

基于 RRT* 算法和 DWA 算法的分层结构路径规划策略

姜涛, 周兴阁, 陈宇

(同济大学机械与能源工程学院, 上海 201804)

摘要: 为优化复杂动态环境下路径规划问题, 提高路径搜索效率, 缩短路径规划时间, 提出一种基于改进 RRT* 算法和 DWA 算法的分层结构路径规划策略; 该路径规划分层结构由全局路径规划层和局部路径规划层组成, 全局路径规划层根据已知静态障碍物信息利用改进 RRT* 算法进行全局路径规划; 当路径中出现未知动态障碍物时, 根据规划策略将启用局部路径规划层, 基于获取的动态障碍物信息, 利用 DWA 算法进行局部路径规划; 最终通过全局与局部路径规划策略相结合得到最终的可行路径; 仿真实验结果表明: 文章提出的分层结构路径规划策略在复杂动态环境中可以高效地完成路径规划任务。

关键词: 路径规划; 分层结构; 改进 RRT* 算法; 动态窗口法; 概率偏移; TEB 算法

Hierarchical Structure Route Planning Strategy Based on RRT* Algorithm and DWA Algorithm

JIANG Tao, ZHOU Xingge, CHEN Yu

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

Abstract: To optimize route planning in complex dynamic environments, and improve the efficiency and shorten the route planning time, a hierarchical structure route planning strategy based on improved rapidly-exploring random tree (RRT*) algorithm and dynamic window approach (DWA) algorithm is proposed. The hierarchical planning strategy is composed of two layers: the global route planning layer and local route planning layer. The global layer adopts the improved RRT* algorithm to implement the global route planning by the the information on the static obstacles. Simultaneously, the planning strategy will be activated when unknown obstacles appear in the route. Based on the acquired information on the dynamic obstacles, the DWA algorithm is used to acquire the local route planning. By integrated the global and local route planning strategies, the final feasible route is obtained. Simulation results show this strategy can efficiently achieve route planning in complex dynamic environments.

Keywords: route planning; hierarchical structure; improved RRT* algorithm; DWA algorithm; probability shift; time elastic band (TEB) algorithm

0 引言

路径规划在机器人自主寻路避障、无人艇海面搜寻目标、自动驾驶汽车等多个领域有着广泛的应用, 是智能化无人装备实现自主航行的关键技术之一, 其算法主要可以分为全局路径规划和局部路径规划两大类^[1]。

经典的全局路径规划算法主要可以分为三类, 即启发式搜索算法、随机采样算法和群体智能算法。启发式搜索算法主要有 Dijkstra^[2], A*^[3], D*^[4] 算法等, 其通过合理的启发函数和评估策略, 能够提高搜索效率、适应不同环境、解决复杂问题, 并具有良好的鲁棒性。随机采样算法主要有蒙特卡洛采样^[5], 概率路线图 (PRM, probabilistic road map)^[6], 快速随机扩展树 (RRT, rapidly-exploring random tree) 等。其中, RRT^[7] 是最为典型的一种算法。由于该类算法采用随机采样的策略, 因而无需将地图栅格

化且兼具运算速度快的优点, 但采样的随机性也使该算法难以得到较优且光滑的路径。针对 RRT 算法存在的问题, 现有很多研究在此算法的基础上做了改进, 如文献 [8] 设计了一种基于双采样域的 RRT* 算法, 该算法考虑了约束条件、动态障碍物和外部干扰等问题, 结合模型预测控制 (MPC, model predictive control) 方法建立了分层式路径规划策略, 实现了约束条件下的路径规划; 文献 [9] 在 RRT 算法的基础上引入了自适应步长因子和目标引力, 提高了 RRT 算法的计算速度和寻优能力; 文献 [10] 采用基于 RRT 算法的渐近区域采样方法, 融合偏向目标采样机制防止随机采样点的反向搜索, 并用动态圆规则化使规划路径有效避免碰撞和通过狭窄区域。群体智能算法是从自然界的群体发展规律中得到启发, 通过模仿自然界中群体行为得到的一类算法。典型的群体智能算法包括蚁群算法^[11]、

收稿日期: 2023-08-27; 修回日期: 2023-10-08。

作者简介: 姜涛 (1969-), 男, 博士, 副教授。

引用格式: 姜涛, 周兴阁, 陈宇. 基于 RRT* 算法和 DWA 算法的分层结构路径规划策略[J]. 计算机测量与控制, 2024, 32(9): 241-248.

粒子群算法^[12]、遗传算法^[13]和狼群算法^[14]等。该类算法通常具有适应性强、鲁棒性好的特点,但对参数的设置相对比较敏感,不合理的参数设计会使算法得到的路径陷入局部最优。针对这一问题,如今的群体智能算法大多在此基础上进行了一些改进,如采用自适应参数、自适应步长等方法,以避免算法过早地收敛以陷入局部最优解的情况发生。

局部路径规划通常使用基于运动学的方法,如速度控制、力控制、轨迹规划等,在保证安全性的同时,实现最优的局部路径规划。现有的局部路径规划算法主要有人工势场法(APF, artificial potential field)^[15]、动态窗口法(DWA, dynamic window approach)^[16]等。

全局路径规划通可以计算出一条全局最优路径,从而保证路径能够避开已知障碍物,顺利抵达目标点,但并不应对实时变化的环境。局部路径规划可以处理动态障碍物和未知障碍物的问题,保证路径的实时性,但由于得到的信息有限,容易陷入局部最优解,难以得到全局最优路径。全局路径规划和局部路径规划各有其优劣,因此本文提出一种基于RRT*算法和DWA算法的分层结构路径规划策略,实现复杂环境中高效、安全、合理的路径规划。

