

文章编号: 2095-2163(2020)08-0076-08

中图分类号: TP311.13

文献标志码: A

# 面向观测任务的无人艇路径规划策略

孟祥根

(哈尔滨工业大学 计算机科学与技术学院, 哈尔滨 150000)

**摘要:** 随着无人技术和机器人技术的发展,作为其重要环节的全覆盖路径规划技术越发关键。与普通的机器人全覆盖路径规划不同,由于海况、天气等环境因素的影响,水面全覆盖路径规划需要考虑路径完成时间、路径平滑度、路径安全性等问题。目前对于机器人全覆盖路径规划问题已经有许多算法,如基于A星算法的路径规划、基于改进蚁群算法的路径规划、基于回溯法的路径规划,但这些算法在水面路径规划上存在路径转弯点过多的问题。本文针对一种水面全覆盖路径规划问题,提出一种路径规划算法,通过全局规划和局部规划两阶段确定路径。全局规划路径确定大致运动路线,保证覆盖率;局部规划在使用传感器探测到目标后,聚类处理并根据目标规划出新路径,保证路径的安全性。

**关键词:** 水面路径规划; 全覆盖问题; 聚类算法; 卡尔曼滤波

## Unmanned surface vessel path planning strategy for observation task

MENG Xianggen

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

**[Abstract]** With the development of unmanned vehicle and robot technology, full coverage path planning technology, which is an important part of it, has become increasingly critical. Due to the influence of sea conditions, weather and other environmental factors, the path planning of full water surface coverage needs to consider the path completion time, path smoothness, path safety and other issues. This article proposes a path planning algorithm for a full-coverage path planning problem. The path is determined by two stages: global planning and local planning. The global planning path determines the approximate motion route to ensure coverage. Local planning uses sensors to detect the target. Then clustering processing is performed and a new path is planned according to the goal to ensure the security of the path.

**[Key words]** path planning on water; full coverage problem; clustering algorithm; Kalman filter

### 0 引言

水面无人艇是一种智能水面航行设备,一般采用遥控操作或自主航行的模式执行任务,其中自主航行模式属于更智能的模式,解决自主航行遇到的问题水面无人艇研究领域的重点。作为无人艇航行时依赖的基本技术之一,路径规划在无人艇研究领域占有不可或缺的地位,目前已有许多路径规划算法,包括基于A星算法的路径规划、基于改进蚁群算法的路径规划、基于回溯法的路径规划等,总结起来包括深度搜索算法、启发式算法和元启发式算法。在全覆盖路径规划领域也有许多算法被提出,螺旋生成树覆盖算法,基于栅格构建无向连接图,从而根据生成树确定运动轨迹,并根据拍卖算法确定最终的运动轨迹<sup>[1]</sup>;利用回溯机制的局部区域覆盖算法,用于未覆盖子区域的记录和为区域衔接路径选取目标点<sup>[2]</sup>,这部分全覆盖算法应用于移动机器人,对于体型不大、移动灵活的设备具有较高的应用价值,规划出的路径较为密集,而水面无人艇由

于体积大、转向不灵活等特点,不适合直接采用诸如此类的方法。在无人设备领域,除无人艇外,还有无人机、扫地机器人等设备的研究较为广泛。

在以往提出的路径规划算法中,栅格法建模被大量采用,因为栅格法具有结构直观、便于处理的特点,代价是大量存储空间的开销。在航行过程中,并非所有栅格都存在障碍物,大部分栅格结构中存储的数据差异很小,信息量很小,导致存储空间的浪费。而对于全覆盖路径规划算法,在实际应用中也存在各种问题,如重复率高、路径拐点过多、覆盖率不全等。

针对全覆盖水面路径规划中的一种问题,本文提出一种方法,将路径规划分为全局路径预规划和局部路径重规划两部分。在该问题中,无人艇被指定在一个矩形区域内运动,矩形区域内随机分布多个事先未知但可以实时获取到的坐标点,每个坐标点具有一个不可进入的圆形区域,设备需要确定路径,确保可以获取到所有坐标点的数据,靠近坐标点

**作者简介:** 孟祥根(1997-),男,硕士研究生,主要研究方向:水面无人设备路径规划。

**收稿日期:** 2020-06-01

到一定距离,并避免进入不可进入区域内。

对于以上问题,在全局路径预规划的过程中,算法根据先验已知的任务区域设定数据,计算整体运动路径,保证行进过程中区域的覆盖率;在局部路径重规划过程中,根据行进过程中传感器实时获取的后验障碍物数据对路径进行局部修改,保证抵近障碍物观察的同时避免与障碍物发生碰撞。实验表明,本文方法可以在保证区域覆盖率的同时使路径具有一定的平滑度,并保证区域内的障碍物都能进行抵近观察。

