

# 控制与决策

Control and Decision

## 基于已知地形信息的海底机器人路径规划

高存璋, 谷海涛

引用本文:

高存璋, 谷海涛. 基于已知地形信息的海底机器人路径规划[J]. *控制与决策*, 2022, 37(9): 2296–2304.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.0343>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 非平坦地形下移动机器人安全路径规划

Safe path planning of mobile robot in uneven terrain

控制与决策. 2022, 37(2): 323–330 <https://doi.org/10.13195/j.kzyjc.2020.1221>

#### 基于改进蚁群和DWA算法的机器人动态路径规划

Robot dynamic path planning based on improved ant colony and DWA algorithm

控制与决策. 2022, 37(9): 2211–2216 <https://doi.org/10.13195/j.kzyjc.2021.1804>

#### 机器人信息增益RRT环境探索算法

Robot RRT based on information gain for environment exploration

控制与决策. 2021, 36(11): 2683–2689 <https://doi.org/10.13195/j.kzyjc.2020.1007>

#### 基于改进RRT\*FN算法的机器人路径规划

Robot path planning based on improved RRT\*FN algorithm

控制与决策. 2021, 36(8): 1834–1840 <https://doi.org/10.13195/j.kzyjc.2019.1713>

#### 凸优化与A\*算法结合的路径避障算法

Convex optimization and A-star algorithm combined path planning and obstacle avoidance algorithm

控制与决策. 2020, 35(12): 2907–2914 <https://doi.org/10.13195/j.kzyjc.2019.0351>

# 基于已知地形信息的海底机器人路径规划

高存璋<sup>1,2,3</sup>, 谷海涛<sup>1,2†</sup>

- (1. 中国科学院沈阳自动化研究所 机器人学国家重点实验室, 沈阳 110016;  
2. 中国科学院机器人与智能制造创新研究院, 沈阳 110169; 3. 中国科学院大学, 北京 100049)

**摘要:** 路径规划是实现机器人智能化的重要组成部分, 规划路径的优劣在很大程度上决定了机器人执行任务的效果. 传统的路径规划算法, 例如基于图搜索的 *dijkstra* 算法和其改进后的 *A\** 算法, 以及基于采样的 *RRT* (rapidly-exploring random tree) 算法和其改进后的 *RRT\** 算法, 仅仅考虑了避障问题; 基于插值曲线的算法可以产生较为光滑的轨迹; 基于数值优化的算法可以将机器人速度、加速度等加入损失函数, 通过优化求解, 产生动力学特性较好的轨迹. 然而, 面对当前越来越精确、丰富的先验地形信息, 鲜有算法可以充分利用他们. 对此, 基于海底数字高程地图 (digital elevation map, DEM), 提出扩展 *A\** 算法及 *FM* (fast marching) 算法改进算法, 能够利用先验地形信息提高路径规划的效果. 通过仿真分析, 对比 3 种算法: 扩展 *A\** 算法、*TC FM* (terrian cared fast marching) 和 *TC FM\** 算法, 仿真结果表明, 扩展 *A\** 算法求解速度更快、局部规划能力更强, *TC FM* 和 *TC FM\** 算法所求得的路径更短、更光滑.

**关键词:** 海底机器人; 路径规划; 数字高程地图; 扩展 *A\** 算法; *TC FM* 和 *TC FM\** 算法

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.0343

开放科学(资源服务)标识码(OSID):



引用格式: 高存璋, 谷海涛. 基于已知地形信息的海底机器人路径规划[J]. 控制与决策, 2022, 37(9): 2296-2304.

## Seabed robot path planning based on priori terrain information

GAO Cun-zhang<sup>1,2,3</sup>, GU Hai-tao<sup>1,2†</sup>

- (1. State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; 2. Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China; 3. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** Path planning is an important part of realizing robot autonomous movement. The planned path largely determine the performance of the robot in work. Traditional path planning algorithms, such as the *dijkstra* algorithm based on graph search and its improved version the *A\** algorithm, as well as the sampling-based *RRT* (rapidly-exploring random tree) algorithm and its improved version the *RRT\** algorithm, only consider obstacle avoidance problems. Algorithms based on interpolation curves can generate smoother trajectories. Algorithms based on numerical optimization can add robot speed, acceleration, etc. to the loss function, which can generate trajectories with better dynamic properties through optimal solving. In current, topographic information are richer and more accurate, and few algorithms can make full use of them. Therefore, based on the digital elevation map (DEM) of the seabed, this paper proposes an extended *A\** algorithm and an improved *FM* (fast marching) algorithm, using prior geographic information to improve the effect of path planning. Through simulation analysis, three algorithms are compared: the extended *A\** algorithm, the *TC FM* (terrian cared fast marching) and the *TC FM\** algorithm. Simulation results show that the extended *A\** algorithm solves faster, its the local planning ability is stronger, and the *TC FM* and *TC FM\** algorithms find the path shorter and smoother.

**Keywords:** seabed robot; path planning; DEM; extend *A\** algorithm; *TC FM* and *TC FM\** algorithm

## 0 引言

机器人想要实现智能化, 需要解决自主导航、决策的问题. 这需要一系列模块的配合: 首先, 环境感

知模块需要采集当前环境的信息, 完成定位和实时建图; 其次, 基于当前已知信息, 决策模块需要完成运动规划和行为规划等一系列决策任务; 然后, 控制

收稿日期: 2021-03-01; 录用日期: 2021-06-17.

基金项目: 中国科学院海洋信息技术创新研究院前沿基础研究项目 (QYJC201913); 十三五预研项目 (2020107/2002).

责任编辑: 警斌.

†通讯作者. E-mail: ght@sia.cn.

