

基于 TR-RRT 算法的机械臂路径规划研究

宋仁捷¹, 葛长青², 张东阳¹, 苗建军¹

(1. 沈阳理工大学 装备工程学院, 沈阳 110159; 2. 中国人民解放军 32752 部队, 湖北 襄阳 441100)

摘要: 为使机器人在复杂环境中高效执行任务, 不仅要求其具备一定的算力基础, 还需对路径规划算法进行有效优化。针对传统 RRT 算法用于复杂环境时存在计算量庞大及路径搜索效率低下等问题, 提出一种目标约束 RRT (target restraint RRT, TR-RRT) 算法, 通过引入目标偏置、约束点引导、冗余点移除、动态步长、三次样条插值等策略, 增强搜索能力, 提高搜索效率, 并对规划的路径进行平滑处理。为验证本文改进算法的性能, 分别在二维、三维环境以及 Gazebo 环境中进行仿真实验, 并与 RRT、RRT-Connect、Informed-RRT* 算法进行比较, 结果表明, 本文改进算法在不同实验环境下的规划时间和路径长度及节点数量均优于对比算法, 显著提高了路径规划的效率与稳定性。

关键词: RRT 算法; 路径规划; 目标偏置; 动态步长

中图分类号: TP242 **文献标志码:** A **DOI:** 10.3969/j.issn.1003-1251.2026.01.003

Research on Path Planning of Robotic Arm Based on TR-RRT Algorithm

SONG Renjie¹, GE Changqing², ZHANG Dongyang¹, Miao Jianjun¹

(1. Shenyang Ligong University, Shenyang 110159, China; 2. Unit 32752 of the People's Liberation Army of China, Xiangyang 441100, China)

Abstract: To enable robots to perform tasks efficiently in complex environments, it is necessary for them not only to have a certain computing power foundation but also to effectively optimize the path planning algorithm. For the problems of large computational load and low path search efficiency when the traditional RRT algorithm is applied in complex environments, a target constraint RRT algorithm (TR-RRT) is proposed. By introducing strategies such as target bias, constraint point guidance, redundant point removal, dynamic step size, and cubic spline interpolation, the search ability is enhanced, the search efficiency is improved, and the planned path is smoothed. To verify the performance of the improved algorithm, simulation experiments are conducted in two-dimensional, three-dimensional environments, and the Gazebo environment, and compared with the RRT, the RRT-Connect, and the Informed-RRT* algorithm. The results show that the improved algorithm outperforms the comparison algorithms in terms of planning time, path length, and node number in different experimental environments, significantly improving the efficiency and stability of path planning.

Key words: RRT algorithm; path planning; target bias; dynamic step size

随着科技的迅速发展, 机器人因出色的灵活性、精确度和稳定性, 在工业领域中的应用日益广

泛, 其中最常见的是多自由度机械臂。为使机器人高效地执行任务, 需要在避障的同时找到一条

最优的可行路径,这不仅要求机器人具备较高的计算能力,还需要在算法设计上进行有效优化,以应对复杂的作业环境。因此,机器人的路径规划研究具有重要的理论和实际意义。

目前常用的路径规划算法主要有 A* 算法^[1-3]、RRT* 算法^[4-5]、RRT 算法^[6-8]等。A* 算法是典型的全局搜索算法,其借助启发式函数引导搜索方向,能够在静态环境中保证路径的最优性与完备性,但在三维工作环境中易因状态空间爆炸导致计算复杂度呈指数级增长,出现计算困难、效率低下等问题。RRT* 算法是采样类算法的代表,其通过随机采样方式突破高维空间的维数灾难限制,具备在复杂非结构化环境中运行的能力,但其随机采样特性会导致前期路径质量较差,且在三维空间应用时需进行大量碰撞检测与路径优化,同样面临计算量庞大的瓶颈。RRT 算法通过在工作空间中随机生成树状结构来探索路径,相较于其他算法,具有较强的搜索能力和较快的搜索速度,特别是在较单一的环境中能够快速生成路径,但其在实践中仍存在路径质量较差、路径不够平滑、算法稳定性不足等问题,尤其在复杂环境中容易陷入局部搜索而导致无法找到有效路径。此外,RRT 算法的路径规划结果通常会包含大量冗余路径点,极大地降低了机器人执行任务的效率。

