



(12)发明专利

(10)授权公告号 CN 106273480 B

(45)授权公告日 2019.02.22

(21)申请号 201610654586.2

(22)申请日 2016.08.11

(65)同一申请的已公布的文献号

申请公布号 CN 106273480 A

(43)申请公布日 2017.01.04

(73)专利权人 福建农林大学

地址 350002 福建省福州市仓山区上下店路15号

(72)发明人 林甲祥 陈日清 舒兆港 吴丽萍

(74)专利代理机构 北京方圆嘉禾知识产权代理有限公司 11385

代理人 董芙蓉

(51)Int.Cl.

B29C 64/386(2017.01)

B33Y 50/00(2015.01)

(56)对比文件

CN 103528585 A, 2014.01.22,

CN 101996414 A, 2011.03.30,

CN 102110306 A, 2011.06.29,

US 2014/0291893 A1, 2014.10.02,

CN 102193998 A, 2011.09.21,

审查员 范维

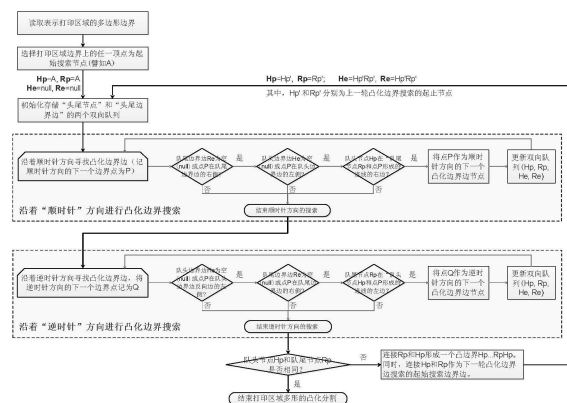
权利要求书1页 说明书5页 附图3页

(54)发明名称

可凹区域的3D快速打印路径规划方法

(57)摘要

本发明公开了可凹区域的3D快速打印路径规划方法。首先,任选区域边界上的一点作为凸化边界搜索的起始点,初始化存储起止边界点和起止边界边的两个双端队列。然后,采用贪心策略,沿打印区域多边形的边界“顺时针”方向进行凸化边界搜索,直到不满足条件的边界点出现或搜索结束;接着,采用贪心策略,沿打印区域多边形的边界“逆时针”方向进行凸化边界搜索,直到不满足条件的边界点出现或搜索结束;最后,判断起止边界点是否相同,若是则结束;否则,连接队头和队尾边界点,形成下一轮搜索的起始条件,进行下一轮边界凸化。本发明避免了多边形打印区域路径规划时可能面临的激光中断和继续问题。



1. 一种可凹区域的3D快速打印路径规划方法,其特征在于,包括下列步骤:

步骤1.任意选取区域边界上的一点A作为凸化边界边起始搜索节点;

步骤2.初始化存储“头尾节点”和“头尾边界边”的两个双向队列,队头节点 $H_p=A$,队尾节点 $R_p=A$,队头边界边 $H_e=null$,队尾边界边 $R_e=null$;

步骤3.沿着区域边界的“顺时针”方向进行凸化边界搜索,将顺时针方向的下一个边界点记为P,判断点P是否符合凸化边界条件,如果是,则更新所述两个双向队列,将点P作为顺时针方向的下一个凸化边界边节点,重复步骤3,否则,进行到步骤4;

步骤4.沿着区域边界的“逆时针”方向进行凸化边界搜索,将逆时针方向的下一个边界点记为Q,判断点Q是否符合凸化边界条件,如果是,则更新所述两个双向队列,将点Q作为逆时针方向的下一个凸化边界边节点,重复步骤4,否则,进行到步骤5;

步骤5.判断队头节点和队尾节点是否相同,如果不同,则连接队头节点 H_p 和队尾节点 R_p 作为下一轮凸化边界边搜索的起始搜索边界边,将这两个节点作为下一轮凸化边界边搜索的头尾节点,初始化所述两个双向队列,返回到步骤3。

2. 根据权利要求1所述的可凹区域的3D快速打印路径规划方法,其特征在于,点P符合凸化边界的条件如下:

队尾边界边 R_e 为空或点P在队尾边界边的右侧,并且

队头边界边 H_e 为空或点P在队头边界边的左侧,并且

队头节点 H_p 在队尾节点 R_p 和点P形成的连线的右边。

3. 根据权利要求1所述的可凹区域的3D快速打印路径规划方法,其特征在于,点Q符合凸化边界的条件如下:

队头边界边 H_e 为空或点P在队头边界边反向边的左侧,并且

队尾边界边 R_e 为空或点P在队尾边界边的右侧,并且

队尾节点 R_p 在队头节点 H_p 和点P形成的连线的左边。

4. 根据权利要求2或3所述的可凹区域的3D快速打印路径规划方法,其特征在于,令点P或点Q的坐标为 $P(x, y)$,则根据如下步骤来判断点 $P(x, y)$ 是在有向线段 \overline{AB} 的左侧还是右侧:

计算由 $P(x, y)$, $A(x_1, y_1)$, $B(x_2, y_2)$ 组成的两个向量 \overline{AP} 和 \overline{BP} 的叉积 $v(P, A, B)$ 。若 $v > 0$,则所述点在有向线段 \overline{AB} 的右侧;若 $v < 0$,则所述点在有向线段 \overline{AB} 的左侧;若 $v = 0$,则所述点与有向线段 \overline{AB} 共线。

5. 根据权利要求4所述的可凹区域的3D快速打印路径规划方法,其特征在于,所述叉积 $v(P, A, B)$ 的计算公式如下:

$$v(P, A, B) = (x - x_1) \times (y_2 - y_1) - (x_2 - x_1) \times (y - y_1)。$$

可凹区域的3D快速打印路径规划方法

技术领域

[0001] 本发明涉及3D打印领域,具体而言,涉及可凹区域的3D快速打印路径规划方法。

背景技术

[0002] 3D打印中的路径规划算法就是对切片分层所获得的截面轮廓进行扫描填充,合理的路径规划不仅能够提高3D打印速度,而且还可以节省打印材料。传统的3D打印路径生成算法在处理凹多边形打印区域时可能面临激光中断和继续问题。

发明内容

[0003] 为了解决现有技术中的上述问题,本发明提出了一个多边形打印区域的顺时针&逆时针双向凸化分割算法(Clockwise and Counterclockwise Two-way Convexify Segment Algorithm,CCTCSA),简称CCTCSA算法。提高了激光器的使用效率,延长了激光器的寿命。

[0004] CCTCSA算法的基本思想是将多边形打印区域分割成若干个非凹的凸形区域,以此避开多边形打印区域路径规划时可能面临的激光中断和继续问题。

[0005] 为此,本发明提供一种可凹区域的3D快速打印路径规划方法,包括下列步骤:

[0006] 步骤1.任意选取区域边界上的一点A作为凸化边界边起始搜索节点;

[0007] 步骤2.初始化存储“头尾节点”和“头尾边界边”的两个双向队列,队头节点 $H_p=A$,队尾节点 $R_p=A$,队头边界边 $H_e=null$,队尾边界边 $R_e=null$;

[0008] 步骤3.沿着区域边界的“顺时针”方向进行凸化边界搜索,将顺时针方向的下一个边界点记为P,判断点P是否符合凸化边界条件,如果是,则更新所述两个双向队列,将点P作为顺时针方向的下一个凸化边界边节点,重复步骤3,否则,进行到步骤4;

[0009] 步骤4.沿着区域边界的“逆时针”方向进行凸化边界搜索,将逆时针方向的下一个边界点记为Q,判断点Q是否符合凸化边界条件,如果是,则更新所述两个双向队列,将点Q作为逆时针方向的下一个凸化边界边节点,重复步骤4,否则,进行到步骤5;

[0010] 步骤5.判断队头节点和队尾节点是否相同,如果不同,则连接队头节点 H_p 和队尾节点 R_p 作为下一轮凸化边界边搜索的起始搜索边界边,将这两个节点作为下一轮凸化边界边搜索的头尾节点,初始化所述两个双向队列,返回到步骤3。

[0011] 优选地,点P符合凸化边界的条件如下:

[0012] 队尾边界边 R_e 为空或点P在队尾边界边的右侧,并且

[0013] 队头边界边 H_e 为空或点P在队头边界边的左侧,并且

[0014] 队头节点 H_p 在队尾节点 R_p 和点P形成的连线的右边。

[0015] 优选地,点Q符合凸化边界的条件如下:

[0016] 队头边界边 H_e 为空或点P在队头边界边反向边的左侧,并且

[0017] 队尾边界边 R_e 为空或点P在队尾边界边的右侧,并且

[0018] 队尾节点 R_p 在队头节点 H_p 和点P形成的连线的左边。

[0019] 优选地,令点P或点Q的坐标为 $P(x, y)$,则根据如下步骤来判断点 $P(x, y)$ 是在有向线段 \overline{AB} 的左侧还是右侧:

[0020] 计算由 $P(x, y)$, $A(x_1, y_1)$, $B(x_2, y_2)$ 组成的两个向量 \overline{AP} 和 \overline{BP} 的叉积 $v(P, A, B)$ 若 $v > 0$,则所述点在有向线段 \overline{AB} 的右侧;若 $v < 0$,则所述点在有向线段 \overline{AB} 的左侧;若 $v = 0$,则所述点与有向线段 \overline{AB} 共线。

[0021] 优选地,所述叉积 $v(P, A, B)$ 的计算公式如下:

[0022] $v(P, A, B) = (x - x_1) \times (y_2 - y_1) - (x_2 - x_1) \times (y - y_1)$ 。

附图说明

[0023] 将参考附图中所说明的示范性的实施例而在下文中更详细地解释本发明。

[0024] 图1是CCTCSA算法的处理流程图。

[0025] 图2是以A为起点进行多边形打印区域的凸化边界搜索过程图。

[0026] 图3示出了多边形打印区域的另一个凸化边界“EHGFE”。

具体实施方式

[0027] 图1示出了CCTCSA算法进行打印区域多边形的凸化分割流程。其中,CCTCSA算法将可能为凹的多边形打印区域转变为一系列凸形区域的方法是:

[0028] 首先,任意选取区域边界上的一点作为凸化边界搜索的起始点,初始化存储起止边界点和起止边界边的两个双端队列 H_p, R_p, H_e, R_e 。

[0029] 然后,沿着打印区域多边形的边界“顺时针”方向进行凸化边界搜索,采用贪心策略,直到不满足条件的边界点出现或搜索结束;

[0030] 紧接着,采用贪心策略,沿着打印区域多边形的边界“逆时针”方向进行凸化边界搜索,直到不满足条件的边界点出现或搜索结束;

[0031] 最后,根据起止边界点是否相同,判断打印区域多边形的边界凸化过程是否结束。若是,则结束算法;否则,连接队头和队尾边界点,形成下一轮搜索的起始条件,进行下一轮边界凸化。

[0032] 以图2所示的多边形打印区域为例,若以A为起始搜索点、且先沿着打印区域多边形边界顺时针方向进行搜索,后沿着打印区域多边形边界逆时针方向进行搜索,则多边形打印区域的双向凸化分割过程如图2所示。

[0033] 初始时刻队头边界点为 $Head-PNT=A$,队尾边界点为 $Rear-PNT=A$,队头边界边为 $Head-EDGE=null$,队尾边界边为 $Rear-EDGE=null$,如图2(1)所示。

[0034] 步骤1:采用贪心策略,沿着顺时针方向进行凸化边界搜索。如图2(2~5)所示。

[0035] 步骤1.1:获得顺时针方向的第一个边界顶点K,形成第一条边界边 \overline{AK} ,并更新队头边界点 $Head-PNT$ 、队尾边界点 $Rear-PNT$,队头边界边 $Head-EDGE$,队尾边界边 $Rear-EDGE$,使得 $Head-PNT=A$, $Rear-PNT=K$, $Head-EDGE=\overline{AK}$, $Rear-EDGE=\overline{AK}$ 。如图2(2)所示。

[0036] 步骤1.2:循环。沿着顺时针方向寻找凸化边界点,直到碰到不满足条件的边界点或边界点搜索结束。如图2(2~5)所示。

[0037] 首先,找到边界点J,判断:(1)点J是否在当前队尾边界边 $Rear-EDGE=\overline{AK}$ 的右侧(顺时针方向)?(2)点J是否在队头边 $Head-EDGE=\overline{AK}$ 的反向边 \overline{KA} 的左侧(逆时针方向)?