## 1 全局路径预规划

为了更清晰地表述算法过程,使用更加形式化的语言描述问题。已知一矩形区域  $S$ , 矩形四个顶点的坐标为  $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ , 其中  $i = 1, 2, 3, 4, x_i$  表示经度坐标,  $y_i$  表示纬度坐标。在区域  $S$  的边界上有两个指定坐标点  $(x_{in}, y_{in})$  和  $(x_{out}, y_{out}), (x_{in}, y_{in})$  为进入区域  $S$  的坐标点,  $(x_{out}, y_{out})$  为离开区域  $S$  的坐标点。区域  $S$  内存在若干待观察点  $c_i$ , 构成待观察点集  $C = \{(clon_i, clat_i) \mid i = 1, 2, 3, \dots, n\}$ , 其中  $i = 1, 2, 3, \dots, n, c_i$  表示一个未知但可通过设备传感器探测到的待观察坐标点,  $clon_i$  表示该点的经度坐标,  $clat_i$  表示该点的纬度坐标。对于任意  $c_i \in C$ , 有  $r_i > 0$ , 以  $c_i$  为圆心,  $r_i$  为半径的圆形区域  $o_i$  为不可进入区域;对于任意  $c_i \in C$ , 有  $R_i > r_i$ , 以  $c_i$  为圆心,  $R_i$  为半径的圆形区域  $O_i$  为观察区域,即设备需要在运动过程中进入此圆形区域,方可认定坐标点  $c_i$  已被观察。对于任意  $c_i \in C, c_i$  与区域  $S$  边界的距离不小于常数  $\varepsilon_0$ 。设备传感器可覆盖的观察半径为  $r_0$ 。要求算法求解一个坐标点序列  $V = [v_1, v_2, \dots, v_n]$ 。设  $V$  中每两个相邻的坐标点构成线段边集为  $E = \{v_1v_2, v_2v_3, \dots, v_{n-1}v_n\}$ ,  $V$  和  $E$  构成路径  $P = \{V, E\}$ , 其中  $i = 1, 2, 3, \dots, n, v_i$  表示一个位于路径上的坐标点,则路径  $P$  需要满足以下条件:

- $v_1$  和  $(x_{in}, y_{in})$  重合,  $v_n$  和  $(x_{out}, y_{out})$  重合;
- $\forall o_i, \forall v_jv_{j+1} \in E, v_jv_{j+1}$  与  $o_i$  无交叠部分;
- $\forall O_i, \exists v_jv_{j+1} \in E, v_jv_{j+1}$  与  $O_i$  有交叠部分。

条件(1)表示路径从进入点开始,到离开点结束;条件(2)表示路径不能经过任一待观察点的不可进入区域;条件(3)表示路径需要经过每一个待观察点的观察区域。

### 1.1 总体路径曲线

为了保证区域内的所有待观察点都能被发现,算法需要保证设备沿路径运动过程中,以设备  $w$  为

圆心,传感器探测距离  $r_0$  为半径的圆形区域可以覆盖以  $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$  为顶点的矩形区域。为了保证这一点,需要确定一种曲线作为全局预规划的总体行进路线,保证路径可以在粗粒度上覆盖矩形区域。

以线的形式覆盖矩形区域有很多方式,目前已经提出很多空间填充曲线,如希尔伯特曲线、皮亚诺曲线等。在本文研究的问题中,路径所服务的对象需要在水面运动,过于复杂的曲线并不适用于该问题,因为在实际水面上无法进行复杂的控制,简单曲线如蛇形曲线、Z形曲线更适用于该问题,可以有效减少路径长度,提高设备的运行速度。为了确定蛇形曲线和Z形曲线对于本问题的适用性,从路径长度、拐角大小以及重复覆盖面积等方面对两种曲线进行比较。

如图 1 所示,图 1(a)为蛇形曲线路径示意图,图 1(b)为Z形曲线路径示意图。对于相同长度的一次往返,设一个单程长度为  $a$ , 两个单程的间隔距离为  $b$ , 则蛇形曲线的路径长度为  $2a + b$ , Z形曲线的路径长度为  $2a + \sqrt{a^2 + b^2}$ 。根据三角不等式,Z形曲线的路径长度大于蛇形曲线的路径长度,因此蛇形曲线的路径具有路程短的优点。

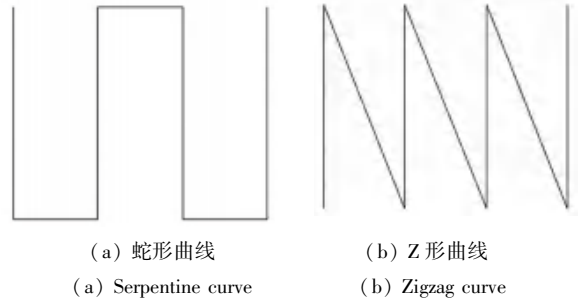


图 1 总体路径曲线示意图

Fig. 1 Diagram of overall path curve

如图 2 所示,蛇形曲线的转弯角度为  $\pi - \alpha$ , Z形曲线的转弯角度为  $\pi - \beta$ 。无论何种情况,  $\alpha$  均为直角,  $\beta$  均为锐角,因此  $\alpha > \beta$ , 可以得出  $\pi - \alpha < \pi - \beta$ 。因此蛇形曲线具有转弯角度小的优点,设备在蛇形曲线上运动可以有更好的稳定性。

为了保证区域覆盖率,两个单程的间隔距离应该根据传感器的观察半径确定。设路径上的一个单程长度为  $a$ , 两个单程的间隔距离为  $b$ , 传感器的探测半径为  $r_0$ , 则当  $b = 2r_0$  时,传感器的观察区域可将整个区域完整覆盖,此时两个单程的衔接处会有区域被重复覆盖。为了估计重复覆盖的区域在整个观察过程中需要的代价,将两个单程的衔接路径长

