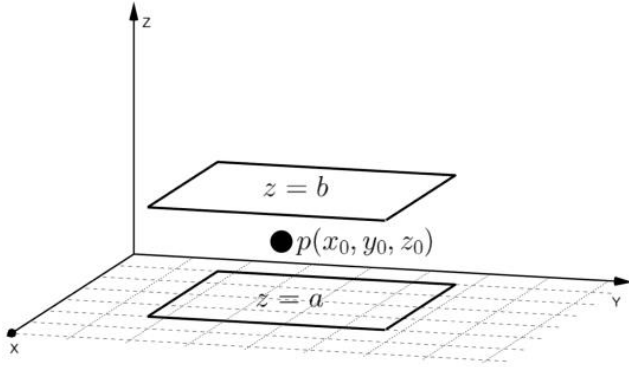


图1 平行 XOY 面的两平面

Figure 1 Two planes parallel to XOY plane

(2) 平面平行 X 轴或 Y 轴或 Z 轴

假设空间中现有两平面 $y+z=a$, $y+z=b$ (a, b 为任意常数, 且 $a < b$) 平行 X 轴 (同理 Y 轴/Z 轴), 如图 2 所示,

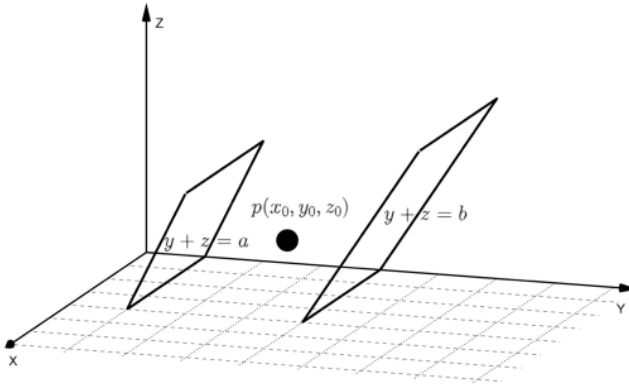


图2 平面过 X 轴

Figure 2 Plane passing X-axis

围成的空间区域 M 平行 X 轴, 则位于该区域中的任意一点 $p(x_0, y_0, z_0)$ 满足条件:

$$\begin{cases} y_0 + z_0 > a & \text{点 } p \text{ 位于平面 } y+z=a \text{ 的上方} \\ y_0 + z_0 < b & \text{点 } p \text{ 位于平面 } y+z=b \text{ 的下方} \end{cases} \quad (3)$$

(3) 平面不平行任一面也不平行任一轴

假设空间中现有两平面 $x+y+z=a$ 和 $x+y+z=b$ (a, b 为任意常数, 且 $a < b$), 如图 3 所示,

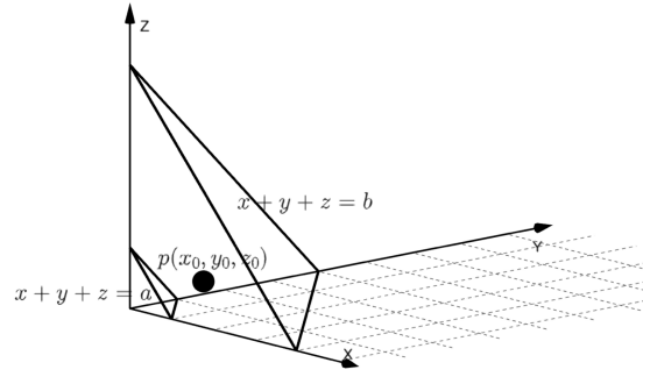


图3 既不平行 XOY 面也不过 X 轴

Figure 3 Neither parallel to XOY plane nor X-axis

围成空间区域为 M, 则位于该区域中的任意一点 $p(x_0, y_0, z_0)$ 满足条件:

$$\begin{cases} x_0 + y_0 + z_0 > a & \text{点 } p \text{ 位于平面 } x+y+z=a \text{ 的上方} \\ x_0 + y_0 + z_0 < b & \text{点 } p \text{ 位于平面 } x+y+z=b \text{ 的下方} \end{cases} \quad (4)$$

算法 1: 点与平面位置关系判别算法 upOrDown

输入: 封闭平面 geoPanel 和待判定点 p

输出: 点与平面的位置关系

1) result ← 计算点 p 所在平行平面 D 值

2) If result 大于 D then

3) Return 1; // 点在平面后 (左) 方

4) Else If result 小于 D then

5) Return -1; // 点在平面前 (右) 方

6) Else If p 不在封闭平面内 then

7) Return 0; // 点在平面区域外

8) Return 2; // 点在平面区域内

2.2 点与凸多面体地理围栏位置关系

2.2.1 凸多面体中点与各平面满足的线性方程组

实际应用场景中, 地理围栏多数为多面体形状, 且凸多面体居多。因此, 假设某地理围栏为如下图 5 所示的多面体区域, 该地理围栏由六个平面围成, 设任意一平面的平面方程为 $ax+by+cz+d=0$, 则该平面方程满足的函数为: $f(x, y, z) = ax+by+cz+d$, 因此在多面体 ABCDEFGH 中。

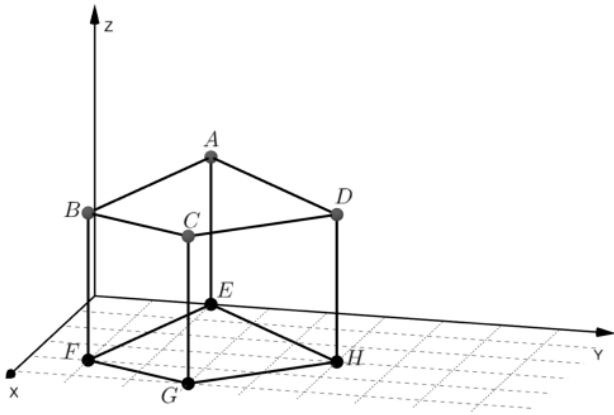


图4 凸多面体

Figure 4 Convex polyhedron

平面 ABCD 对应函数为:

$$f_{ABCD}(x, y, z) = a_1x + b_1y + c_1z + d_1 \quad (5)$$

平面 ABFE 对应函数为:

$$f_{ABFE}(x, y, z) = a_2x + b_2y + c_2z + d_2 \quad (6)$$

平面 ADHE 对应函数为:

