

DOI:10.19651/j.cnki.emt.2314642

基于 RRT* 算法的移动机器人安全光滑路径生成*

李文君 李忠伟 罗 懿

(中国石油大学(华东)海洋与空间信息学院 青岛 266400)

摘要: 在多障碍物复杂工厂环境中,针对快速探索随机树算法(RRT*)生成的路径存在冗余点、贴近障碍物且存在锯齿状转折的问题,改进得到了安全-光滑 RRT* (Safe-Smooth RRT*) 路径规划算法。首先,引入目标偏置策略;其次,该算法利用融合目标点引力思想的新节点扩展方式以及改进的近邻点度量策略以减少树的盲目扩展,提高生长的目标性;随后,引入节点安全约束,将安全节点加入树中;改进路径简化方法,剔除冗余点的同时兼顾了安全性;最后通过 B 样条局部平滑来改善路径的平滑性。在 MATLAB 仿真实验中分别与标准 RRT* 算法、自适应目标偏向性 RRT 算法和改进 RRT 算法相比,在平均路径长度方面最大下降了 7.1%,在平均有效节点数方面最大下降了 64.1%,且所得路径始终与障碍物保持一定的安全距离,结果表明改进算法有效提升了路径的光滑性和安全性。

关键词: 移动机器人;路径规划;RRT* 算法;近邻节点度量;节点安全约束;改进路径简化;局部平滑

中图分类号: TP242 **文献标识码:** A **国家标准学科分类代码:** 520.60

Safe and smooth path generation of mobile robot based on RRT* algorithm

Li Wenjun Li Zhongwei Luo Cai

(School of Marine and Spatial Information, China University of Petroleum, Qingdao 266400, China)

Abstract: In complex factory environments with multiple obstacles, in order to solve the problem that the path generated by rapidly-exploring random tree star algorithm(RRT*) has redundant points, is close to the obstacle and has jagged turns, the path planning algorithm of SSRRT* [Safe-Smooth RRT*] is improved. Firstly, a target biasing strategy is introduced. Secondly, the algorithm utilizes a new node expansion approach that combines the concept of target point attraction and an improved nearest neighbor point metric to reduce the blind expansion of the tree and accelerate growth towards the target point; Node security constraints are then imposed to add the security nodes to the tree; Improved path simplification eliminates redundant points while taking into account security; Finally, the local smoothing of the B-spline is used to improve the smoothness of the path. By comparing with the standard RRT* algorithm, the adaptive target bias RRT algorithm, and improved RRT algorithm, the maximum decrease in average path length is 7.1%, the maximum decrease in average effective node number is 64.1%, and always keep a safe distance from obstacles. The results indicate that the improved algorithm effectively improves the smoothness and safety of the path.

Keywords: mobile robots; path planning; RRT* algorithm; neighbor node metric; node safety constraints; improved path simplification; local smoothing

0 引言

移动机器人具有高操作性和灵活性,可以代替人类在复杂环境中从事各种任务,提高生产生活的效率。路径规划^[1]对自主移动机器人至关重要,路径的优劣将直接影响机器人移动的安全性和稳定性。常用的路径规划方法有: Dijkstra

算法^[2]、A* 算法^[3-4]、人工势场法^[5]、遗传算法^[6-7]、蚁群算法^[8-9]、粒子群算法^[10]、基于采样的路径规划算法^[11]以及上述算法的变体等。常见的采样算法是 RRT 算法,其无需对环境显式建模,空间搜索性能强,适用于复杂高维空间,且具有概率完备性和分辨率完整性。但其规划出的路径冗余点和无效分支较多、路径不平滑且与障碍物距离较近。

收稿日期:2023-09-22

* 基金项目:国家自然科学基金面上项目(60271491)、自主创新科研计划项目(理工科)战略专项(22CX01004A)资助

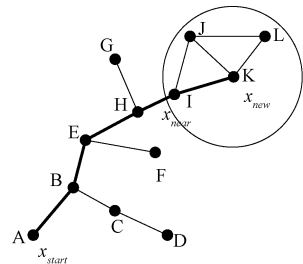
因此致力于优化规划结果的 RRT 变体算法^[12-13]被提出。王杨斌等^[14]对 Informed RRT* 施加自适应步长进行全局探索,但仍存在较多冗余点;文献^[15]采用二阶贝塞尔曲线改善了 RRT 算法路径波折的问题;张丹萌等^[16]改进 RRT-Connect 算法提高了算法收敛速度,但仍存在较多冗余点;刘奥博等^[17]将目标偏置与双向搜索结合以提高 RRT* 规划效率,但该方法未能很好地解决扩展规模的问题;文献^[18]考虑了机器人运动动力学约束,但 OBVP 问题计算复杂,消耗时间长;文献^[19]的 NRRT* 算法通过学习不同属性的最优路径概率分布以找到最优解,但需要大量成功的路径规划案例来训练 CNN 模型;马小陆等^[20]利用 JPS 搜索策略减少 RRT* 路径中的冗余节点,但仍存在折线转折。张瑞等^[21]通过添加新节点到障碍物的安全约束来剔除危险节点,但简化处理时没有考虑机器人的实际尺寸;许万等^[22]利用最小转弯半径的 B 样条曲线平滑路径,虽然路径质量较好,但该方法通过删除占用栅格少的障碍物来简化地图,没有考虑机器人的实际尺寸,无法保证最终路径的安全性;王海芳等^[23]改进 RRT* FN 算法,通过膨胀障碍物来保证路径的安全性,但该方法受限于地图精度。

上述等人对 RRT* 算法的改进,在提高了搜索树的目标导向性、降低冗余节点以及保证安全性方面进行了改进,但往往侧重单一方面的改善,未对路径的平滑性和长度以及移动机器人的实际尺寸等因素进行综合考虑。因此本文旨在改进传统 RRT* 算法,提出了 SSRRT* (Safe-Smooth RRT*) 的路径规划算法,所提方法综合考虑了节点数的优化、移动机器人的实际尺寸和路径光滑性因素。首先引入目标偏置采样;其次,通过融合目标点引力思想的新节点扩展方式,以及改进的候选近邻点度量策略减少树的盲目扩展,加快向目标点的生长;此外,考虑到机器人的实际尺寸,引入节点安全约束,以确保路径的安全性;然后改进路径简化方法,简化初始路径并为后端优化提供安全保障;同时对简化后的路径进行二次 B 样条平滑,使最终路径平滑连续,便于跟踪控制。最后使用 MATLAB 在不同仿真环境下运行以验证改进后算法的有效性。

