



(12) 发明专利申请

(10) 申请公布号 CN 102213767 A

(43) 申请公布日 2011.10.12

(21) 申请号 201110088548.2

(22) 申请日 2011.04.08

(71) 申请人 浙江工业大学

地址 310014 浙江省杭州市下城区朝晖六区

申请人 杭州普诺科技有限公司

(72) 发明人 徐志江 庄壮 孟利民 王佳阳

(74) 专利代理机构 杭州天正专利事务所有限公
司 33201

代理人 王兵 王利强

(51) Int. Cl.

G01S 19/42 (2010.01)

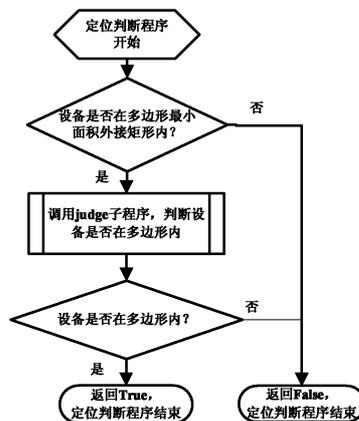
权利要求书 2 页 说明书 5 页 附图 4 页

(54) 发明名称

一种车载 GPS 封闭区域的定位控制方法

(57) 摘要

一种车载 GPS 封闭区域的定位控制方法,封闭区域为多边形,包括以下步骤:(1) 构建所述多边形的凸包;(2) 构建所述多边形的最小面积外接矩形;(3) 定位判断过程如下:3.1) 判断当前点是否在所述最小面积外接矩形内,如果点在最小面积外接矩形外,判定当前点不在所述封闭区域内;如果点在最小面积外接矩形内,进入下一步;3.2) 判断点是否在简单多边形内。本发明提供一种简化计算、速度快、大大减少占用 CPU 资源的车载 GPS 封闭区域的定位控制方法。



1. 一种车载 GPS 封闭区域的定位控制方法,所述封闭区域为多边形,其特征在于:所述定位控制方法包括以下步骤:

(1) 构建所述多边形的凸包,步骤为:

1.1) 在所有点中选取 y 坐标最小的一点作为基点 p0,如果存在多个点的 y 坐标都为最小值,则选取 x 坐标最小的一点;然后按照其它各点 p 和基点构成的向量 $\vec{p_0p}$ 与 x 轴的夹角由小到大进行排序,得到 p1, p2, ..., 如果有相同夹角的情况,则选取与基点距离较远的点;

1.2) 将 p0, p1, p2, ... 加入堆栈 S,作为堆栈的初始值;

1.3) 假如堆栈的栈顶为 B,栈顶前的点为 A,加入一个新的点 C,计算 \vec{AB} 与 \vec{BC} 的叉积是否大于 0,如果大于 0,则把新点 C 加入堆栈;如不是,则将 B 删除,将 C 加入堆栈;

1.4) 反复执行步骤 1.3),直到所有的点都被扫描完;堆栈中保留的点就是凸包的顶点;

(2) 构建所述多边形的最小面积外接矩形,步骤为:

2.1) 选取凸包中一条边作为起始边并对该凸包选中边的左端点为中心旋转使该边平行于坐标横轴,计算并保存其最小绑定矩形的坐标、该边的编号及其旋转角度。

2.2) 顺序选择其它边,按照 2.1) 的方法计算并保存其最小绑定矩形的最表、该边的编号及其旋转角度;

2.3) 比较所得最小绑定矩形的面积,其中,面积最小者按其记录的旋转角度以该边的左端点为中心逆向旋转即为所求的最小面积外接矩形;

(3) 定位判断过程如下:

3.1) 判断当前点是否在所述最小面积外接矩形内,如果点在最小面积外接矩形外,判定当前点不在所述封闭区域内;如果点在最小面积外接矩形内,进入下一步;

3.2) 判断点是否在简单多边形内。

2. 如权利要求 1 所述的一种车载 GPS 封闭区域的定位控制方法,其特征在于:所述步骤 3.2) 包括以下子步骤:

3.2.1) 先设定计数器 count = 0,以待定位设备所在点为端点,作自左向右的射线 L;

3.2.2) 选取多边形的一条边,看点是否在该边上。如果点在边上,则直接判定点在多边形中;如果点不再边上,则进入下一步;

3.2.3) 判断该边是否平行于横轴,如果平行的话,则直接忽略该边,选取下一条边,重复步骤 3.2.2);如果不平行的话,则进入下一步;

3.3.4) 判断该边是否与射线相交,如果不相交,则选取下一条边重复步骤 3.2.2);如果相交,并且交点不是该边的下端点,则计数器 count = count+1;