$$f_{ADHE}(x, y, z) = a_3x + b_3y + c_3z + d_3 \quad (7)$$

平面 BCGF 对应函数为:

$$f_{BCGF}(x, y, z) = a_4x + b_4y + c_4z + d_4 \quad (8)$$

平面 CDHG 对应函数为:

$$f_{CDHG}(x, y, z) = a_5x + b_5y + c_5z + d_5 \quad (9)$$

平面 HGFE 对应函数为:

$$f_{HGFE}(x, y, z) = a_6x + b_6y + c_6z + d_6 \quad (10)$$

对空间内任意一点 $p(x_0, y_0, z_0)$, 如果同时满足

$$\begin{cases} f_{ABCD}(x_0, y_0, z_0) > 0 \\ f_{ABFE}(x_0, y_0, z_0) > 0 \\ f_{ADHE}(x_0, y_0, z_0) > 0 \\ f_{BCGF}(x_0, y_0, z_0) < 0 \\ f_{CDHG}(x_0, y_0, z_0) < 0 \\ f_{HGFE}(x_0, y_0, z_0) < 0 \end{cases} \quad (11)$$

说明点 p 在多面体 ABCDEFGH 的区域内部, 只要公式 (11) 中任意一个条件不满足, 则表示点 p 在多面体区域外部或者区域边界上。

2.2.2 点与凸多面体位置关系判别

假设现有一个平面数为 n 的凸多面体地理围栏, 取围栏外任意一点为坐标原点, 以该点纬度延长线为 X 轴、经度延长线为 Y 轴和地面向上延长线为 Z 轴(延长线以围栏所在方向为正向), 构建一个空间直角坐标系, 因此, 多面体的平面序列(按从下到上, 逆时针方向)为 $M = \{M_0, M_1, \dots, M_{n-1}, M_n\}$ ($n \in \mathbb{N}$), 其中 M_i ($i \in \mathbb{N}$) 由 m ($m \geq 3$) 个不共线顶点按逆时针方向构成, 待判定点为 $P_p(x_p, y_p, z_p)$, 如图 5 所示。

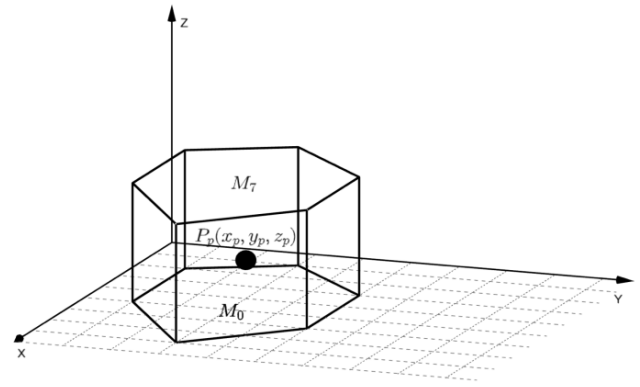


图5 凸多面体地理围栏

Figure 5 Convex polyhedron geographic fence

本文提出的线性方程算法判断点在凸多面体内外的具体求解步骤如下:

(1) 构建围栏平面所在的线性方程组

依次求解地理围栏中所有平面所满足的线性方程。由第 2.1 小节分析可知, 若平面 M 与 XOY 面平行, 即方程 $ax + by + cz + d = 0$ 中 $a = 0$ 且 $b = 0$, 则点 $p_i(x_i, y_i, z_i)$ ($i \in \mathbb{N}$) 构成的平面 M 所满足的平面方程为:

$$z = d \quad (d \text{ 为任意常数}) \quad (12)$$

若平面满足 2.1 节(2)(3) 情况, 则由点 $p_i(x_i, y_i, z_i)$ ($i \in \mathbb{N}$) 构成的平面 M 所满足的平面方程为:

$$ax + by + cz + d = 0 \quad (d \text{ 为任意常数}) \quad (13)$$

其中, 已知平面 M 由三个不共线顶点(按逆时针)构成, 空间中不共线的点 $p_i(x_i, y_i, z_i)$, $p_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$ 与 $p_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$, 则由点 p_{i-1} 、点 p_i 和点 p_{i+1} 构成的矢量分别为:

$$\begin{aligned}\overrightarrow{p_i p_{i-1}} &= (x_{i-1} - x_i, y_{i-1} - y_i, z_{i-1} - z_i) \\ \overrightarrow{p_i p_{i+1}} &= (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i)\end{aligned}\quad (14)$$

由平面法向量的定义, 可得:

$$\overrightarrow{p_i p_{i-1}} \times \overrightarrow{p_i p_{i+1}} = \begin{vmatrix} i & j & k \\ x_{i-1} - x_i & y_{i-1} - y_i & z_{i-1} - z_i \\ x_{i+1} - x_i & y_{i+1} - y_i & z_{i+1} - z_i \end{vmatrix} \quad (15)$$

则平面方程参数为:

$$\begin{cases} a = (y_{i-1} - y_i) \times (z_{i+1} - z_i) - (y_{i+1} - y_i) \times (z_{i-1} - z_i) \\ b = (x_{i+1} - x_i) \times (z_{i-1} - z_i) - (x_{i-1} - x_i) \times (z_{i+1} - z_i) \\ c = (x_{i-1} - x_i) \times (y_{i+1} - y_i) - (x_{i+1} - x_i) \times (y_{i-1} - y_i) \\ d = -(ax_i + by_i + cz_i) \end{cases} \quad (16)$$

(2) 判断待判定点 $P_p(x_p, y_p, z_p)$ 与围栏每个平面的位置关系

在空间中任取三个不共线的点求得平面法向量, 存在内法向量和外法向量两种方向, 为了快速判别点与多面体位置关系, 本文将统一所有平面法向量的方向, 即外法向量[17]。本文规定多面体中所有平面每个顶点都将以逆时针方向排列, 则任取该点集合中的三点所求得的平面方程亦为外法向量, 且逆时针排序的点集合与其外法向量方向满足右手定则。