模块根据决策信息来控制机器人的运动, 以实现期望目标. 本文主要关注于决策模块中的路径规划任务. Mac等<sup>[1]</sup>将路径规划方法分为经典方法和启发式方法: 经典方法又分为基于图搜索的算法、基于采样的算法、人工势场(APF)、插值曲线等方法; 启发式算法分为神经网络、模糊控制、自然启发式算法(蚁群算法、粒子群算法等)及其混合算法. 作者统计发现: 2000年之前, 经典方法在路径规划方面占统治地位; 之后, 启发式方法日趋流行. 然而, 这种流行主要存在于学术研究方面, 在实际工程问题上, 特别是自动驾驶领域, 广泛使用的是经典算法. González等<sup>[2]</sup>总结了自动驾驶领域的路径规划算法, 并指出, 研究机构和企业最广泛关注的是插值曲线的方法, 因为插值曲线可以考虑车辆的约束、速度和加速度等因素, 产生较为平滑的曲线, 这对于乘客的乘坐体验是非常重要的. 在实际应用中, 基于图搜索的方法是第2种被广泛应用的方法.

经典的基于图搜索的路径规划算法有 dijkstra 算法、A\* 算法<sup>[3]</sup>、D\* 算法<sup>[4]</sup>、D\*Lite 算法<sup>[5]</sup>、LifelongPlanning A\* 算法<sup>[6]</sup>、Anytime D\* 算法<sup>[7]</sup>、Field D\* 算法<sup>[8]</sup>等. dijkstra 算法是一种广泛使用的广度优先图遍历算法, 将其应用于路径规划问题, 存在边缘无用图节点也纳入计算的问题, 因此, 受加入目标点对于搜索方向的启发, 人们提出了 A\* 算法. 然而, 面对动态变化的环境, 上述图搜索算法都显得无能为力, 而 D\* 算法、Field D\* 算法等具有动态路径规划功能的算法解决了这个问题.

上述基于图搜索的算法仅仅考虑了避障、简单的机器人运动学约束问题以及插值曲线方法可以产生较平滑的轨迹, 但这对于已有的高精度地图提供的丰富地形信息都没有加以利用. FM 算法最早由 Sethian<sup>[9]</sup> 提出, 可以通过解 Eikonal 方程模拟波的前向传播, 是水平集方法的一种, 被广泛应用于路径规划问题. 文献 [10-12] 将其应用于无人机路径规划, Petres 等<sup>[13]</sup> 将其应用于水下机器人路径规划, Chen 等<sup>[14]</sup> 将其应用于无人船路径规划. 近些年, FM 算法和其他算法相结合<sup>[15-18]</sup>, 被用于最优运动规划和无人机的区域覆盖任务, 展现了 FM 算法的强大功能. 通过 FM 算法, 可以综合利用地形信息, 定义地图上每一点波的传播速度, 通过解 Eikonal 方程, 产生一个由目标点单调增加的势场, 然后利用梯度下降方法可以求得一条最优轨迹.

本文对 A\* 算法和 FM 算法加以改进, 提出扩展 A\* 算法、TC FM 和 TC FM\* 算法, 使其能够利用先验

地形信息进行路径规划, 同时对 3 种算法进行对比分析, 3 种算法都可以基于数字高程地图 (DEM) 进行路径规划. 仿真分析表明, 扩展 A\* 算法求解速度更快、局部规划能力更强, TC FM 和 TC FM\* 算法求得路径更短、更光滑.

## 1 背景知识

### 1.1 数字高程地图

数字高程地图 (DEM) 是通过地面的高程数据展示地形地貌等地理特征, 在地理信息系统学科中, 数字高程地图的存储包含多种规范格式, 一般包含经纬度、海拔高度等基本信息, 其数据表述如图 1 所示.

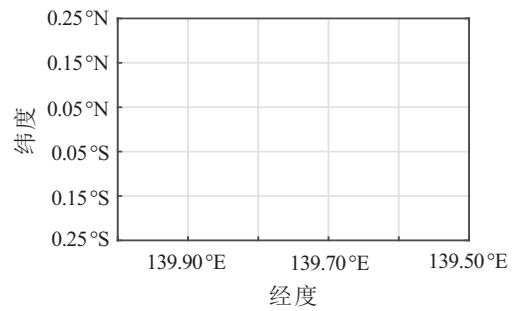


图 1 DEM 数据格式

图 1 中: 横坐标为经度值 (longitude), 纵坐标为纬度值 (latitude).  $peakData(x_i, y_j) \{i = 1, 2, \dots, M; j = 1, 2, \dots, N\}$  表述经度为  $x_i$ 、纬度为  $y_j$  处海拔高度 (海水深度). 本文使用的是 GEBCO<sup>[19]</sup> 提供的数字海底高程地图, 如图 2 所示, 为  $0.25^\circ S \sim 0.25^\circ N$ ,  $139.5^\circ E \sim 140.0^\circ E$  范围内的地形情况.

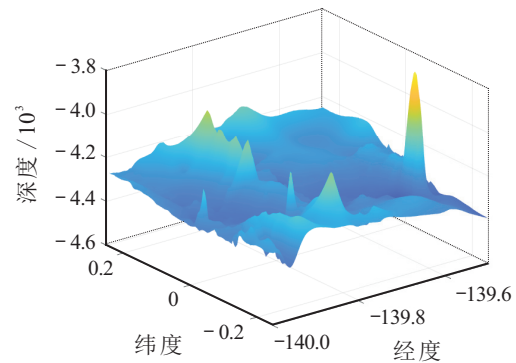


图 2 某海域海底地形情况

### 1.2 A\* 算法和 FM 算法

A\* 算法是一类经典的路径规划算法, 其伪代码如下 algorithm 1 所示.

**algorithm 1** A\* 算法.

input: node\_start, node\_goal;

output: prev.