度与路径总长度之比作为近似重复率,路径的近似重复率越大,说明被重复覆盖的区域面积越大,传感器观察坐标点效率越低。在如图2所示的两个单程中,蛇形曲线的路径总长度为 $2a + b$ ,衔接处路径长度为 $b$ ,近似重复率为 $\frac{b}{2a + b}$ ;Z形曲线的路径总长度为 $2a + \sqrt{a^2 + b^2}$ ,衔接处路径长度为 $\sqrt{a^2 + b^2}$ ,近似重复率为 $\frac{\sqrt{a^2 + b^2}}{2a + \sqrt{a^2 + b^2}}$ 。根据三角不等式,有 $\sqrt{a^2 + b^2} > b$ 成立,即 $2a\sqrt{a^2 + b^2} + b\sqrt{a^2 + b^2} > 2ab + b\sqrt{a^2 + b^2}$ ,提取公因子可变形为 $(2a + b)\sqrt{a^2 + b^2} > (2a + \sqrt{a^2 + b^2})b$ ,将其转化为分式的形式即为 $\frac{\sqrt{a^2 + b^2}}{2a + \sqrt{a^2 + b^2}} > \frac{b}{2a + b}$ 。由此可以得出蛇形曲线的近似重复率更小,在运动过程中,传感器可以更加高效地对坐标点进行观察。

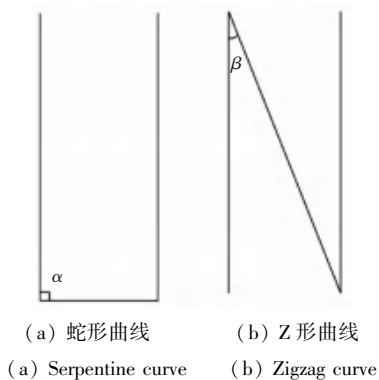


图2 曲线转弯角度示意图

Fig. 2 Diagram of turning angle

## 1.2 分区策略

在确定使用蛇形曲线作为总体路径曲线后,有两件主要的事情需要考虑,其一是保证所有目标点可以被探测到,其二是根据目标点的位置对路径进行重规划,保证路径满足条件(3)。在重规划过程中,需要为待观察点确定观察顺序,根据确定好的顺序对各个待观察点进行路径重规划,否则会产生重复覆盖范围过大,导致路径过长,观测效率低的问题。根据蛇形曲线的特点,路径由多个单程路径构成,多个单程路径间由一段较短的转弯路径衔接,因此使用分区策略来为路径重规划提供策略基础。

如图3所示,矩形区域PMNQ为任务区域,将MN边划分为若干长为 $r_0$ 的短边,最后不足 $r_0$ 的部分单独作为一条短边。以各条短边以及PM边的平行线为边长,矩形区域可被划分为若干子区域,路径

的各个单程与各个子区域的中线重合。由于传感器的探测半径为 $r_0$ ,因此当设备在每个子区域内沿总体路径运动时,子区域可被完整覆盖。由于设备在运动过程中具有惯性,所以运动路线在转弯处无法保证与规划路线重合,因此在路径与边界之间的转弯处,将路径从边界回收一定的距离 $\eta$ ,保证设备在运动过程中不会超出矩形区域的约束。

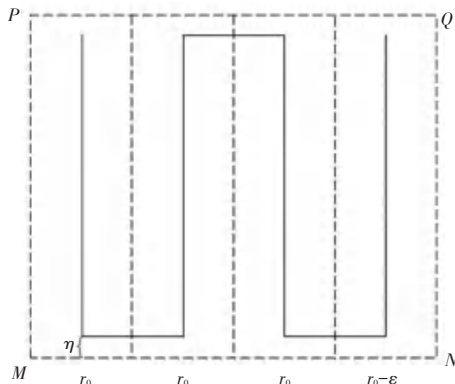


图3 分区示意图

Fig. 3 Diagram of partition

## 2 局部路径重规划

在全局路径预规划阶段对路径进行了初步规划,设定了总体航行路径。剩下的问题是如何在航行过程中发现所有的待观察点,并将路径重规划,使路径能够经过待观察点的观察区域,但不进入待观察点的不可进入区域。对于这个问题,首先需要保证数据的准确性,其次要根据准确的数据制定策略,通过算法确定重规划后的路径,进而使路径满足上述条件。本文主要根据数据预处理、单点处理、多点处理的流程对处理该问题。其中数据预处理主要目的是将粗糙的数据经过预处理转化为较为准确的数据,单点处理是对每个单独的待观察点进行路径重规划,多点处理是将区域内已知的多个待观察点的路径进行连接。

### 2.1 数据预处理

在水面上使用传感器探测特定位置的坐标点,通常采用回波的方法实现,但由于水面存在波浪的影响,无人艇处于不断运动的状态,回波获取到的坐标点数据会被干扰,获取到无关数据,或者使探测到的坐标点发生偏移,存在较大的误差。为了保证路径的准确性,要对传感器探测到的数据进行预处理,过滤掉无关数据,并使用卡尔曼滤波跟踪有效数据,实时获取到正确的坐标点。

卡尔曼滤波是一种去除噪声,还原真实数据的数据处理方法,通过给定具有测量误差的数据测量



值和系统对数据的预估值,对待求数据进行迭代,得到较为准确的数据。卡尔曼滤波只需要考虑上一次迭代的数据和当前数据,无需考虑更早的数据,使用过程较为简便,被广泛应用在导航领域。卡尔曼滤波包括两个步骤,分别为预测和更新,用公式(1)~(5)表示:

$$X_k^- = \Phi_{k-1,k} X_{k-1} + \omega_{k-1}, \quad (1)$$

$$P_k^- = \Phi_{k-1,k} P_{k-1} \Phi_{k-1,k}^T + Q_{k-1,k}, \quad (2)$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \quad (3)$$

