

# 基于改进 RRT\*算法的仓储机器人路径规划优化方法

闫转英<sup>1</sup> 李奇<sup>2</sup> (通讯作者)

1 浩鲸云计算科技股份有限公司, 江苏南京, 210000;

2 南京工业大学浦江学院计算机与通信工程学院, 江苏南京, 211200;

**摘要:** 针对 RRT 算法在室内环境中路径规划时存在收敛速度慢、轨迹曲折等问题, 本文提出一种结合目标导向采样、自适应步长与贝塞尔曲线平滑的改进方法。该方法在保持 RRT 渐进最优性的基础上, 通过调整采样策略减少无效探索, 依据局部障碍密度动态调节扩展步长, 并对生成路径进行几何平滑处理。在典型室内场景下的仿真实验表明, 与标准 RRT\*相比, 所提方法平均规划时间减少 31.2%, 路径长度缩短 12.5%, 转弯次数降低 45.1%。结果说明该方法可在有限计算资源下获得更适用于智能车执行的路径。

**关键词:** 路径规划; 改进 RRT\*算法; 智能车; 采样优化; 轨迹平滑

**DOI:** 10.69979/3041-0673.26.01.102

## 引言

室内移动智能车的自主导航依赖于可靠的路径规划模块。在结构化但包含静态家具与动态行人等障碍的环境中, 规划算法需在合理时间内生成可行且便于跟踪的轨迹。传统图搜索方法受限于离散化建模, 在连续空间中难以精确表达车辆运动约束。采样-based 方法因其对高维连续状态空间的天然适应性, 成为此类场景的常用选择。

RRT\* (Rapidly-exploring Random Tree Star) 作为 RRT 的渐进最优变体, 理论上可在无限采样下收敛至最优路径, 已被用于多种移动平台的规划任务。但在实际应用中, 其性能受采样效率与环境复杂度影响较大。均匀随机采样易导致大量节点分布于非关键区域, 延长收敛时间; 生成路径常含高频方向变化, 增加底层控制器负担; 固定步长在不同密度区域表现不一致; 面对动态障碍时缺乏高效重规划机制。

本文围绕上述问题, 对 RRT\*进行三项改进: 引入目标方向引导的混合采样策略, 设计基于局部障碍密度的自适应步长机制, 并采用三次贝塞尔曲线对原始路径进行后处理平滑。改进后的算法在典型室内布局中进行测试, 结果用于评估其在规划效率与路径可执行性方面的变化。

## 1 RRT\*算法基本原理

RRT\*由 Karaman 与 Frazzoli 于 2011 年提出, 是一种基于增量采样的渐进最优路径规划算法。设配置空间为  $C \subseteq R^2$ , 障碍物区域为  $C_{obs}$ , 自由空间为  $C_{free} = C - C_{obs}$ 。给定起始点  $C_{start} \in C_{free}$  与目标点  $x_{goal} \in C_{free}$ , 算法维护一棵以  $x_{start}$  为根的树  $T = (V, E)$ , 其中  $V$  为已采样节点集合,  $E$  为连接边集合。

算法每轮迭代执行如下操作: 首先以概率  $p_{goal}$  (通

常取 0.1-0.3) 令  $x_{rand} = x_{goal}$ , 否则在  $C_{free}$  中均匀随机采样一点; 随后在  $V$  中查找欧氏距离最近的节点  $x_{nearest} = \operatorname{argmin}_{v \in V} \|v - x_{rand}\|$ ; 沿方向  $x_{nearest} \rightarrow x_{rand}$  以固定步长  $\eta$  生成新节点  $x_{new} = x_{nearest} + \eta \cdot \frac{x_{rand} - x_{nearest}}{\|x_{rand} - x_{nearest}\|}$ ; 若线段  $(x_{nearest}, x_{new})$  与  $C_{obs}$  无交集, 则将  $x_{new}$  加入  $V$ , 并将边  $(x_{nearest}, x_{new})$  加入  $E$ ; 最后, 在半径  $r_{rewire}$  邻域内对所有节点  $v \in V$  执行重连接操作: 若通过  $x_{new}$  到达  $v$  的代价更低, 则更新  $v$  的父节点并递归更新其子树代价。

该算法具备概率完备性与渐进最优性, 但其实际性能依赖于采样分布与环境几何特性。在室内环境中, 家具、墙体等静态障碍形成非均匀自由空间, 使得均匀采样效率下降, 初始路径质量不稳定。

## 2 改进 RRT\*算法设计

### 2.1 混合采样策略

为提升搜索导向性, 本文将采样过程分为三类: 目标采样、引导采样与随机采样。设采样概率分别为  $p_{goal} = 0.3$ 、 $p_{base} = 0.4$ 、 $p_{rand} = 0.3$ 。当触发引导采样时, 计算当前最近邻节点  $x_{nearest}$  指向目标的方向向量, 并沿该方向延伸一段距离得到引导点:

$$x_{bias} = x_{nearest} + \beta \cdot \frac{x_{goal} - x_{nearest}}{\|x_{goal} - x_{nearest}\|} \quad (1)$$