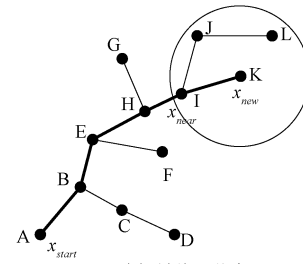
1 RRT* 算法

RRT* 算法引入重选父节点和重布线策略以解决路径最优性的问题,寻径的过程中考虑了路径代价,通过不断迭代来改善路径质量。

RRT* 算法的重选父节点策略:具体过程如图 1(a)~(b)所示,以新节点 x_{new} 为圆心,设置半径 R 为搜索范围寻找 x_{new} 的近邻作为候选父节点,分别为 I、J、L,依次计算每个近邻节点到起点的路径代价加上 x_{new} 到每个近邻的路径代价,把代价最小的点作为新的父节点。经过对比发现,路径 A→B→E→H→I→K 比 A→B→E→H→I→J→K 以及 A→B→E→H→I→J→L→K 的长度都要小,故选择 I 作 K 的父节点加入到树中。



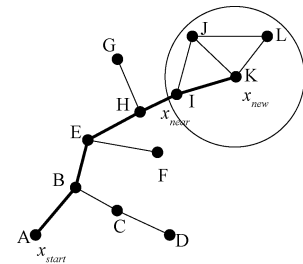
(a) 搜索候选父节点



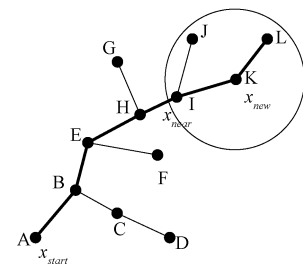
(b) 选择最佳父节点

图 1 重选父节点过程

如图 2(a)~(b)所示, RRT* 算法为了进一步优化路径,部署了重布线过程。该过程试图为 x_{new} 寻找下一个可连接的近邻点,如果该近邻点的父节点改为 x_{new} 后,路径要比原先到达该近邻点的路径短,那么就将该点的父节点改为 x_{new} 。路径 A→B→E→H→I→K→L 比 A→B→E→H→I→J→L 短,因此将 L 的父节点改为 K,即新节点 x_{new} 成为新的随机树。



(a) 优化路径



(b) 更新节点和代价

图 2 重布线过程

2 改进的 RRT* 算法

2.1 目标偏置策略

设定目标偏置概率 p_{bias} , 按照均匀概率随机产生一个

概率值 p , 若 $p < p_{bias}$, 则将目标点 x_{goal} 作为随机采样点 x_{rand} , 反之就在 ω_{free} 进行随机采样, 目标偏置策略如式(1)所示。

$$x_{rand} = \begin{cases} x_{goal}, & p < p_{bias} \\ \text{sampleFree}(\omega_{free}), & p \geq p_{bias} \end{cases} \quad (1)$$

式中: $\text{sampleFree}(\omega_{free})$ 为随机采样函数。 ω_{free} 表示自由空间。

2.2 改进的节点扩展方式

在传统新节点扩展过程中, 新节点 x_{new} 的扩展方向是由随机采样点决定的, 这样会导致新节点 x_{new} 的生成有较大的随机性和盲目性。因此, 受文献[24]的启发, 将 APF 的引力场思想融入 RRT* 的节点扩展中, 这样做考虑了目标点对新节点的“引力”作用, 使得新节点的生成不只依赖于采样点方向。这种方法可以有效减少树在无关区域的生长, 加速树在目标附近的生长, 从而加快路径规划过程, 如图3所示。新的 x_{new} 扩展函数如式(2)所示。

$$x_{new} = x_{nearest} + \Delta s \times \left(C_1 \times \frac{x_{goal} - x_{nearest}}{\|x_{goal} - x_{nearest}\|} + C_2 \times \frac{x_{rand} - x_{nearest}}{\|x_{rand} - x_{nearest}\|} \right) \quad (2)$$

式中: C_1 是目标方向权重, C_2 是采样点方向权重, $C_1 + C_2 = 1$ 。该方法综合考虑了采样点方向和目标点方向, 结合了随机性和引导性, 通过给二者分配不同权重以兼顾到路径规划问题的探索性和优化性, 更加智能地引导了树的生长。

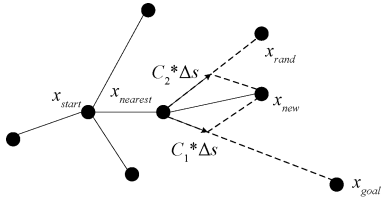


图3 新节点扩展

2.3 近邻节点度量方程

传统 RRT* 算法在选择近邻节点时, 将候选近邻节点 $x_{nearest}$ 到采样点 x_{rand} 的欧氏距离作为度量方法来选取近邻节点, 由于采样点的随机性, 导致近邻节点的位置也是随机产生的, 因此采用新的评估策略来选取近邻点以减少随机性, 引导近邻节点朝目标点的方向进行选取, 如图4所示。近邻点的筛选通过以下两种评估方法并赋予相应权重, 按照 $metric$ 值的大小进行排序, 选出最佳的节点作为近邻点:

1) 计算候选节点中的每个节点与目标点的欧氏距离;

$$D = \sqrt{(x_{goal} - x_i)^2 + (y_{goal} - y_i)^2} \quad (3)$$

2) 计算候选节点中的每个节点与随机点组成的向量 v_1 以及与目标点组成的向量 v_2 形成的方向代价;

$$M = 1 - \frac{v_1 \cdot v_2}{\|v_1\| \cdot \|v_2\|} \quad (4)$$

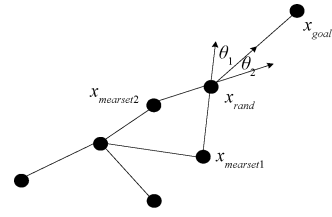


图4 近邻节点选择

3) 计算出 $metric_i$ 的值, 并从小到大进行排列, 筛选出候选近邻节点集合 $X_{nearest}$ 中满足无碰撞且 $metric_{i_{min}}$ 所对应的 $X_{nearest}(i) = (x_i, y_i)$ 作为近邻节点;

$$metric_i = \lambda_1 \times D + \lambda_2 \times M \quad (5)$$

其中, D 为距离代价, M 为方向代价, (x_i, y_i) 为候选近邻点。 λ_1 表示欧氏距离的权重系数, λ_2 表示方向代价的权重系数, $\lambda_1 + \lambda_2 = 1$, 根据这种方式也可以避免较多冗余路径产生。

