

# 一种移动机器人路径规划新算法

朱宏辉, 王嘉豪

(武汉理工大学 物流工程学院, 武汉 430063)

**摘要:** 快速搜索随机树 (Rapidly-exploring random Tree Star, RRT\*) 算法在移动机器人实际应用中规划路径在转向部分存在较多的冗余转折点, 导致移动机器人在移动转向过程中出现多次停顿与转向, 为剔除规划路径中的冗余路径点, 提高机器人移动流畅性, 提出一种改进的 RRT\* 算法; 算法将局部逆序试连法引入移动机器人路径规划, 在确保 RRT\* 算法概率完备性和渐进最优性的前提下, 剔除规划路径中的冗余路径节点, 使最终路径更加接近最短路径; 通过 MATLAB 仿真实验证明, 规划路径平均长度缩短 4%, 算法耗时缩短 35%, 改进后的 RRT\* 算法能缩短规划路径且转向部分路径更加平滑; 最后, 使用改进后的 RRT\* 算法在室内环境下进行移动机器人路径规划实验; 实验结果表明: 规划路径上无冗余路径点, 且移动机器人沿路径移动流畅。

**关键词:** 路径规划; 快速搜索随机树; 局部逆序试连; 机器人; 路径优化

## A New Algorithm for Mobile Robot Path Planning

Zhu Honghui, Wang Jiahao

(School of Logistic Engineering, Wuhan University of Science and Technology, Wuhan 430063, China)

**Abstract:** The Rapidly-exploring random Tree Star algorithm is used in the planning of mobile robots in practical applications, but there are many redundant turning points in the turning part of the planning path, which leads to multiple pauses and turns of mobile robot in the process of mobile turning. In order to eliminate the redundant path points in the planning path and improve the mobile fluency of the robot, an improved RRT\* algorithm is proposed. In the algorithm, the local reverse order trial connection method is introduced into the path planning of mobile robot, and on the premise of ensuring the probability completeness and progressive optimality of RRT\* algorithm, redundant path nodes in the planning path are eliminated, and the final path is closer to the shortest path. Through MATLAB simulation and experiments, it is proved that the average length of planning path is reduced by 4% and the algorithm time is reduced by 35%, and the improved RRT\* algorithm can shorten the planning path and turn to part of the path more smoothly. Finally, the improved RRT\* algorithm is used to conduct mobile robot path planning experiment in indoor environment. The experimental results show that there is no redundant path point on the planning path, and mobile robot moves smoothly along the path.

**Keywords:** path planning; RRT\*; local reverse connection; robot; path optimization

## 0 引言

轮式机器人是通过驱动电机带动轮子转动来使得机器人发生移动行为的机器人, 由于其具有结构简单、控制方便、模型易建、承重大、自重轻、行走速度快等特点, 在生活、物流、交通等领域应用最为广泛。路径规划是移动机器人完成各种功能的基础, 具备路径规划能力的移动机器人才真正具备实用性。机器人路径规划是在已知自身位置和目标点或者目标区域的条件下, 在机器人运动空间寻找一条能够无碰撞抵达目标点或者区域的线路<sup>[1-3]</sup>。路径规划能够极大地扩展移动机器人工作空间, 提高移动机器人的实际应用价值。

传统的路径规划算法有: Dijkstra 算法、A\* 算法、蚁群算法、粒子群算法等。Dijkstra 算法简单, 但搜索速率较慢, 耗时较长。A\* 算法通过启发式图搜索方式寻找起始点到终点之间的最低代价路径, 降低了算法的复杂度, 提高了算法效率, 但算法对启发式函数选取较为严苛, 且随着场景规模的增大, 算法运行空间呈指数增长。蚁群算法与粒子群算法等仿生学路径规划算法易与其他算法结合, 但算法运行效率低, 不能满足移动机器人实时性要求, 且容易陷入局部最小值。这些传统算法效率低下, 求解困难, 且无法应用于高维复杂空间<sup>[4-5]</sup>。

1998 年, LaValle 等提出快速扩展随机树 (rapidly-exploring random tree, RRT) 算法<sup>[6]</sup>, 它通过不断在状态空间随机性选取样本点并进行选择来规划一条可行路径<sup>[7]</sup>。RRT 算法无需对状态空间显式建模, 规划速度快且能够考虑机器人的各种约束, 这种算法能够很好解决复杂环境下的路径规划问题。相比其他算法而言, RRT 算法适用于高维复杂空间的路径规划问题, 且算法路径搜索效率更高。不过, RRT 算法也存在一些缺陷, 如收敛速度缓慢、路径