3.3.5) 选取完所有边之后,判断计数器 count 是否为偶数,如果是奇数,则判断点在多边形内;如果 count 是偶数,则判断点在多边形外。

3. 如权利要求 1 或 2 所述的一种车载 GPS 封闭区域的定位控制方法,其特征在于:所述步骤 3.1) 中,判断当前点是否在所述最小面积外接矩形的过程为:假设当前点为 P,最小面积外接矩形顶点依逆时针顺序分别为 P₁、P₂、P₃、P₄,若各点之间满足条件 $(\vec{P_1P} \times \vec{P_1P_4}) \bullet (\vec{P_2P} \times \vec{P_2P_3}) \leq 0$ 且 $(\vec{P_2P} \times \vec{P_2P_1}) \bullet (\vec{P_3P} \times \vec{P_3P_4}) \leq 0$, 其中, $\vec{P_1P}$ 为从点 P₁

到点 P 的向量, $\vec{P_1P_4}$ 为从点 P₁ 到点 P₄ 的向量, $\vec{P_2P}$ 为从点 P₂ 到点 P 的向量, $\vec{P_2P_3}$ 为从点 P₂ 到点 P₃ 的向量, $\vec{P_2P}$ 为从点 P₂ 到点 P 的向量, $\vec{P_2P_1}$ 为从点 P₂ 到点 P₁ 的向量, $\vec{P_3P}$ 为从点 P₃ 到点 P 的向量, $\vec{P_3P_4}$ 为从点 P₃ 到点 P₄ 的向量, 符号“×”为向量的叉积运算符, 符号“·”表示向量的点积运算符, 则判定点在最小面积外接矩形内, 否则判定在最小面积外接矩形外。

一种车载 GPS 封闭区域的定位控制方法

技术领域

[0001] 本发明涉及车载 GPS 控制技术,尤其是一种封闭区域的定位控制方法。

背景技术

[0002] 近年来, GPS 监控技术在各行业得到了普遍的运用,车载 GPS 监控设备在各种交通运输行业中也得到了深远的发展。在驾驶培训行业,管理单位对车辆需要区域监控,规定了车辆只能在一个封闭的区域内,进入区域或者离开这个区域就要报警。因此设备需要根据当前的 GPS 经纬度,判定是否在封闭区域内。在实际工作中,常常假定封闭区域为一简单多边形,因此问题可以归结到判定设备是否在简单多边形内。设备每秒钟都要刷新当前 GPS 经纬度,也需要每秒钟根据当前的 GPS 经纬度判断设备是否在封闭区域内,这也将会给设备的 CPU 带来沉重的负担。

发明内容

[0003] 为了克服已有的车载 GPS 封闭区域的定位控制方法的计算复杂、速度慢、占用大量 CPU 资源的不足,本发明提供一种简化计算、速度快、大大减少占用 CPU 资源的车载 GPS 封闭区域的定位控制方法。

[0004] 本发明解决其技术问题所采用的技术方案是:

[0005] 一种车载 GPS 封闭区域的定位控制方法,所述封闭区域为多边形,所述定位控制方法包括以下步骤:

[0006] (1) 构建所述多边形的凸包,步骤为:

[0007] 1.1) 在所有点中选取 y 坐标最小的一点作为基点 p_0 ,如果存在多个点的 y 坐标都为最小值,则选取 x 坐标最小的一点;然后按照其它各点 p 和基点构成的向量 $\overrightarrow{p_0p}$ 与 x 轴的夹角由小到大进行排序,得到 p_1, p_2, \dots ,如果有相同夹角的情况,则选取与基点距离较远的点;

[0008] 1.2) 将 p_0, p_1, p_2, \dots 加入堆栈 S,作为堆栈的初始值;

[0009] 1.3) 假如堆栈的栈顶为 B,栈顶前的点为 A,加入一个新的点 C,计算 \overrightarrow{AB} 与 \overrightarrow{BC} 的叉积是否大于 0,如果大于 0,则把新点 C 加入堆栈;如不是,则将 B 删除,将 C 加入堆栈;

[0010] 1.4) 反复执行步骤 1.3),直到所有的点都被扫描完;堆栈中保留的点就是凸包的顶点;

[0011] (2) 构建所述多边形的最小面积外接矩形,步骤为:

[0012] 2.1) 选取凸包中一条边作为起始边并对该凸包选中边的左端点为中心旋转使该边平行于坐标横轴,计算并保存其最小绑定矩形的坐标、该边的编号及其旋转角度。