1 路径规划算法

1.1 环境建模

路径规划主要包括环境建模和引入算法并生成路径两个过程,环境模型构建作为研究路径规划的首要任务,通常涉及到对环境特征信息的提取,包括空间物体的位置及形状信息,构建合理的环境模型有助于避开障碍物、降低路径规划的复杂度,提高规划路径质量的同时降低时空开销。目前常用的环境建模方法有栅格法^[17]、链接图法^[18-19]、可视图法^[20]等。

采用栅格法对地图环境进行建模。在环境建模中使用的栅格地图主要用于对地图进行单元划分,用大小相等的矩形表示。

如图 1 所示为一典型的栅格地图,其中自由栅格为白色,表示可通行区域,障碍栅格为黑色,表示不可通行区域。栅格地图简单且易于实现,并且不用经过复杂的数学计算就可处理障碍物边界,因此被广泛用于环境建模中。

由于地图上的已知障碍物信息与实际障碍物可能存在一定的偏差,直接在栅格图中进行全局路径规划,难以保证避障避碰的安全性。为保证路径能够安全的与障碍物避让,需要与障碍物间留有一定的安全距离。因而在路径规划前需要对地图上的障碍物边界进行膨胀处理,即向外扩充地图上障碍物边界,以保证规划的路径与真实障碍物间存在一定间隔。

定义一个结构元素 $B(x, y)$, 如 1 个 3×3 的矩形, $A(x, y)$ 为待处理的矩阵, 则用 $B(x, y)$ 对 $A(x, y)$ 执行膨胀的过程定义为:

$$(A \oplus B)(x, y) = \max\{A(x+x', y+y') \mid (x', y') \in D_B\} \quad (1)$$

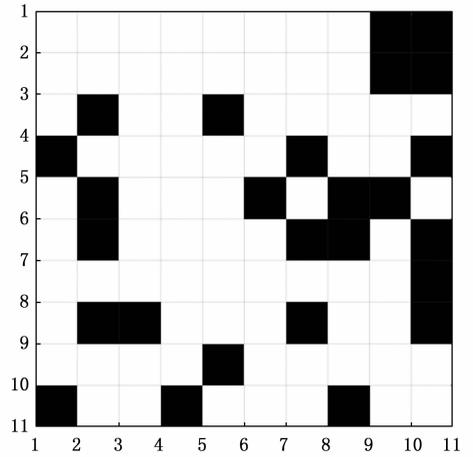


图 1 栅格地图示例

式中, D_B 为结构元素 B 的邻域, 使用结构元素 B 对矩阵 A 进行膨胀相当于矩阵 A 中的每个元素在其结构元素 B 的邻域内取最大值。

1.2 RRT* 算法介绍

RRT* 算法是一种用于求解无人艇路径规划问题的算法, 其原理示意图如图 2 所示。该算法的核心思想是构建一棵无向树, 其中树的节点代表状态, 边代表在不同状态之间的移动, 通过随机采样和局部优化策略来不断扩展树的节点, 从而找到一条从起始状态到目标状态的合理路径。

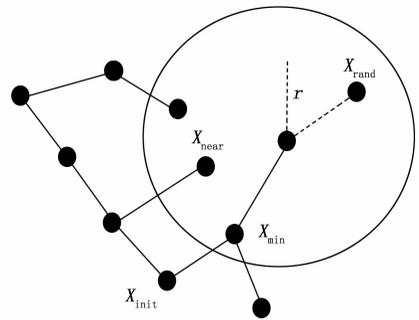


图 2 RRT* 算法原理示意图

RRT* 算法的具体流程如下:

- 1) 定义起始状态和目标状态, 并将其作为树的根节点。
- 2) 随机生成一个范围在 $(0, 1)$ 之内的数作为策略概率 P , 并按策略概率 P 从状态空间中随机获取一个采样状态。
- 3) 在树上寻找距离采样最近的节点, 从该节点向新的采样状态前进步长 S 后得到一个新的节点, 连接两个节点形成一条新边, 判断该边产生的路径是否与障碍物发生碰撞, 如发生碰撞则返回 2)。
- 4) 在树中对新加入的节点进行优化, 将其与其附近距离为 r 以内的父节点进行比较, 选择代价最小的父节点, 重新连接父节点和子节点。搜索半径 r 的计算公式如下:

$$r = \min(\gamma(\frac{\log(n)}{n})^{1/d}, \eta) \quad (2)$$

式中, γ 为搜索半径 r 的半径常值系数, n 为目前的节点数, d 为状态空间维度, η 为节点间最大连接距离值。

5) 判断是否到达目标状态, 如果新加入的节点的状态距离目标状态的距离满足收敛条件, 则算法终止, 否则返回至 2)。

6) 从树的根节点开始, 按照父子关系从终止节点向根节点遍历, 得到从起始状态到目标状态的可行路径。

2 局部路径规划

全局路径规划通常计算速度快、效率高, 但难以应对动态障碍物和未知障碍物等突发状况, 而局部路径规划可以处理动态障碍物和未知障碍物的问题。在环境信息有限的前提下, 局部路径规划可以根据收集到的局部信息寻找合适的路径并避开障碍物。

动态窗口算法 (DWA, dynamic window approach) 依据机器人的运动学模型进行推导, 根据最大推力和最大转动力矩得出其所能达到的最大加速度以及角加速度, 建立物理约束, 由此预测下一采样时刻机器人所能达到的速度以及角速度, 进而得到采样时间间隔内以当前速度为中心的有效速度矢量空间。根据速度矢量空间中的有效速度模拟下一采样时间间隔内机器人的运动轨迹, 通过轨迹评价函数对所得到的多组模拟轨迹进行评价筛选, 选取其中得分最高的一条运动轨迹作为最终轨迹。

DWA 算法的物理约束主要包括速度和角速度约束、加速度约束以及碰撞检测, 如下所示^[21]。

1) 速度与角速度约束: 根据机器人物理性能和环境限制, 机器人的速度存在最大值和最小值约束。此时采样速度空间为:

$$V_m = \{v \in [v_{\min}, v_{\max}], [\omega_{\min}, \omega_{\max}]\} \quad (3)$$