2.4 节点安全约束

RRT* 算法为了获得最短路径, 有时会在障碍物附近生成节点, 虽然这些节点并没有与障碍物直接发生碰撞, 但它距离障碍物较近, 在实际移动中可能会增加碰撞的风险。为提高路径安全性, 常用的方法有膨胀障碍物、构造目标函数、施加安全距离约束等。膨胀障碍物的方法需要对图像进行预处理, 对地图精度要求较高; 目标函数法能够考虑较多风险因素, 但需要精确设计目标函数; 而施加安全距离约束的方法相对简单易实现, 但可能会增加路径长度。

为了确保路径能与障碍物保持一定的安全距离, 采用施加节点安全约束的方法, 设置节点与障碍物之间的安全距离 d_{safe} , 其大小为机器人的车身半径 d 。采用节点安全约束检测的方法代替传统 RRT* 中的碰撞检测机制, 将节点到障碍物的距离 d_{obs} 与 d_{safe} 进行比较, 如果节点到障碍物的距离 d_{obs} 大于或等于 d_{safe} , 则将节点加入树中, 否则寻找新的安全节点。这种方法考虑了机器人的实际尺寸, 使规划的路径与障碍物保持一定的安全距离。

2.5 路径简化方法

该算法已经找到了一条安全可行的初始路径, 但经大量仿真实验发现这些路径会存在冗余转折, 且由于引入了安全约束, 路径长度有所增加。为此, 本文改进了路径简化方法, 考虑到了机器人的实际尺寸, 在检查当前点与待检查点组成的路径分支是否碰撞的同时, 评估该分支的潜在碰撞风险, 这一改进既减少了冗余节点, 又为后端路径平滑提供一定的安全保障。潜在碰撞风险计算如式(6)所示。

$$C_{safe} = \begin{cases} \frac{1}{(l_{obs} - d_{safe})^2}, & l_{obs} \geq d_{safe} \\ +\infty, & l_{obs} < d_{safe} \end{cases} \quad (6)$$

其中, l_{obs} 表示路径分支到障碍物的距离。

改进路径简化方法的过程如下: 首先, 将初始路径定义为 $results = [P_1, P_2, \dots, P_N]$, 定义两个列表 openlist 和

closestlist, 先将 P_1 由 openlist 移至 closestlist, 然后将 P_1 作为当前节点, 依次向后连接寻找新的待检查节点直到 P_1 与某个待检查点的连线与障碍物发生碰撞为止, 当发生碰撞时, 将 P_1 与该节点之间的其他点加入到 openlist 中, 计算路径分支的潜在碰撞风险, 选取代价低的分支上的待检查点推入到 closestlist 中, 然后将该节点置为当前节点继续向后寻找新的子节点直至连接到目标点为止, 如图 5 所示。

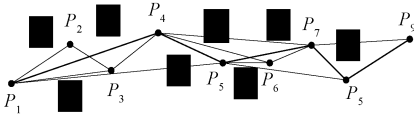


图 5 关键点提取流程

当从点 P_4 向后选取时, $P_4 \rightarrow P_5$ 和 $P_4 \rightarrow P_6$ 都满足无碰撞的条件, 但 $P_4 \rightarrow P_5$ 的路段与障碍物的间隙要比 $P_4 \rightarrow P_6$ 的大, 安全性更高。所以根据计算 C_{safe} 后, 选出简化后路径为 $P_1 \rightarrow P_4 \rightarrow P_5 \rightarrow P_7 \rightarrow P_8 \rightarrow P_9$ 。为后续的平滑过程提供了安全保证。

2.6 转折点局部平滑

生成的初始路径呈锯齿状, 会导致机器人在转折点处多次启停, 降低了机器人的稳定性, 因此需要进行平滑处理。在障碍物较多或机器人存在高转向角的情况下, 进行全局优化时, 由于缺少障碍物边界约束, 可能会增加碰撞风险。而局部优化仅在路径转折处进行平滑处理, 使转折点两侧的路径能够平滑过渡, 且与原路径高度拟合。虽然高次 B 样条曲线能够保证路径的平滑性, 但需要更多的控制点来定义, 增加了数据处理负担。为了满足平滑性需求、实现较高程度拟合并减少计算资源开销, 选择分段二次 B 样条曲线进行局部平滑处理, 在其进行局部平滑时只需要很少的控制点就可以消除锯齿状转折、产生一条平滑的曲线。如图 6 所示为全局优化和局部优化的对比, 虚线表示局部优化, 点划线表示全局优化, 可以看出局部优化只在拐弯处平滑, 减少了不必要的转向, 同时能够高度拟合原路径, 降低了碰撞风险。

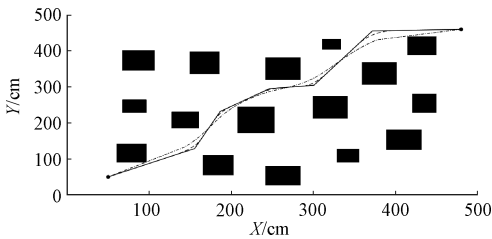


图 6 全局优化和局部优化对比

B 样条曲线由 $n + 1$ 个控制点 B_i 和一个节向量 u_i 定义, 公式如式(7)所示。

$$C(u) = \sum_{i=1}^{n+1} N_{i,k}(u)B_i \quad (7)$$

其中, 由 DeBoor-Cox 递归式(8)~(9)可以得到 B 样条曲线的基函数 $N_{i,k}(u)$:

$$N_{i,k}(u) = \begin{cases} 1, & x_i \leq u \leq x_{i+1} \\ 0, & \text{其他} \end{cases} \quad (8)$$

$$N_{i,k}(u) = \frac{(u - x_i)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i,k-1}(u)}{x_{i+k} - x_{i+1}} \quad (9)$$

其中, x_i 是节点值, $u \in [0, 1]$ 。

二次 B 样条曲线的基函数为式(10)所示。

$$\begin{cases} N_{0,2}(u) = \frac{1}{2}(1 - u)^2 \\ N_{1,2}(u) = \frac{1}{2}(-2u^2 + 2u + 1) \\ N_{2,2}(u) = \frac{1}{2}u^2 \end{cases} \quad (10)$$