[0013] 2.2) 顺序选择其它边,按照 2.1) 的方法计算并保存其最小绑定矩形的最表、该边的编号及其旋转角度;

[0014] 2.3) 比较所得最小绑定矩形的面积,其中,面积最小者按其记录的旋转角度以该边的左端点为中心逆向旋转即为所求的最小面积外接矩形;

[0015] (3) 定位判断过程如下：

[0016] 3.1) 判断当前点是否在所述最小面积外接矩形内，如果点在最小面积外接矩形外，判定当前点不在所述封闭区域内；如果点在最小面积外接矩形内，进入下一步；

[0017] 3.2) 判断点是否在简单多边形内。

[0018] 进一步，所述步骤 3.2) 包括以下子步骤：

[0019] 3.2.1) 先设定计数器 $\text{count} = 0$ ，以待定位设备所在点为端点，作自左向右的射线 L ；

[0020] 3.2.2) 选取多边形的一条边，看点是否在该边上。如果点在边上，则直接判定点在多边形中；如果点不再边上，则进入下一步；

[0021] 3.2.3) 判断该边是否平行于横轴，如果平行的话，则直接忽略该边，选取下一条边，重复步骤 3.2.2)；如果不平行的话，则进入下一步；

[0022] 3.3.4) 判断该边是否与射线相交，如果不相交，则选取下一条边重复步骤 3.2.2)；如果相交，并且交点不是该边的下端点，则计数器 $\text{count} = \text{count} + 1$ ；

[0023] 3.3.5) 选取完所有边之后，判断计数器 count 是否为偶数，如果是奇数，则判断点在多边形内；如果 count 是偶数，则判断点在多边形外。

[0024] 再进一步，所述步骤 3.1) 中，判断当前点是否在所述最小面积外接矩形的过程为：假设当前点为 P ，最小面积外接矩形顶点依逆时针顺序分别为 P_1 、 P_2 、 P_3 、 P_4 ，若各点之间满足条件 $(\overrightarrow{P_1P} \times \overrightarrow{P_1P_4}) \cdot (\overrightarrow{P_2P} \times \overrightarrow{P_2P_3}) \leq 0$ 且 $(\overrightarrow{P_2P} \times \overrightarrow{P_2P_1}) \cdot (\overrightarrow{P_3P} \times \overrightarrow{P_3P_4}) \leq 0$ ，其中， $\overrightarrow{P_1P}$ 为从点 P_1 到点 P 的向量， $\overrightarrow{P_1P_4}$ 为从点 P_1 到点 P_4 的向量， $\overrightarrow{P_2P}$ 为从点 P_2 到点 P 的向量， $\overrightarrow{P_2P_3}$ 为从点 P_2 到点 P_3 的向量， $\overrightarrow{P_2P_1}$ 为从点 P_2 到点 P_1 的向量， $\overrightarrow{P_3P}$ 为从点 P_3 到点 P 的向量， $\overrightarrow{P_3P_4}$ 为从点 P_3 到点 P_4 的向量，符号“ \times ”为向量的叉积运算符，符号“ \cdot ”表示向量的点积运算符，则判定点在最小面积外接矩形内，否则判定在最小面积外接矩形外。

[0025] 本发明的技术构思为：先作出一个合理的假设：假定封闭区域的更新频率是非常低的。这种假设是合理的，符合实际情况。这意味着封闭区域所规定的简单多边形不会频繁改变。本发明解决其技术问题所采用的技术方案是：在简单多边形的基础上构建凸包，在凸包上再构建矩形。这意味着矩形包含凸包，凸包又包含简单多边形。构建顶点数为 n 的多边形的凸包的时间复杂度是 $O(n \log n)$ ，而构建顶点数为 m ($m \leq n$) 的凸包的最小面积外接矩形的时间复杂度是 $O(m^2)$ 。

[0026] 在判定的时候，先判定点（设备）是否在矩形中。如果点不在矩形内，则可判定设备在封闭区域外；如果点在矩形内，则通过一般的方法判定点是否在简单多边形内。为了降低 CPU 计算压力，把整个范围内的位置判定分为包含多边形的最小矩形外的判定和最小矩形内的判定两部分。当待判点在矩形外时，所需的时间复杂度为 $O(1)$ ，而传统的方法的时间复杂度为 $O(n)$ ；当待判点在矩形内时，所需的时间复杂度为 $O(n)$ ，与传统的方法相同。这样当 GPS 在矩形外，判定速度比原来的快，计算复杂度低的多；当 GPS 在多边形内，计算速度和复杂度与原来的一样（只多了点是否在矩形内的判定的计算，可以忽略）。这样就降低了整体判定的运算量，对于经常需要重复计算的系统有很好的效果。