根据向量判别法思想, 待判定点在凸多面体围栏内部时, 该点皆位于围栏平面的外法向量所指方向的后方, 如图 6 所示。

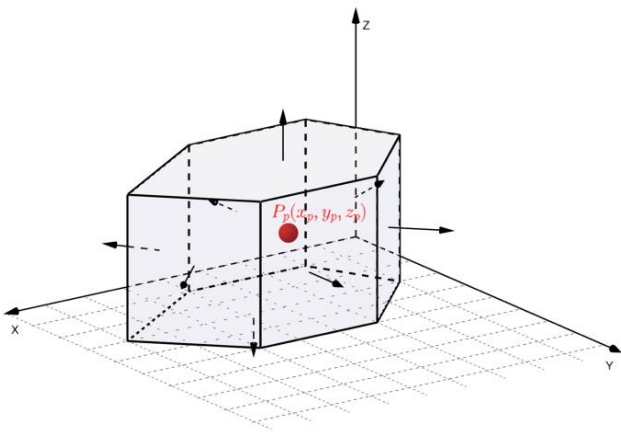


图 6 平面外法向量

Figure 6 Out of plane normal vector

若存在一个平面的外法向量所指方向位于待判定点的前方, 则待判定点一定在该围栏外, 若围栏所有

平面的外法向量所指方向位于待判定点的后方, 则待判定点一定在该围栏内。

算法 2: 点与凸多面体位置关系算法 isInPlane

输入: 多面体模型 panelList 和待判定点 p

输出: 待判定点 p 与多面体的位置关系

1) Foreach geoPanel in panelList do

2) If (geoPanel.upOrDown(p) == -1 ||

geoPanel.upOrDown(p) == 0)

3) Return false; // 点在多面体外部

4) End For

5) Return true; // 点在多面体内

3 简单非凸多面体地理围栏的凸剖分算法

地理围栏的优势在于可准确勾勒出小区、商业楼以及其它目标体的实际形状, 而多数地理围栏的形状可看做是不规则的多面体, 包含许多凹棱, 为了使本文提出的线性方程算法更具有普遍性, 需要先对实际应用场景涉及的简单非凸多面体(无环、无孔)地理围栏进行预处理, 再采用凸剖分技术消除原多面体的凹棱, 将其分割为多个子凸多面体。本章基于 BSP 树对点与非凸多面体的位置关系进行详细分析与实现, 包括多面体地理围栏的凹凸性判定、围栏分割平面与分割点确定和凸剖分完整流程。

3.1 多面体地理围栏的凹凸性判定

若空间内现有一个多面体地理围栏, 如图 7 所示, 由 2.2.2 小节的公式(14~16)可知, 平面 M_i 的外法向量 $\vec{n}_i = (x_i, y_i, z_i)$, 其平面顶点为 $C_{i-2}(x_{i-2}, y_{i-2}, z_{i-2})$ 、 $C_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$ 、 $C_i(x_i, y_i, z_i)$ 和 $C_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$ 。取点 $C_p(x_p, y_p, z_p)$ 为除此 4 个顶点之外的任一顶点, 则由点 C_i 、 C_p 构成的矢量为:

$$\overrightarrow{C_i C_p} = (x_p - x_i, y_p - y_i, z_p - z_i) \quad (17)$$

由空间内矢量点积可知, \vec{n}_i 、 $\overrightarrow{C_i C_p}$ 的矢量点积为:

$$\vec{n}_i \cdot \overrightarrow{C_i C_p} = x_i \times (x_p - x_i) + y_i \times (y_p - y_i) + z_i \times (z_p - z_i) \quad (18)$$

根据向量判别法, 若其余点 C_p 皆满足:

(1) $\vec{n}_i \cdot \overrightarrow{C_i C_p} < 0$, 说明除构成平面 M_i 的其余点皆

在同一侧，则该多面体为凸多面体；

(2) $\vec{n}_i \cdot \overrightarrow{C_i C_p} > 0$ ，说明除构成平面 M_i 的其余点存在异侧，则该多面体为凹多面体；

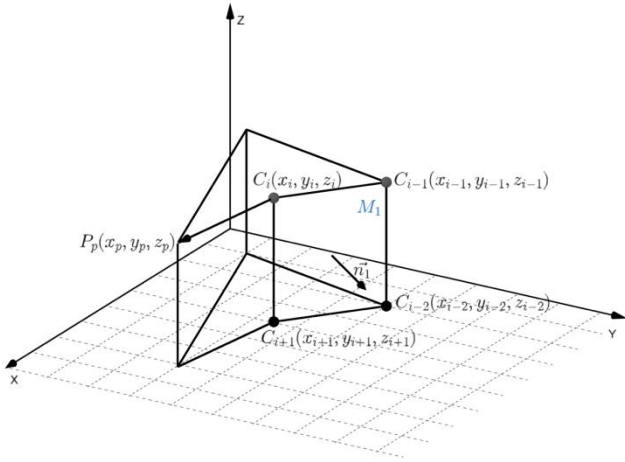


图 7 判断多面体凹凸性

Figure 7 Convexity and concavity of polyhedron

当多面体地理围栏为凹多面体时， $\overrightarrow{C_i C_p}$ 边所在平面且该平面为 M_i 的相邻平面，则该平面与平面 M_i 相交棱即为凹棱。