/\* 存储已遍历节点的父节点 \*/

1) node\_start.f\_g  $\leftarrow$  0, node\_start.state  $\leftarrow$

```

FROZEN;
/* 节点的  $f_g$  属性,表示此节点到初始点的距离 */
2) foreach node  $\in$  node_start.neighbours do
3)   node.state  $\leftarrow$  NARROW BAND;
4)   narrow_band.insert(node);
5)   node.f_g  $\leftarrow$  distance(node_start, node);
6)   prev(node)  $\leftarrow$  node_start;
7) end
8) while  $\sim$  empty(narrow_band) do
9)   current_node  $\leftarrow$  narrow_band.popfirst();
10)  current_node.state  $\leftarrow$  FROZEN;
11)  if current_node = node_goal then
12)    break;
13)  end
14)  foreach node  $\in$  current_node.neighbours do
15)    if node.state  $\neq$  FROZEN then
16)      f_g  $\leftarrow$  current_node.f_g +
        distance(current_node, node)
17)      if node.state = UNKNOW then
18)        node.state  $\leftarrow$ ;
        NARROW BAND
19)        narrow_band.insert(node);
20)      end
21)      if f_g  $\geq$  node.f_g then
22)        continue;
23)      end
24)      prev(node)  $\leftarrow$  current_node;
25)      node.f_g  $\leftarrow$  f_g;
26)      narrow_band.resort();
27)    end
28)  end
29) end

```

在  $A^*$  算法中,搜索空间中的节点,存在以下3种状态.

1) FROZEN: 已经确定此节点到出发点的最短距离,且此距离不再改变;

2) NARROW BAND: 活动节点,已经计算了到出发点的距离,但此距离可能不是最短距离,随附近节点的更新,此距离还可以改变;

3) UNKNOW: 还未到达的节点,初始化为到出发点距离正无穷大.

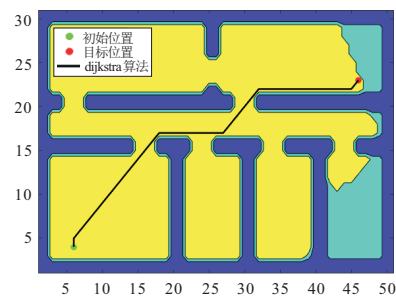
相较于 dijkstra 算法,  $A^*$  算法主要改进在

narrow\_band.popfirst() 上. dijkstra 算法 NARROW BAND 队列按到初始点的距离  $h(x_i, y_j)$  升序排列,而  $A^*$  算法衡量 NARROW BAND 各节点的损失函数为

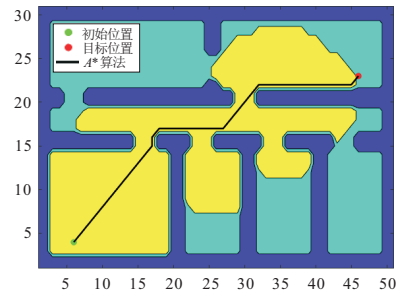
$$f(x_i, y_j) = \alpha_1 h(x_i, y_j) + \alpha_2 g(x_i, y_j). \quad (1)$$

其中:  $h(x_i, y_j)$  表示  $(x_i, y_j)$  点到初始点的距离,  $g(x_i, y_j)$  表示  $(x_i, y_j)$  点到目标点的欧氏距离. 由于  $g(x_i, y_j)$  的加入,对于搜索方向给予了一定的启发性,某些情况可以加快求解速度.  $\alpha_1$  和  $\alpha_2$  用来调节目标点方向启发性的的大小.

在建立的仿真室内环境中,二者的表现如图3所示. 深蓝色区域为围墙,深绿色区域为算法未遍历区域,黄色区域为遍历区域.



(a) dijkstra 算法



(b)  $A^*$  算法

图3 dijkstra 算法和  $A^*$  算法在室内路径规划表现

由图3可以发现, dijkstra 算法需要遍历大约90%的节点才能得到可行路径,而  $A^*$  算法只需要遍历大约55%的节点.

同样环境下,可以使用 FM 算法进行路径规划. 首先,通过求解如下 Eikonal 方程为仿真环境建立势场:

$$1 = F(x)|\nabla T(x)|. \quad (2)$$

其中:  $F(x)$  表示位置  $x$  处的速度(这里设置非障碍处为1,障碍处为0),  $T(x)$  表示波传播到位置  $x$  处的时间.

在上述产生的势场中,使用梯度下降可以求得最优路径. 整个算法伪代码如 algorithm 2 所示.

**algorithm 2** FM 算法.

input: node\_start, node\_goal;

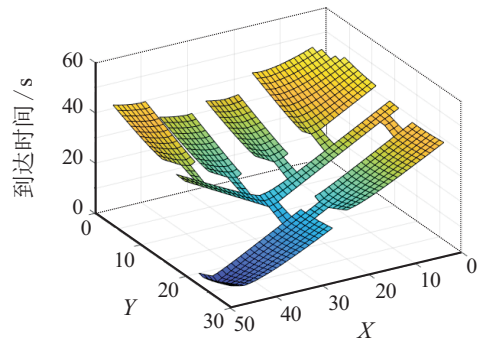
```

output: map T.
/*存储波传播到各点的时间*/
1) foreach node ∈ node_goal.neighbours do
2)   node.state ← NARROW BAND;
3)   narrow_band.insert(node);
4)   mapT.node ← solveEikonal(node);
5) end
6) while ~ isempty(narrow_band) do
7)   current_node ← narrow_band.popfirst();
8)   current_node.state ← FROZEN;
9)   foreach node ∈ current_node.neighbours do
10)    if node.state = FROZEN then
11)      skip;
12)    end
13)    else
14)      mapT.node ← solveEikonal(node);
15)      if node.state = NARROW BAND
16)        then
17)          narrow_band.resort();
18)        end
19)        if node.state = UNKNOWN then
20)          node.state ←
21)            NARROW BAND;
22)          narrow_band.insert(node);
23)        end
24)      end

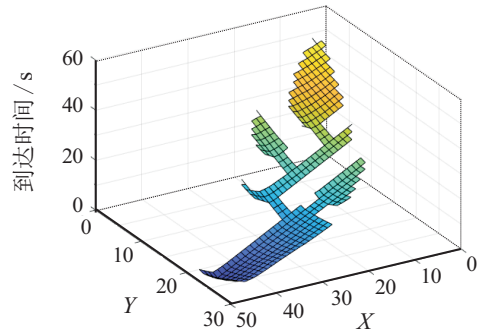
```

类似于A\*算法的思路,可以不对整个环境建立势场,通过对NARROW BAND各节点的排序中引入到初始点的距离来实现该目的,这就是FM\*算法.具体实现可以参考Valero-Gomez等<sup>[20]</sup>的分析.二者建立的势场和产生的规划路径如图4所示.由图4(c)可以发现,FM\*算法虽然遍历了较少的节点,但规划路径与FM算法差别不大.