[0027] 本发明的有益效果主要表现在：简化计算、速度快、大大减少占用 CPU 资源。

附图说明

[0028] 图 1 是车载 GPS 封闭区域的定位控制方法构建多边形最小外接矩形程序的流程图。

[0029] 图 2 是车载 GPS 封闭区域的定位控制方法定位判断程序的流程图。

[0030] 图 3 是构建凸包子程序的流程图。

[0031] 图 4 是构建凸包的最小面积外接矩形子程序的流程图。

[0032] 图 5 是判断点是否在简单多边形内的子程序的流程图。

具体实施方式

[0033] 下面结合附图对本发明作进一步描述。

[0034] 参照图 1~图 5, 一种车载 GPS 封闭区域的定位控制方法, 所述封闭区域为多边形, 所述定位控制方法包括以下步骤:

[0035] (1) 构建所述多边形的凸包, 步骤为:

[0036] 1. 1) 在所有点中选取 y 坐标最小的一点作为基点 p_0 , 如果存在多个点的 y 坐标都为最小值, 则选取 x 坐标最小的一点; 然后按照其它各点 p 和基点构成的向量 $\overrightarrow{p_0p}$ 与 x 轴的夹角由小到大进行排序, 得到 p_1, p_2, \dots , 如果有相同夹角的情况, 则选取与基点距离较远的点;

[0037] 1. 2) 将 p_0, p_1, p_2, \dots 加入堆栈 S, 作为堆栈的初始值;

[0038] 1. 3) 假如堆栈的栈顶为 B, 栈顶前的点为 A, 加入一个新的点 C, 计算 \overrightarrow{AB} 与 \overrightarrow{BC} 的叉积是否大于 0, 如果大于 0, 则把新点 C 加入堆栈; 如不是, 则将 B 删除, 将 C 加入堆栈;

[0039] 1. 4) 反复执行步骤 1. 3), 直到所有的点都被扫描完; 堆栈中保留的点就是凸包的顶点;

[0040] (2) 构建所述多边形的最小面积外接矩形, 步骤为:

[0041] 2. 1) 选取凸包中一条边作为起始边并对该凸包选中边的左端点为中心旋转使该边平行于坐标横轴, 计算并保存其最小绑定矩形的坐标、该边的编号及其旋转角度。

[0042] 2. 2) 顺序选择其它边, 按照 2. 1) 的方法计算并保存其最小绑定矩形的最表、该边的编号及其旋转角度;

[0043] 2. 3) 比较所得最小绑定矩形的面积, 其中, 面积最小者按其记录的旋转角度以该边的左端点为中心逆向旋转即为所求的最小面积外接矩形;

[0044] (3) 定位判断过程如下:

[0045] 3. 1) 判断当前点是否在所述最小面积外接矩形内; 如果点在最小面积外接矩形外, 判定当前点不在所述封闭区域内; 如果点在最小面积外接矩形内, 进入下一步;

[0046] 3. 2) 判断点是否在简单多边形内。

[0047] 图 1 是本发明的构建多边形的最小面积外接矩形的程序的流程图。在判定一个点是否在封闭区域内之前, 首先获取简单多边形 (也就是封闭区域) 的每个顶点的坐标, 根据这些顶点利用 Graham 扫描法, 构建包含简单多边形的凸包, 获得了凸包的各个顶点的坐标; 在凸包的基础上, 求解凸包的最小面积外接矩形。

[0048] 图 2 是本发明的定位判断程序的流程图。本发明将判断过程分为两步。第一步,

先判断点是否是多边形的最小面积外接矩形内。如果点在矩形外,则无需作第二步,直接可以判定点在简单多边形外;如果点在矩形内,则还不能得知点是否在简单多边形内,还需要第二步进行判断。第二步,判断点是否在简单多边形内。