收稿日期:2020-04-21; 修回日期:2020-05-12。

基金项目:国家自然科学基金项目(51805395)。

作者简介:朱宏辉(1956-),男,湖北黄冈人,博士生导师,教授,主要从事智能控制以及嵌入式技术方向的研究。

通讯作者:王嘉豪(1994-),男,湖北黄冈人,研究生,主要从事机器人以及机器视觉方向的研究。

节点冗余等<sup>[8-9]</sup>。2000 年, Kuffner 和 LaValle 等提出了 RRT-connect 算法, 提高了节点的扩展效率, 并于第二年提出双向搜索树 (Bidirectional-RRT) 算法, 由起点和终点同时延伸搜索树, 寻找最优路径。2011 年, Karaman 等<sup>[10]</sup>提出 RRT\* 算法, 通过优化节点扩展方法使得算法具备概率完备性, 随着采样点的不断增加, 总能在空间中寻找一条由起始点到可到达点的路径。同时, RRT\* 算法那还具备渐进最优性, 通过增加采样点逐步优化规划的路径, 但是这种算法在复杂场景下冗余节点较多, 且收敛时间较长。2013 年, M. Jordan 和 A. Perez 提出了将双向扩展的思想同 RRT\* 算法结合的 B-RRT\* (Bidirectional-RRT\*) 算法, 加快了算法的收敛速度, 但是算法在扩展过程中缺乏方向性并且规划路径距离障碍物过近<sup>[11]</sup>。2015 年 Qureshi A H 等人使用智能样本插入函数使得算法快速收敛到最佳路径<sup>[12]</sup>。

针对 RRT\* 算法在室内应用场景下规划路径存在较多冗余路径点导致移动机器人移动过程中出现多次停顿及转向的问题, 本文提出一种基于局部逆序试连的路径优化策略的改进 RRT\* 算法, 通过局部逆序试连法剔除冗余路径点, 缩短了规划路径长度, 且提高了算法运行速度, 避免了实际应用中冗余路径点导致机器人在转弯过程中出现多次停顿和换向现象。

## 1 RRT\* 算法原理

RRT\* 算法是在 RRT 算法基础上进行了相应的改进, 而使其保留算法概率完备性的同时, 还具备渐近最优性的显著特征<sup>[13]</sup>。RRT 算法虽然提升了路径搜索效率, 但是由于节点扩展过程中的随机采样, 导致路径规划过程的不确定性, 算法运行效率时高时低。相同起始点和终点的条件下规划路径差别较大, 可复用性差, 同时规划路径往往与最佳路径偏差较大。由于 RRT 算法获取的最终路径并非全局最优路径, Karaman 和 Frazzoli 提出 RRT\* 算法来解决这个问题<sup>[14]</sup>。RRT\* 算法同样采用随机采样方式扩展随机树, 但其在新产生路径节点附近寻找更优的父节点, 以保证从起点到新路径点的路径代价为最小, 然后对随机树进行重布线。RRT\* 算法通过寻找最优父节点和随机树重布线来不断优化路径, 使得整个路径规划过程渐进优化。算法的实现原理如下:

设  $C$  表示系统的全局状态空间, 路径规划也就是在空间  $C$  内寻找能从起点  $X_{init}$  到目标区域  $X_{goal} \subset C$  或者目标点  $X_{goal}$  的一条由多个节点连接而成的线路, 且线路不经过或接触空间内的任何障碍物区域。全局空间  $C$  由障碍区域和安全区域两部分组成, 若  $C_{obs} \subset C$  表示障碍区域,  $C_{free} \subset C$  表示安全区域, 则应满足条件  $C_{obs} \cup C_{free} = C$  且  $C_{obs} \cap C_{free} = \emptyset$ 。  $T = (V, E)$  表示一棵包含  $k$  个路径节点的随机树, 随机树定义式中  $V$  代表路径节点集合,  $E$  代表连接两节点的边, 满足条件  $T \in C_{free}$ 。定义任意两节点之间的代价值为它们之间的欧几里得距离  $Cost(\varphi(X_i, X_j))$ ,  $Cost \|X_j - X_i\|$  表示节

点  $X_i$  按照随机树  $T$  中节点顺序到节点  $X_j$  的路径长度。

RRT\* 算法寻找节点的方法如下: 首先将起点  $X_{init}$  作为根节点放入随机树中, 构建初始随机树  $T_1$ 。然后, 通过随机函数在安全区域  $C_{free}$  内获取一个随机点  $X_{rand}$ , 并在随机树中寻找与  $X_{rand}$  距离最近的节点  $X_{nearest}$ 。令步长为  $\rho$ , 在  $X_{nearest}$  与  $X_{rand}$  连线之间寻找一点  $X_{new}$ , 使得  $X_{nearest}$  与  $X_{new}$  之间的欧几里得距离为  $\rho$ 。如果  $X_{nearest}$  与  $X_{new}$  连线没有与障碍区域  $C_{obs}$  相交, 则将  $X_{new}$  作为新的路径节点放入随机树  $T$  中, 形成新的随机树。重复上述过程, 随机树扩展过程如图 1 所示。