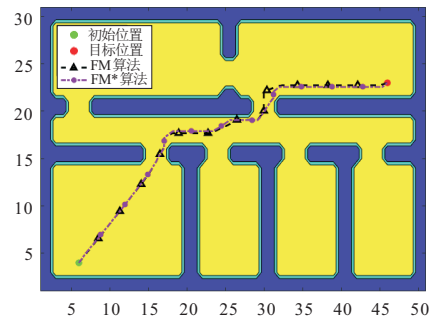
上述A\*算法、FM算法和FM\*算法仅具有在避障条件下求最短路径的能力.然而,海底机器人运行的实际海洋环境中,不仅有难以通过的障碍物,还有陡坡、深坑等地形,海底机器人通过这些地区时存在较大的侧翻、被困的风险.如图5所示,A区域为障碍物,A\*、FM、FM\*算法在最短路径原则下规划路径均通过B区域,而B区域明显存在深坑,海底机器人面临发生意外的风险.此时,经由障碍物A左上角,虽然路径距离稍长,但地形十分平坦,适宜海底机器人



(a) FM算法室内仿真环境产生的势场



(b) FM\*算法室内仿真环境产生的势场



(c) 规划路径

图4 势场和规划路径

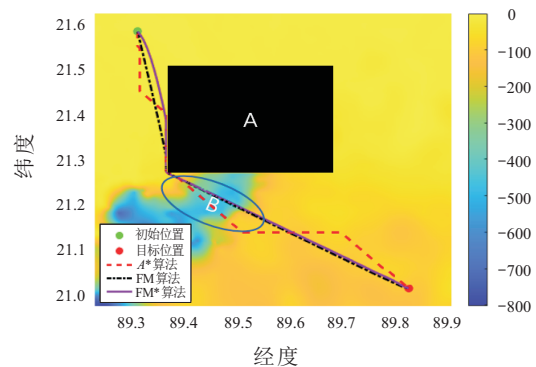


图5 A\*、FM、FM\*算法局限

运动.面对上述问题,有必要开发新的算法以解决这些问题.

## 2 基于DEM的扩展A\*和改进FM算法

1.2节介绍了A\*和FM及FM\*算法,可以发现,上述算法可以完成避障条件下的路径规划的任务.但是,随着现代传感器测量精度的不断提高和人类足迹

范围的扩大,即使火星表面和几千米深的海底,人类都已获得精度较高的地图.然而,上述算法不能利用这些地图数据,这显然是一种浪费.依靠这些先验信息,可以明显提高规划路径的质量,降低海底机器人发生侧翻、被困的机率.

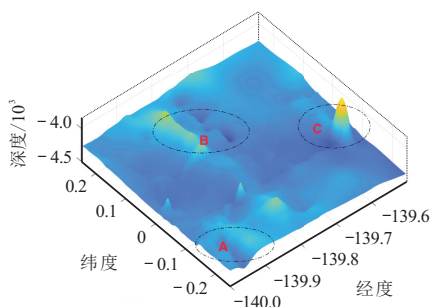
本文为实现这一目的,提出一种扩展A\*算法,并与改进后的FM及FM\*算法进行对比分析.基于海底数字高程地图,本文参考Garrido等<sup>[21]</sup>的方法,引入3种地理信息:1)海底表面坑洼度;2)海底表面坡度;3)高度变化.本文将上述参数归一化后,引入损失函数中,用来提高路径规划的效果.

假设机器人运动初始点为 $(x_s, y_s)$ ,目标点为 $(x_g, y_g)$ ,对于地图中所有节点 $(x_i, y_j) \{i = 1, 2, \dots, M, j = 1, 2, \dots, N\}$ ,分别求取如下参数.

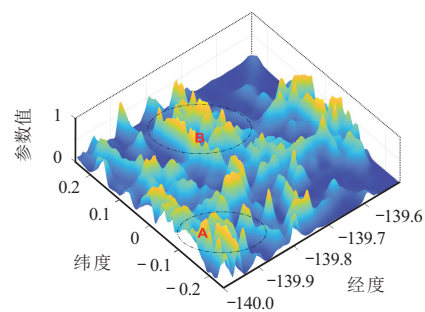
#### 1) 海底表面坑洼度.

Castejón等<sup>[22]</sup>通过计算某点处附近区域表面法向量方向偏差的方法描述这一点附近的表面粗糙度.可以直观地发现:对于一个水平面,处处法向量相互平行,法向量方向偏差为零;而对于一个球面,表面各点处法向量方向都不相同,故法向量方向偏差较大.对于一片区域,可以通过高程数据提取这一区域的一簇法向量 $(x_i, y_i, z_i), i = 1, 2, \dots, n$ (都为单位向量),因此可以定义

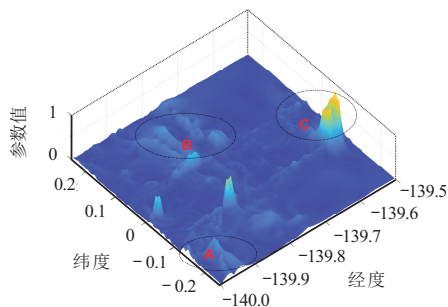
$$\begin{cases} R = \frac{1}{n} \sqrt{\left(\sum_{i=1}^n x_i\right)^2 + \left(\sum_{i=1}^n y_i\right)^2 + \left(\sum_{i=1}^n z_i\right)^2}, \\ C_R = 1 - R, \end{cases} \quad (3)$$