为克服上述算法存在的问题,研究人员提出了多种改进算法,如 Bi-RRT 算法^[9]和 Informed-RRT* 算法^[10]等,但这些算法仍然存在一定的局限性,尤其在路径的平滑性和稳定性等方面。

针对现有 RRT 及其衍生算法仍存在搜索效率低下、搜索范围不精确及最终路径不够平滑等问题,本文提出一种目标约束 RRT(target restraint RRT, TR-RRT)算法。通过设定的约束点对搜索树的搜索方向进行引导,并引入目标偏置策略,设置偏置阈值,以显著提高搜索效率;在节点扩展阶段,根据环境状态动态调整步长,增强算法的探索能力;使用基于 Dijkstra 算法的冗余点移除策略进行路径剪枝;使用三次 B 样条插值策略对规划的路径进行平滑优化。通过二维、三维环境及 Gazebo 环境下的仿真实验验证本文改进算法的性能。

1 相关算法简介

1.1 RRT 算法

RRT 算法在智能终端与无人驾驶系统的自主寻路功能中发挥着重要作用。该算法通过空间探

索与树状结构生长相结合的方式构建可行轨迹,主要流程如下。

1) 构建搜索结构框架:以初始位置为根节点建立空结构,每个节点对应实际坐标系中的具体坐标,连接线表示相邻节点间的轨迹。

2) 空间随机选点:迭代过程中按预设步长在工作区域内生成随机目标点,该点的空间分布遵循均匀概率原则。

3) 邻近点定位:完成随机选点后,采用空间索引结构快速定位当前树状结构中目标点几何距离最近的节点,该过程可借助多维空间分割技术进行优化。

4) 路径延伸操作:确定最近点后,沿目标点方向按固定步长进行延伸探索,该过程需进行障碍物接触判定,若延伸路径无障碍物阻隔,则将新节点纳入路径网络并建立连接。

5) 终点区域判定:每当生成新节点时,系统将检测该点是否进入目标范围,当新节点与终点的间距小于设定步长时,建立终点连接并确认路径构建完成。

6) 轨迹逆向重构:当路径网络成功延伸至目标区域后,系统通过从终点向起点逐级追溯父节点的方式,形成完整的可执行轨迹。

1.2 RRT-Connect 算法

RRT-Connect 算法^[11-12]是路径规划领域中的经典概率采样算法,该算法在传统 RRT 算法基础上引入双向搜索策略,显著提升了路径搜索效率,特别适用于高维空间的运动规划问题。

RRT-Connect 继承了 RRT 算法的核心思想,通过构建随机树探索未知空间。其创新点在于同时生成两棵随机树:一棵树(T_a)从起始点 q_{init} 生长,另一棵树(T_b)从目标点 q_{goal} 生长。两棵树通过交替扩展策略实现双向搜索,当两棵树的扩展区域出现交叠时即判定为找到可行路径。具体流程为:主循环中首先生成随机采样点 q_{rand} , T_a 树执行扩展操作,寻找最近邻节点 q_{near} ,沿 q_{near} 方向扩展新节点 q_{new} ;若扩展成功,则 T_b 树执行连接操作,即从当前节点连续扩展,直至到达 T_a 方向或达到最大步长;每次迭代后交换两棵树角色,实现动态平衡搜索;当某棵树的节点进入另一棵树的邻域时,验证两点间直线路径是否有碰撞,若无障碍则建立连接,完成路径规划。与 RRT 算法相比,RRT-Connect 算法使用双搜索树策略,能够更快找到最优路径,具有更高的搜索效率。此外,RRT-Connect 还具有良好的并行化能力,

可通过多线程或分布式计算来提升性能。但 RRT-Connect 算法也存在局限性:在复杂环境中或者目标点附近障碍物较多的情况下,可能需要生成更多的节点才能找到最优路径;算法的性能依赖于步长等参数的设定,需要在实际应用中进行调整。