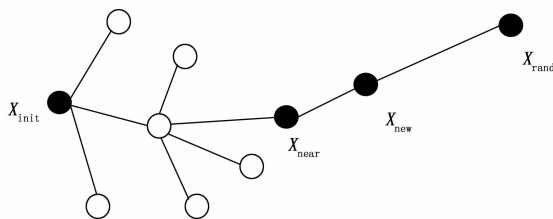


图 1 随机树节点扩展

RRT\* 算法与 RRT 算法随机树扩展过程相同, 而 RRT\* 算法与 RRT 算法的不同之处在于随机树扩展之后还需通过最佳父节点选择和随机树重布线优化规划路径<sup>[4]</sup>。随机树扩展若干次之后的结果如图 2 所示, 假设此时的新节点为  $X_k$ ,  $X_k$  邻域范围内的路径节点集合为  $\{X_{near}\}$ 。以邻域内的节点为  $X_k$  的父节点, 遍历路径节点集合  $\{X_{near}\}$ , 并计算从起始点  $X_{init}$  到父节点  $X_{near}$  与父节点  $X_{near}$  到节点  $X_k$  的代价值之和  $Cost \|X_{near} - X_{init}\| + Cost(\varphi(X_{near}, X_k))$ 。在路径节点集合  $\{X_{near}\}$  中选择使得代价值之和最小的路径节点作为节点  $X_k$  的最优父节点  $X_{min}$ , 从而使新的路径成本最小。RRT\* 算法通过重新选择父节点优化路径, 路径规划生成新的路径节点的过程中将不断优化规划路径, 使得 RRT\* 算法具备渐进最优的特性, 重新选择父节点过程如图 3 所示。

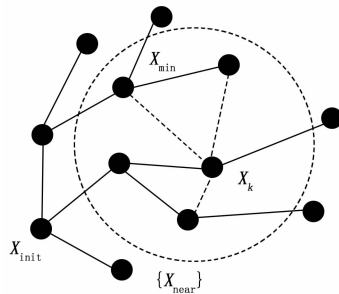


图 2 随机树扩展 k 次后

RRT\* 算法为当前节点重新选择父节点之后, 比较经过最优父节点的路径代价值之和与原路径代价值  $Cost \|X_k - X_{init}\|$  大小, 若前者较小则切断  $X_k$  与原父节点连接, 使用最优父节点  $X_{min}$  替换原有父节点, 并递归改变  $X_k$  子代路径节点连接; 若后者较小, 则保持原有的路径节点连接关系。RRT\* 算法通过对随机树重布线进一步对规

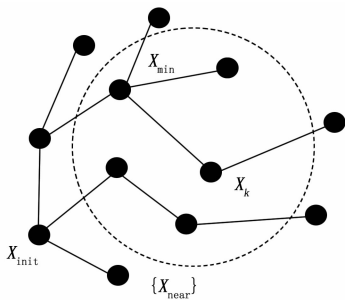


图 3 选择最佳父节点

划路径进行优化, 使得最终生成的规划路径代价值最小, 并且通过随机树重布线减少了随机树中的多余分支, 提高了算法的搜索速率, 随机树重新布线过程如图 4 所示。

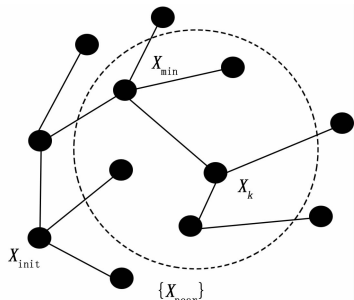


图 4 随机树重布线

重复上述随机树扩展和优化过程, 直到新的路径节点  $X_{new}$  与目标路径点  $X_{goal}$  距离在允许范围内, 迭代终止。每一次迭代过程中, 若随机树重连后任一路径节点连线与障碍物区域相交, 则放弃随机树重连, 直接进入下一次迭代过程。

原始 RRT\* 算法伪代码如表 1 所示, 由原理分析可知算法存如下问题: 采样过程中在全局区域随机生成采样点, 导致在生成路径中存在大量冗余路径节点, 尤其在转向部分路径。