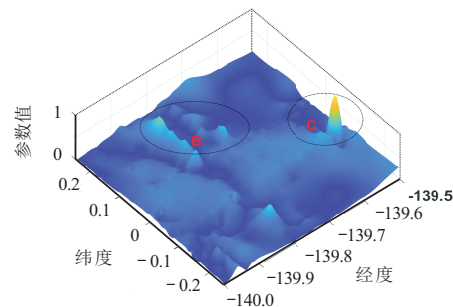
(a) 海底地形情况



(b) 海底表面坑洼度



(c) 海底表面坡度



(d) 高度变化

图6 各参数效果

其中 $C_{R_{i,j}} \in [0, 1]$ .  $C_{R_{i,j}} = 1$ ,表示 $(x_i, y_j)$ 附近区域地面法向量方向差异达到最大,地形起伏不定,坑洼较多;  $C_{R_{i,j}} = 0$ ,表示 $(x_i, y_j)$ 附近区域地面法向量都平行,地形完全是个平面,没有坑洼.

#### 2) 海底表面坡度.

将海底高程数据 $\text{peakData}(x, y)$ 简写为 $\text{pD}(x, y)$ ,则 $(x_i, y_j)$ 处海底表面坡度可以由下式计算:

$$\begin{cases} S_x = \frac{\text{pD}(x_{i+1}, y_j) - \text{pD}(x_{i-1}, y_j)}{2\Delta x}, \\ S_y = \frac{\text{pD}(x_i, y_{j+1}) - \text{pD}(x_i, y_{j-1})}{2\Delta y}, \\ S_{i,j} = \text{norm}([S_x, S_y]), \\ S_{\max} = \max(S_{i,j}), S_{\min} = \min(S_{i,j}), \\ C_{S_{i,j}} = \frac{S_{i,j} - S_{\min}}{S_{\max} - S_{\min}}. \end{cases} \quad (4)$$

其中: $\Delta x$ 、 $\Delta y$ 表示经度和纬度的高程地图栅格精度,  $\text{norm}()$ 表示对向量取模,  $\max()$ 、 $\min()$ 表示对所有元素取极大值和极小值.通过式(4)所示方法对数据进行归一化后,  $C_{S_{i,j}} \in [0, 1]$ .  $C_{S_{i,j}} = 1$ ,表示 $(x_i, y_j)$ 附近坡度较大,不适合机器人航行;  $C_{S_{i,j}} = 0$ ,表示 $(x_i, y_j)$ 附近地形水平,适合机器人运动.

#### 3) 高度变化.

$$\begin{cases} H_{i,j} = |\text{pD}(x_i, x_j) - \text{pD}(x_s, y_s)|, \\ H_{\max} = \max(H_{i,j}), H_{\min} = \min(H_{i,j}), \\ C_{H_{i,j}} = \frac{H_{i,j} - H_{\min}}{H_{\max} - H_{\min}}. \end{cases} \quad (5)$$

同样,经过归一化后,  $C_{H_{i,j}} \in [0, 1]$ , 将它加入损失函数, 将会倾向于得到高度变化小的路径。

对于图2所示地区的海底地形, 各地理信息参数计算结果如图6所示. 由图6可见: A区域明显有一个较深的坑, 坡度变化较大, 坑洼度和坡度参数值在此区域较大; B区域左边有连续海底山脉, 右侧有不规则沟壑, 坑洼度、坡度、高度变化参数值在此区域都较大; C区域有明显突起山峰, 此处坡度、高度变化参数值很大. 上述A、B、C三个区域地形复杂, 显然不适合机器人经过, 对应这些地方, 通过图6(b)~(d)可以发现, 文中所定义的3个参数能很好地表达这些信息, 可以用于接下来的路径规划。

### 2.1 扩展A\*算法

将上述各参数引入衡量各节点的损失函数中, 即

$$\begin{cases} f(x_i, y_j) = \alpha_1 h(x_i, y_j) + \alpha_2 C(x_i, y_j), \\ C(x_i, y_j) = \beta_1 g(x_i, y_j) + \beta_2 C_{R_{i,j}} + \beta_3 C_{S_{i,j}} + \beta_4 C_{H_{i,j}}. \end{cases} \quad (6)$$

其中:  $\beta_1 \sim \beta_4$  为各参数的权重因子, 且  $\beta_i < 1, i = 1, \dots, 4$ . 修改后的算法称为扩展A\*算法 (extended A\* algorithm).

### 2.2 考虑地形信息的FM算法

在1.2节, 将仿真环境内各节点的速度统一为1, 表示在环境内所有非障碍区域, 机器人都能以最大速度运动. 这一节, 将上述3种地理信息引入速度决策中, 有

$$F_{i,j} = 1 - (\gamma_1 C_{R_{i,j}} + \gamma_2 C_{S_{i,j}} + \gamma_3 C_{H_{i,j}}). \quad (7)$$

其中:  $\sum_{q=1}^3 \gamma_q = 1, \gamma_q$  取值大小用来衡量各部分参数的权重. 由于  $C_R, C_S, C_H$  都进行了归一化,  $F_{i,j} \in [0, 1]$ .  $F_{i,j} = 0$ , 表示机器人在  $(x_i, y_j)$  点不能通过;  $F_{i,j} = 1$ , 表示机器人可以在  $(x_i, y_j)$  点以最大速度运动. 上述改进后的FM算法称为考虑地形信息 (terrific) 的FM算法 (TC FM algorithm). 参照FM与FM\*算法关系, 对其增加启发项, 获得TC FM\*算法。

## 3 仿真与分析

为了展现扩展A\*算法、TC FM、TC FM\*算法路径规划的效果, 本文以传感器受限制, 需要水下近底航行的AUV和水下履带机器人为原型, 进行3个环境下的仿真分析, 并且考察不同数据规模下算法的表现. 算法各参数的取值如表1所示。