$$X_k = X_k^- + K_k (Y_k - H_k X_k^-), \quad (4)$$

$$P_k = (I - K_k H_k) P_k^-. \quad (5)$$

公式(1)和公式(2)为预测步骤,公式(3)、公式(4)和公式(5)为更新步骤。其中,  $X_k^-$  表示第  $k$  次迭代待求解参数的预测值;  $X_k$  表示第  $k$  次迭代待求解参数的迭代结果;  $\Phi_{k-1,k}$  表示从第  $k-1$  次迭代到第  $k$  次迭代的状态转移参数;  $\omega_{k-1}$  表示预测过程中引入的误差参数;  $P_k^-$  表示协方差矩阵第  $k$  次迭代的预测值;  $P_k$  表示协方差矩阵的第  $k$  次迭代结果;  $Q_{k,k+1}$  表示预测协方差矩阵的过程噪声;  $K_k$  表示第  $k$  次迭代的增益参数;  $H_k$  表示待求解参数的系数矩阵;  $I$  表示单位矩阵。

在本文所要解决的问题中,待求解参数为待观察点的坐标位置,因此将每次迭代的待求解参数定义为  $X_k = [u_k, v_k, x_k, y_k]$ , 其中  $u_k$  表示目标沿横坐标方向的运动速度;  $v_k$  表示目标沿纵坐标方向的运动速度;  $x_k$  表示目标的横坐标;  $y_k$  表示目标的纵坐标。

如图 4 所示,为了将传感器探测到的目标进行整合聚类,本文使用一个本地目标列表存储水面待观察点。在传感器探测到目标的一个周期内,传感器会向程序输入探测到的传感器目标列表,对每个传感器目标计算与其距离最近的本地目标,判断该目标与当前传感器目标的距离是否在传感器误差距离内。如果在误差距离内,则将当前传感器目标与对应的最近本地目标进行融合,将传感器目标加入该本地目标的关联列表。关联列表中目标的位置,决定了本地目标的位置。在将所有传感器目标加入到本地目标列表后,对每一个本地目标的关联列表求平均位置,作为本次卡尔曼滤波迭代的观测值,对本地目标进行更新。最终通过传感器数据和卡尔曼滤波,系统得到聚类后的待观察目标点较为准确的位置数据。

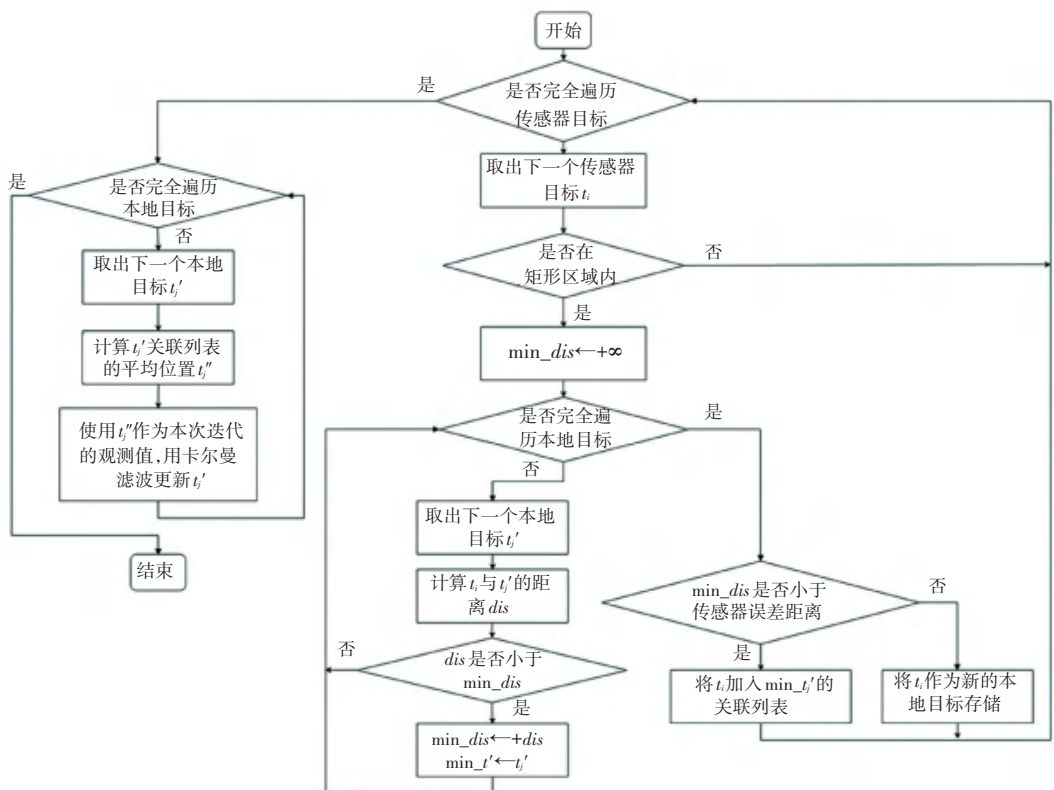


图 4 数据预处理流程图

Fig. 4 Data preprocessing flowchart

## 2.2 单点处理

由于矩形区域内存在多个待观察点,为了由浅入深解决问题,首先将问题简化,考虑其中一个目标点,如何完成路径规划任务。路径需要进入每个待观察点的观察区,同时不能进入任何一个待观察点的不可进入区域,因此需要以待观察点为中心确定该点的外围环绕路径,再根据外围环绕路径与预规划路径的关系,确定将外围路径加入总体路径的切入点和切出点。