根据以上分析，多面体地理围栏凹凸性判定算法如下，当检测出多面体某个面存在凹棱，则返回该平面所在索引值，若多面体每个平面都不存在凹棱，则

表示多面体为凸多面体，返回值-1。

算法 3：多面体凹凸性判定算法 isConvex

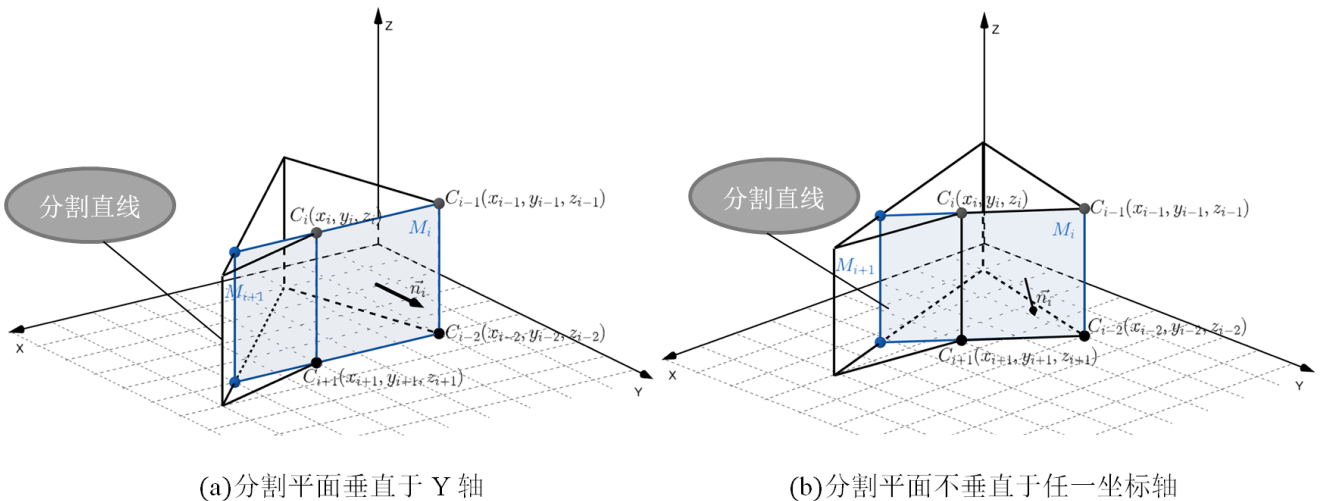
输入：多面体平面集合 panelList

输出：若为凸多面体返回-1，否则返回形成凹棱的第一个平面索引值

- 1) Foreach i in panelList.size() do/
- 2) pVector←panelList.get(i)/
- 3) poVector←(panelList.size().get(i), panelList.size().get(i+1))/
- 4) Return←crossProduct(pVector· poVector) //矢量点积
- 5) If(Return>0) return i;//返回平面索引值
- 6) End For
- 7) Return -1;//说明多面体是凸多面体

3.2 围栏分割平面与分割点确定

若多面体围栏为凹多面体时，假设多面体的某边界面 M_i 、 M_{i+1} 形成的棱 $C_i C_{i+1}$ 为凹棱，其中面 M_i 由顶点 C_{i-2} 、 C_{i-1} 、 C_i 和 C_{i+1} 组成，由 2.2.2 小节公式(14~16)可知，边界面 M_i 所在平面法向量 $\vec{n}_i = (x_i, y_i, z_i)$ 。将面 M_i 延长展开，延长平面第一次与多面体某边界面相交的交线即为分割直线，面 M_i 与分割直线所在平面即为分割平面。分割平面可以分为两种情况：①与坐标轴（X 轴、Y 轴、Z 轴）垂直，②不垂直于任一坐标轴，如图 8 所示。



(a) 分割平面垂直于 Y 轴

(b) 分割平面不垂直于任一坐标轴

图 8 分割平面与分割直线

Figure 8 Division plane and division line

分割平面与多面体围栏某边界面的交线即为分割直线。本文借助向量判别法的思想判断多边形与多边形是否相交，可简要描述为：若多面体某个多边形的所有顶点均在分割平面的同一侧，则这个边界多边形

是否相交，可简要描述为：若多面体某个多边形的所有顶点均在分割平面的同一侧，则这个边界多边形

与分割平面一定没有交点；反之，则分割平面与这个面相交的线即为分割直线，如图 9 所示。其中蓝色为分割平面，多边形 DIJE 与分割平面不相交，多边形 AEJF 与分割平面相交。

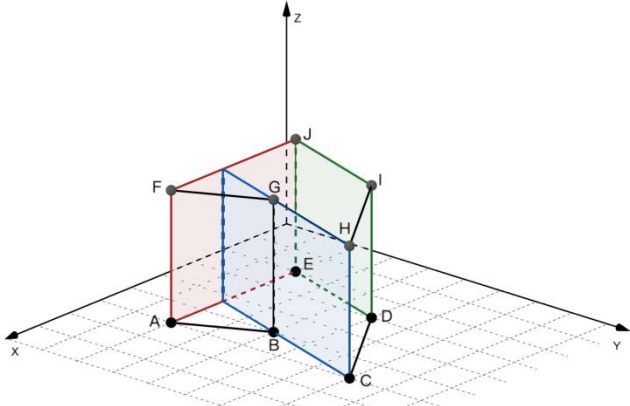


图 9 分割直线确定

Figure 9 Determination of split line

假设空间内现有一个简单凹多面体，已知其某个凹棱顶点为 C_{i-1} 、 C_i ，形成凹棱的边界面分别是面 M_{i-1} 与 M_i ，另外边界面是由顶点 C_{i+1} 、 C_{i+2} 、 C_{i+3} 、 C_{i+4} 组成的面 M_{i+1} ，其中面 M_i 所在平面为分割平面，且法向量为 $\vec{n}=(x,y,z)$ ，由向量判别法可知：

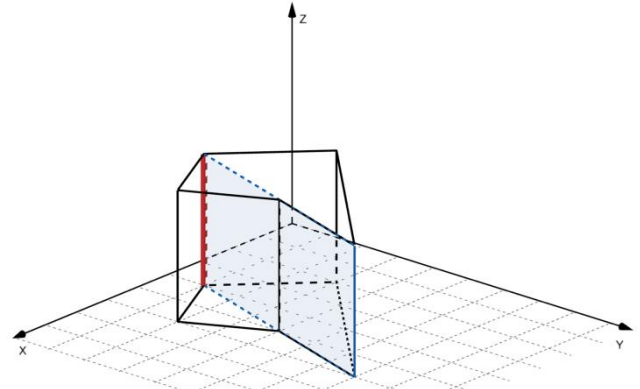
①若分割平面 M_i 与面 M_{i+1} 不相交，则满足：

$$\begin{cases} \vec{n} \cdot \overrightarrow{C_i C_{i+1}} > 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+2}} > 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+3}} > 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+4}} > 0 \end{cases} \text{ 或 } \begin{cases} \vec{n} \cdot \overrightarrow{C_i C_{i+1}} < 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+2}} < 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+3}} < 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+4}} < 0 \end{cases} \quad (19)$$

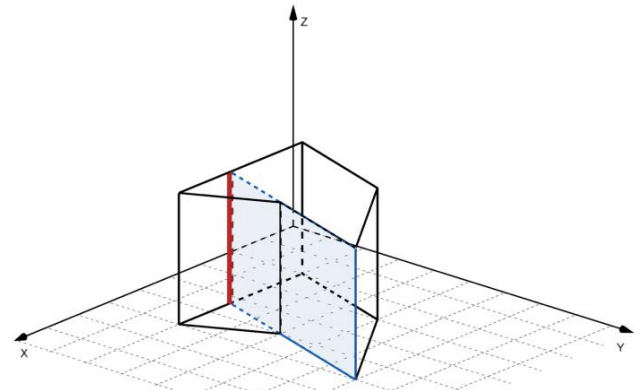
②若分割平面 M_i 与面 M_{i+1} 相交，则满足：

$$\begin{cases} \vec{n} \cdot \overrightarrow{C_i C_{i+1}} \geq 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+2}} \geq 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+3}} < 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+4}} < 0 \end{cases} \text{ 或 } \begin{cases} \vec{n} \cdot \overrightarrow{C_i C_{i+1}} \leq 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+2}} \leq 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+3}} > 0 \\ \vec{n} \cdot \overrightarrow{C_i C_{i+4}} > 0 \end{cases} \quad (20)$$