则二次 B 样条曲线表示为式(11)所示。

$$C(u) = \frac{1}{2}(1 - u)^2 B_0 + \frac{1}{2}(-2u^2 + 2u + 1)B_1 + \frac{1}{2}u^2 B_2 \quad (11)$$

当给定控制点 B_i 就能确定二次 B 样条曲线。

综上所述, SSRRT* 算法实现流程如图 7 所示。

3 算法仿真实验及性能分析

3.1 不同环境下的算法仿真实验

在 Intel(R) Core(TM) i5-6200U CPU@2.30 GHz 的 MATLAB2022b 仿真平台上进行算法验证。首先搭建不同的静态障碍物环境, 大小为 500 cm × 500 cm, 黑色矩形为障碍物, 白色为可行区域, 起点设为 [50, 50], 目标点为 [480, 460]。

经多次实验测试, 确定了 SSRRT* 算法的参数如表 1 所示。其中, 安全距离的取值是基于 turtlebot3 waffle Pi 的车身半径。偏置采样概率取 0.2。目标方向权重 C_1 在 0.32~0.45 取值和 C_2 在 0.55~0.68 之间取值, 为了保持算法高效性并避免因忽略探索性而陷入局部最优, 因此取 C_1 为 0.45, C_2 为 0.55, 欧氏距离权重系数 λ_1 取 0.4, 方向代价权重系数 λ_2 取 0.6。

为验证本文算法的有效性并消除 RRT 类算法的随机性, 在地图 1、地图 2 和地图 3 环境下对标准 RRT*、文献[25]的自适应目标偏向性 RRT 算法、文献[26]的改进 RRT 算法与所提 SSRRT* 算法各进行了 50 次独立实验, 并统计数据的平均值; 实验效果如图 8、9 和 10 所示, 性能指标如表 2 所示。

由图 8~10 以及表 2 分析可得, SSRRT* 算法相对于标准 RRT* 算法、自适应目标偏向性 RRT 和改进 RRT 算法减少了无效的扩展搜索, 表现出较强的目标导向性; 在地图 1、地图 2 和地图 3 中, SSRRT* 的平均路径长度相对于标准 RRT* 的优化效果为 3.22%、3.41% 和 4.99%, 相较于自适应目标偏向性 RRT 的优化效果为 1.93%、0.89% 和 1.24%, 相较于改进 RRT 算法的优化效果为 5.2%、

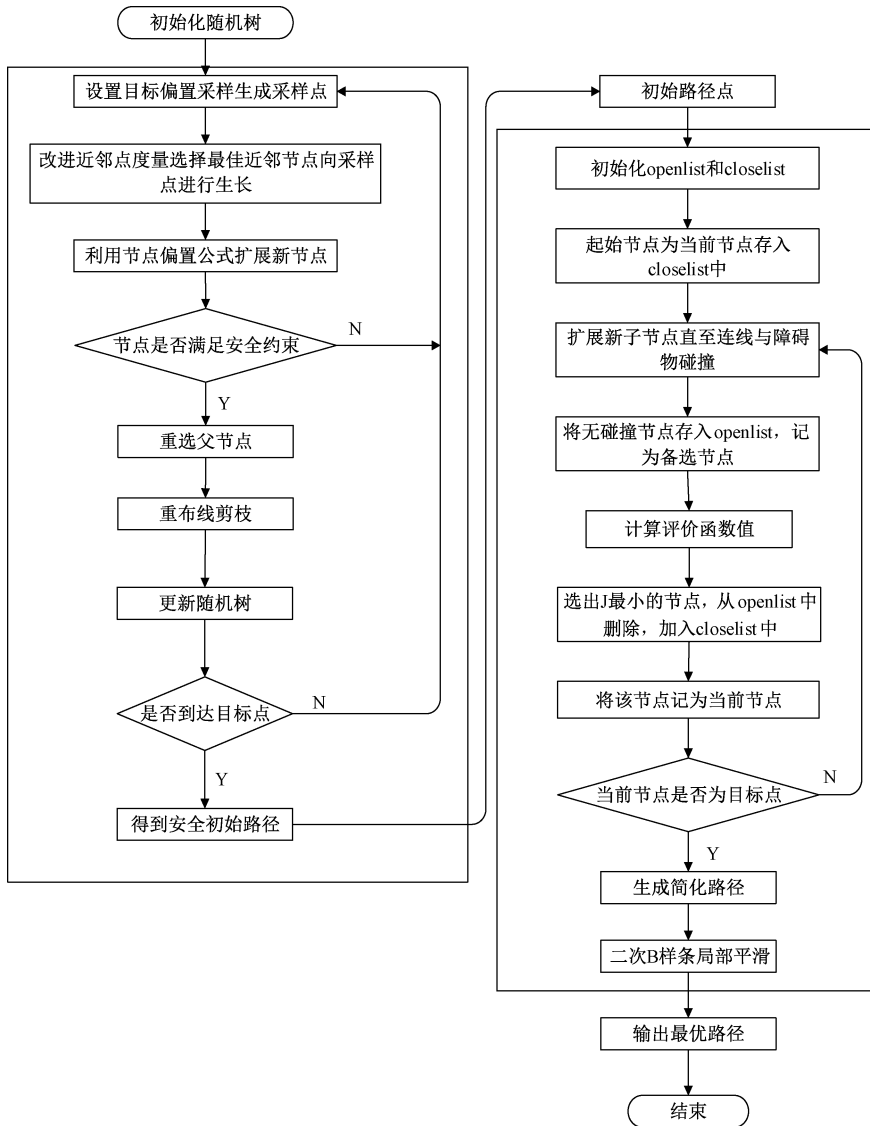


图7 安全-光滑RRT*(SSRRT*)算法实现流程

表1 参数选择

预设参数	取值
安全距离 d_{safe} /cm	15.3
偏置采样概率 p_{bias}	0.2
目标方向权重 C_1	0.45
采样点方向权重 C_2	0.55
欧氏距离权重 λ_1	0.4
方向代价权重 λ_2	0.6

3.2%和7.1%;SSRRT*的平均有效节点数分别为7.76、7.40和7.44,较标准RRT*的优化效果为55.8%、55.6%和58.5%,较自适应目标偏向性RRT的优化效果为57.5%、64.1%和63.3%,与改进RRT算法进行简化操作得到的节点数相差无几;在地图1、地图2和地图3环境中,SSRRT*的平均规划时间相对于RRT*减少了1.0646s、