2) 加速度约束: 受到驱动能力和环境因素限制, 机器人的加速度存在最大值和最小值约束。因此, 考虑加速度时采样速度空间为:

$$V_d = \{(v, \omega), v \in [v_c - v_{\max} \cdot \Delta t, v_c + v_{\max} \cdot \Delta t], \omega \in [\omega_c - \omega_{\max} \cdot \Delta t, \omega_c + \omega_{\max} \cdot \Delta t]\} \quad (4)$$

式中, v_{\max} 为最大线加速度, ω_{\max} 为最大角加速度。

3) 碰撞检测: 实现实时避障功能需要考虑周围障碍物因素, 避免碰撞到障碍物的采样速度约束条件为:

$$V_a = \{(v, \omega), v \leq \sqrt{2 \cdot D_{\text{dist}(v, \omega)} \cdot v_{\max}}, \omega \leq \sqrt{2 \cdot D_{\text{dist}(v, \omega)} \cdot \omega_{\max}}\} \quad (5)$$

式中, $D_{\text{dist}(v, \omega)}$ 为当前模拟轨迹中机器人与障碍物之间的距离。

如图 3 所示, 在一个采样时间间隔内, 动态窗口算法生成了多组速度组合, 相应地将会生成多条预测轨迹。为了保证路径能够安全、合理且高效地避开探测到的动态障碍物, 需要从这些预测轨迹中根据“评价函数”筛选出最

合适的一条预测轨迹。

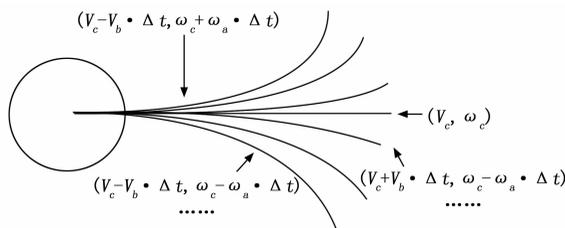


图 3 DWA 算法原理图

评价函数由方位角、距离以及速度评价函数三部分加权构成, 其具体的计算公式如下所示:

$$G(v, \omega) = \sigma[\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega)] \quad (6)$$

式中, α, β, γ 为权重, $\alpha, \beta, \gamma > 0$, $\text{heading}(v, \omega)$, $\text{dist}(v, \omega)$, $\text{velocity}(v, \omega)$ 分别为方位角评价函数、距离评价函数和速度评价函数, σ 为归一化函数。

1) 方位角评价函数:

根据当前选定的采样速度组合, 可以按照预测轨迹终点处的航向角, 定义方位角偏差 θ 为预测轨迹终点处的航向角与预测轨迹终点和目标点之间的连线与 x 轴夹角之间的角度差。若角度差越大, 则偏航越大; 若角度差为零, 则表示该预测轨迹方位角误差为零, 因此采用 $\pi - \theta$ 对方位角进行评价, 评价价值越高, 则方位角偏差 θ 越小, 相应的预测轨迹航向越准确。

2) 距离评价函数:

距离评价函数定义为预测轨迹终点处位置与最近障碍物之间距离。若轨迹附近范围内没有障碍物, 则设置为一个定值, 以确保该评价函数不会过大; 同时, 将预测轨迹中碰撞到或将会碰撞到障碍物的轨迹去除, 以确保机器人不会与障碍物相撞。按此评价函数得到的分数高的轨迹距离障碍物越远, 因而其保障了机器人的安全行驶。

3) 速度评价函数:

速度评价函数定义为运动速度的绝对值。当存在多组安全速度组合时, 速度评价函数会根据速度的大小给出预测轨迹评价价值, 速度越大, 评价价值越大, 这样选出的速度组合可以使得机器人快速地到达目标点, 从而使算法更加高效。

4) 归一化过程:

以上 3 种评价函数具有不同的量纲和量纲单位, 这样的情况会影响到数据分析的结果。为了消除指标之间的量纲影响, 需要进行数据标准化处理, 以解决数据指标之间的可比性。原始数据经过数据标准化处理后, 各指标处于同一数量级, 适合进行综合对比评价。因而以上 3 种评价函数在最终评价函数中并非简单地加权叠加, 而是在进行归一化处理后再进行叠加。具体的归一化过程如下:

$$\begin{cases} \text{normal_heading}(i) = \frac{\text{heading}(i)}{\sum \text{heading}(i)} \\ \text{normal_dist}(i) = \frac{\text{dist}(i)}{\sum \text{dist}(i)} \\ \text{normal_velocity}(i) = \frac{\text{velocity}(i)}{\sum \text{velocity}(i)} \end{cases} \quad (6)$$

3 算法改进

3.1 改进 RRT* 算法

传统 RRT 算法在迭代新节点时采样的随机性,使得算法寻路过程会盲目搜索整个状态空间,寻路过程存在盲目性,寻出的路径往往不是最优路径且极不平滑,不平滑的路径不易跟踪控制。相较于传统的 RRT 算法, RRT* 算法在树的构建过程中引入了路径优化策略,使得树的结构更加合理,能够找到更优的路径。

3.1.1 轨迹光滑与优化

RRT* 算法规划的最终路径虽然可以到达目标点,但其路径曲折不光滑,不利于进行跟踪控制。为了解决上述的问题,采用 TEB (Time elastic band) 算法对 RRT* 算法规划得到的全局路径进行规划。

TEB 算法通过构建一系列带有时间参数的离散位姿形成的路径并将其作为候选轨迹。这些轨迹形成了一条弹性带,连接了路径的起始点和目标点,这条弹性带将在各种约束的作用下发生变形。TEB 算法将对这些离散点进行不断地采样和优化,优化的目标函数将综合考虑运动学约束条件,曲线平滑度,远离障碍物等因素的影响,通过加权的形式形成软约束。在算法的不断迭代下,最终由这些离散位姿组成的轨迹能实现时间短、距离短、安全避障等目标,同时能使优化后的轨迹满足运动学约束条件。