如图5所示,对于一待观察点 $c_i$ ,其不可进入区域半径为 $r_i$ ,观察距离为 $R_i$ 。设一条满足条件的环绕路径为 $[v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,m_i}]$ ,其中 $m_i$ 表示该环绕路径上路径点的个数。若要求路径上的转弯点最少,则相邻两点的连线需要外切于 $c_i$ 的不可进入区域边界圆。以 $v_{i,1}v_{i,2}$ 为例,当 $v_{i,1}v_{i,2}$ 与 $c_i$ 的不可进入区域边界圆相切时,记 $v_{i,1}v_{i,2}$ 所对圆心角的一半为 $\theta$ ,则 $\theta = \arccos \frac{r_i}{R_i}$ 。整个圆的圆心角为 $2\pi$ ,因此

$$m_i = \frac{2\pi}{2\theta} = \frac{\pi}{\theta}, \text{ 与 } \theta \text{ 的表达式联立可以得到: } m_i =$$

$$\frac{\pi}{\arccos \frac{r_i}{R_i}}. \text{ 在已知点数以及两个相邻点所对的圆心角后,只需要确定一点,并确定环绕路径的环绕方向,即可得到整个环绕路径。以正北方向为 } 0^\circ \text{ 方位角,当环绕路径为顺时针时, } v_{i,k} \text{ 相对于 } c_i \text{ 的方位角为 } 2(k-1)\theta, \text{ 与 } c_i \text{ 的距离为 } R_i, \text{ 其中 } k=1, 2, \dots, m_i; \text{ 反之,当环绕路径为逆时针时, } v_{i,k} \text{ 相对于 } c_i \text{ 的方位角为 } 2\pi - 2(k-1)\theta, \text{ 与 } c_i \text{ 的距离为 } R_i, \text{ 其中 } k=1, 2, \dots, m_i。$$

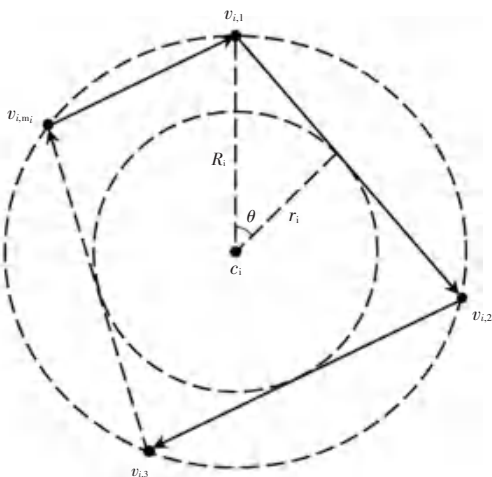


图5 环绕路径示意图

Fig. 5 Diagram of surrounding path

在确定环绕路径的环绕方向时,需要考虑待观察点与当前路径的位置关系,从而保证重规划路径符合分区策略,即重规划后的路径仍然在当前分区内。以无人艇前进方向为路径的正方向,当待观察点位于路径左侧时,环绕路径为逆时针方向;反之,当待观察点位于路径右侧时,环绕路径为顺时针方向。按照这种方法,保证重规划路径是从靠近整体航线的一侧截取而来,从而保证重规划后的路径仍然在当前分区内。

待观察点是在航行过程中探测到的,为了将待观察点对应的重规划路径插入到当前路径中,需要确定其具体的插入位置。为了在待观察点数据较为准确的时间段靠近待观察点,重规划的路径会被插入到当前无人艇在路径中的上一个点 $v_{last}$ 和下一个点 $v_{next}$ 之间,从而保证传感器在探测到准确的目标后较为及时地靠近观察。按照路径最短的原则,在从环绕路径上截取重规划路径时,计算环绕路径上的每一个点 $v_{i,k}$ 与 $v_{last}$ 的距离,取距离最近的点 $v_{i,in}$ 作为重规划路径的切入点;同理,计算环绕路径上的每一个点 $v_{i,k}$ 与 $v_{next}$ 的距离,取距离最近的点 $v_{i,out}$ 作为重规划路径的切出点。

## 2.3 多点处理

多点路径重规划,最主要的问题是多个待观察点的重规划顺序和重规划时机。由于待观察点是未知的,只能通过传感器探测到,因此路径的覆盖率必须得到保证,否则将会导致部分待观察点无法被探测。所以在无人艇航行过程中,即使探测到新的待观察点,也不能无条件地立即将该点的重规划路径插入到总体路径中。如果无条件地插入重规划路径,即每当传感器获取到新的待观察点,就立即观察,那么总体路径将会出现较大改变,产生大量迂回路径,使路径长度剧烈增大,重复覆盖增多,观察效率低;但如果将迂回部分的路径删除,又会导致区域内出现无法覆盖的子区域,使部分待观察点产生无法被探测到的风险。基于以上考虑,采用前文提到的分区策略对多个待观察点进行处理。

在任务中,算法需要处理的具体事项包括:对待观察点进行路径重规划,在进入观察区域后发送待观察点相对于无人艇的方位及距离,在观察完毕后发送停止指令。基于以上任务,算法为每个观察点分配了5个状态:状态“0”表示未规划,状态“1”表示待规划,状态“2”表示已规划未发送方位距离,状态“3”表示需要发送方位距离,状态“4”表示观察完毕。对于处于状态“0”的点,如果该点位于当前无人艇所在分区,则将其状态转移至状态“1”,并加入