1.2014s和1.3601s,相较于自适应目标偏向性RRT减少了0.7483s、1.0095s和0.5949s,相较于改进RRT算法减少了0.3286s、0.3504s和0.2045s。根据障碍物的平均最小距离这一指标可知,SSRRT*相较于其他3种算法生成的路径,与障碍物的间隙更大,能够保持安全距离,提高了路径的安全性。综合考虑,尽管相较于其他3种算法,SSRRT*在时间方面的改进幅度不大,但在路径长度、有效节点数和安全性方面均取得了显著的优化效果。

3.2 算法各模块效果分析

通过实验来分别评估节点安全约束、路径简化以及转折点局部平滑对算法效果的影响。

1) 节点安全约束对算法效果的影响

如图11(a)~(b)所示,展示了添加节点安全约束前后路径的效果对比,将约束后与约束前的算法各进行了50次的仿真实验进行对比并取平均值。实验中取安全距离

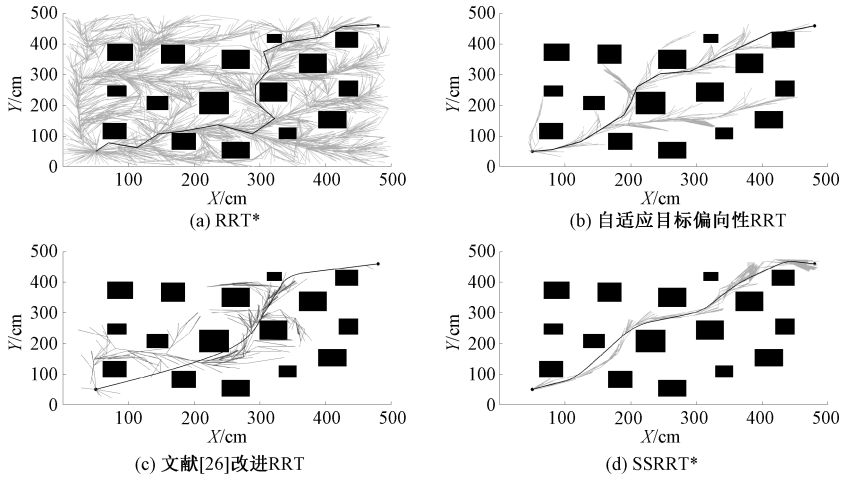


图 8 算法性能比较【地图 1】

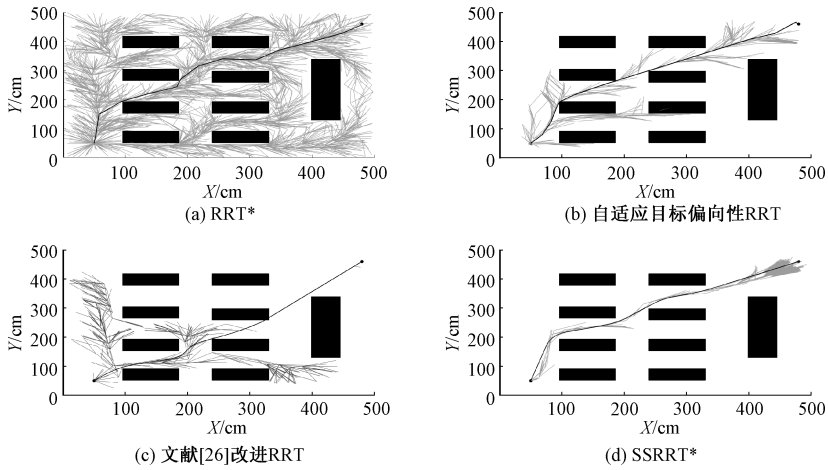


图 9 算法性能比较【地图 2】

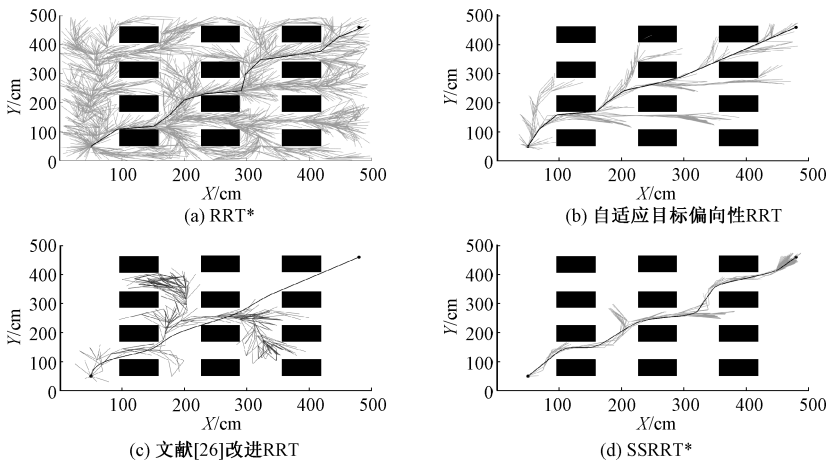


图 10 算法性能比较【地图 3】

$d_{safe} = 15.3 \text{ cm}$ 。由表 3 可知,添加安全约束后牺牲了一定的长度代价,由于需要进行几何距离的判断,平均运行时间增加了 0.160 0 s,但差距较小,在可接受范围内。此外,加入安全约束后,路径与障碍物的最小距离明显增大,增

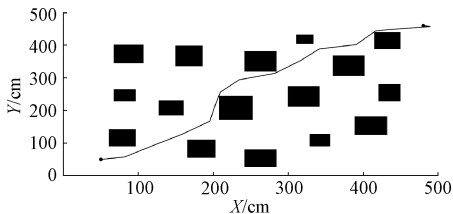
加了约 50.1%,提高了路径的安全性。

2) 路径简化对算法效果的影响

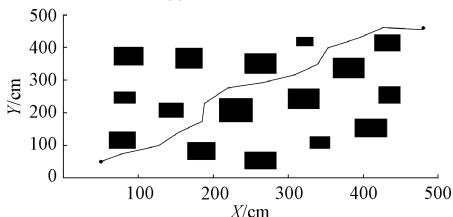
由于添加了节点安全约束,路径长度有所增加,故进行路径简化。图 12 中实线是简化前的路径,虚线是简化