为了定量分析3种算法的性能, 本文设计了3个指标: 1) 路径长度; 2) 平均高度变化; 3) 平均坡度变化. 假设路径规划算法求得的路径为  $\text{Path} = \{p_1,$

表1 仿真参数取值

参数	取值	参数	取值
$\alpha_1$	1	$\beta_4$	0.25
$\alpha_2$	0.6	$\gamma_1$	0.4
$\beta_1$	0.24	$\gamma_2$	0.2
$\beta_2$	0.25	$\gamma_3$	0.4
$\beta_3$	0.45		

$p_2, \dots, p_n\}$ . 其中:  $p_i = (\text{lon}_i, \text{lat}_i) (1 \leq i \leq n)$  为规划路径上的第*i*个点, lon、lat分别表示经、纬度。

#### 1) 路径长度.

赤道上, 经度一度为111 km, 因此, 经度方向的路径长度可以通过  $111 \times \cos(\text{lat}) \times \Delta \text{lon}$  计算; 纬度方向的路径长度可以通过  $111 \times \Delta \text{lat}$  计算. 参照图7, 规划路径第*i*段长度及路径总长度PL可按下式计算:

$$\begin{cases} \Delta x = 111 \times \cos\left(\text{lat}_i + \frac{\Delta \text{lat}_i}{2}\right) \times \Delta \text{lon}_i, \\ \Delta y = 111 \times \Delta \text{lat}_i, \\ \text{PL}_i = \sqrt{\Delta x^2 + \Delta y^2}, \\ \text{PL} = \sum_{i=1}^{n-1} \text{PL}_i. \end{cases} \quad (8)$$

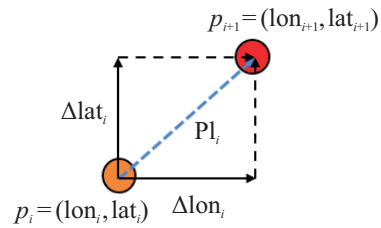


图7 路径长度计算

#### 2) 平均高度变化.

第*i*点的高度变化  $\text{HC}_i$  取当前点相对于路径下一采样点高度的差, 路径每一采样点高度变化及路径平均高度变化 (EHC) 如下式所示:

$$\begin{cases} \text{HC}_i = |\text{pD}(p_i) - \text{pD}(p_{i+1})|, \\ \text{EHC} = \frac{1}{n-1} \sum_{i=1}^{n-1} \text{HC}_i. \end{cases} \quad (9)$$

#### 3) 平均坡度变化.

采用式(4)定义的归一化后参数  $C_{S_{i,j}}$  度量当前点坡度. 与高度变化相似, 第*i*点的坡度变化  $\text{SC}_i$  取当前点相对于路径下一采样点  $C_{S_{i,j}}$  参数的差, 路径每一采样点坡度变化及路径平均坡度变化 (ESC) 如下式所示:

$$\begin{cases} \text{SC}_i = |C(p_i) - C(p_{i+1})|, \\ \text{ESC} = \frac{1}{n-1} \sum_{i=1}^{n-1} \text{SC}_i. \end{cases} \quad (10)$$

### 3.1 仿真分析1(地形1)

首先,在图2所示的海底地形环境下进行仿真,设定初始位置为  $0.23^{\circ}\text{S}, 139.99^{\circ}\text{E}$ , 目标位置为  $0.20^{\circ}\text{N}, 139.53^{\circ}\text{E}$ . 算法规划路径如图8所示.

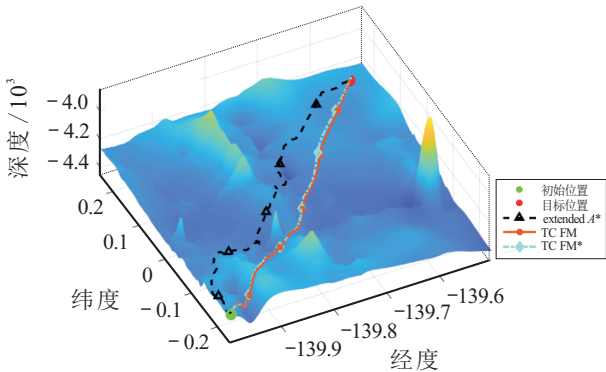


图8 地形1环境下仿真结果

通过图8可以发现:对于第2节指出的A区域(有一个深坑),扩展A\*算法能够规划出一条坡度较缓的路径通过此区域,而TC FM算法和TC FM\*算法不能产生该效果;但是,扩展A\*算法规划路径较长,会浪费机器人更多的能量.对于仿真区域中突出的山峰,3种算法都能进行规避.

表2给出了在地形1中,3种算法规划路径的长度、路径平均高度变化、平均坡度变化.对比3种算法可以发现,TC FM、TC FM\*算法规划路径较短,且平均高度变化和平均坡度变化较小.

表2 地形1中各算法对比

算法	路径长度/km	平均高度变化/m	平均坡度变化
extendedA*	85.9	4.96	0.0119
TC FM	72.4	2.39	0.0048
TC FM*	77.4	2.36	0.0049

### 3.2 仿真分析2(地形2)

地形2选取  $26.90^{\circ}\text{N} \sim 29.36^{\circ}\text{N}, 115.43^{\circ}\text{E} \sim 116.92^{\circ}\text{E}$  范围内的区域进行仿真.此区域内斜对角线方向地形较平坦,另外两侧区域,一侧起伏不平,一侧有陡峭的山坡,高度变化很大.设定初始位置为  $29.23^{\circ}\text{N}, 116.84^{\circ}\text{E}$ , 目标位置为  $27.11^{\circ}\text{N}, 115.56^{\circ}\text{E}$ . 相较于仿真分析1,这次仿真主要验证各算法应对坑洼较多的复杂地形时的路径规划能力.实验结果如图9所示.通过图9可以发现,3种算法都能沿着较平缓的斜对角山谷规划路径.相比较而言:因TC FM算法和TC FM\*算法是在所求得的势场上通过梯度下降的方法获得路径,故规划的路径较平滑;而扩展A\*算法求得的路径方向变化较大,路径较长.表3数据验证了这点.