其中  $\beta$  为引导距离, 取值为以  $x_{nearest}$  为中心、半径 2.0 m 范围内最大无障碍距离的 80%。采样规则如下: 以 0.3 概率返回  $x_{goal}$ , 以 0.4 概率返回  $x_{bias}$ , 其余 0.3 概率均匀采样。该策略使采样点更集中于可行且朝向目标的区域。

### 2.2 自适应步长机制

定义局部障碍密度  $\rho(n)$  为节点  $n$  周围半

径  $R = 2.0m$  内障碍栅格数量与总栅格数之比。步长  $\eta(n)$  根据  $\rho(n)$  线性插值:

$$\eta(n) = \eta_{max} \left(1 - \frac{\rho(n)}{\rho_{max}}\right) + \eta_{min} \frac{\rho(n)}{\rho_{max}} \quad (2)$$

其中  $\eta_{max} = 1.5$  ,  $\eta_{min} = 0.3m$  ,  $\rho_{max} = 1.0$  。在开阔区域 ( $\rho \approx 0$ ) , 步长接近  $\eta_{max}$  , 加快探索; 在密集区域 ( $\rho \rightarrow 1$ ) , 步长趋近  $\eta_{min}$  , 提高安全性。该机制避免了固定步长在异质环境中的适应性不足。

### 2.3 贝塞尔曲线路径平滑

RRT\* 输出的路径为点列  $\{p_0, p_1, \dots, p_k\}$  , 其中  $p_0 = x_{start}$  ,  $p_k = x_{goal}$  。首先使用 Douglas-Peucker 算法简化路径, 保留曲率变化超过阈值的关键点, 得到简化序列  $\{q_0, q_1, \dots, q_m\}$  。随后对每三个连续点  $(q_i, q_{i+1}, q_{i+2})$  构造三次贝塞尔曲线:

$$B_i(t) = (1-t)^3 q_i + 3(1-t)^2 t Q_{i,1} + 3(1-t)t^2 Q_{i,2} + t^3 q_{i+2}, t \in [0, 1], \quad (3)$$

其中控制点

$$Q_{i,1} = q_i + \alpha(q_{i+1} - q_i), \quad Q_{i,2} = q_{i+2} - \alpha(q_{i+2} - q_{i+1}), \quad \alpha = 0.6$$

最终轨迹由各段  $B_i(t)$  拼接而成。

为确保安全性, 对平滑轨迹以  $\Delta s = 0.1m$  间隔采样, 检查每个采样点是否满足  $dist(p, C_{obs}) \geq \delta$  , 其中安全裕度  $\delta = 0.75m$  。若某段违反约束, 则保留原始折线段  $(q_i, q_{i+1}, q_{i+2})$  。

### 2.4 算法流程与复杂度

完整算法在标准 RRT\* 框架上嵌入上述改进。每次迭代中, 先执行混合采样, 再计算自适应步长, 扩展新节点后执行重连接。路径生成后调用平滑模块。时间复杂度仍为  $O(n \log n)$  ( $n$  为采样次数), 但因采样效率提升, 实际运行时间减少。空间复杂度为  $O(n)$  , 与原算法一致。

## 3 仿真实验与结果分析

### 3.1 实验设置

构建  $20m \times 20m$  室内仿真环境, 采用  $0.1m$  分辨率栅格地图, 包含墙体、桌椅、沙发等静态障碍, 覆盖面积约 35%。动态障碍设为 5 个圆形实体, 半径  $0.4m$ , 速度  $0.6 - 1.2m/s$ , 沿预设或随机轨迹移动。起始点与目标点在自由空间内随机生成, 共 50 组独立任务。对比算法包括: 标准 RRT、RRT\*、Informed RRT\*。实验平台为 Intel i7-11800H CPU、32 GB RAM, 算法基于 Python 3.9 与 OMPL 库实现, 最大采样次数设为 5000。仿真环境步图如图 1 所示。

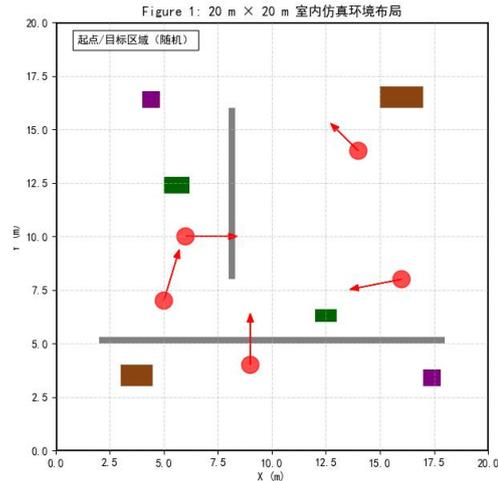


图 1 20m\*20m 室内仿真环境布局

评估指标包括: (1) 规划时间 (ms); (2) 路径长度 (m); (3) 转弯次数 (相邻三段夹角变化超过  $15^\circ$ ); (4) 动态场景任务成功率 (成功避障并抵达目标的比例)。