表2 4种地图环境下的指标比较

环境	算法	平均时间/ s	平均路径 长度/cm	平均有效 节点数	平均 拐点数	与障碍物的平均 最小距离/cm
地图1	RRT*	7.643 2	678.23	17.66	15.66	8.049 2
	文献[25]算法	7.326 9	669.29	18.16	15.08	6.904 3
	文献[26]算法	6.907 2	692.35	7.76	5.76	8.384 1
	SSRRT*	6.578 6	646.39	7.72	5.72	15.401 1
地图2	RRT*	7.733 4	691.69	16.72	14.72	8.265 2
	文献[25]算法	7.541 5	674.07	20.64	16.58	7.421 5
	文献[26]算法	6.882 4	690.12	6.42	4.38	7.024 1
	SSRRT*	6.532 0	668.09	6.40	4.40	15.658 7
地图3	RRT*	8.403 2	690.14	17.94	15.94	8.062 2
	文献[25]算法	7.638 0	663.94	20.28	16.44	7.119 7
	文献[26]算法	7.247 6	705.46	7.46	5.46	7.465 7
	SSRRT*	7.043 1	655.71	7.44	5.44	15.323 4



(a) 加节点安全约束前



(b) 加节点安全约束后

图11 添加节点安全约束前后对比

表3 施加安全约束前后对比

算法	平均规划 时间/s	平均路径 长度/cm	与障碍物的 最小距离/cm
加安全约束前	6.201 6	653.42	7.697 6
加安全约束后	6.361 6	676.22	15.402 9

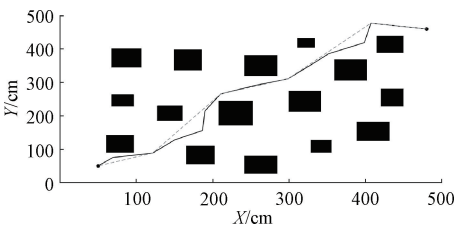


图12 路径简化前后对比图

后的路径。将简化前与简化后的算法各进行了50次的仿真实验进行对比并取平均值,由表4可知,路径简化后比

表4 路径简化前后的指标对比

简化	平均运行 时间/s	路径 长度/cm	有效 节点数	拐点数	与障碍物的 最小距离/cm
简化前	6.370 6	686.22	17.24	15.24	15.534 5
简化后	6.491 6	651.74	7.76	5.76	15.320 6

简化前在运行成本上略微增加了0.1210s,但增加幅度很小,可以接受。简化后的路径相较于简化前的路径在路径长度、节点数两个方面的优化效果显著,分别为5.5%和58.8%,路径长度和节点数量明显减少,降低了计算负担。尽管与障碍物的最小距离略微减小,但仍大于安全距离,路径安全性有所保障。

3) 转折点局部平滑对算法效果的影响

如图13是进行转折点局部平滑的示例图。

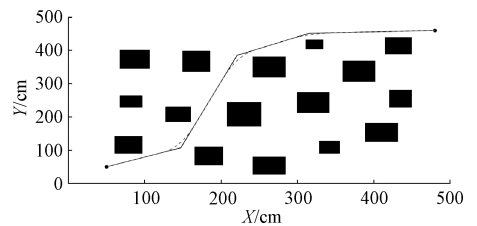


图13 局部平滑效果图

经过节点安全约束和路径简化处理后,路径仍存在锯齿状转折,路径效果如图13所示,因此将局部平滑前和平滑后的算法各进行了50次仿真实验对比并取平均值,由表5看出,优化后比优化前在时间上略微增加了0.056s,但优化后的算法成功消除了路径的锯齿状转折,提高了路径平滑性。由于局部平滑拟合度较高,路径长度有所减小,但减小幅度不大,优化效果约为1.1%,在防止路径长度增加的同时,确保了路径的平滑性。

表 5 B 样条优化前后对比

步骤	平均运行时间/s	平均路径长度/cm
优化前	6.502 6	652.04
优化后	6.558 6	645.19

3.3 算法各阶段组合规划效果的综合评估

通过组合引入节点安全约束、路径简化和转折点局部平滑来评估这些阶段对路径规划性能的综合影响。设置原始 RRT* 算法作为基准参考,记为实验组 1,将综合考虑节点安全约束、路径简化和转折点局部平滑的算法记为实验组 2,其余实验组依次编号。经过 50 次仿真实验取平均值得到实验数据记入表中。

1) 引入节点安全约束和路径简化对算法的影响

将组合考虑节点安全约束和路径简化的算法记为实验组 3。

如图 14(a)~(b) 所示是 3 个实验组效果对比图。图 14(a) 为原始 RRT* 规划出来的路径图,图 14(b) 中实线是实验组 2 规划出来的路径,虚线是实验组 3 规划出来的路径,由图 14 和表 6 分析得出,在平均规划时间方面,实验组 2 和实验组 3 相比实验组 1 略有减少,但实验组 2 与实验组 1 在时间上基本保持一致,表明转折点局部平滑操作的引入对规划时间的影响较小;实验组 2 和实验组 3 的平均有效节点数相比实验 1 从 18.62 减少到 7.44,有明显的减少,这表明路径简化操作的引入使得算法在搜索空间中更高效的扩展;在平均路径长度方面,实验组 2 和实验组 3 相比与实验组 1 有明显的减少,分别减少了 33.856 6 cm、31.843 5 cm,实验组 2 相比与实验组 3 在长度方面也略有减少,仅减少了 2.013 1 cm,表明路径简化操作缩短了路径长度;在安全性方面,实验组 2 和实验组 3 与障碍物的平均最小距离基本保持一致,且能保持安全距离,而实验组 1 与障碍物的距离较近,这表明添加节点安全约束有助于提高安全性,使路径不紧贴障碍物。实验组 3 考虑了引入节

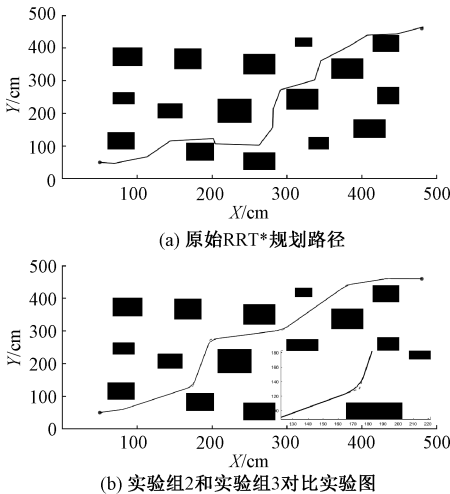


图 14 3 个实验组效果对比图

点安全约束和路径简化对算法的影响,改进了路径的有效节点数并保持安全距离,但由于缺乏平滑操作,相比于 3 个点综合考虑的路径来说,仍存在锯齿状转折,路径平滑度仍需要改进。