表 1 RRT\* 算法伪代码

序号	伪代码
1	$T \leftarrow \text{RRT}^*(X_{init}, X_{goal})$
2	$V \leftarrow (X_{init}), E \leftarrow \phi; T \leftarrow (V, E)$
3	for $i=0$ to $i=N$ do
4	$X_{rand} \leftarrow \text{Sample}(i)$
5	$X_{nearest} \leftarrow \text{Nearest}(X_{rand}, V)$
6	$X_{new} \leftarrow \text{Steer}(X_{nearest}, X_{rand})$
7	ChooseBestPNode and Rewiring( $X_{nearest}, X_{new}, T$ )
8	$\psi(T) \leftarrow \text{FindBestPath}(X_{init}, X_{goal}, T)$
9	Return $\psi(X_{init}, X_{goal}, T)$

## 2 RRT\* 算法改进

RRT\* 算法通过不断寻找最佳父节点和随机树重连来逐渐优化路径, 但是因为迭代过程中对全局状态空间进行采样, 产生了大量没有必要的节点, 特别是在转弯部分的路径, 如图 5 所示。产生的冗余节点不仅降低了算法的收敛速

度, 还导致算法在实际应用中使得机器人在运动拐角处产生多次停顿和换向运动, 降低了机器人运动的流畅度。

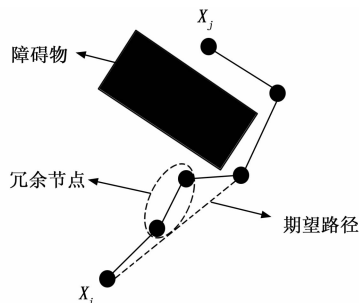


图 5 规划路径中的冗余节点

在实际应用场景中, 由于 RRT\* 算法规划的路径中在障碍物附近的拐角处产生了许多冗余节点, 导致移动机器人在拐角附近运动时, 产生多次停顿并换向。这种现象导致移动机器人的运动流畅性差, 同时降低了机器人的运动效率。同时, 移动机器人频繁停顿和转向产生的机械冲击将增加移动机器人各机械部件的性能损耗, 缩短了机器人的机械寿命。为避免这种现象出现, 本文提出一种局部逆序试连的方法对 RRT\* 算法规划的路径进行优化, 通过局部逆序试连法剔除规划路径中的冗余路径节点, 从而缩短规划路径的长度, 使得转向部分路径更加平滑。在实际应用场景中减少移动机器人停顿和转向的次数, 可以提高移动机器人的运动流畅性, 延长机器人机械寿命, 缩短规划路径, 算法流程如下:

首先在扩展随机树  $T$  中寻找与障碍区域中心距离小于等于  $R$ , 假设找到的所有节点集合为  $\{X_k\}_{k=i_1}^N, \{X_k\}_{k=i_2}^N, \dots, \{X_k\}_{k=i_m}^N, m$  表示节点集合序号。令某一节点集合  $\{X_k\}_m$  中的所有节点依次为  $X_k, X_{k+1}, \dots, X_j$ , 其中  $X_k$  为局部初始节点,  $X_j$  为局部目标节点, 各路径节点按照顺序连接形成局部路径, 更新路径节点集合为  $\{X'\}_m$ , 如图 6 所示。

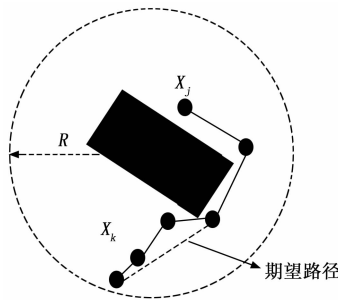


图 6 局部逆序试连

1) 令尝试连接节点  $X_a$  初始值为  $X_k$ , 尝试连接节点  $X_b$  初始值为  $X_j$ , 判断  $X_a$  与  $X_b$  连线是否接触到障碍区域, 如果连线接触到障碍区域, 则将  $X_{j-1}$  赋值给  $X_b$ , 再次判断  $X_a$  与  $X_b$  连线是否接触障碍区域。当  $X_a$  与  $X_b$  之间连线没有接触障碍区域或者  $X_b = X_k$  时, 将此时的节点  $X_b$  放入到与之对应的更新路径节点集合  $\{X'\}_m$  中;

2) 令  $X_a = X_b, X_b = X_j$ , 重复上述步骤 1) 中的过程,

将满足条件的新节点放入到对应的更新路径节点集合  $\{X'\}_m$  中;

3) 重复上述步骤 2), 当  $X_b = X_j$  时,  $X_a$  与  $X_b$  的连线没有接触障碍区域, 则将  $X_a$  放入更新路径节点集合  $\{X'\}_m$  中;

4) 将得到的更新节点集合替换原始的节点集合, 并以更新节点集中的节点替换随机树中原始节点集中的节点, 即完成了一次局部路径平滑。使用下一个节点集合替换此时的节点集合, 重复上述步骤, 进行下一次局部路径平滑。