待规划队列；当程序未处于重规划航行状态时，将待规划队列的状态为“1”的目标转移至状态“2”，并对这些点进行路径重规划，进入重规划航行状态；对于处于状态“2”的点，将其加入等待方位距离队列；对于等待方位距离队列中的点，如果程序不处于观察状态，选择距离最近的点转移至状态“3”；对处于状态“3”的点，如果其在航线上投影点位于无人艇投影点的后方，则转移至状态“4”，发送停止指令，否则持续计算方位距离并发送。

算法中使用两个队列存储待观察点，分别为待重规划队列和等待方位距离队列，其目的是分批对目标点进行处理。在每次路径重规划前，要判断当前是否位于重规划路径上，若位于重规划路径上，则

当前节拍不能进行新的重规划，以此来避免重规划路径偏离预规划路径过多。在进行重规划时，输入的点为待重规划队中的待观察点，因为在加入待重规划队列前，这些点经过判断，保证其位于当前分区内，符合分区策略。在对多个点进行重规划时，也要对这些点按行进方向从近到远排序，按照顺序连接每个点的重规划路径，避免产生迂回路径。在发送方位距离时，由于每次只能观察一个待观察点，因此使用队列的形式将点存储，只有上一个点观察完毕，才会取出队列中的下一个点，计算方位距离进行观察。

多点处理流程图如图 6 所示。

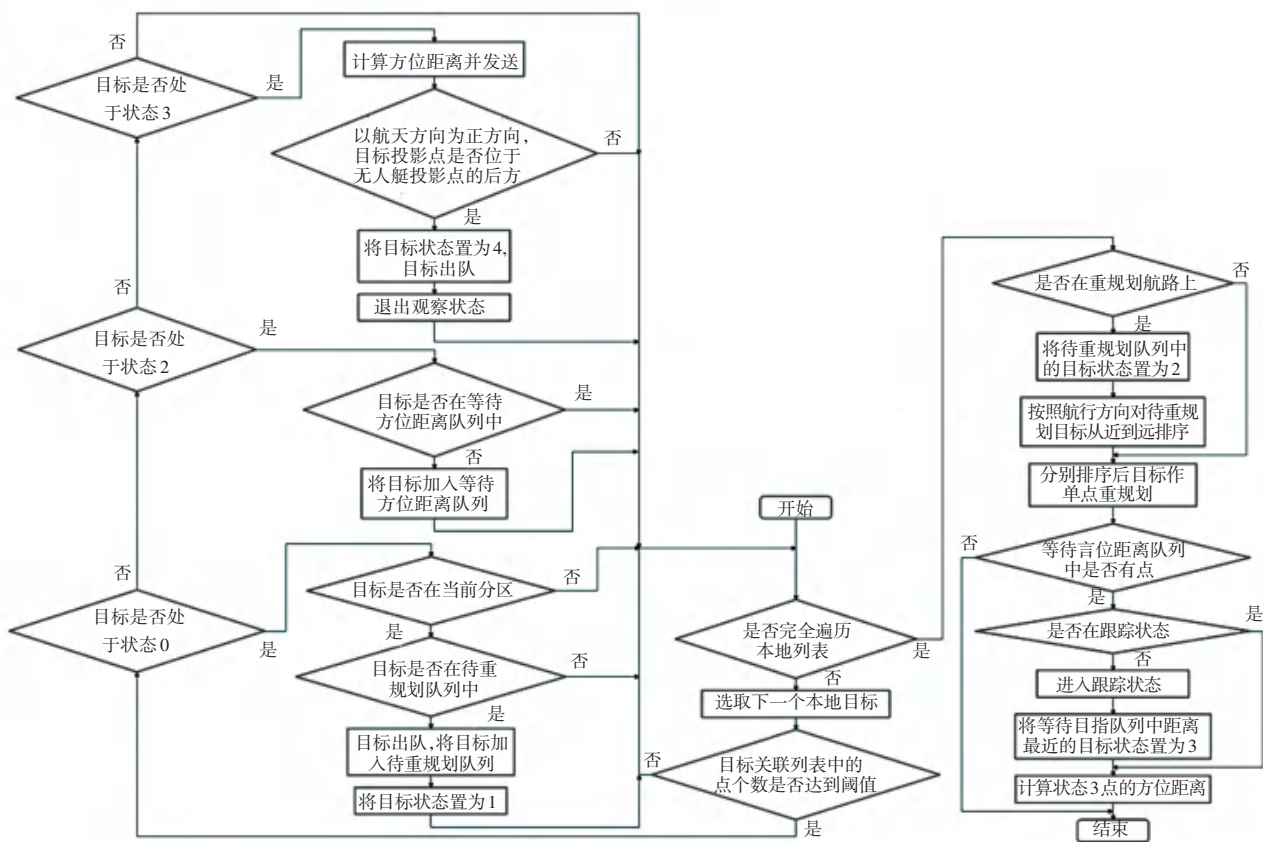


图 6 多点处理流程图

Fig. 6 Multi-point processing flowchart

### 3 实验与结果分析

实验采用三组数据考察算法在不同场景下的运行情况。数据一是常规数据，各目标点距离适中，目标点分布较为均匀；数据二中，部分目标点位于同一分区，但位于预规划路径的不同侧；数据三为极端数据，考察在不满足问题条件时算法对数据的执行效果。在实验中，本文在 0~100 之间生成数据点来模拟应用场景中的地理位置坐标点。

#### 3.1 实验一

表 1 为数据一的关键点位置分布。本组数据主要考察在最朴素情况下，两阶段路径规划对任务执行的效果。在数据一中，目标点的不安全区域半径为 250 m，观察区域半径为 260 m，所有目标点均位于其他目标点的观察区域外。图 7 为数据一对应的仿真结果。根据仿真结果可以看出，目标点由于预规划路径对区域的覆盖率为 100%，重规划过程在