表 6 实验组指标对比

实验组	平均规划时间/s	平均有效节点数	平均路径长度/cm	与障碍物的平均最小距离/cm
实验组 1	7.110 7	18.62	693.741 7	9.001 3
实验组 2	6.695 1	7.44	659.885 1	15.301 4
实验组 3	6.689 1	7.44	661.898 2	15.301 6

2) 引入节点安全约束和转折点局部平滑对算法的影响

将组合考虑节点安全约束和转折点局部平滑的算法记为实验组 4。

如图 15(a)~(b) 所示是 3 个实验组效果对比图。图 15(a) 是原始 RRT* 规划出来的路径图,图 15(b) 中实线是实验组 2 规划出来的路径,虚线是实验组 4 规划出来的路径。由图 15 和表 7 分析得出,在平均规划时间方面,实验组 2 和实验组 4 相比实验组 1 基本保持一致,说明引入的改进点并没有显著抑制规划时间;在平均有效节点数方面,实验组 4 由于相比实验组 2 的节点数多了 8.00,表明由于没有加入路径简化操作,使得规划出来的路径仍存在冗余节点,而实验组 2 中路径的搜索更加高效,比实验组 1 所得路径的节点数减少了 10.34;在平均路径长度方面,实验组 4 由于没有加入路径简化操作,得到的路径与原始 RRT* 所得路径的长度相差不大,但是实验组 2 在路径简化的作用下,长度得到显著缩短,相比实验组 1 和实验组 4 分别减少了 46.008 6 cm、31.152 6 cm;在安全性方面,由于实验组 2 和实验组 4 都加入了节点安全约束,二者与障

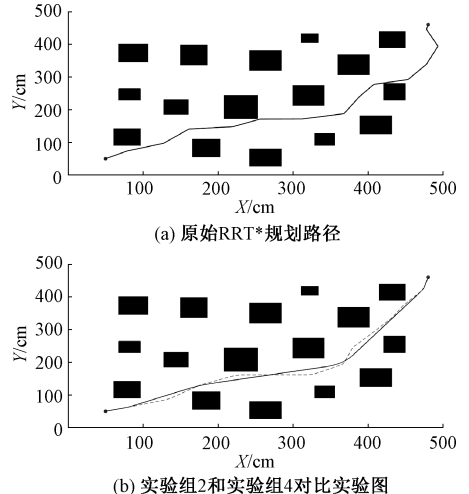


图 15 3 个实验组效果对比图

障碍物的最小距离明显增加,路径的安全性得到显著提升,实验组1由于没有进行安全约束,所以规划出来的路径与障碍物距离过近,安全性低;并且从图15(b)中可以看出,实验组2和实验组4都加入了转折点局部平滑,所以路径的平滑性相比实验组1有显著提升,消除了锯齿状转折。

表7 实验组指标对比

实验组	平均规划时间/s	平均有效节点数	平均路径长度/cm	与障碍物的平均最小距离/cm
实验组1	6.734 4	16.76	677.254 4	8.555 4
实验组2	6.713 1	6.42	631.245 8	15.312 5
实验组4	6.698 4	14.42	662.398 4	15.313 2

3) 引入路径简化和转折点局部平滑对算法的影响

将组和考虑路径简化和转折点局部平滑的算法记为实验组5。

如图16(a)~(c)所示是3个实验组效果对比图。图16(a)是原始RRT*规划出来的路径图,图16(b)是实验组5规划出来的路径,图16(c)是实验组2规划出来的路径。由图16和表8分析得出,在平均规划时间方面,实验组2和实验组5比实验组1略微减少,减少幅度不大,但由于实验组2引入了节点安全约束,涉及到安全距离的计算,所以相比实验组5略微增加,仅增加了0.053 s;由于实验组2和实验组5都引入了路径简化操作,所以相比实验组1的有效节点数减少了7.80,搜索更加高效,在平均路

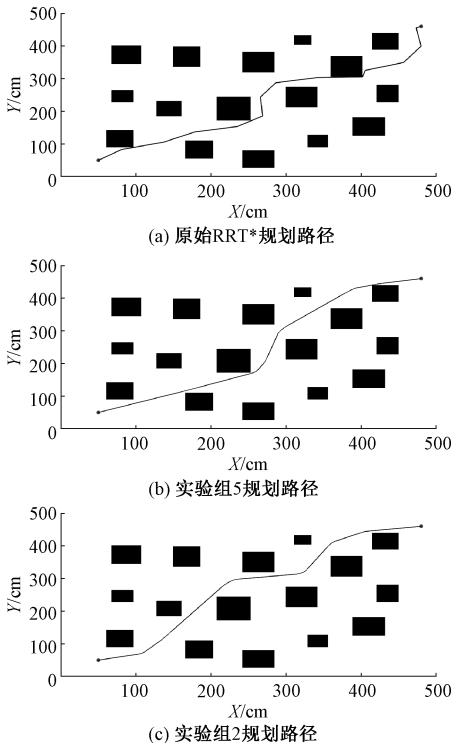


图16 3个实验组效果对比图

表8 实验组指标对比

实验组	平均规划时间/s	平均有效节点数	平均路径长度/cm	与障碍物的平均最小距离/cm
实验组1	6.851 4	15.22	675.092 6	7.769 5
实验组2	6.735 5	7.42	631.245 8	15.301 3
实验组5	6.682 5	7.42	631.098 4	8.731 4

径长度方面,相比实验组1分别减少了43.846 8 cm、43.994 2 cm,实验组2和实验组5在路径长度方面基本相同;由图16的路径规划图可知,由于实验组2和实验组5都加入了转折点局部平滑操作,路径的平滑度显著提升,相比实验组1得到的路径,锯齿状转折得到消除;由于实验组2加入了节点安全约束,得到的路径与障碍物的始终保持一定的安全距离,安全性得到显著提高,而实验组1和实验组5由于缺少节点安全约束,导致规划出来的路径与障碍物距离较近,安全性低。