5) 利用更新路径节点集中所有节点依次替换扩展随机树  $T$  中与障碍物距离小于等于  $R$  的节点, 将更新后的扩展随机树  $T'$  中的路径节点依次连接, 即可得到最终的规划路径。

通过引入局部逆序试连路径优化方法, 有效剔除了 RRT\* 算法规划路径中的冗余路径节点, 缩短了规划路径长度。改进后的 RRT\* 算法在移动机器人路径规划实际应用中, 能够规划出更接近最优路径的路径, 且能够避免移动机器人在转向时候出现多次停顿和换向, 使得移动机器人运动更加流畅。

### 3 实验与结果分析

为了验证对原始 RRT\* 算法改进的有效性和性能更优, 首先需要通过仿真软件进行路径规划模拟实验。仿真实验平台为安装在 64 位 Windows7 系统 (电脑配置: Intel Core 处理器, 主频 3.7 GHz, 内存 32 GB) 上的 MATLAB 2018a。仿真实验中在不同状态空间条件下分别使用 RRT\* 算法和改进后 RRT\* 算法进行路径规划实验, 通过分析在相同条件下两种算法得到的规划路径的对比结果, 验证算法改进的有效性。仿真实验中模拟地图大小为  $1\ 000 \times 1\ 000$ , S 表示机器人起始位置, G 表示目标位置, 黑色阴影区域表示障碍区域, 白色区域为安全区域。

RRT\* 算法的随机采样步长和邻域半径决定了算法的收敛速度和路径的准确度, 适当减少搜索步长和邻域半径可以提高算法的准确性, 但同时降低了搜索效率<sup>[15]</sup>。实验中将随机采样步长设置为 25, 邻域半径为 50, 以使得路径规划实验效果最佳。在相同状态空间条件下, 分别使用 RRT\* 算法和改进的 RRT\* 算法进行 20 次路径规划实验。路径规划过程中迭代次数越多, 表示规划路径所用时间越长<sup>[16]</sup>, 通过对比两种算法规划路径所用迭代次数和规划路径长度可以直观地比较两种算法性能优劣。

在相同状态空间中使用不同的 RRT 改进算法进行路径规划仿真实验, 通过对比不同路径规划算法的仿真实验结果, 分析比较算法性能, 仿真实验中各算法计算得到的规划路径如图 7 所示。其中图 7 为 RRT-Connect 算法规划的路径, 由路径规划结果可知该算法搜索效率较高, 但其规划路径会出现与理想路径有较大的偏差的现象。图 8 为 B-RRT\* 算法路径规划仿真结果, 其路径规划过程中搜索效率更高, 且收敛速度更快, 但是路径在局部区域容易出现与障碍物贴近的情况, 在移动机器人实际应用当中容易出

现碰撞事故, 不适合直接用于室内服务机器人的路径规划。图 9 中 RRT\* 算法规划路径与 B-RRT\* 算法生成的路径相似, 但其生成路径与障碍物能够保持一定的距离, 更适合移动机器人实际使用。但是 RRT\* 算法规划的局部路径存在较多冗余路径点, 导致规划路径曲折, 同时收敛速度较慢。通过局部逆序试连方法对 RRT\* 算法进行优化, 可以得到冗余转折节点大幅度减少的平滑路径, 并使得路径缩短, 如图 10 所示。增加障碍物种类, 使用改进后的 RRT\* 算法再次规划移动机器人运动路径, 如图 11 所示。实验结果表明, 在增加障碍物数目和种类的情况下, 改进后的 RRT\* 算法仍能够正确规划路径, 且规划路径更接近最优路径。