[0049] 因为矩形也是一种简单的多边形,所以从本质上说,上述判断过程的第一步和第二步其实是一样的,都是判断点是否在简单多边形内。但是矩形是特殊的简单多边形,其结构非常简单、有规律,判断点是否在矩形内的计算量较一般的简单多边形低很多。假设当前点为 P, 矩形顶点依逆时针顺序分别为 P_1 、 P_2 、 P_3 、 P_4 , 若各点之间满足条件 $(\overrightarrow{P_1P} \times \overrightarrow{P_1P_4}) \cdot (\overrightarrow{P_2P} \times \overrightarrow{P_2P_3}) \leq 0$ 且 $(\overrightarrow{P_2P} \times \overrightarrow{P_2P_1}) \cdot (\overrightarrow{P_3P} \times \overrightarrow{P_3P_4}) \leq 0$, 其中, $\overrightarrow{P_1P}$ 为从点 P_1 到点 P 的向量, $\overrightarrow{P_1P_4}$ 为从点 P_1 到点 P_4 的向量, $\overrightarrow{P_2P}$ 为从点 P_2 到点 P 的向量, $\overrightarrow{P_2P_3}$ 为从点 P_2 到点 P_3 的向量, $\overrightarrow{P_2P_1}$ 为从点 P_2 到点 P_1 的向量, $\overrightarrow{P_3P}$ 为从点 P_3 到点 P 的向量, $\overrightarrow{P_3P_4}$ 为从点 P_3 到点 P_4 的向量, 符号“ \times ”为向量的叉积运算符, 符号“ \cdot ”表示向量的点积运算符, 则判定点在矩形内, 否则判定在矩形外。将原本只需要一步的判断过程分为两步, 在很大程度上减少判断所需的计算量。

[0050] 图 3 是本发明的构建凸包子程序的流程图。本发明利用 Graham 扫描法构建一个简单多边形的凸包, 步骤为:

[0051] (1) 在所有点中选取 y 坐标最小的一点作为基点 p_0 。如果存在多个点的 y 坐标都为最小值, 则选取 x 坐标最小的一点。然后按照其它各点 p 和基点构成的向量 $\overrightarrow{p_0p}$ 与 x 轴的夹角由小到大进行排序。如果有相同夹角的情况, 则选取与基点距离较远的点, 其它点不考虑。

[0052] (2) 将 p_0, p_1, p_2 加入堆栈 S, 作为堆栈的初始值。

[0053] (3) 假如堆栈的栈顶为 B, 栈顶前的点为 A。加入一个新的点 C, 计算 \overrightarrow{AB} 与 \overrightarrow{BC} 的叉积是否大于 0。如果大于 0, 则把新点 C 加入堆栈; 如不是, 则将 B 删除, 将 C 加入堆栈。

[0054] (4) 反复执行上步, 直到所有的点都被扫描完。这时堆栈中保留的点就是凸包的顶点。

[0055] 图 4 是本发明的构建简单多边形的最小面积外接矩形子程序的流程图。多边形最小面积外接矩形有两个性质, 第一个性质是任意一个多边形的最小面积外接矩形和其凸包的最小面积外接矩形是等价的, 第二个性质是凸包的最小面积外接矩形至少有一边经过凸包的一边。依据这两个性质得到了任意多边形最小面积外接矩形的算法:

[0056] (1) 计算多边形的最小凸包。

[0057] (2) 选取凸包中一条边作为起始边并对该凸包选中边的左端点为中心旋转使该边平行于坐标横轴, 计算并保存其最小绑定矩形的坐标、该边的编号及其旋转角度。

[0058] (3) 顺序选择其它边, 按照第二步的方法计算并保存其最小绑定矩形的面积、该边的编号及其旋转角度。

[0059] (4) 比较所得最小绑定矩形的面积, 其中面积最小者按其记录的旋转角度以该边的左端点为中心逆向旋转即为所求的最小面积外接矩形。

[0060] 图 5 是本发明的判断点是否在简单多边形内的子程序的流程图。在本发明中, 判断的方法是射线法, 步骤为:

[0061] (1) 先设定计数器 $\text{count} = 0$, 以设备所在点为端点, 作自左向右的射线 L 。

[0062] (2) 选取多边形的一条边, 看点是否在该边上。如果点在边上, 则直接判定点在多边形中; 如果点不再边上, 则进行下一步。

[0063] (3) 判断该边是否平行于横轴。如果平行的话, 则直接忽略该边, 选取下一条边, 重复步骤 (2); 如果不平行的话, 则进行下一步。

[0064] (4) 判断该边是否与射线相交。如果不相交, 则选取下一条边重复步骤 (2); 如果相交, 并且交点不是该边的下端点, 则计数器 $\text{count} = \text{count} + 1$ 。

[0065] (5) 选取完所有边之后, 判断计数器 count 是否为偶数。如果是奇数, 则可判断点在多边形内; 如果 count 是偶数, 则判断点在多边形外。