与 A*PS 加多次样条曲线插值的传统算法相比, TEB 算法可以有效避免路径节点过少时,样条曲线或多项式插值曲线难以准确控制的问题。

3.1.2 引入概率偏移机制

改进 RRT* 算法引入概率偏移机制,即每次采样时均有一定的概率直接选取目标点作为采样点。通过概率偏移机制的设立,可以有效减少全局路径规划算法中无效采样点的个数,加快算法的收敛速度。

概率偏移机制首先预设偏移概率值 P_c ,在对状态空间进行采样前随机生成一个范围在 $(0,1)$ 之内的数,即为概率值 P 。如果 P 大于概率偏移值 P_c ,则从状态空间中随机获取一个采样状态,否则选取目标状态作为采样状态。图 4 为引入概率偏移机制的 RRT* 算法流程图。

3.2 分层路径规划

针对全局路径规划和局部路径规划算法各自的优缺点,提出一种分层式的路径规划策略。第一层为全局规划层,基于改进的 RRT* 算法实现全局路径规划;第二层为局部

规划层,基于 DWA 算法实现在线局部路径规划。基于分层结构的路径规划策略的框架如图 5 所示。

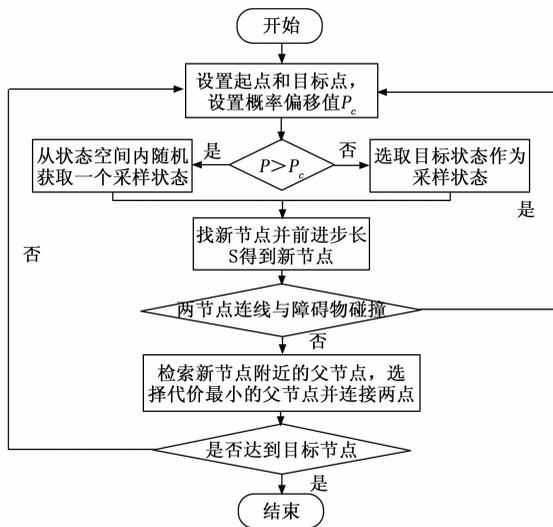


图 4 引入概率偏移机制的 RRT* 算法流程图

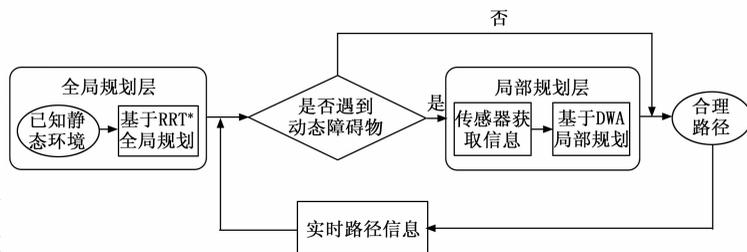


图 5 基于分层结构的无人艇路径规划策略流程图

算法实现的具体步骤如下:

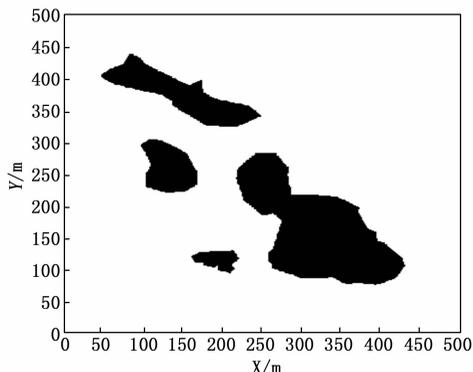
- 1) 加载地图环境,获取已知的起点和终点状态,启动全局规划层,使用基于 RRT* 的全局规划算法规划出一条较优的全局路径,避开静态障碍物。
- 2) 检测在全局路径上是否出现动态障碍物,若有动态障碍物,则启动局部规划层,获取传感器采集的数据信息并使用基于 DWA 算法的局部规划方法进行局部路径规划,避开障碍物。
- 3) 在局部路径规划结束后,重新启动全局规划层,生成一条新的全局路径。
- 4) 到达目标点后循环结束,得到最终路径。

4 仿真与分析

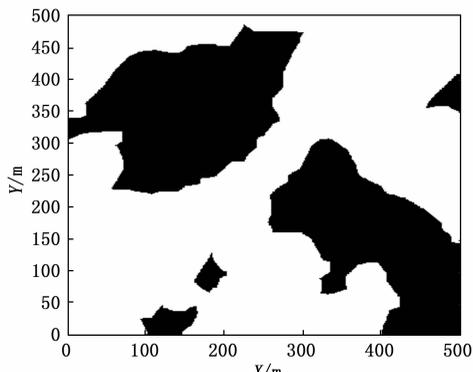
为了验证基于 RRT* 的全局路径规划算法的效果,采用了如图 6 所示的两张地图进行了测试。在本章中的所有地图中,黑色栅格均为障碍物,白色栅格均为可通行区域。两张测试地图来源于实际环境,其地图大小均为 $500 \text{ m} \times 500 \text{ m}$,地图的分辨率为 1 m 。

1) 测试地图 1 仿真测试结果:

首先对测试地图 1 进行膨胀处理,所选的膨胀核为 10×10 大小的矩形结构元素。然后在膨胀地图上进行全局路



(a) 测试地图 1

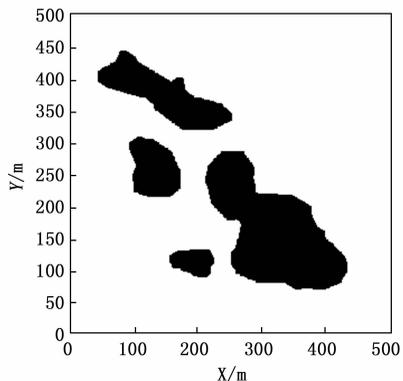


(b) 测试地图 2

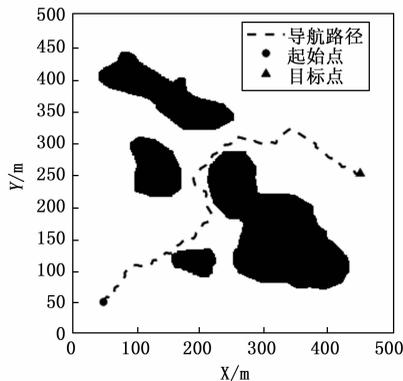
图 6 全局路径规划测试地图

径规划, 目标点的概率偏移值为 0.1, 随机生成树的最大步长为 5, 截止条件为 $d \leq 1$, 迭代步数为 10 000 步。由于 RRT* 算法具有随机性, 因此对每一个实验条件下的仿真进行三次规划, 并将规划后的曲线进行优化处理, 最终在未经膨胀的原地图上展示。

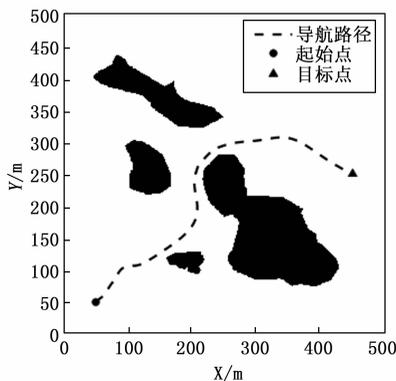
RRT* 算法的规划路径与优化处理后的最终路径效果如图 7 所示, 6 组实验测试相应路径的数据结果如表 1 所示。其中, 实验 1 到实验 3 中的起始点为 (50, 50), 目标点为 (450, 250), 实验 4 到实验 6 中的起始点为 (450, 50), 目标点为 (30, 450)。从图 7 可以看出, RRT* 算法可以在膨胀后的地图上正确地规划一条安全有效的全局路径, 而在



测试地图 1 膨胀后的结果



(a) RRT* 规划路径



(b) 优化后路径

图 7 测试地图 1 实验 1 路径结果

经过 TEB 算法优化后原始路径中的转折明显减少, 且不存在明显的尖角, 符合实际运动学约束和轨迹需求。

从表 1 中可以看出, 优化后的路径长度明显地小于原始路径的长度, 在实验 1 到实验 6 中, 优化后的路径长度相较于原始路径的长度而言, 分别减小 14.44%, 14.67%, 14.28%, 14.49%, 14.77% 和 14.69%。

表 1 测试地图 1 实验数据

实验	原始路径节点数	原始路径长度/m	优化后路径长度/m
1	137	676.276 3	578.602 7
2	129	637.850 2	544.231 2
3	130	641.461 3	549.873 1
4	160	791.962 8	677.178 6
5	155	769.223 2	655.581 0
6	158	784.463 5	669.247 6

2) 测试地图 2 仿真测试结果:

与测试地图 1 类似, 我们先对测试地图 2 进行同样的膨胀处理。然后在膨胀地图上进行全局路径规划, 所用的 RRT* 算法的参数与测试地图 1 中相同。仿真结果如图 8 所示, 六组实验测试相应路径的数据结果如表 2 所示。其中, 实验 1 到实验 3 中的起始点为 (50, 50), 目标点为 (450, 250), 实验 4 到实验 6 中的起始点为 (390, 70), 目标点为 (30, 250)。

表 2 测试地图 2 实验数据

实验	原始路径节点数	原始路径长度/m	优化后路径长度/m
1	143	707.144 8	602.136 6
2	129	640.000 0	552.301 6
3	139	688.192 6	587.527 2
4	111	548.243 9	468.964 4
5	101	498.437 7	446.210 0
6	107	527.612 6	466.409 0

从表 2 中可以看出, 在测试地图 2 中, 优化后的路径长度明显地小于原始路径的长度, 在实验 1 到实验 6 中, 优化后的路径长度相较于原始路径的长度而言, 分别减小 14.44%, 14.67%, 14.28%, 14.49%, 14.77%, 14.69%。