分区策略的基础上保证了预规划阶段的覆盖完整形,因此无人艇在目标点附近均产生近距离观察的动作,满足了观测任务的执行要求。

表1 数据一各关键点位置

Tab. 1 Location of key points of data one

	横坐标	纵坐标
区域点 1	100.00000000	100.00000000
区域点 2	100.00000000	0.00000000
区域点 3	0.00000000	0.00000000
区域点 4	0.00000000	100.00000000
目标点 1	70.73170731	73.12500000
目标点 2	70.73170731	29.37500000
目标点 3	41.15853658	29.37500000
目标点 4	41.15853658	73.12500000

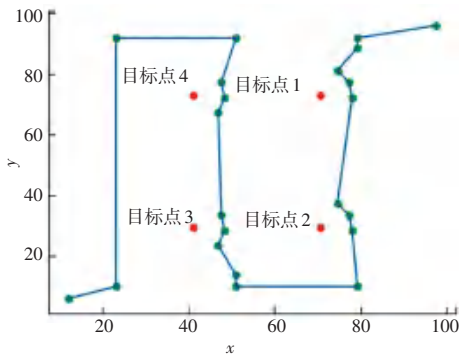


图7 数据一仿真结果

Fig. 7 Simulation result of data one

### 3.2 实验二

表2为数据二的关键点位置分布。本组数据主要考察在最朴素情况下,两阶段路径规划对任务执行的效果。在数据二中,目标点的不安全区域半径为250 m,观察区域半径为260 m,所有目标点均位于其他目标点的观察区域外。图8为数据二对应的仿真结果。由仿真结果可以看出,所有目标点均被传感器探测到,并进行了抵近观测的动作,最终行驶出的完整路径对各个目标点均有重规划路径段。在对目标点3和目标点4的观察动作产生的路径中,能够验证分区策略被有效执行,观测路径的环绕方向保证了重规划路径没有对预规划路径的完整覆盖造成破坏。

表2 数据二各关键点位置

Tab. 2 Location of key points of data two

	横坐标	纵坐标
区域点 1	0.00000000	100.00000000
区域点 2	100.00000000	100.00000000
区域点 3	100.00000000	0.00000000
区域点 4	0.00000000	0.00000000
目标点 1	24.62121212	66.87500000
目标点 2	33.61742424	48.12500000
目标点 3	45.45454545	25.62500000
目标点 4	61.07954545	43.75000000
目标点 5	82.38636364	46.25000000

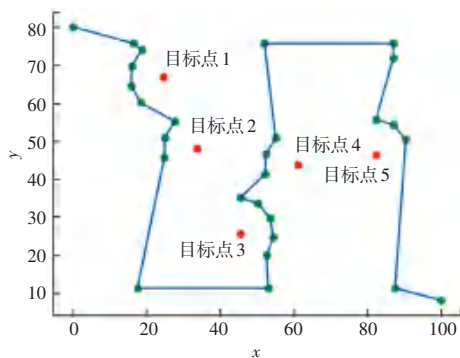


图8 数据二仿真结果

Fig. 8 Simulation result of data two

### 3.3 实验三

表3为数据三的关键点位置分布。数据三中,目标点的不可进入区域半径为250 m,观察区域半径为260 m,其中目标点1和目标点2的距离为143 m,目标点4和目标点5与区域边界距离小于任务指定的常数 $\varepsilon_0$ 。设置此组不满足问题要求的数据,主要考察本文提出策略对于不满足问题要求的情况的执行效果,发现策略的问题,确定未来的研究方向。图9为数据三对应的仿真结果。根据仿真结果,算法在重规划时对目标点1做出的重规划路径进入了目标点2的不可进入区域,同时在边界附近进入了目标点4和目标点5的不可进入区域。在这组数据中,由于目标点1和目标点2距离小于不可进入区域的半径,因此要保证两个点均可被观察到,会导致路径进入其中一点的不可进入区域;对于目标点4和目标点5,基于问题中目标点与区域边界距离的约束,算法的分区策略在单一方向上分区,因此在转弯路径上未对目标点重规划,导致路径穿过目标点的不可进入区域。基于数据三对于算法执行情况考察,未来在此工作的基础上,对于不满足问题约束的部分数据,工作重点可以放在多个目标点的不可进入区域重叠的情况上,以及目标点距离边界过近的情况上的处理流程。

表3 数据三各关键点位置

Tab.3 Location of key points of data 3

	横坐标	纵坐标
区域点 1	0.00000000	100.00000000
区域点 2	100.00000000	100.00000000
区域点 3	100.00000000	0.00000000
区域点 4	0.00000000	0.00000000
目标点 1	31.72348485	62.50000000
目标点 2	26.51515152	62.50000000
目标点 3	17.99242424	26.87500000
目标点 4	40.40403030	14.37500000
目标点 5	68.81312121	76.25000000
目标点 6	75.04736364	37.50000000

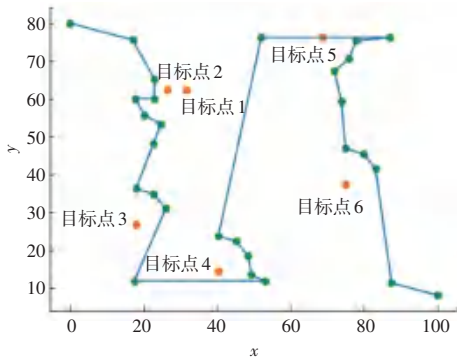


图 9 数据三仿真结果

Fig. 9 Simulation result of data three

#### 4 结束语

根据实验结果,在满足问题约束的常规数据下,算法可以合理规划出路径,探测到所有待观察点,并

(上接第 75 页)