(3) 队头边界点 $Head-PNT=A$ 是否在形成的边界边 \overline{KJ} 的右侧 (顺时针方向)? 因为三个条件皆成立, 因此, 点 J 符合凸化边界条件, \overline{KJ} 为下一条凸边界边, 并更新队尾边界点 $Rear-PNT=J$ 和队尾边界边 $Rear-EDGE=\overline{KJ}$ 。如图2 (3) 所示。

[0038] 然后, 找到边界点 H , 判断: (1) 点 H 是否在当前队尾边界边 $Rear-EDGE=\overline{KJ}$ 的右侧 (顺时针方向)? (2) 点 H 是否在队头边 $Head-EDGE=\overline{AK}$ 的反向边 \overline{KA} 的左侧 (逆时针方向)? (3) 队头边界点 $Head-PNT=A$ 是否在形成的边界边 \overline{JH} 的右侧 (顺时针方向)? 因为三个条件皆成立, 因此, 点 H 符合凸化边界条件, \overline{JH} 为下一条凸边界边, 并更新队尾边界点 $Rear-PNT=H$ 和队尾边界边 $Rear-EDGE=\overline{JH}$ 。如图2 (4) 所示。

[0039] 紧接着, 找到边界点 G , 判断: (1) 点 G 是否在当前队尾边界边 $Rear-EDGE=\overline{JH}$ 的右侧 (顺时针方向)? (2) 点 G 是否在队头边 $Head-EDGE=\overline{AK}$ 的反向边 \overline{KA} 的左侧 (逆时针方向)? (3) 队头边界点 $Head-PNT=A$ 是否在形成的边界边 \overline{HG} 的右侧 (顺时针方向)? 由于条件 (1) 不成立, 即边界点 G 在当前队尾边界边 $Rear-EDGE=\overline{JH}$ 的左侧, 不构成凸化边界, 且边界点 G 不是队头边界点 $Head-PNT=A$, 因此结束顺时针方向的边界点搜索, 进入下一步骤“逆时针方向的凸化边界点搜索”。如图2 (5) 所示。

[0040] 步骤2: 采用贪心策略, 沿着逆时针方向进行凸化边界搜索。如图2 (6~10) 所示。

[0041] 步骤2.1: 循环。沿着逆时针方向寻找凸化边界点, 直到碰到不满足条件的边界点或边界点搜索结束。如图2 (6~10) 所示。

[0042] 首先, 找到边界点 B , 判断: (1) 点 B 是否在当前队头边界边 $Head-EDGE=\overline{AK}$ 反向边 \overline{KA} 的左侧 (逆时针方向)? (2) 点 B 是否在当前队尾边界边 $Rear-EDGE=\overline{JH}$ 的右侧 (顺时针方向)? (3) 队尾边界点 $Rear-PNT=H$ 是否在形成的边界边 \overline{AB} 的左侧 (逆时针方向)? 因为三个条件皆成立, 因此, 点 B 符合凸化边界条件, \overline{AB} 为下一条凸化边界边, 并更新队头边界点 $Head-PNT=B$ 和队头边界边 $Head-EDGE=\overline{BA}$ 。如图2 (6) 所示。

[0043] 然后, 找到边界点 C , 判断: (1) 点 C 是否在当前队头边界边 $Head-EDGE=\overline{BA}$ 反向边 \overline{AB} 的左侧 (逆时针方向)? (2) 点 C 是否在当前队尾边界边 $Rear-EDGE=\overline{JH}$ 的右侧 (顺时针方向)? (3) 队尾边界点 $Rear-PNT=H$ 是否在形成的边界边 \overline{BC} 的左侧 (逆时针方向)? 因为三个条件皆成立, 因此, 点 C 符合凸化边界条件, \overline{CB} 为下一条凸化边界边, 并更新队头边界点 $Head-PNT=C$ 和队头边界边 $Head-EDGE=\overline{CB}$ 。如图2 (7) 所示。

[0044] 紧接着, 找到边界点 D , 判断: (1) 点 D 是否在当前队头边界边 $Head-EDGE=\overline{CB}$ 反向边 \overline{BC} 的左侧 (逆时针方向)? (2) 点 D 是否在当前队尾边界边 $Rear-EDGE=\overline{JH}$ 的右侧 (顺时针方向)? (3) 队尾边界点 $Rear-PNT=H$ 是否在形成的边界边 \overline{CD} 的左侧 (逆时针方向)? 因为三个条件皆成立, 因此, 点 D 符合凸化边界条件, \overline{CD} 为下一条凸化边界边, 并更新队头边界点 $Head-PNT=D$ 和队头边界边 $Head-EDGE=\overline{DC}$ 。如图2 (8) 所示。