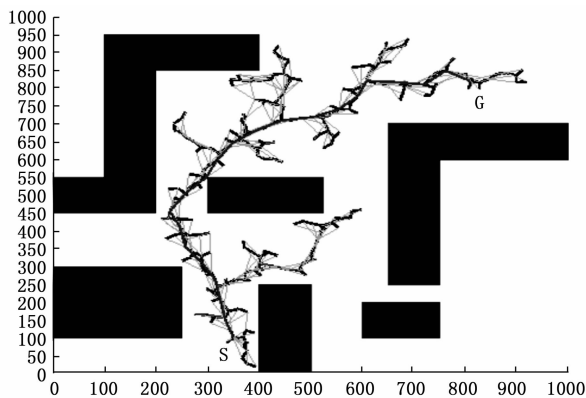


图 7 RRT-Connect 算法规划路径

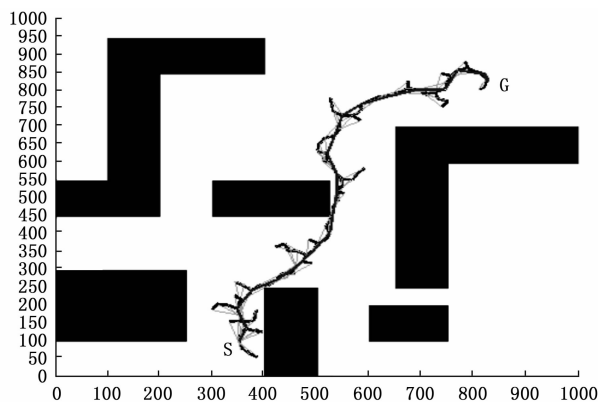


图 8 B-RRT\* 算法规划路径

图 10 和图 11 中路径 1 为原始 RRT\* 算法规划路径, 路径 2 为改进后的 RRT\* 算法规划路径。由观察可知, 改进后的 RRT\* 算法的规划路径中冗余路径节点远少于原始 RRT\* 算法生成路径中冗余路径节点数目, 不仅有效缩短了规划路径的长度, 同时使得规划路径更为平滑。

图 12 为 RRT\* 算法改进前后在多种障碍类型区域下的实验数据对比结果折线图。由折线图可知, 改进的 RRT\* 算法路径规划过程所需迭代次数更少, 并且在任意状态空间条件下其规划路径长度也更短。仿真实验数据统计结果如表 2 所示, 统计结果表明在不同状态空间下改进后的

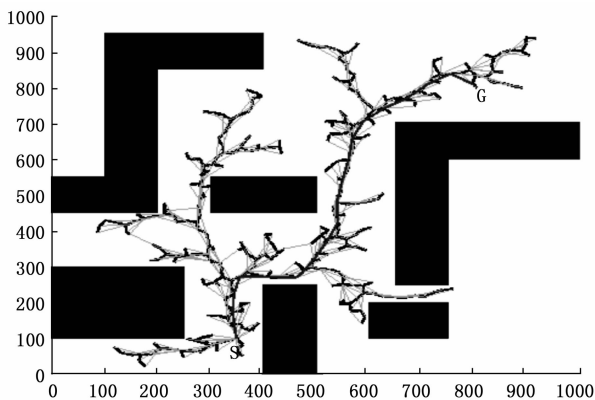


图 9 RRT\* 算法规划路径

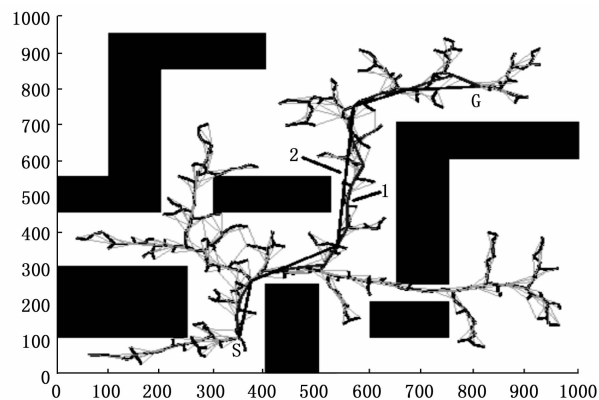


图 10 RRT\* 算法与改进算法规划路径对比 (单一类型障碍物)

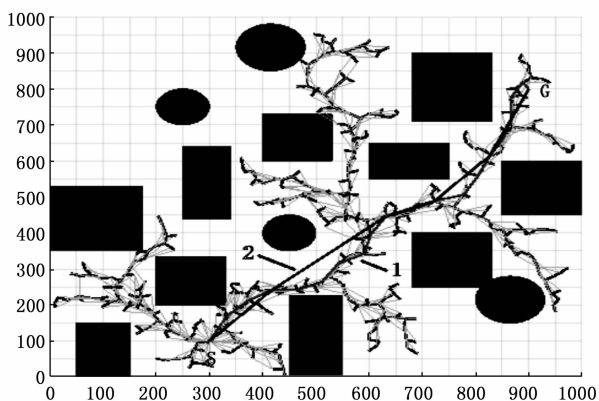
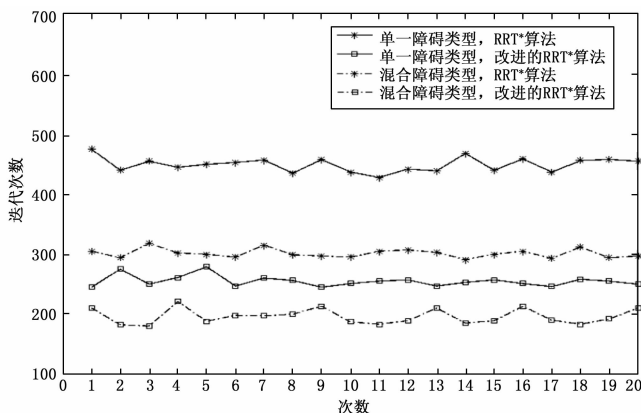