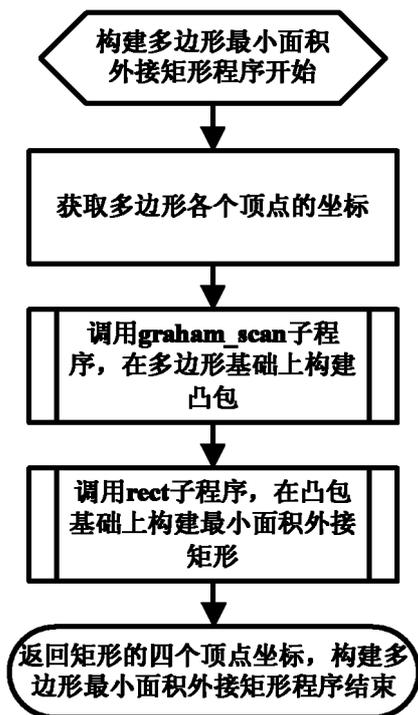


图 1

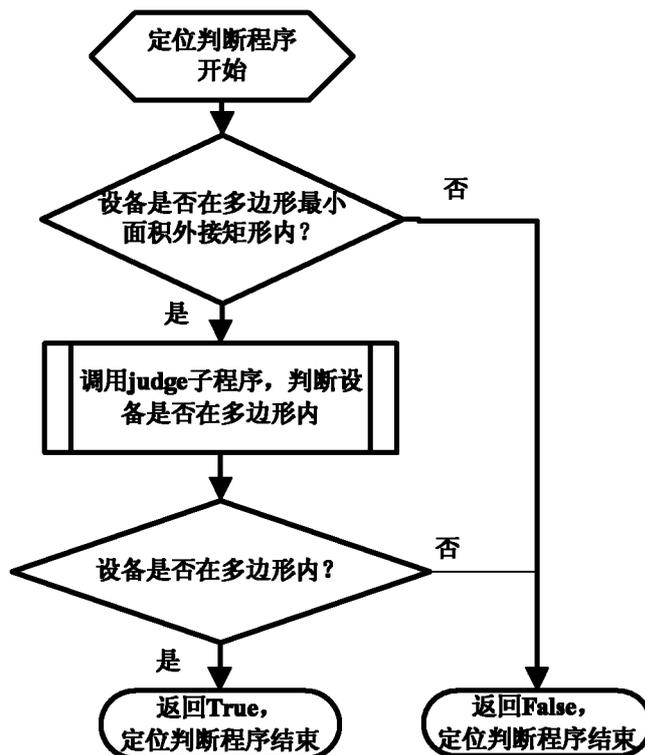