实验组2即节点安全约束、路径简化和转折点局部平滑都考虑的路径规划算法表现出更佳的安全性和平滑性,是因为3个点都考虑的结果。在考虑了节点安全约束后,路径与障碍物的距离增加,安全性提高,在路径平滑性方面,引入路径简化和转折点局部平滑使路径更加平滑、减少了路径的曲折性。基于实验结果分析,实验组2是最佳的规划方案,因为它综合考虑了安全性和平滑性两个因素,提高了所规划路径的整体质量和性能。

4 结 论

本文提出了一种改进的基于采样的全局规划器SSRRT*,该算法通过增加目标偏置概率、改进新节点和近邻点的生成方式来降低搜索的盲目性,施加节点安全约束以生成与障碍物有足够间隙的路径。为了进一步提高路径质量,采用改进路径简化操作对生成的初始路径进行简化处理,并对简化后的路径进行了二次B样条局部平滑,生成一条适合机器人移动的安全光滑可行路径。实验结果表明,SSRRT*算法生成的路径更加安全光滑,质量更高。然而该算法仅考虑了静态障碍物情况,并未考虑动态障碍物条件对规划任务的影响。因此下一步工作将在现有环境中增加动态障碍物,设计局部路径规划器,并与所提算法融合以进一步提高其适用性。

参考文献

- [1] 王旭,朱其新,朱永红. 面向二维移动机器人的路径规划算法综述[J/OL]. 计算机工程与应用:1-18[2023-07-13]. <http://kns.cnki.net/kcms/detail/11.2127.TP.20230506.1511.008.html>.
- [2] DAN D Z, QING J S. A new algorithm based on dijkstra for vehicle path planning considering intersection attribute[J]. IEEE Access,2021,9.

- [3] 张建光,张方,陈良港,等.基于改进 A* 算法的自动引导车的路径规划[J].国外电子测量技术,2022,41(1):123-128.
- [4] JIAN ZH Q, ZHANG S Y, CHEN SH T, et al. A global-local coupling two-stage path planning method for mobile robots[J]. IEEE Robotics and Automation Letters, 2021, 6(3):5349-5356.
- [5] 徐劲力,许建宁,黄丰云,等.基于改进人工势场法的移动机器人路径规划[J].计算机应用与软件,2023,40(7):289-293.
- [6] 柯清派,史训涛,袁智勇,等.基于改进遗传算法的变电站巡检机器人路径规划[J].电测与仪表,2023,60(8):144-149,156.
- [7] YONG T, YUFANG W, HE G, et al. A path-planning method for wall surface inspection robot based on improved genetic algorithm[J]. Electronics, 2022, 11(8):1192.
- [8] 郝兆明,安平娟,李红岩,等.增强目标启发信息蚁群算法的移动机器人路径规划[J].科学技术与工程,2023,23(22):9585-9591.
- [9] 王宏健,鄂鑫,张凯,等.改进蚁群算法解决 UAV 集群任务规划问题[J].仪器仪表学报,2022,43(9):238-254.
- [10] 郝琨,邓晁硕,赵璐,等.基于区域搜索粒子群算法的机器人路径规划[J].电子测量与仪器学报,2022,36(12):126-135.
- [11] LAVALLE S M. Rapidly-exploring random trees: A new tool for path planning[J]. Research Report, 1999, DOI:http://dx.doi.org/.
- [12] 马新国,马希青.融合改进 RRT 和 Dijkstra 算法的机器人动态路径规划[J].组合机床与自动化加工技术,2023(2):5-9.
- [13] 侯宇翔,高焕兵,汪子健,等.改进 RRT 的移动机器人路径规划算法[J].电子测量技术,2022,45(16):47-52.
- [14] 王杨斌,章伟,王为科,等.改进 Informed-RRT* 的动态环境路径规划算法[J].电光与控制,2022,29(5):28-32.
- [15] 张亮,王鑫,张建锋.多障碍场景下改进 RRT 算法路径规划[J].计算机工程与设计,2023,44(6):1706-1713.
- [16] 张丹萌,甄子洋,陈桢.基于改进 RRT-Connect 的协同航迹规划[J].电光与控制,2021,28(9):25-29.
- [17] 刘奥博,袁杰.目标偏置双向 RRT* 算法的机器人路径规划[J].计算机工程与应用,2022,58(6):234-240.
- [18] WEBB J D, BERG D V J. Kinodynamic RRT* : Optimal motion planning for systems with linear differential constraints [J]. ArXiv Preprint, 2012, ArXiv:1205.5088.
- [19] WANG J, CHI W, LI C, et al. Neural RRT* : Learning-based optimal path planning [J]. IEEE Transactions on Automation Science and Engineering, 2020, 17(4):1748-1758.
- [20] 马小陆,梅宏,王兵,等.基于 JPS 策略的改进 RRT* 移动机器人全局路径规划算法[J].中国惯性技术学报,2020,28(6):761-768.
- [21] 张瑞,周丽,刘正洋.融合 RRT* 与 DWA 算法的移动机器人动态路径规划[J/OL].系统仿真学报:1-12[2023-09-01]. https://doi.org/10.16182/j.issn1004731x.joss.22-1543.
- [22] 许万,杨晔,余磊涛,等.一种基于改进 RRT* 的全局路径规划算法[J].控制与决策,2022,37(4):829-838.
- [23] 王海芳,崔阳阳,李鸣飞,等.基于改进 RRT* FN 的移动机器人路径规划算法[J].东北大学学报(自然科学版),2022,43(9):1217-1224,1249.
- [24] 张伟民,付仕雄.基于改进 RRT* 算法的移动机器人路径规划[J].华中科技大学学报(自然科学版),2021,49(1):31-36.
- [25] 刘恩海,高文斌,孔瑞平,等.改进的 RRT 路径规划算法[J].计算机工程与设计,2019,40(8):2253-2258.
- [26] 顾子倡,刘宇,岳广,等.基于改进 RRT 算法的快速路径规划[J].兵器装备工程学报,2022,43(10):294-299.

作者简介

李文君, 硕士研究生, 主要研究方向为移动机器人路径规划。

E-mail: 3542417683@qq.com

李忠伟, 博士、教授, 主要研究方向为大数据处理与人工智能算法及其智慧应用。

E-mail: lizhongwei@vip.163.com

罗偲(通信作者), 博士、副教授, 主要研究方向为人工智能、机器学习等以及仿生无人系统设计相关技术的应用开发。

E-mail: luo_cai@upc.edu.cn