因此，求得的分割直线可以分为两种情况：①分割直线与多面体某条边重合，②分割直线落在多面体某个面上，与棱不重合，如图 10 所示。



(a) 分割直线与多面体某条棱重合



(b) 分割直线在多面体某个边界面上

图 10 分割直线的位置

Figure 10 Location of the dividing line

若分割平面与多面体围栏的某面相交，相交直线为 L ，直线与该围栏面的边的交点即为分割点。由图 10 可知，直线 L 存在两种形式，当直线 L 与多面体围栏的某条棱重合，则无需新增分割点，分割点即为该棱顶点；当直线 L 与围栏的任一棱皆不重合，则直线 L 与围栏面的相交点位于某棱上。由此，分割点具体求解如下：

(1) 判断分割点是否为围栏顶点

根据公式 (14~16) 可知，分割平面的方程为 $ax+by+cz+d=0$ ，若与分割平面相交的围栏面的顶点序列为 $\{C_{i-2}, C_{i-1}, C_i, C_{i+1}, C_{i+1}\}$ ，其中顶点 C_i 坐标为 (x_i, y_i, z_i) ，则由点与平面位置关系，可知：

$$\begin{cases} ax_i + by_i + cz_i + d = 0 \text{ 表示点 } C_i \text{ 在分割平面上} & (a) \\ ax_i + by_i + cz_i + d \neq 0 \text{ 表示点 } C_i \text{ 不在分割平面上} & (b) \end{cases} \quad (21)$$

当顶点序列中存在一个或两个顶点满足公式 21 (a)，说明该顶点在分割平面上，即分割平面与该面

的相交点（分割点）是顶点序列中的某顶点；反之，顶点序列中所有顶点皆满足公式 21 (b) 时，说明分割平面与该面的分割点是位于该围栏的某棱上。

(2) 确定分割点位置

假设围栏面的某条边顶点 $C_i(x_i, y_i, z_i)$ 、 $C_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$ ，根据点与平面位置关系，当线段(边) $C_i C_{i+1}$ 满足公式 (22)，说明顶点 C_i 、 C_{i+1} 位于分割平面的同一侧，即该线段上不存在分割点；当线段 $C_i C_{i+1}$ 满足公式 (23)，说明顶点 C_i 、 C_{i+1} 位于分割平面的异侧，即分割点在该线段上。

①顶点 C_i 、 C_{i+1} 位于平面同侧，有：

$$\begin{cases} ax_i + by_i + cz_i + d > 0 \\ ax_{i+1} + by_{i+1} + cz_{i+1} + d > 0 \end{cases} \quad \text{或} \quad \begin{cases} ax_i + by_i + cz_i + d < 0 \\ ax_{i+1} + by_{i+1} + cz_{i+1} + d < 0 \end{cases} \quad (22)$$

②顶点 C_i 、 C_{i+1} 位于平面异侧，有：

$$\begin{cases} ax_i + by_i + cz_i + d > 0 \\ ax_{i+1} + by_{i+1} + cz_{i+1} + d < 0 \end{cases} \quad \text{或} \quad \begin{cases} ax_i + by_i + cz_i + d < 0 \\ ax_{i+1} + by_{i+1} + cz_{i+1} + d > 0 \end{cases} \quad (23)$$

(3) 确定分割点坐标

假设线段 $C_i C_{i+1}$ 的两顶点位于分割平面异侧，则线段 $C_i C_{i+1}$ 与分割平面的交点为分割点，交点坐标为 $C_m(x_m, y_m, z_m)$ ，根据直线与平面关系，求解分割点坐标的具体步骤如下：

① 线段 $C_i C_{i+1}$ 所在直线的方向向量为 $\overrightarrow{v_l} = (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i)$ ，则直线参数方程为：

$$\begin{cases} x = x_i + (x_{i+1} - x_i)t \\ y = y_i + (y_{i+1} - y_i)t \\ z = z_i + (z_{i+1} - z_i)t \end{cases} \quad \text{其中 } t \in (0, 1) \quad (24)$$

②将直线参数方程和分割平面的平面方程联立求解，其中点 $C_i(x_i, y_i, z_i)$ 为直线上的一点，点 $C_k(x_k, y_k, z_k)$ 为分割平面上的一点，由此，可得参数 t ：

$$t = \frac{(x_k - x_i) \times a + (y_k - y_i) \times b + (z_k - z_i) \times c}{a \times (x_{i+1} - x_i) + b \times (y_{i+1} - y_i) + c \times (z_{i+1} - z_i)} \quad (25)$$

③根据公式 (24~25) 可得分割点 $C_m(x_m, y_m, z_m)$ 为：

$$\begin{cases} x_m = x_i + (x_{i+1} - x_i)t \\ y_m = y_i + (y_{i+1} - y_i)t \\ z_m = z_i + (z_{i+1} - z_i)t \end{cases} \quad (26)$$

3.3 凸剖分流程

根据上述原理，基于 BSP 树的简单非凸多面体地理围栏凸剖分算法流程可以描述为：

(1) 输入多面体围栏 M 的面集合 List。

(2) 判断围栏 M 的凹凸性。若遇到第一个形成凹棱的平面，则表示围栏 M 是凹多面体，停止判断，并返回该平面在 List 集合中的索引值 index，进入步骤(3)；若判断结束后，仍未遇到能形成凹棱的平面，则表示围栏 M 为凸多面体，进入步骤(6)。

(3) 记录形成凹棱的第一个平面 M_1 和第二个平面 M_2 ，其中平面 M_1 即为分割平面 panel。

(4) 根据分割平面 panel 求得与围栏底面、顶面相交的分割直线，继而求出分割点，同时将新增分割点加入平面 panel 的顶点序列中，并生成一个新的树节点 N，节点 N 的类型为“L”，节点值 Value 为分割平面 panel 的顶点序列。