[0045] 再者, 找到边界点 E , 判断: (1) 点 E 是否在当前队头边界边 $Head-EDGE=\overline{DC}$ 反向边 \overline{CD} 的左侧 (逆时针方向)? (2) 点 E 是否在当前队尾边界边 $Rear-EDGE=\overline{JH}$ 的右侧 (顺时针方向)? (3) 队尾边界点 $Rear-PNT=H$ 是否在形成的边界边 \overline{DE} 的左侧 (逆时针方向)? 因为三个条件皆成立, 因此, 点 E 符合凸化边界条件, \overline{DE} 为下一条凸化边界边, 并更新队头边界点 $Head-PNT=E$ 和队头边界边 $Head-EDGE=\overline{ED}$ 。如图2 (9) 所示。

[0046] 最后, 找到边界点 F , 判断: (1) 点 F 是否在当前队头边界边 $Head-EDGE=\overline{ED}$ 反向

边 \overline{DE} 的左侧(逆时针方向)?(2)点F是否在当前队尾边界边 $Rear-EDGE = \overline{JH}$ 的右侧(顺时针方向)?(3)队尾边界点 $Rear-PNT = H$ 是否在形成的边界边 \overline{EF} 的左侧(逆时针方向)?因为条件(1)成立、条件(2)不成立,因此,点F不符合凸化边界条件, \overline{DE} 不是下一条凸化边界边,逆时针凸化边界搜索结束。如图2(10)所示。

[0047] 步骤3:顺时针和逆时针凸化边界搜索都结束后,若队头节点E和队尾节点H不相同,则连接E和H两个节点,形成一个局部凸化边界。如图2(11)所示。

[0048] 经过上述步骤1~3之后,因为寻找到的凸化边界的起始节点分别为E和H,不相同。因此,连接E和H两个节点形成一个局部凸化边界。之后,以凸化边界 \overline{EH} 为队头边界边,开始新一轮的凸化边界搜索,便可找到下一个凸化边界EHGF。如图3所示,具体过程在此省略。

[0049] CCTCSA算法实现的核心步骤和伪代码如以下清单。

[0050]

输入:(1) List<Point>表示的一个多边形打印区域,记为 $AB...JK$ 。

(2) 3D 打印光斑的直径,记为 d 。

输出: List<Point>表示的一个打印路径,记为 $P=\overline{L_1R_1R_2L_2...L_iR_iR_{i+1}L_{i+1}...}$ 。

对于给定的打印区域多边形,打印路径生成算法的核心步骤如下:

(1) 将打印区域多边形分割为一系列凸剖分,记为 $\mathfrak{R} = \{\chi_i | i = 1, 2, \dots\}$,其中 χ_i 为一个凸形区域。

(2) 采用贪心策略,沿着“顺时针”方向进行凸化边循环搜索,直到碰到不满足条件的边界节点或搜索结束。然后,对凸剖分 \mathfrak{R} 中的每个凸形区域 χ_i ,使用经典的路径生成算法,实现打印路径规划。

(3) 采用贪心策略,沿着“逆时针”方向进行凸化边循环搜索,直到碰到不满足条件的边界节点或搜索结束。然后,对凸剖分 \mathfrak{R} 中的每个凸形区域 χ_i ,使用经典的路径生成算法,实现打印路径规划。

(4) 若打印区域多边形对应的凸剖分 \mathfrak{R} 中的所有凸形区域都已进行打印路径规划,则联接所有凸形区域的打印路径,形成给定多边形打印区域的最终打印路径,并返回。

[0051] CCTCSA算法使用到的核心操作是:判断点 $P(x, y)$ 是在有向线段 \overline{AB} 的左侧?还是右侧?其原理是计算三个点 $P(x, y)$, $A(x_1, y_1)$, $B(x_2, y_2)$ 组成的2个向量 \overline{AP} 和 \overline{BP} 的叉积 $v(P, A, B)$ 。若 $v > 0$,则点P在有向线段 \overline{AB} 的右侧;若 $v < 0$,则点P在有向线段 \overline{AB} 的左侧;若 $v = 0$,则点P与有向线段 \overline{AB} 共线。其中,叉积 $v(P, A, B)$ 的计算公式如下:

[0052] $v(P, A, B) = (x - x_1) \times (y_2 - y_1) - (x_2 - x_1) \times (y - y_1)$