图 2

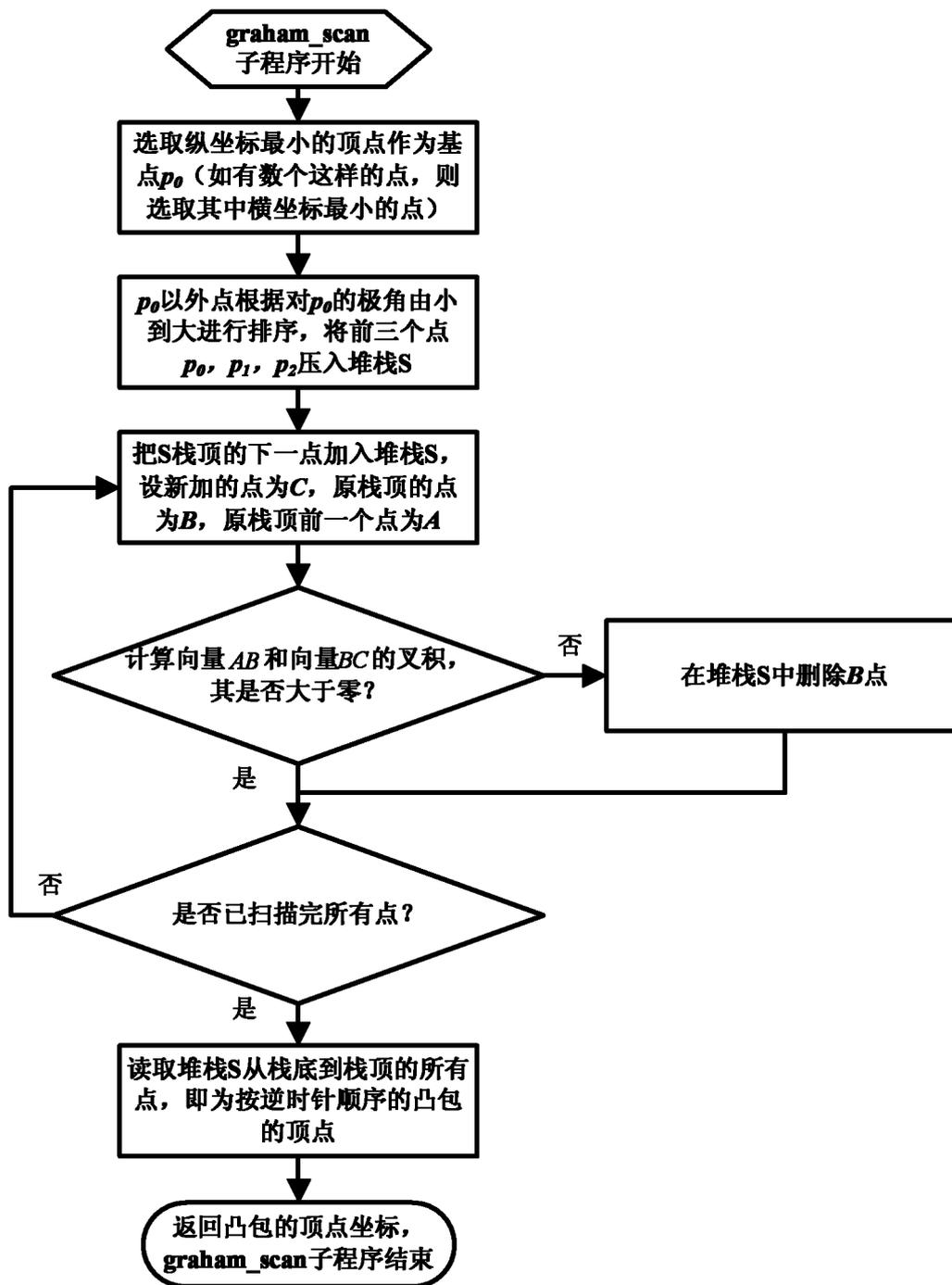


图 3