(5) 根据分割平面 panel，依次判断围栏 M 中每个边界面位于分割平面的右（上）方还是左（下）方。以分割平面 panel 为界限，将面集合 List 分为两部分：leftC、rightC，其中 leftC 包含分割点及凹点形成的平面和位于分割平面左（下）方的围栏平面，将该面集合放入节点 N 的左子树；rightC 包含分割平面以及位于分割平面右（上）方的围栏平面，将该面集合放入节点 N 的右子树。然后递归循环步骤(2)、(3)、(4)、(5)。

(6) 若多面体围栏 M 为凸多面体，生成并返回一个新的树节点 N，节点 N 的类型为“P”，节点值 Value 为多面体 M 的面集合 List。

直到所有分解得到的子多面体 M_i 均为凸多面体，则表示凸剖分完成，算法结束。凸剖分算法流程图如图 11 所示。

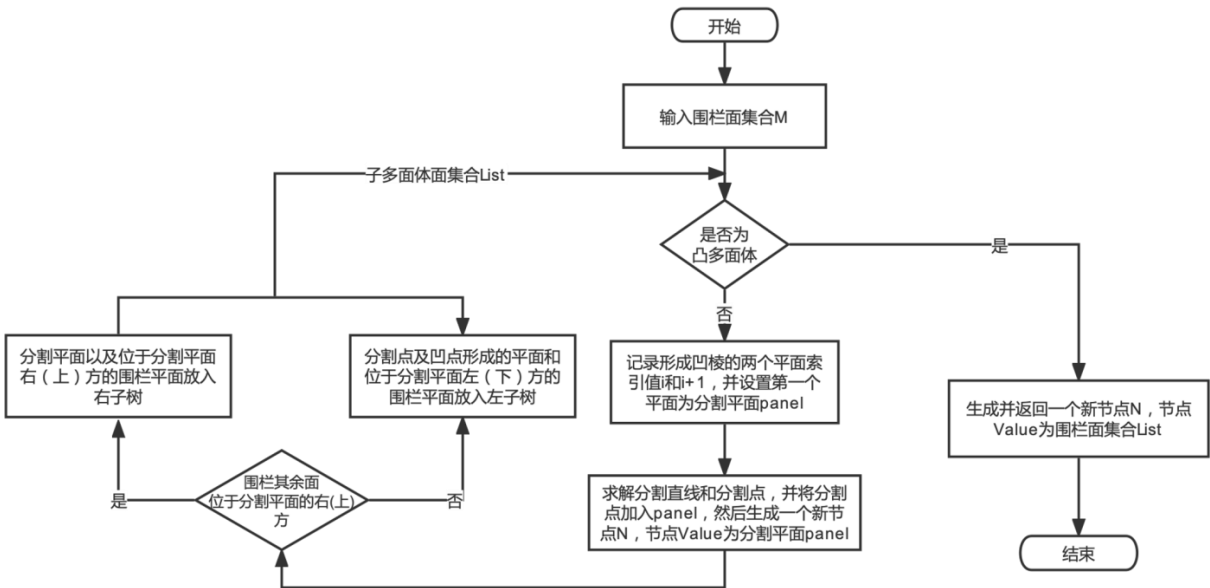


图 11 基于 BSP 树的凸剖分算法流程图

Figure 11 BSP Tree Based Convex Partition Algorithm Flow Chart

根据图 11 提出的基于 BSP 树对简单非凸多面体地理围栏进行凸剖分的原理，本文算法采用二叉树的插入逻辑递归实现围栏的凸剖分，算法如下：

```
算法 4：多面体的凸剖分算法 put
输入：多面体模型 panelList
输出：基于 BSP 的树形结构
1) index←isConvex(panelList); //判断多面体的凹凸性
2) If(index>-1) //表示为凹多面体
3) //获得多面体的分割平面
4) botPanel←panelList.size().get(index)
5) newP←计算底面新增分割点
6) botPanel←keepPal(botPanel,newP); //重构底面顶点数组
7) //构建 BSP 树的左右子树
8) leftC←分割平面左侧的顶点数组
9) rightC←分割平面右侧的顶点数组
10) //返回左右子树的顶点数组
11) subPolyhe (botPanel,leftC,rightC);
12) leftPanel,rightPanel←对 leftC,rightC 进行重构多面体
13) x = new Node("L",botPanel); //新增平面节点
14) x.left = put(x.left, leftPanel); //构建左子树
15) x.right = put(x.right,(Value) rightPanel); //构建右子树
16) Else //为凸多面体
```

```
17) Return new Node("P",panelList); //新增凸多面体节点
```

4 算法测试与结果分析

本文所提基于线性方程的三维地理围栏算法是解决点包含问题的一个新思路、新方法。为验证本文线性方程算法的有效性，本文将从测试围栏形状、测试围栏顶点数量以及待判定点数量等多个方面与改进射线法（待判定点位于多面体边界面或顶点的情况均视为该点位于多面体内部）进行测试比较。测试环境如表 1 所示。

表 1 测试环境配置

Table 1 Test Environment Configuration

名称	配置
操作系统	MacBook Pro(13-inch)
处理器	Apple M1
内存	16GB
测试工具	IntelliJ IDEA 2021.3.3 x64

4.1 凸多面体地理围栏

本文将从多个角度测试线性方程算法与改进射线法的时间耗时、准确度对比情况。将从以下两个纬度展开测试：

(1) 待判定点数量固定，测试围栏顶点数量与算法耗时的关系。

本次测试选取 5 个凸多面体围栏，围栏顶点的数量分别为 10、20、30、40、50，测试围栏附近生成 10

万个待判定点，依次判断待判定点与测试围栏之间的位置关系，线性方程算法与改进射线法判断耗时对比图如图 12 所示。从顶点数量与算法耗时情况可看出，随着测试围栏顶点数量的增加，本文所提线性方程算法和改进射线法的耗时皆在不断增加，但在不同围栏顶点个数的测试中线性方程算法耗时始终低于改进射线法。经计算，线性方程算法在判别效率上比改进射线法提升了 40%-48.49%。