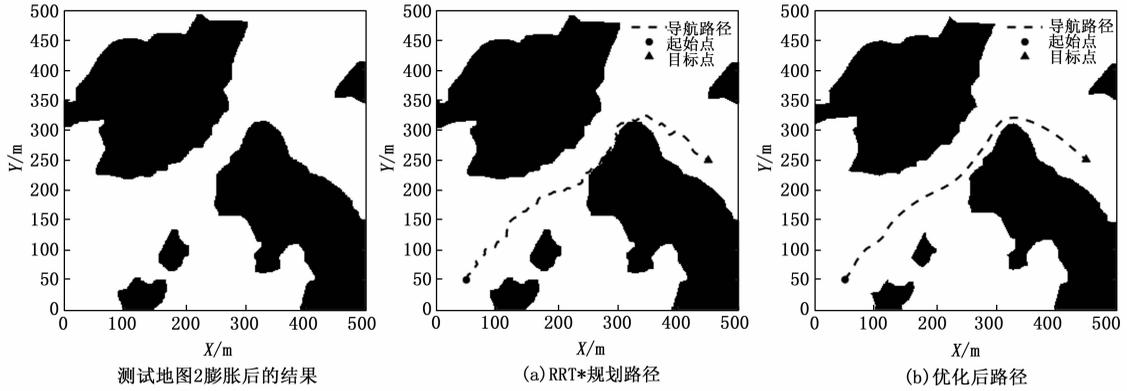


图 8 测试地图 2 实验 1 路径结果

这说明了所设计的路径优化算法对于不同的环境都可以提升算法性能，缩短路径长度。

综上所述，所设计的全局路径规划算法可以获得一条合理、高效、安全的全局轨迹。

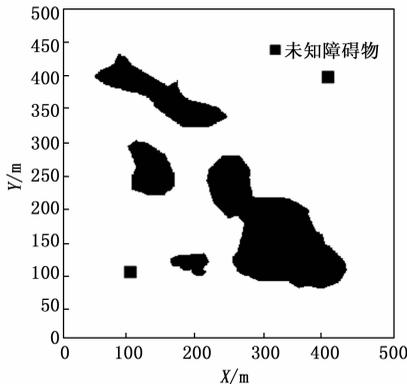
为了验证提出的分层式结构路径规划策略的有效性，在图 6 的两张地图上分别加入了未知静态障碍物和未知动态障碍物，形成了如图 9 所示的地图。分别对两种环境进行路径规划的仿真。

长为 10 m。在该地图上采用基于分层式结构的路径规划策略进行仿真实验，仿真所用的参数和运动学约束如表 3 所示，初始点和目标点分别为 (50, 50) 和 (450, 450)，初始速度均为 0，仿真结果如图 10 所示。

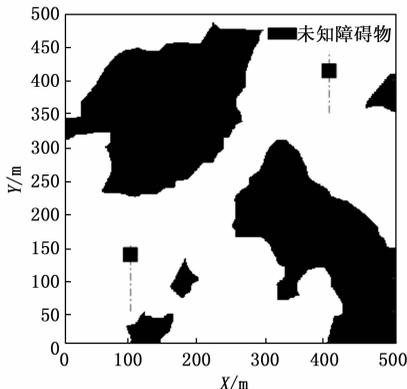
如图 10 所示，在初始时刻 $t = 0$ s 时，首先执行全局规划层的策略，将原有地图膨胀，并在膨胀后的地图上采用 RRT* 算法和 TEB 算法进行全局路径规划。在未检测到未知障碍物时，保持期望路径，当 $t = 36$ s 时检测到未知障碍物，启用局部规划层，采用 DWA 算法进行局部路径规划，实现动态避障功能。当 $t = 63$ s 时，成功避开未知障碍物，重新启动全局规划层进行全局路径规划，如图 10 (c) 所示。如此反复，最终可以根据障碍物的类型选择不同的规划算法进行路径规划，降低了算法的复杂度，且实现了对已知障碍物和未知障碍物的高效避障。

表 3 分层式结构路径规划策略测试中的参数设置与动力学约束

名称	值	名称	值
目标点概率偏移	0.1	RRT* 最大循环次数	10 000
RRT* 最大步长	5 m	膨胀核	10×10 的矩形
最大速度	2 m/s	最大加速度	0.5 m/s ²
最大角速度	$\pi/6$ rad/s	最大角加速度	$\pi/3$ rad/s ²
速度分辨率	0.01 m/s	角速度分辨率	$\pi/180$ rad/s
时步	1 s	预测时间	5 s
$[\alpha, \beta, \gamma]$	[0.45, 1, 1]	传感器探测范围	20 m



(a) 分层式结构路径规划策略测试地图 1



(b) 分层式结构路径规划策略测试地图 2

图 9 分层式结构路径规划策略测试地图

在如图 9 (a) 所示的地图中，有两个未知的方形静态障碍物，其位置位于 (100, 100) 和 (400, 400) 处，边

在如图 9 (b) 所示的地图中，有两个移动的未知方形障碍物，其边长为 10 m，且沿黑色虚线进行往复运动。障碍物运动的速度为 1 m/s，障碍物中心的初始位置位于 (100, 150) 和 (400, 400) 处，其 x 坐标保持不变， y 坐标的运动的范围为 [100, 150] 和 [350, 450]。在该地图上采用基于分层式结构的路径规划策略进行仿真实验，仿真所用的参数和运动学约束与测试地图 1 中相同，初始点和目标点分别为 (50, 50) 和 (450, 450)，初始速度均为 0，仿真的结果如图 11 所示。