[0053] 对于规模为 n 的打印区域多边形,因为CCTCSA算法在进行边界凸化时每个边界节点只搜索一次,核心计算时间是判断点与边界边之间的左右侧向(顺逆时针方向)关系,由于叉积 v 的计算可以视为常量时间,因此,规模为 n 的打印区域多边形的边界凸化的时间复杂度为 $O(n)$,而打印路径规划所使用的核心算法为经典的方法,可见CCTCSA算法的时间复杂度与经典的路径规划算法一样。

[0054] 上面结合附图和实施例对本发明做了详细的说明。但是,应当理解,本发明的实施例并不限于所公开的特定实施例,并且对该实施例的修改和其他实施例也意图被包含在所附权利要求书的范围内。尽管此处使用了特定术语,但是它们仅在通用和描述性意义上使用,而非为了限制的目的。

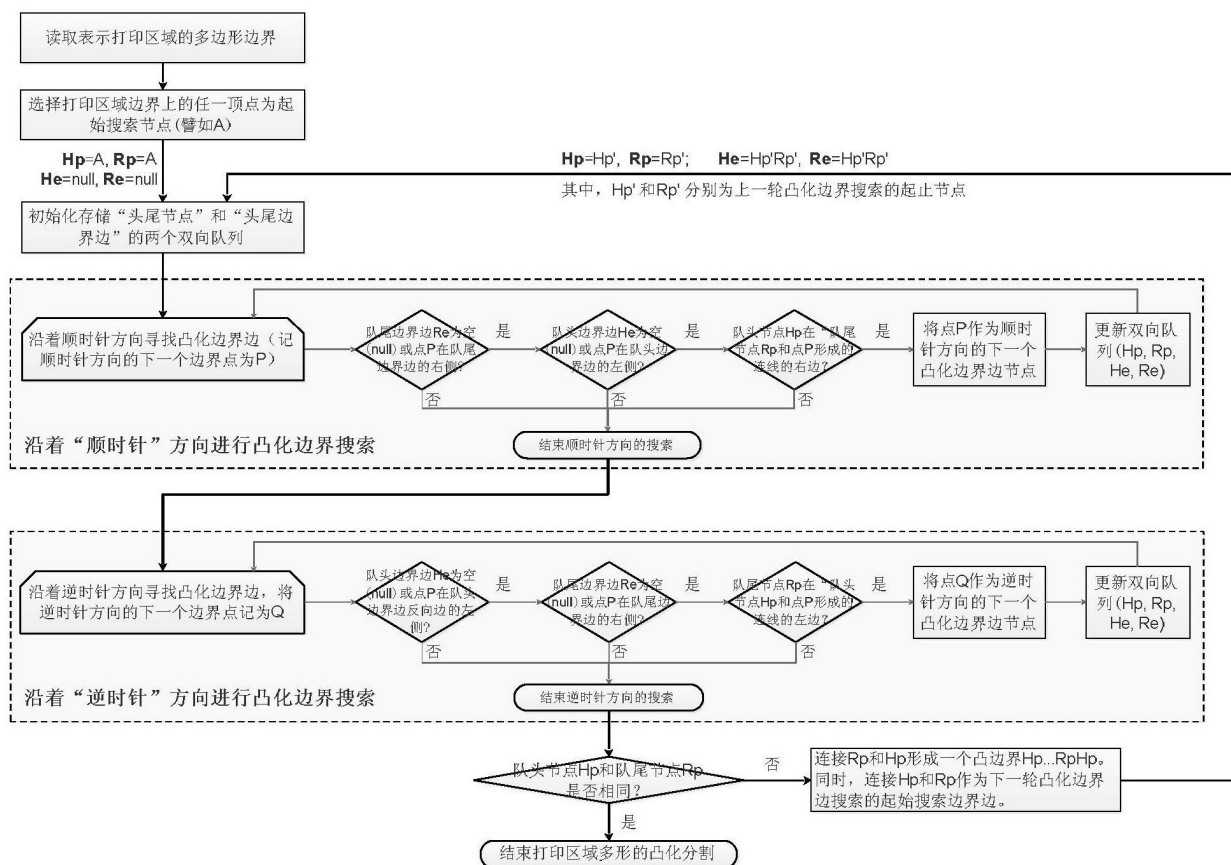


图1

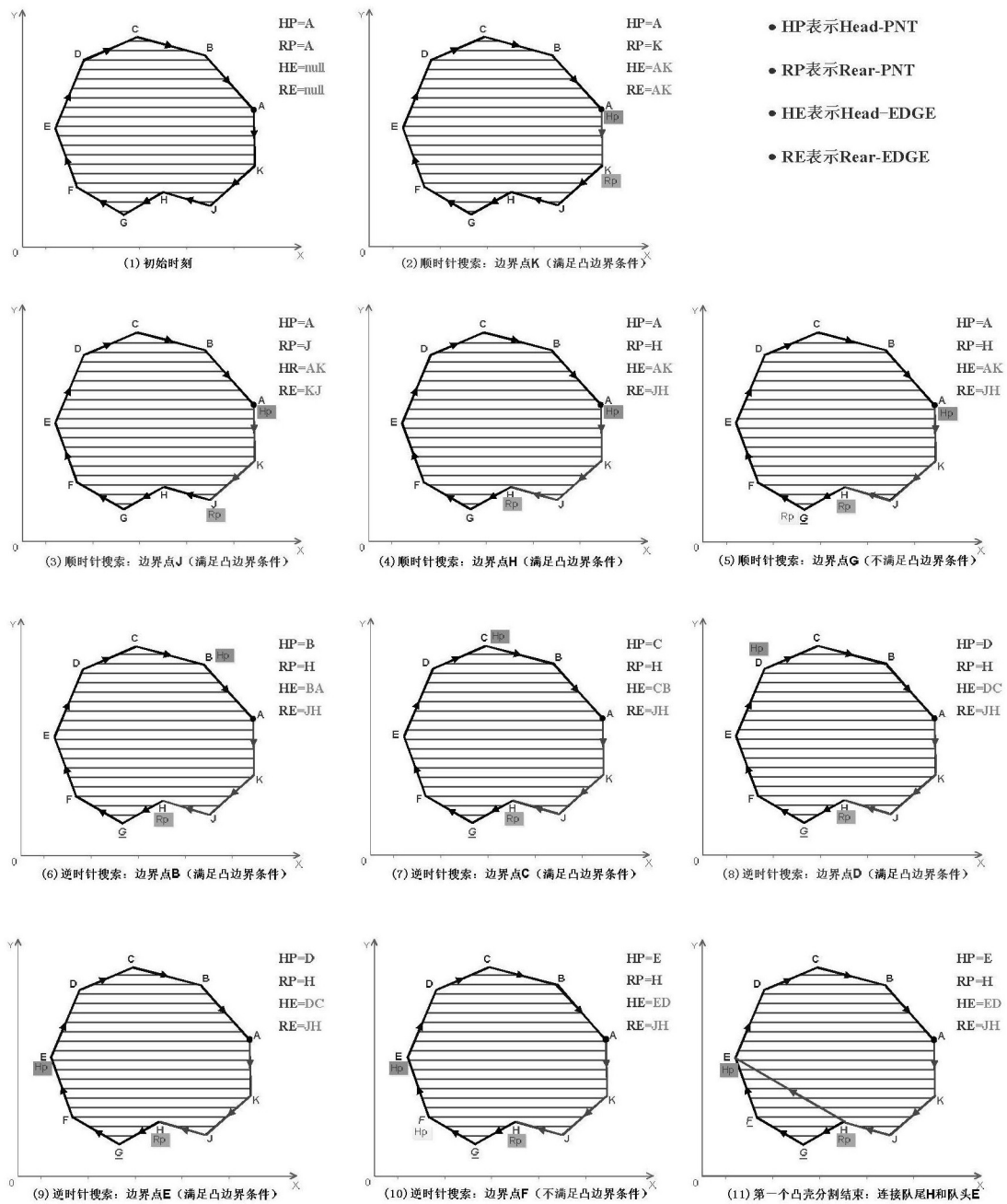


图2

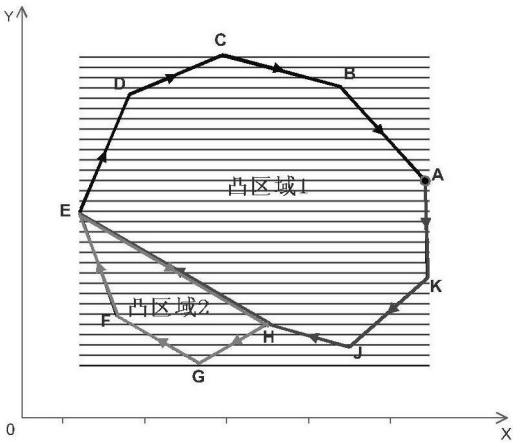


图3