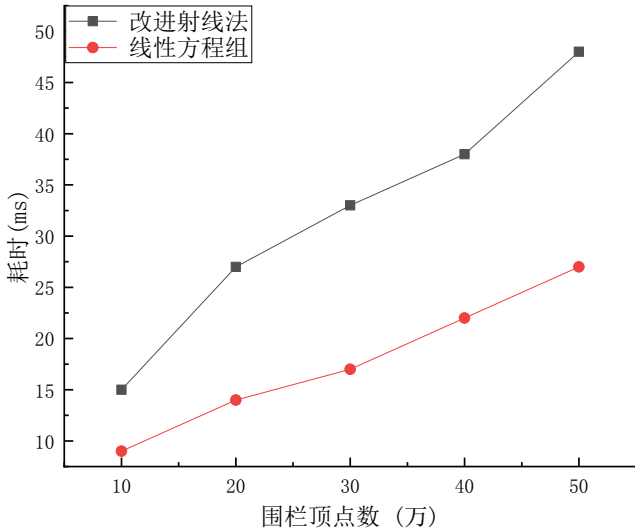


图 12 凸多面体围栏顶点数量与算法耗时的关系

Figure 12 The relationship between the number of fence vertices and the algorithm time consumption

(2) 围栏顶点数量固定，测试待判定点数量与算法耗时的关系。

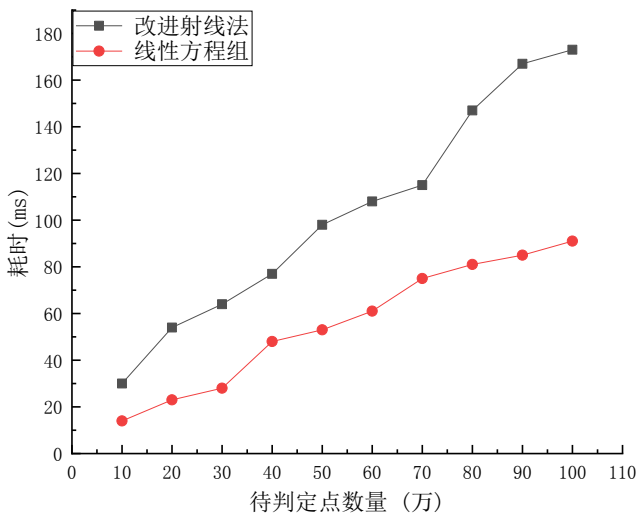


图 13 凸多面体围栏中待判定点数量与算法耗时的关系

Figure 13 The relationship between the number of points to be determined and the algorithm time consumption

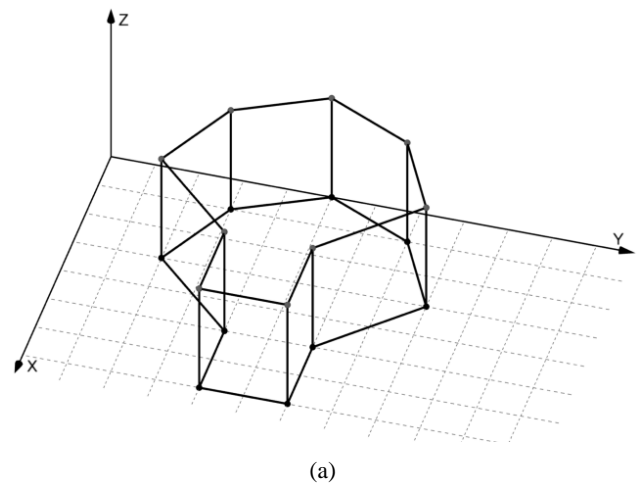
本次测试选取围栏顶点数量为 30 的测试围栏，以该围栏区域为主体，在其附近随机生成 10 万~100 万个待判定点，并对其进行内外点测试。待判定点数量与算法耗时关系如图 13 所示。

从算法耗时对比图 13 可以看出，随着待判定点数量的增加，线性方程算法和改进射线法进行点包含判断时，其耗时均在不断增加，但线性方程算法耗时始终比改进射线法耗时少。经计算，线性方程算法在判别效率上提升了 34.78%-57.41%。

经过本小节测试可知，以凸多面体地理围栏为测试对象，无论是地理围栏不同顶点数还是待判定点不同数量，线性方程算法与改进射线法在耗时上相比始终优于改进射线法，在位置判断效率上本算法有一定程度提升。根据射线法原理，其判断点在地理围栏内外问题时，需对围栏每面进行判断，例如，一个有 N 面的多面体围栏，射线法需要对比 N 次后才能得到结果，其平均时间复杂度为 $O(N^2)$ 。而本文提出的线性方程算法在最好情况下，只需判断一次即可得到结果，此时时间复杂度为 $O(1)$ 。通常情况下，本算法平均时间复杂度亦为 $O(N)$ ，因此，线性方程算法更具有优势。

4.2 凹多面体地理围栏

本小节测试在凹多面体围栏顶点数量固定时，线性方程算法、改进射线法与待判定点数量的耗时关系。以图 14(a)为测试围栏，该围栏共有 18 个顶点，2 条凹棱，经线性方程分割后，得到三个子凸多面体，如图 14(b)所示。



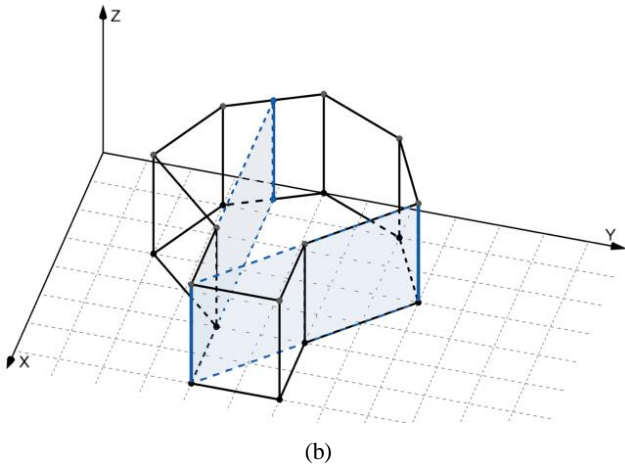


图 14 凹多面体测试围栏

Figure 14 Concave polyhedron test fence

随机在测试围栏周围生成 10 万~50 万个待判定点数量，并对该围栏进行内外点测试，得到待判定点数量与算法耗时的关系如图 15 所示。

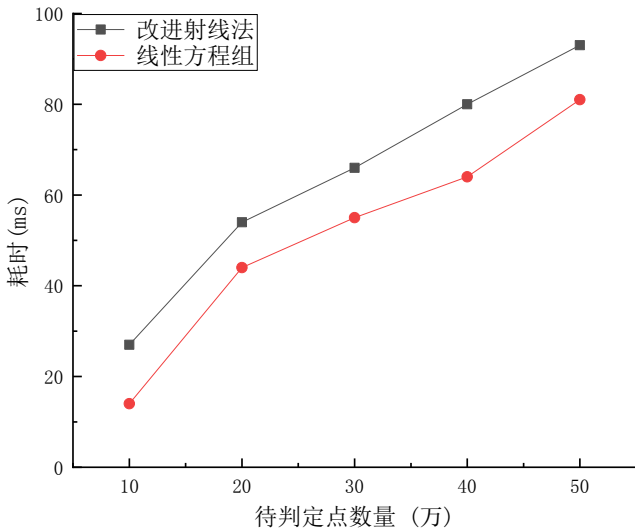


图 15 凹多面体围栏中待判定点数量与算法耗时的关系

Figure 15 The relationship between the number of points to be determined and the algorithm time consumption