从图 11 中可以看出，规划的路径可以有效地避开动态的未知障碍物。结合测试地图 1 的结果可知，所提出的基

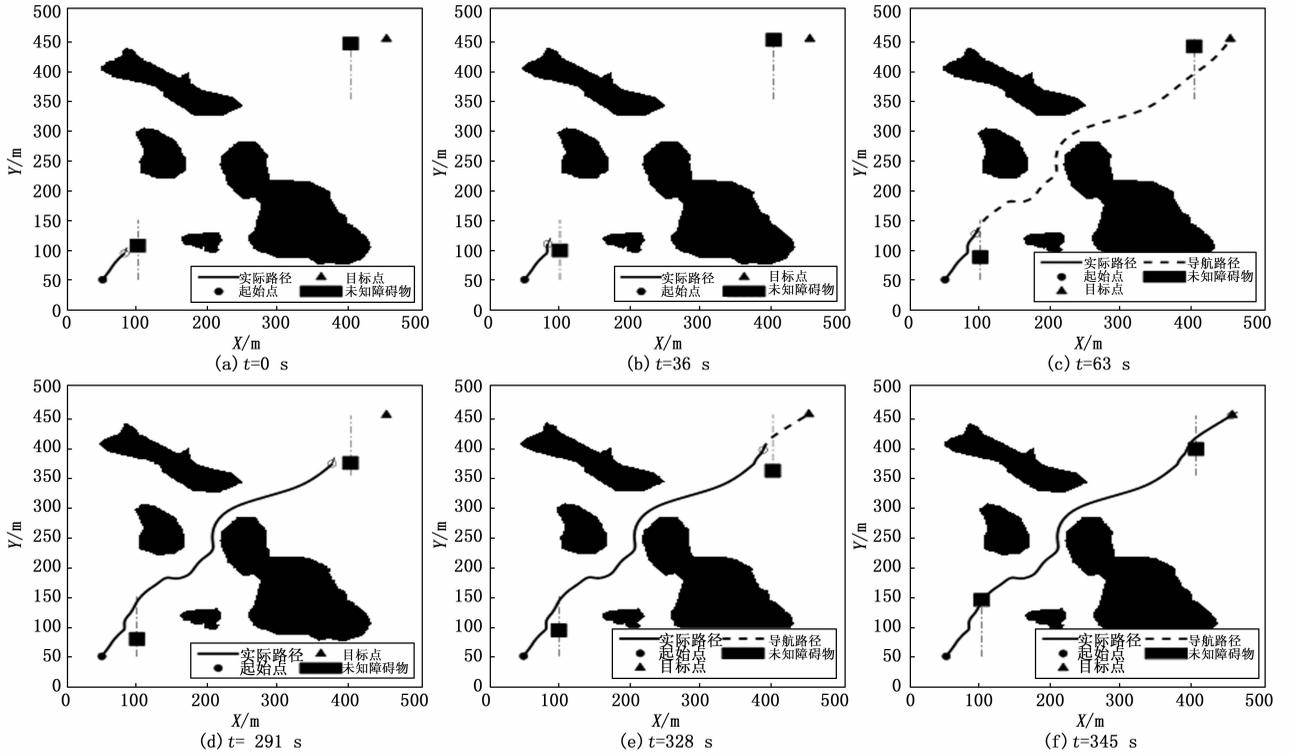


图10 分层式结构路径规划策略测试地图1仿真结果

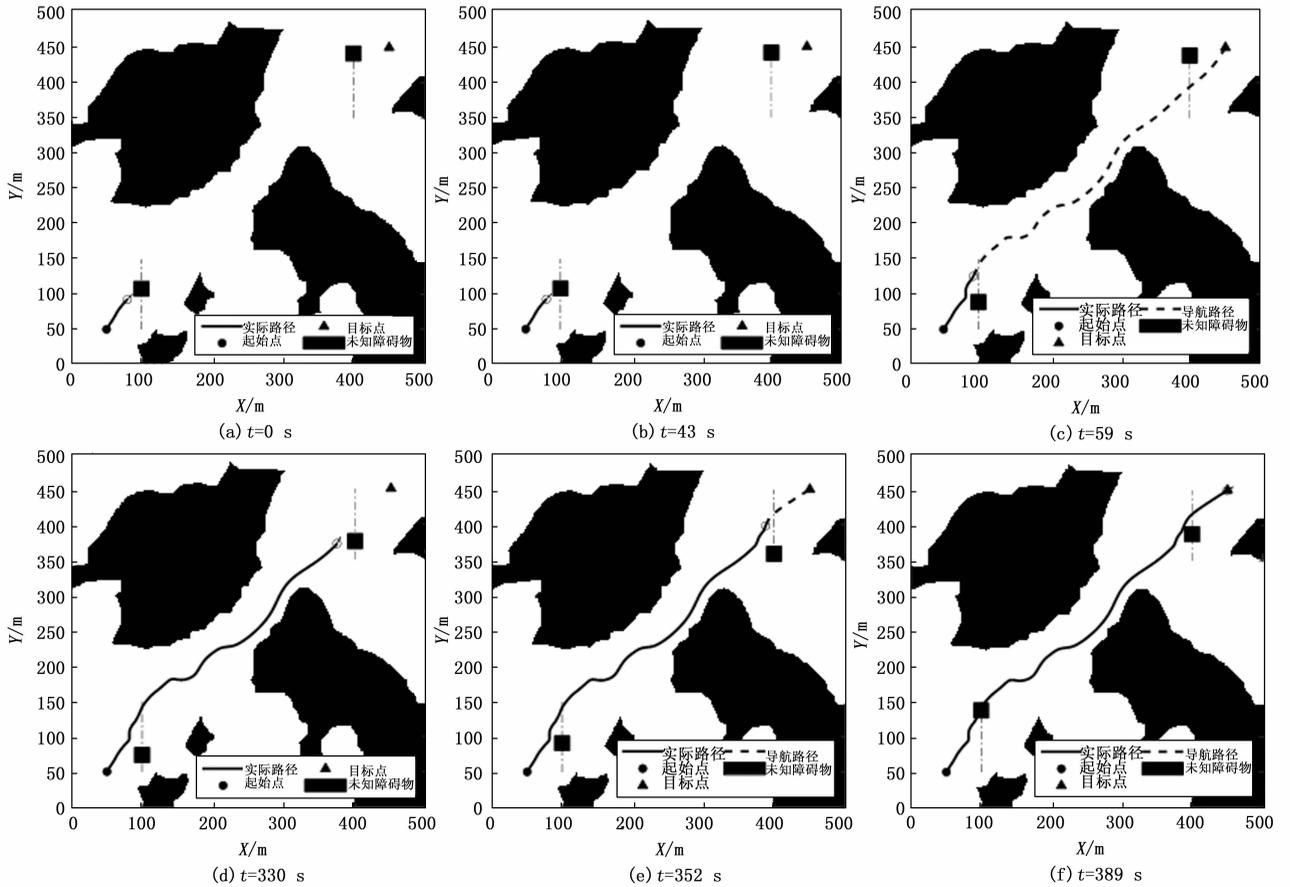


图11 分层式结构路径规划策略测试地图2仿真结果

于分层式结构的路径规划策略可以找到一条合理的全局路径，同时实现对未知动态障碍物的高效避障。

5 结束语

本文针对基于 RRT* 算法的全局路径规划和基于 DWA 算法的局部路径规划过程存在的不足之处，对 RRT* 算法进行了改进，并进一步对两算法进行融合改进研究。引入概率偏移机制，有效减少全局路径规划算法中无效采样点的个数，加快算法的收敛速度；采用 TEB (Time elastic band) 算法对 RRT* 算法规划得到的全局路径进行规划，综合考虑运动学约束条件，曲线平滑度，远离障碍物等因素的影响，通过加权的形式形成软约束，保证优化后的轨迹满足运动学约束条件，同时剔除冗余节点，平滑曲线；建立分层结构路径规划策略，实现对静态障碍物和动态障碍物的高效避障。最后，通过仿真实验对所提出的方法进行分析评价，结果表明，改进的 RRT* 算法相比于传统 RRT* 算法，在进行全局路径规划时，路径平滑度提高，路径长度缩短 13% 以上，性能有效提升，并且所提出的分层结构轨迹规划策略有较好的全局路径规划能力，对动态障碍物可以有效规避。