2 算法改进

2.1 目标偏置策略

为提高路径规划的效率和精度,本文采用目标偏置策略对 RRT 算法进行优化。目标偏置策略的数学表达式为

$$\mathbf{X}_{\text{new}} = \begin{cases} \mathbf{x}_{\text{goal}}, & r \leq Q \\ \mathbf{x}_{\text{sample}}, & r > Q \end{cases} \quad (1)$$

式中: \mathbf{X}_{new} 为新采样点的位置; \mathbf{x}_{goal} 表示目标点的位置; $\mathbf{x}_{\text{sample}}$ 表示随机生成采样点的位置; r 表示随机数; Q 为目标偏置阈值。

在搜索树算法利用随机数生成采样点的过程中,设置一个目标偏置阈值 Q ,以增加采样点接近目标区域的概率。当生成采样点时,先生成一个 $0 \sim 1$ 之间的随机数 r 。如果 r 的值不大于 Q ,选择目标点 \mathbf{x}_{goal} 作为采样点 \mathbf{X}_{new} 的优先扩展方向,以快速引导搜索树向目标区域扩展;如果 r 的值大于 Q ,通过传统方式生成一个新随机点 $\mathbf{x}_{\text{sample}}$ 作为新采样点 \mathbf{X}_{new} ,以保证搜索空间的全面性,避免过度集中在目标区域附近。

通过目标偏置策略可达到目标区域和整个工作空间探索的有效平衡,提高路径规划的效率。

2.2 约束点引导策略

RRT 算法采用随机生成采样点的方式进行搜索,可能导致搜索时间过长且效率低下,本文引入约束点引导策略解决该问题,将约束点(通常为特定点或目标点)用作参考点,使算法在扩展树时优先向这些点附近进行探索。节点搜索的数学表达式为

$$\mathbf{X}_{\text{new}} = \mathbf{x} + h(t) \times \frac{(\mathbf{t}_{\text{goal}} - \mathbf{x})}{\|\mathbf{t}_{\text{goal}} - \mathbf{x}\|} \quad (2)$$

式中: \mathbf{x} 表示当前节点的位置; $h(t)$ 为步长,其中 t 表示迭代次数; \mathbf{t}_{goal} 为约束点(本文选取目标点)的位置。

通过约束点引导策略,RRT 算法在扩展树时优先向目标点附近进行探索,从而更快到达目标点,显著减少因随机采样带来的盲目搜索,同时提高算法在复杂环境中进行路径规划的效率。

2.3 基于 Dijkstra 算法的冗余点移除

RRT 算法生成的路径不仅较为曲折,而且会因为冗余节点的存在产生大量的拐角,机械臂沿规划路径行进时会加剧其关节减速器、轴承等机械结构的磨损,且其行程延长、能耗与作业时间增加。

为消除冗余节点、减少路径拐角,本文采用基于 Dijkstra 算法^[13-14]的冗余点移除策略。Dijkstra 算法的基本原理为:创建一个距离数组用于存储所有节点到起点的最短距离,默认值为无穷大,起点的距离设为 0,使用优先队列(最小堆)管理待处理节点;从优先队列中取出距离最小的节点,遍历其所有相邻节点,计算每个相邻节点的实时距离;若当前节点到达该相邻节点的路径距离比该相邻节点当前记录在数组中的最短距离更短,则更新相邻节点的最短距离记录,并将其加入队列;当目标节点被处理或队列为空时结束,输出各节点的距离,目标节点在距离数组中对应的数值即为起点到目标节点的最短距离。

冗余点移除步骤如下。

1) 在算法生成初始路径 L 后,对冗余点进行第一次剪裁。将初始路径分解为一系列离散的节点集合 $\mathbf{P} = \{P_1, P_2, \dots, P_N\}$,对于任意两节点 P_a 和 P_b (其中 $a, b = 1, 2, \dots, N$,且 $a \neq b$),若两点之间的路径不与障碍物发生碰撞,则认为可以形成一条有效的路径,并计算该段路径的运动代价,将这些信息构建为一个带权有向图。通过 Dijkstra 算法在该图中找到从起点到终点的最优路径,得到第一次剪裁后的优化路径 L_1 。