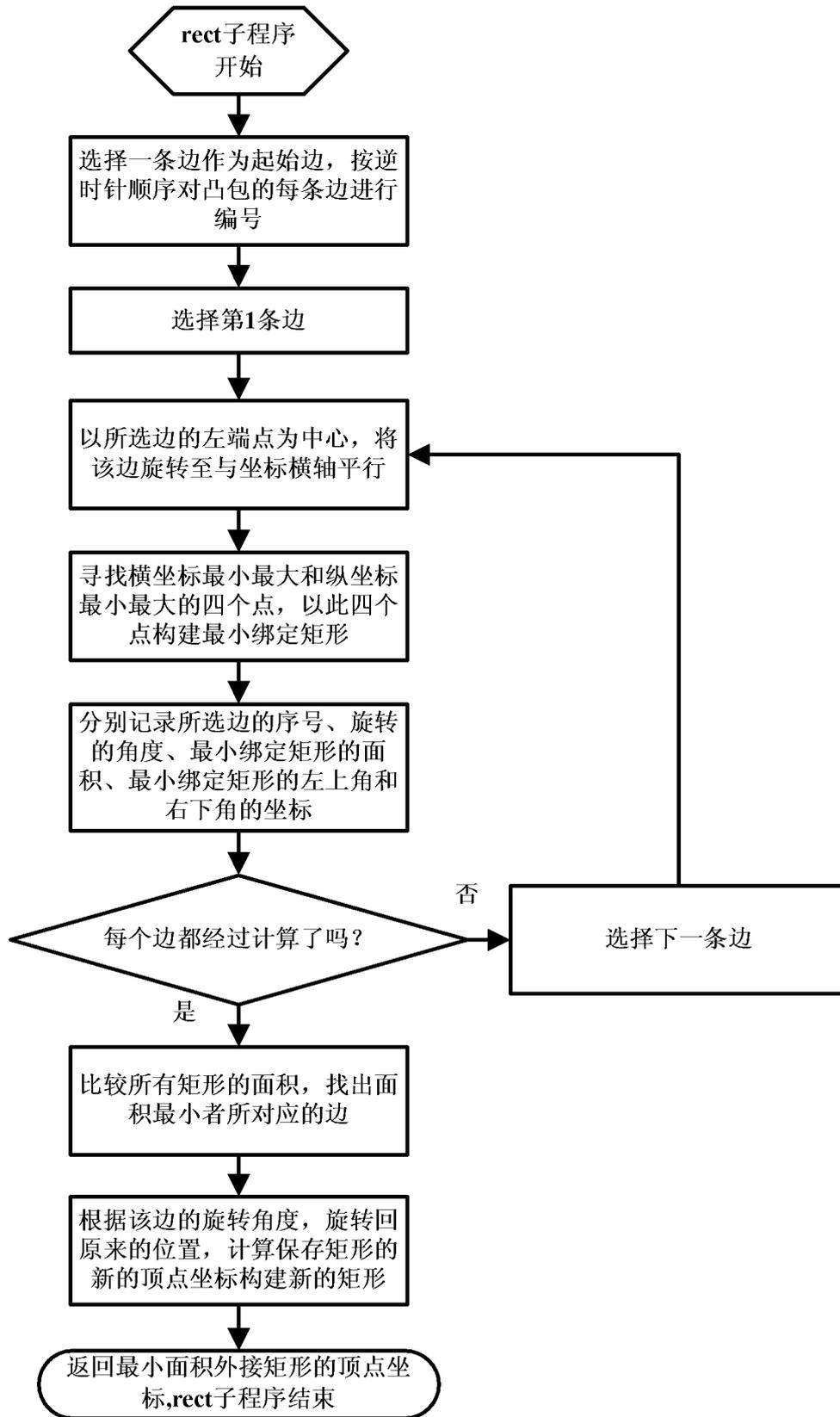


图 4

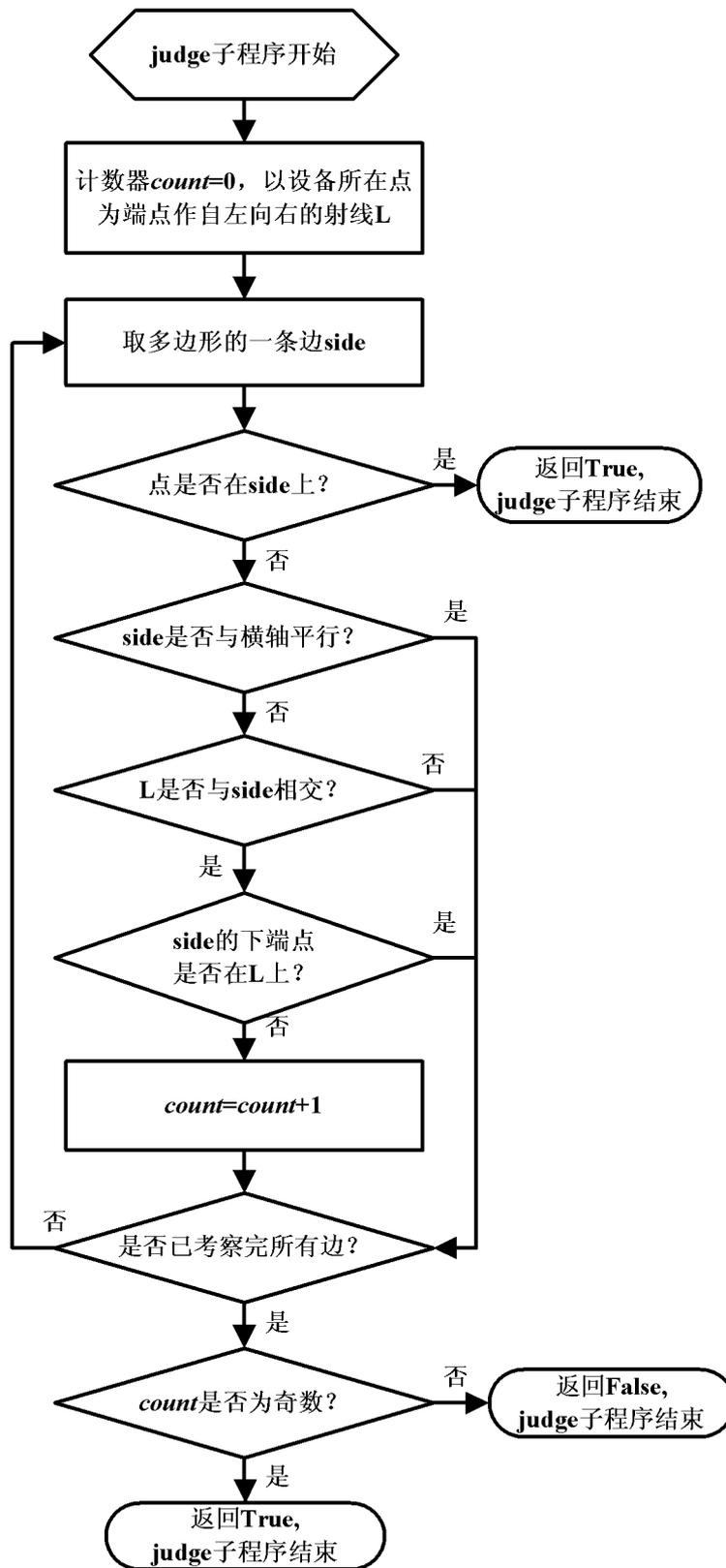


图 5