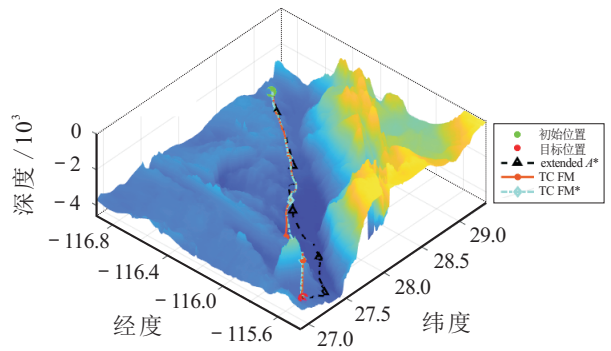


图9 地形2环境下仿真结果

表3 地形2中各算法对比

算法	路径长度/km	平均高度变化/m	平均坡度变化
extendedA*	283.9	14.91	0.0021
TC FM	258.1	13.10	0.0017
TC FM*	257.5	12.02	0.0016

对于临近目标点的凸起山丘,扩展A\*算法能够沿着坡度较小的区域规划路径,而TC FM算法和TC FM\*算法直接越过了山丘,从而展现了扩展A\*算法更强的局部规划能力.

### 3.3 仿真分析3(地形3)

地形3选取  $32.06^{\circ}\text{N} \sim 33.03^{\circ}\text{N}, 119.90^{\circ}\text{E} \sim 120.63^{\circ}\text{E}$  范围内区域进行仿真.此区域左下方地形平坦,右上方水深较浅,两部分之间地形剧烈变化,且在水深较浅的右上角区域有一处深沟(图10方框标出).设定初始位置为  $32.16^{\circ}\text{N}, 120.59^{\circ}\text{E}$ , 位于图10左下角;目标位置为  $32.96^{\circ}\text{N}, 119.95^{\circ}\text{E}$ , 位于图10右上角.在此环境下,仿真实验结果如图10所示.

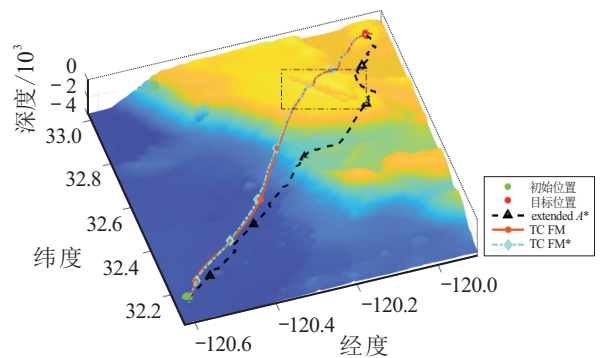


图10 地形3环境下仿真结果

表4 地形3中各算法对比

算法	路径长度/km	平均高度变化/m	平均坡度变化
extendedA*	120.7	29.74	0.0078
TC FM	109.8	13.34	0.0051
TC FM*	109.6	13.98	0.0052



通过图10和表4可以发现:扩展A\*算法规划路径相较于TC FM算法和TC FM\*算法,距离较长;但是,对于图10方框标出的沟壑地形,A\*算法能够牺牲距离,绕过坡度较大的沟壑,规划出平坦的轨迹。

### 3.4 数据规模对算法性能的影响

前面已经分析了扩展A\*算法、TC FM、TC FM\*算法的路径规划效果,本节主要分析各算法应对于不同数据规模时的求解时间。仿真计算机配置为AMD R7 4800H处理器,主频2.9 GHz,16 G内存,64位Windows10操作系统,使用仿真软件为Matlab 2019b。应用上述3种算法在高程图不同栅格数量下进行仿真计算,结果如图11所示。

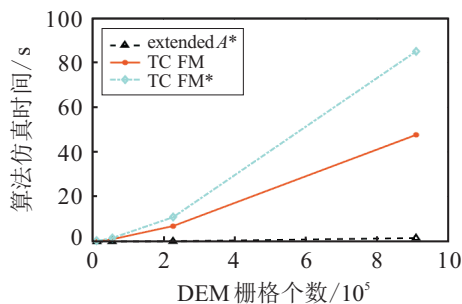


图11 各算法仿真时间

由图11可以发现,3种算法中,扩展A\*算法仿真消耗时间最短。其实由algorithm 1、algorithm 2也可以推测,扩展A\*算法是一种图搜索算法,由于扩展A\*算法在A\*算法的基础上改进而来,相较于dijkstra算法,增加了启发性,减小了需要遍历节点的数量。Garrido等<sup>[21]</sup>将FM算法归类为人工势场(APF)一类的算法,此算法主要计算量在于求解Eikonal方程建立势场方面,计算量相对于基于图搜索的扩展A\*算法明显更大。由图11也可以发现,增加启发性的FM\*算法在求解Eikonal方程建立势场时,虽然也加入了目标点对搜索方向的启发性,但求解时间相对于FM算法,不降反升。这主要是由于在algorithm 2 narrow\_band.popfirst()中,FM\*算法需要导入当前节点到目标点的距离。对于10的5次方数量级的栅格节点,扩展A\*算法可以在2s内完成求解,明显优于TC FM算法和TC FM\*算法,这也一定程度上展现了基于图搜索的路径规划算法的高效性。

## 4 结论

路径规划是机器人决策控制的重要组成部分,是展现机器人的智能化水平的重要方面。随着现代社会传感器精度和种类的不断增多,机器人可以获得更丰富的环境信息;同时面对当前人类已经获得的环

境测绘信息,将他们应用于路径规划中势在必行。然而,当前的路径规划算法的主要关注点还停留在如何规避障碍,如何节约能量以获得最短路径方面,而对于丰富的先验地形信息并没有加以利用。此外,海底机器人运行的实际海洋环境并不是障碍区域、可通行区域非此即彼的,而是需要在最短路径、节省能量、行驶安全性等方面加以权衡。为了解决这个问题,本文提出了扩展A\*、TC FM和TC FM\*算法,通过3个环境下的仿真实验得到如下结论:

1) TC FM、TC FM\*算法由于采用Eikonal方程建立势场与梯度下降相结合的方式路径规划,求得规划路径长度较短,同时路径较平滑,平均高度、坡度变化较小。然而,面对大尺度、栅格地图精度不高的情况,局部极端地形规划能力较差。

2) 扩展A\*算法相较于TC FM、TC FM\*算法,在低精度栅格地图情况下,规避陡峭山峰和沟壑能力更强,且其基于图搜索原理求解速度更快,能在更短时间内得到规划路径。但是,存在规划路径较长、轨迹不够平滑的问题。

本文基于海底数字高程地图,主要针对海底机器人的路径规划进行了研究。以往算法一般为解决避障条件下求最短路径问题,本文试图将地形环境信息加入路径规划算法中,综合考虑各类因素,求得更适合海底机器人运行的路径。相似的思路,可以应用到月球车、火星车的路径规划中。基于数字高程地图,本文主要提取了3类参数,用于衡量当前环境。未来可以考虑如何提取更能表现当前环境的信息,同时将更多的传感器测量数据加入进来,使得路径规划效果更好。下一步的研究重点是将扩展A\*算法与TC FM、TC FM\*算法相结合,在大尺度海底环境、栅格地图精度受限的条件下,求得距离较短、高度和坡度变化较小、轨迹平滑、能够规避局部极端地形的适合海底机器人运动的路径。

### 参考文献(References)

- [1] Mac T T, Copot C, Tran D T, et al. Heuristic approaches in robot path planning: A survey[J]. Robotics and Autonomous Systems, 2016, 86: 13-28.
- [2] González D, Pérez J, Milanés V, et al. A review of motion planning techniques for automated vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 1135-1145.
- [3] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968,

- 4(2): 100-107.
- [4] Martial H Hebert, Charles Thorpe, Anthony Stentz. Intelligent unmanned ground vehicles[M]. Boston: Springer, 1997: 203-220.
- [5] Koenig S, Likhachev M. D\* lite[J]. *Aaai/iaai*, 2002, 15: 476-483.
- [6] Koenig S, Likhachev M, Furcy D. Lifelong planning A[J]. *Artificial Intelligence*, 2004, 155(1/2): 93-146.
- [7] Likhachev M, Ferguson D I, Gordon G J, et al. Anytime dynamic A\*: An anytime, replanning algorithm[C]. *International Conference on Automated Planning and Scheduling (ICAPS)*. Monterey, 2005, 5: 262-271.
- [8] Ferguson D, Stentz A. Using interpolation to improve path planning: The field D\* algorithm[J]. *Journal of Field Robotics*, 2006, 23(2): 79-101.
- [9] Sethian J A. Fast marching methods[J]. *SIAM Review*, 1999, 41(2): 199-235.
- [10] González V, Monje C A, Moreno L, et al. Fast marching square method for UAVs mission planning with consideration of dubins model constraints[J]. *IFAC-PapersOnLine*, 2016, 49(17): 164-169.
- [11] González V, Monje C A, Moreno L, et al. UAVs mission planning with flight level constraint using fast marching square method[J]. *Robotics and Autonomous Systems*, 2017, 94: 162-171.
- [12] Li S D, Zhou H H, Hu J, et al. A fast path planning approach for unmanned aerial vehicles[J]. *Concurrency and Computation: Practice and Experience*, 2015, 27(13): 3446-3460.
- [13] Petres C, Pailhas Y, Patron P, et al. Path planning for autonomous underwater vehicles[J]. *IEEE Transactions on Robotics*, 2007, 23(2): 331-341.
- [14] Chen P F, Huang Y M, Papadimitriou E, et al. Global path planning for autonomous ship: A hybrid approach of Fast Marching Square and velocity obstacles methods[J]. *Ocean Engineering*, 2020, 214: 107793.
- [15] Shi S L, Chen J K, Xiong Y L. A fast marching gradient sampling strategy for motion planning using an informed certificate set[C]. *2020 IEEE International Conference on Robotics and Automation*. Paris, 2020: 1163-1168.
- [16] Xu J, Song K C, Zhang D F, et al. Informed anytime fast marching tree for asymptotically optimal motion planning[J]. *IEEE Transactions on Industrial Electronics*, 2021, 68(6): 5068-5077.
- [17] Gonzalez V, Monje C A, Garrido S, et al. Coverage mission for UAVs using differential evolution and fast marching square methods[J]. *IEEE Aerospace and Electronic Systems Magazine*, 2020, 35(2): 18-29.
- [18] Xu J, Song K C, Dong H W, et al. A batch informed sampling-based algorithm for fast anytime asymptotically-optimal motion planning in cluttered environments[J]. *Expert Systems with Applications*, 2020, 144: 113124.
- [19] GEBCO Compilation Group. GEBCO[Z]. 2020 grid. DOI:10.5285/a29c5465-b138-234d-e053-6c86abc040b9.
- [20] Valero-Gomez A, Gomez J V, Garrido S, et al. The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories[J]. *IEEE Robotics & Automation Magazine*, 2013, 20(4): 111-120.
- [21] Garrido S, Moreno L, Martín F, et al. Fast marching subjected to a vector field-path planning method for Mars rovers[J]. *Expert Systems with Applications*, 2017, 78: 334-346.
- [22] Castejón C, Boada B L, Blanco D, et al. Traversable region modeling for outdoor navigation[J]. *Journal of Intelligent and Robotic Systems*, 2005, 43(2/3/4): 175-216.

### 作者简介

高存璋(1996—),男,硕士生,从事机器人运动规划、智能控制的研究, E-mail: cunzhang1203@163.com;

谷海涛(1981—),男,研究员,博士,从事海洋机器人多学科优化设计方法、USV与AUV协作共融、自主回收等研究, E-mail: ght@sia.cn.

(责任编辑:李君玲)