图 11 RRT\* 算法与改进算法规划路径对比 (多种类障碍物)

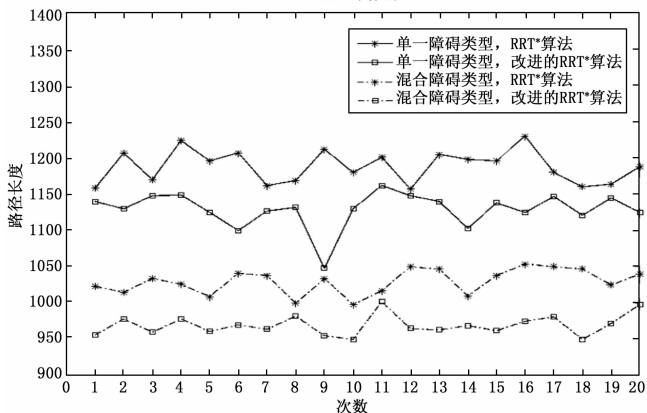
RRT\* 算法平均迭代次数减少了 38% 左右, 平均规划路径长度缩短了 4% 左右, 算法耗时减少 35% 左右。

由仿真实验结果可知, 改进的 RRT\* 算法保留了原始 RRT\* 算法概率完备性和渐进最优的特点, 成功剔除规划路径中的冗余路径节点, 缩短了规划路径的长度, 使得路径规划结果更接近最优路径。

为进一步验证改进 RRT\* 算法在实际应用中的正确性和实用性, 使用该算法进行移动机器人路径规划实验。实验所用移动机器人 X2BOT V2.0, 实验环境为一般室内



(a) 迭代次数



(b) 路径长度

图 12 两种算法实验数据对比结果

表 2 实验结果对比分析表

算法	单一障碍			多种障碍		
	平均迭代次数	平均路径长度	平均耗时/(s)	平均迭代次数	平均路径长度	平均耗时/(s)
原始 RRT* 算法	441.5	1 104.83	11.987	312.60	1 019.38	19.886
改进算法	253.10	1 070.16	8.132	201.43	977.57	12.428
性能增幅	42.6%	3.1%	32.2%	35.6%	4.1%	37.5%

房间, 如图 13 所示。上位机为安装 Ubuntu18.04 系统的笔记本电脑 (Intel Core 处理器, 主频 2.6 GHz, 内存 8 G), 程序运行环境为 ROS。图 14 为机器人在控制软件中的路径规划图, 图中箭头表示机器人在环境地图中的起始位置, 实心方块表示目标点, 算法生成的规划路径由若干路径节点构成。从图 14 中能够看出, 机器人规划出的路径较为平滑, 规划路径中没有多余的路径点。移动机器人在转弯过程中, 没有出现停顿和多次转向的动作, 机器人从起始点到目标点整个移动过程运动流畅。实验结果证明, 改进的 RRT\* 算法通过逆序试连法剔除规划路径中的冗余路径节点, 在实际应用中能够有效避免移动机器人在移动过程出现多次停顿和转向动作, 提高机器人运动连贯性。

### 4 结束语

RRT\* 算法能够规划一条较为接近最短的可行路径,

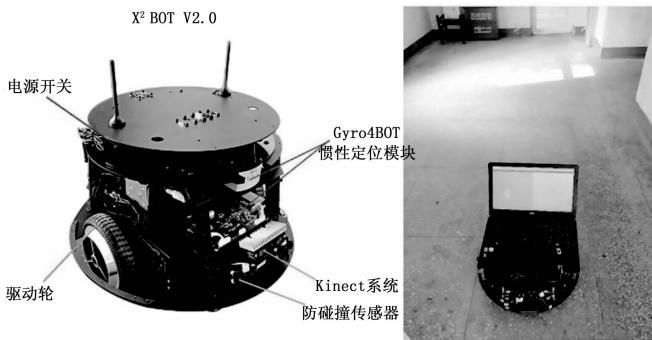
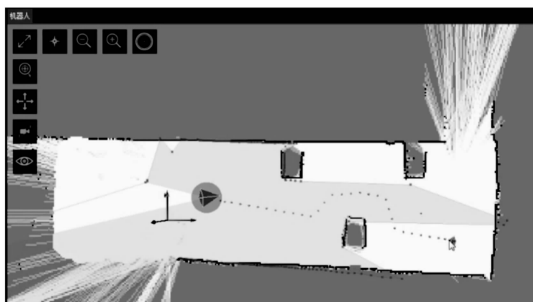
图 13 X<sup>2</sup> BOT V2.0 实物图