从图 15 可知，随着待判定点数量的增加，线性方程算法与改进射线法消耗时长均有增加，但线性方程算法耗时始终低于改进射线法。经计算，线性方程算法在判别效率上比改进射线法提升了 12.5%-20%。

5 结论

在移动互联网技术蓬勃发展的时代背景下，与时空轨迹相关的数据信息和应用技术成为行业内的

研究热点之一。地理围栏技术作为 LBS 主流技术，本文对现有二维地理围栏相关核心算法进行深入研究，并分析其存在的局限性，提出一种基于线性方程的三维地理围栏算法。此算法利用线性方程思想，结合 BSP 树和凸剖分技术实现复杂多面体与点位置关系的判别，相较于改进射线法在性能和准确度方面均有一定提升。结合实验表明，线性方程算法原理简单，能高效、稳定处理任意奇异性，具有普遍适用性。

尽管如此，本文算法仍有需要进一步改进与优化的地方，如线性方程算法在凸剖分时会额外产生新的平面和顶点，构成的 BSP 树不是完全平衡二叉树且剖分方式不适用于有洞的复杂多面体等。

参考文献

- [1] 郭磊, 王晓烨, 田晓龙, 霍德啸. 浅谈物联网时代 LBS 发展应用新模式 [J]. 智能城市, 2020, 6 (07): 19-20. DOI: 10.19301/j.cnki.zncs.2020.07.007.
- [2] 金保可. 室内位置信息服务平台的研究与实现 [D]. 北京邮电大学, 2016.
- [3] 玉山江·艾孜木, 王鑫, 徐畅玥, 曹新辰, 刁雅静. 基于文献计量的地理围栏研究脉络与热点分析 [J]. 现代信息技术, 2021, 5 (24): 108-112+116. DOI: 10.19850/j.cnki.2096-4706.2021.24.028.
- [4] 王静, 刘飞. 基于地理围栏的景点信息推送设计与实现 [J]. 科技视界, 2022 (21): 16-18. DOI: 10.19694/j.cnki.issn2095-2457.2022.21.05.
- [5] 成凯. 基于移动终端传感器的室内地理围栏的研究 [D]. 华中师范大学, 2017.
- [6] 张华伟, 史久琛, 李啸宇, 董苏. 基于 Android 的儿童运动分析及监控系统设计 [J]. 现代计算机 (专业版), 2017 (34): 57-61.
- [7] 黄俊. 基于 Android 系统的远程监控与控制系统的设计与实现 [D]. 宁波大学, 2014.
- [8] 谢东岑, 梁晓龙, 张佳强, 付其喜, 张凯. 无人机地理围栏越界探测算法改进与分析 [J]. 航空工程进展, 2020, 11 (02): 207-213. DOI: 10.16615/j.cnki.1674-8190.2020.02.008.
- [9] 侯岳奇, 陶浩, 龚俊斌, 梁晓龙, 张诺. 多约束条件下无人艇和无人机集群协同航迹规划 [J]. 中国舰船研究, 2021, 16 (01): 74-82. DOI: 10.19693/j.issn.1673-3185.02091.
- [10] 石刚杰, 魏冠楠, 葛修川. 一种航空三维地理信息采集器 [P]. 山东省: CN217424372U, 2022-09-13.

- [11] Vaclav Skala. Point-in-convex polygon and point-in-convex polyhedron algorithms with $O(1)$ complexity using space subdivision [J]. AIP Conference Proceedings, 2016, 1738 (1).
- [12] Wang Wei, Gong Shuiqing, Li Pei Lin, Chui Mingwei. Research on Convex Polyhedron Collision Detection Algorithm Based on Improved Particle Swarm Optimization [P]. mems-12, 2012.
- [13] 魏延生, 张树清, 李华朋, 丁小辉, 刘照. 基于点圆理论的 2D/3D 点包含高效判定 [J]. 中国科学院大学学报, 2018, 35 (03): 353-361.
- [14] 翟艳, 徐卫亚, 张强. 点与多边形或多面体的拓扑关系判断 [J]. 计算机工程与设计, 2015, 36 (04): 972-976. DOI: 10.16208/j.issn1000-7024.2015.04.026.
- [15] 章磊, 何芬, 李鸿赞. 一种基于奇异射线法检测点在多边形内的方法 [J]. 计算机应用研究, 2020, 37 (S2): 133-135.
- [16] 章磊, 何芬, 李鸿赞. 一种基于空间位置进行警情高发统计的方法 [J]. 科技与创新, 2021 (01): 60-61+63. DOI: 10.15913/j.cnki.kjycx.2021.01.020.
- [17] 张洪, 武威, 关晓飞, 张迎宾, 郑路, 武艳强, 陈光齐. 基于局部凸分解的一般多面体接触搜索方法 [J]. 岩石力学与工程学报, 2018, 37 (12): 2709-2720.

作者简介

刘美

1996 年生, 硕士研究生. 研究方向为企业信息化工程.

E-mail: 3321114089@qq.com

高佳宁

1998 年生, 硕士研究生. 研究方向为教育大数据与云计算.

E-mail: gaojn0923@163.com

孙芊

1996 年生, 硕士研究生. 研究方向为软件开发技术.

E-mail: 471658195@qq.com

杨富钧

1996 年生, 硕士研究生. 研究方向为企业信息化工程.

E-mail: 136265871@qq.com

廖雪花

1976 年生, 硕士, 副教授. 研究方向为大数据存储、内存化技术、数据计算与分析.

E-mail: liaoxuehua@163.com