2) 第一次剪裁后的新路径仍然可能保留了初始路径 L 中部分冗余点,为进一步提升路径质量,需要对路径 L_1 进行插值处理。在相邻节点之间以固定间隔(如 0.1 m)进行取样操作,生成节点更为密集的路径 L_2 。针对路径 L_2 ,再次采用 Dijkstra 算法对冗余点进行第二次剪裁,以有效消除路径中多余的节点,进一步降低路径的整体运动代价,使轨迹更加平滑和经济。

通过上述两阶段的优化处理,可有效去除算法生成路径中的大量冗余点,使最终路径达到更优的状态。

2.4 动态步长策略

为减少路径规划所需时间,增强算法的探索能力,根据当前环境状态自主地调整算法的步长,即采用动态步长策略。步长 $h(t)$ 计算公式为

$$h(t) = h_{\min} + (h_{\max} - h_{\min}) \times f \quad (3)$$

式中: h_{\min} 为最小步长; h_{\max} 为最大步长; f 为权重因子。权重因子 f 的具体计算式为

$$f = \omega_1 \times g_1 + \omega_2 \times g_2 \quad (4)$$

式中: ω_1 和 ω_2 为权重系数,用于平衡障碍物密度和目标位置可达性对步长的影响, $\omega_1 + \omega_2 = 1$; g_1 和 g_2 为障碍物密度与目标位置可达性的评估系数。

为确保路径的优化和收敛性,权重因子 f 按以下公式更新。

$$f_{\text{new}} = a \times f_{\text{old}} + (1 - a) \times f_{\text{current}} \quad (5)$$

式中: f_{new} 为更新后的权重因子; a 为遗忘因子,用于控制权重的衰减速度; f_{old} 为旧的权重因子; f_{current} 为当前环境权重因子。

综上,在路径生成过程中节点位置 X_{new} 的计算公式为

$$X_{\text{new}} = x + [h_{\min} + (h_{\max} - h_{\min}) \times f_{\text{new}}] \times \frac{(t_{\text{goal}} - x)}{\|t_{\text{goal}} - x\|} \quad (6)$$

通过动态调整步长,可使算法在复杂环境中生成高质量的路径。

2.5 基于三次B样条曲线的路径优化

虽然RRT算法具有一定的探索能力,但其生成的路径通常由一系列直线组成,路径平滑性较差,难以直接应用于机械臂的实际运动。为此,本文采用三次B样条曲线^[15]对RRT算法初步生成的路径进行平滑处理。三次B样条曲线具有局部可控性和连续性,能够在不改变路径整体趋势的前提下生成光滑且连续的路径。

B样条曲线的数学表达基于控制点和基函数的组合。给定一组控制点 P_0, P_1, \dots, P_n , B样条曲线 $C(u)$ 可表示为

$$C(u) = \sum_{i=0}^n N_{i,k}(u) P_i \quad (7)$$

式中: u 为参数变量,取值范围为 $[0, 1]$; $N_{i,k}(u)$ 为B样条基函数,其中 k 为样条的阶数,本文中 $k = 3$ 。

基函数通过递归定义, $k = 0$ 时表示为

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

若 $k \geq 1$,则表示为

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

式中 u_i 为节点向量 U 中第 i 个节点值,节点向量需满足非递减性, $U = \{u_0, u_1, \dots, u_m\}$,节点数量 $m = n + k + 1$ 。

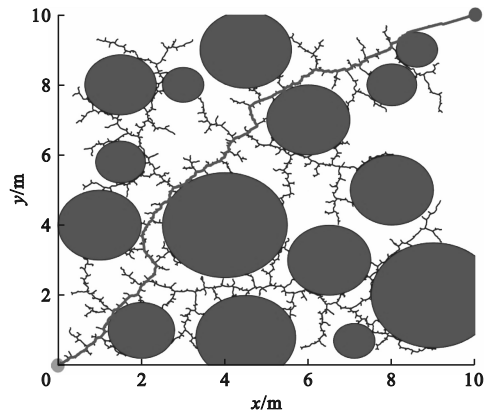
通过将三次B样条插值策略引入改进RRT算法,在保持探索能力的同时,还可以生成更加平滑且适合机械臂运动的路径,不仅提高了路径的质量,还增加了算法的实用性。