图 14 移动机器人路径规划

但是生成路径存在大量不必要的冗余转折节点, 导致机器人移动不连贯和实用性差。为此, 本文提出的一种改进 RRT\* 算法, 通过引入局部逆序试连策略, 能够减少规划路径中的冗余节点, 缩短路径长度, 使得规划路径更加平滑, 更适用于室内移动机器人路径规划。仿真实验和机器人路径规划实验表明, 改进后的 RRT\* 算法规划的路径更接近最优路径, 且路径规划耗时更少。在实际应用中, 能够有效避免移动机器人在转向区域出现多次停顿和换向, 保证机器人移动连贯性。

#### 参考文献:

- [1] 朱大奇, 颜明重. 移动机器人路径规划技术综述 [J]. 控制与决策, 2010, 25 (7): 4-10.
- [2] Jeddisaravi K, Alitappeh R J, Apimenta L C, et al. Multi-objective approach for robot motion planning in search tasks [J]. Applied Intelligence, 2016, 45 (2): 305-321.
- [3] Schmid T M, Cox D C. Robust frequency and timing synchronization for OFDM [J]. IEEE Trans Commun, 1997, 45 (12): 1613-1621.
- [4] Minn H, Zeng M. On timing offset estimation for OFDM systems [J]. IEEE Communications Letters, 2000, 4 (7): 242-244.
- [5] Park B, Cheon H, Kang C, et al. A novel timing estimation method for OFDM systems [J]. IEEE Communications Letters, 2003, 7 (5): 239-241.
- [6] Ren G, Chang Y, Zhang H, et al. Synchronization method

- [3] 张红梅, 李明龙, 杨乐. 基于改进 A\* 算法的移动机器人安全路径规划 [J]. 计算机仿真, 2018, 35 (4): 324-329.
- [4] 裴以建, 杨超杰, 杨亮亮. 基于改进 RRT\* 的移动机器人路径规划算法 [J]. 计算机工程, 2019, 45 (5): 291-296.
- [5] 任子玉. 智能车自主避障路径规划研究综述 [J]. 软件导刊, 2017, 16 (10): 209-212
- [6] Lavelle S M. Rapidly-Exploring Random Trees: A New Tool for Path Planning [R]. Ames, USA: Iowa State University, 1998.
- [7] 宋宇, 王志明. 改进 RRT 移动机器人路径规划算法 [J]. 长春工业大学学报, 2018, 39 (6): 32-36
- [8] 杜明博, 梅涛, 陈佳佳, 等. 复杂环境下基于 RRT 的智能车辆运动规划算法 [J]. 机器人, 2015 (4): 61-68.
- [9] 王全, 王维, 李焱, 等. 基于混合策略的轮式机器人路径规划方法 [J]. 计算机工程与应用, 2014 (4): 45-49.
- [10] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning [M]. Sage Publications, Inc. 2011.
- [11] M. Jordan & A Perez. Optimal Bidirectional Rapidly Exploring Random Trees. [R]. Computer Science and Artificial Intelligence Laboratory Technical Report. 2013.
- [12] Ahmed Hussain Qureshi, Yasar Ayaz, Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments [J]. Robotics and Autonomous Systems, 2015, 68: 1-11.
- [13] Sertac Karaman, Matthew R. Walter, Alejandro Perez, et al. Anytime Motion Planning using the RRT [A]. 2011 IEEE International Conference on Robotics and Automation [C]. 2011: 1478-1483.
- [14] Karaman S, Frazzoli E. Incremental Sampling-based Algorithms for Optimal Motion Planning [A]. Robotics: Science and Systems Conference [C]. 2010, 267-274.
- [15] Jeong I B, Lee S J, Kim J H. RRT\* - Quick: A Motion Planning Algorithm with Faster Convergence Rate [J]. advances in intelligent systems & computing, 2015, 345: 67-76.
- [16] Gammell J D, Srinivasa S S, Barfoot T D. Informed RRT\*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic [J]. Computer Science, 2014 (4): 2997-3004.
- [7] Meng J, Kang G. A novel OFDM synchronization algorithm based on CAZAC sequence [A]. international conference on computer application and system modeling [C]. 2010.
- [8] 张磊. LTE 上行链路 SC-FDE 同步技术研究 [D]. 杭州: 杭州电子科技大学, 2012.
- [9] 王香利. SC-FDE 系统中的频域均衡技术研究 [D]. 郑州: 河南工业大学, 2010.

(上接第 231 页)