表 3 不同分割方法结果

Tab. 3 Values of different segmentation results

算法	coins			cameraman		
	分割阈值 ( $s^*, t^*$ )	最大值	时间/s	分割阈值 ( $s^*, t^*$ )	最大值	时间/s
GOA	(200, 198)	11.14	1.63	(102, 97)	8.14	1.72
IGOA	(150, 133)	14.48	0.65	(98, 100)	13.53	0.67

#### 5 结束语

针对原蝗虫优化算法收敛慢、寻优精度低等问题,本文在初始化阶段采用贝塔分布来生成初始个体,提高了初始种群均匀分布,有助于全局勘探。通过 2 个非线性惯性权重策略来取代本文中的线性递减策略,第一个惯性权重主要是为了前期全局搜索,第二个是为了后期快速收敛。通过基准函数试验结果得知,改进算法在寻优精度及收敛速度上都有明显体现。将改进的蝗虫优化算法(IGOA)用于图像分割,解决了图像分割中阈值选取计算量大而导致分割速度慢的问题。通过与原算法分割结果图的比较,证明改进算法优于原算法。下一步将主要研究如何将改进算法应用于多阈值图像分割的问题。

#### 参考文献

[1] SARKAR S, SEN N, KUNDU A, et al. A differentialevolutionary multilevel segmentation of near infra-redimages using Renyi's entropy [C]//Proceedings of theinternational conference on frontiers of intelligentcomputing: Theory and applications (FICTA). Springer, Berlin, Heidelberg, 2013:699-706.

[2] ZHOU C, TIAN L, ZHAO H, et al. A method of two-dimensional Otsu image threshold segmentation based onimproved firefly algorithm [C]//IEEE InternationalConference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015:1420-1424.

[3] 曹建农. 图像分割的熵方法综述[J]. 模式识别与人工智能, 2012, 25(6):958-971.

[4] 于洋, 孔琳, 虞闯. 自适应粒子群集优化二维 OTSU 的图像阈值分割算法[J]. 电子测量与仪器学报, 2017, 31(6):827-832.

对待观察点做出合理的重规划路径,满足进入观察区域但不进入不可进入区域的约束。而对于不满足问题约束的部分数据,算法会做出带有风险的决策,导致路径进入待观察点的不可进入区域。在未来的工作中,工作重点可以放在多个待观察点的不可进入区域重叠的情况下,以及待观察点距离边界过近的情况上的处理流程。

#### 参考文献

[1] 高冠强, 辛斌 Bin. A-STC:一种基于拍卖的多机器人协同螺旋生成树覆盖运动规划方法[J]. Frontiers of Information Technology & Electronic Engineering, 2019, 20(1): 18-31.

[2] FAN Yunsheng, ZHAO Yongsheng, SHI Linlong. Global path planning of unmanned surface vehicle based on electronic chart rasterization[J]. Navigation of China, 2017, 40(1): 47-52, 113.

[5] HORNG M H. Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization[J]. Expert System with Applications, 2010, 37(6):4580-4592.

[6] 张金果, 郭海涛, 吴君鹏, 等. 改进的最小交叉 Tsallis 熵的小目标声呐图像分割[J]. 吉林大学学报(工学版), 2014, 44(3): 834-839.

[7] 吴诗娅, 吴一全, 周建江. 分解的二维非对称 Tsallis 交叉熵图像阈值选取[J]. 图学学报, 2015, 36(5):763-770.

[8] 马英辉, 吴一全. 基于二维 Renyi 交叉熵的刀具磨损图像分割[J]. 电子测量与仪器学报, 2016, 30(12):1869-1876.

[9] 聂方彦, 李建奇, 张平风, 等. 一种基于 Tsallis 相对熵的图像分割阈值选取方法[J]. 激光与光电子学进展, 2017, 54(7): 137-144.

[10] 魏晶茹, 马瑜, 夏瑞, 等. 基于分数阶粒子群的 Otsu 图像分割算法[J]. 计算机工程与设计, 2017(12):3284-3290.

[11] 宋锦, 高浩, 王保云. 改进人工蜂群算法在图像分割中的应用[J]. 电视技术, 2016, 40(8):8-14, 25.

[12] 李贤阳, 黄婵. 一种结合改进 OTSU 法和改进遗传算法的图像分割方法[J]. 实验室研究与探索, 2012, 31(12):57-61, 112.

[13] 陈恺, 陈芳, 戴敏, 等. 基于萤火虫算法的二维熵多阈值快速图像分割[J]. 光学精密工程, 2014, 22(2):517-523.

[14] SAREMI S, MIRJALILI S, LEWIS A. Grasshopperoptimisation algorithm:Theory and application[J]. Advances in Engineering Software, 2017(105):30-47.

[15] 张达敏, 徐航, 王依柔, 等. 嵌入 Circle 映射和逐维小孔成像反向学习的鲸鱼优化算法[J/OL]. 控制与决策:1-8. https://doi.org/10.13195/j.kzyjc.2019.1362.

[16] 黄洋, 鲁海燕, 许凯波, 等. 一种动态调整惯性权重的简化均值粒子群优化算法[J]. 小型微型计算机系统, 2018, 39(12): 2590-2595.

[17] 华东师范大学数学系. 数学分析(下册)(第 3 版)[M]. 高等教育出版社, 2001.