参考文献:

[1] 赵亮, 王芳, 白勇. 水面无人艇路径规划的现状与挑战 [J]. 船舶工程, 2022, 44 (4): 1-7.

[2] DIJKSTRA E W. A note on two problems in connexion with graphs [M]. Edsger Wybe Dijkstra: His Life, Work, and Legacy, 2022.

[3] HART P E, NILSSON N J, RAPHAEL B. Correction to “a formal basis for the heuristic determination of minimum cost paths” [J]. ACM SIGART Bulletin, 1972 (37): 28-29.

[4] STENTZ A, MELLON I C. Optimal and efficient path planning for unknown and dynamic environments [J]. International Journal of Robotics and Automation, 1995, 10 (3): 89-100.

[5] 韩雨薇, 郑安迪, 朱俞竹, 等. 蒙特卡洛粒子滤波算法应用研究 [J]. 科技资讯, 2020, 18 (35): 12-13.

[6] 邱添, 张志安, 王海龙. 移动机器人的栅格概率路径图法路径规划 [J]. 机床与液压, 2022, 50 (21): 14-19.

[7] LAVALLE S M. Planning algorithms [M]. Cambridge University Press, 2006.

[8] 张振亚, 王进, 程红梅, 等. 基于余弦相似度的文本空间索引方法研究 [J]. 计算机科学, 2005 (9): 160-163.

[9] 李迎辉. Python 开发 Rails 框架——Django 框架介绍 [J]. 程序员, 2006 (11): 107-110.

[10] 肖文娟, 王加胜. 基于 Vue 和 Spring Boot 的校园记录管理 Web App 的设计与实现 [J]. 计算机应用与软件, 2020, 37 (4): 25-30.

[11] 朱兆丰, 刘静琦, 周振虎等. 基于 STM32 的室内环境监测系统设计 [J]. 物联网技术, 2021, 11 (6): 6-9.

[12] GHOSH R K, K DAS S. A survey on sensor localization

[8] WEN N, ZHANG R, WU J, et al. Online planning for relative optimal and safe paths for USVs using a dual sampling domain reduction-based RRT* method [J]. International Journal of Machine Learning and Cybernetics, 2020, 11: 2665-2687.

[9] ZHANG Z, WU D, GU J, et al. A path-planning strategy for unmanned surface vehicles based on an adaptive hybrid dynamic stepsize and target attractive force-RRT algorithm [J]. Journal of Marine Science and Engineering, 2019, 7 (5): 132.

[10] 姚相滢, 许伦辉, 林世城. 基于改进 RRT 算法的智能车辆路径规划研究 [J]. 计算机仿真, 2023, 40 (4): 165-169.

[11] DORIGO M, DI CARO G, GAMBARDELLA L M. Ant algorithms for discrete optimization [J]. Artificial Life, 1999, 5 (2): 137-172.

[12] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory [C] // The 6th International Symposium on Micro Machine and Human Science, 1995: 39-43.

[13] HOLLAND J H. Adaptation in natural and artificial systems [M]. Bradford: A Bradford Book, 1992.

[14] CHENGUANG YANG, XUYAN TU, JIE CHEN. Algorithm of marriage in honey bees optimization based on the wolf pack search [C] // International Conference on Intelligent Pervasive Computing, 2007: 462-467.

[15] KHATIB O. Real-time obstacle avoidance for manipulators and mobile robots [J]. International Journal of Robotics Research, 1986, 5 (1): 90-98.

[16] FOX D, BURGARD W. The dynamic window approach to collision avoidance [J]. IEEE Robotics and Automation Magazine, 1997, 4 (1): 23-33.

[17] 马鸣, 董佳伟, 尹勇, 等. 基于电子海图的无人艇集群区域覆盖路径规划 [J]. 中国舰船研究, 2024, 19 (1): 211-219.

[18] 李满. 移动机器人环境建模与路径规划的研究 [D]. 秦皇岛: 燕山大学, 2012.

[19] 吴杰. 海上突发事件应急救援路径规划算法 [J]. 舰船科学技术, 2020, 42 (10): 19-21.

[20] 李霜琳, 何家皓, 敖海跃, 等. 基于鸽群优化算法的火星飞行器智能可视图法 [J]. 飞行力学, 2020, 38 (5): 90-94.

[21] 严浙平, 黄俊儒, 吴迪. 基于 RRT 和 DWA 的欠驱动 UUV 路径规划 [J]. 数字海洋与水下攻防, 2020, 3 (3): 258-264.

[22] Journal of Control Theory and Applications, 2010, 8 (1): 2-11.

[23] 王群. 车联网的安全机制及关键技术研究 [D]. 南京: 南京理工大学, 2016.

[24] 曾浩. 基于 Python 的 Web 开发框架研究 [J]. 广西轻工业, 2011, 27 (8): 124-125.

[25] 王军丰. 基于 Web 的工作流管理系统的设计与实现 [D]. 西安: 西安建筑科技大学, 2003.

[26] 彭钰皓, 亓相涛. 基于 ESP8266 的天气数据分析系统设计 [J]. 电脑编程技巧与维护, 2019 (9): 75-79.

(上接第 219 页)