### 3.2 结果与讨论

如表 1 汇总了平均性能数据, 所提方法平均规划时间为  $186.3ms$ , 低于 RRT 的  $270.8ms$ ; 路径长度为  $24.7m$ , 短于 RRT 的  $28.2m$ ; 平均转弯次数为  $5.2$  次, 少于 RRT 的  $9.5$  次; 在动态场景中, 任务成功率为  $92.4\%$ , 高于 RRT 的  $73.5\%$ 。

表 1. 路径规划算法平均性能对比 (50 组任务, 动态环境)

性能指标	RRT	RRT*	本文方法(改进 RRT)
平均规划时间 (ms)	270.8	228.5	186.3
平均路径长度 (m)	28.2	28.0	24.7
平均转弯次数	9.5	9.4	5.2
动态场景任务成功率 (%)	73.5	85.2	92.4

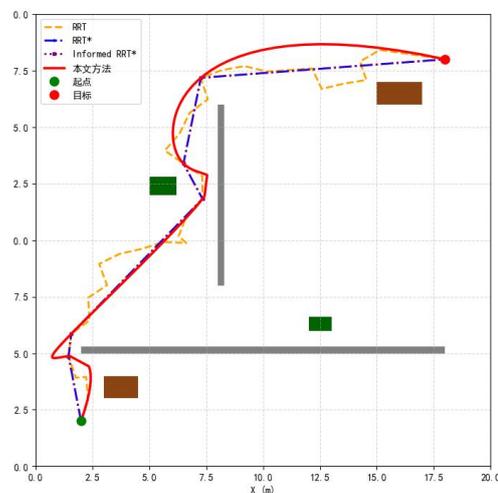


图 2 三种算法路径对比

图2展示了典型路径对比。标准RRT路径明显非最优；RRT\*虽经重连接优化，但仍呈锯齿状；本文方法生成轨迹更为平直，转向过渡自然。消融实验进一步验证各模块贡献：仅引入混合采样，规划时间减少22.1%；加入自适应步长后，狭窄通道通过率提升18.3%；完整算法在转弯次数上比仅采样改进版本再降26.7%，说明平滑模块对路径可执行性有直接作用。

实验还测试了不同动态障碍数量下的表现。当障碍数从2增至8时，所提方法成功率从96.1%降至85.3%，下降幅度小于RRT\*（从82.4%降至61.7%），表明其对动态环境具有一定鲁棒性。

## 4 结论

本文对RRT算法进行了三项改进，并在室内智能车路径规划任务中进行了验证。混合采样策略减少了无效节点生成，自适应步长提升了环境适应能力，贝塞尔平滑改善了路径可执行性。在测试场景中，改进方法在规划时间、路径长度与转弯次数上均优于标准RRT。后续工作将考虑多智能体协同规划、动态障碍预测以及算法在嵌入式系统中的部署优化。

### 参考文献

- [1]Azadeh, K., de Koster, R., & Roy, D. (2019). Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4), 967-999.
- [2]Le-Anh, T., & De Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, 171(1), 1-23.
- [3]Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100-107.
- [4]Yap, P., Burch, N., & Holte, R. (2011). Dynamic route planning in video games. In *Proceedings of the Seventh Conference on Artificial Intelligence and Interactive Digital Entertainment* (pp. 142-147). AAAI Press.
- [5]Sturtevant, N. R. (2012). Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2), 144-148.
- [6]Liu, B., He, L., & Atkin, J. A. (2018). Multi-objective path planning for automated guided vehicles with energy consumption and travel time trade-offs. *International Journal of Production Research*, 56(15), 5117-5133.
- [7]Van den Berg, J., Lin, M., & Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation* (pp. 1928-1935).
- [8]Phillips, M., & Likhachev, M. (2011). SIPP: Safe interval path planning for dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 5628-5635).
- [9]Kim, J., Zhang, S., & Sun, C. (2017). Trajectory smoothing and velocity planning for autonomous vehicle navigation. *IEEE Transactions on Intelligent Transportation Systems*, 18(9), 2446-2455.
- [10]Rosmann, C., Hoffmann, F., Bertram, T., & Knoth, O. (2017). Trajectory modification considering dynamic constraints of autonomous robots. In *ROBOTIK 2017; 10th German Conference on Robotics* (pp. 1-8). VDE.
- [11]Dehghan, M., & Lee, C. G. (2021). Simulation-based optimization for warehouse robot picking systems. *Computers & Industrial Engineering*, 160, 107573.
- [12]Wang, Y., & Li, X. (2020). Energy-efficient path planning for autonomous mobile robots in manufacturing systems. *Journal of Manufacturing Systems*, 57, 286-296.

资助项目：南京工业大学浦江学院校级课题（secbr-zt-11）