3 仿真实验

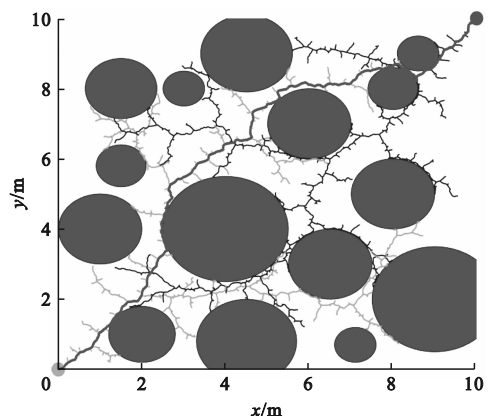
通过仿真实验验证本文改进算法的性能。硬件配置为: Intel(R) i5-12500H CPU, 16.00 GB 内存。使用的仿真软件为 Matlab 与 Gazebo, 分别在二维环境、三维环境与 Gazebo 环境中进行仿真。

3.1 二维空间仿真

设置 $10 \text{ m} \times 10 \text{ m}$ 二维工作空间,起始点为 $(0, 0)$,目标点为 $(10, 10)$,本文TR-RRT算法的目标偏置阈值 Q 设置为 0.3 。分别采用RRT算法、RRT-Connect算法、Informed-RRT*算法、TR-RRT算法进行仿真实验,结果如图1所示。图中圆形图案代表障碍物,细实线表示各算法运行过程中搜索树的扩展过程,连接起始点和目标点的粗实线为各算法规划的最优路径。



(a) RRT算法



(b) RRT-Connect算法

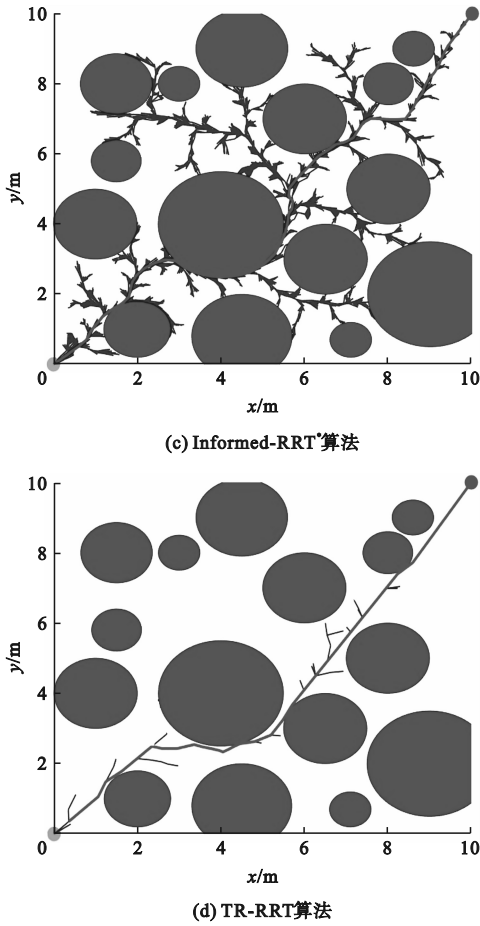


图 1 不同算法在二维空间的仿真结果

Fig. 1 Simulation results of different algorithms in 2D space

由图 1 可以看出, RRT 算法、RRT-Connect 算法、Informed-RRT* 算法规划出的路径不仅有较多曲折, 且路径较长, 而本文提出 TR-RRT 算法规划的路径更加平滑、更短。进行 50 次重复实验, 记录各算法的规划时间、节点数目和路径长度, 并取平均值, 结果如表 1 所示。

表 1 不同算法在二维空间仿真的平均规划时间、平均路径长度及平均节点数量

Table 1 Average planning time, path length and node number of different algorithms in 2D space

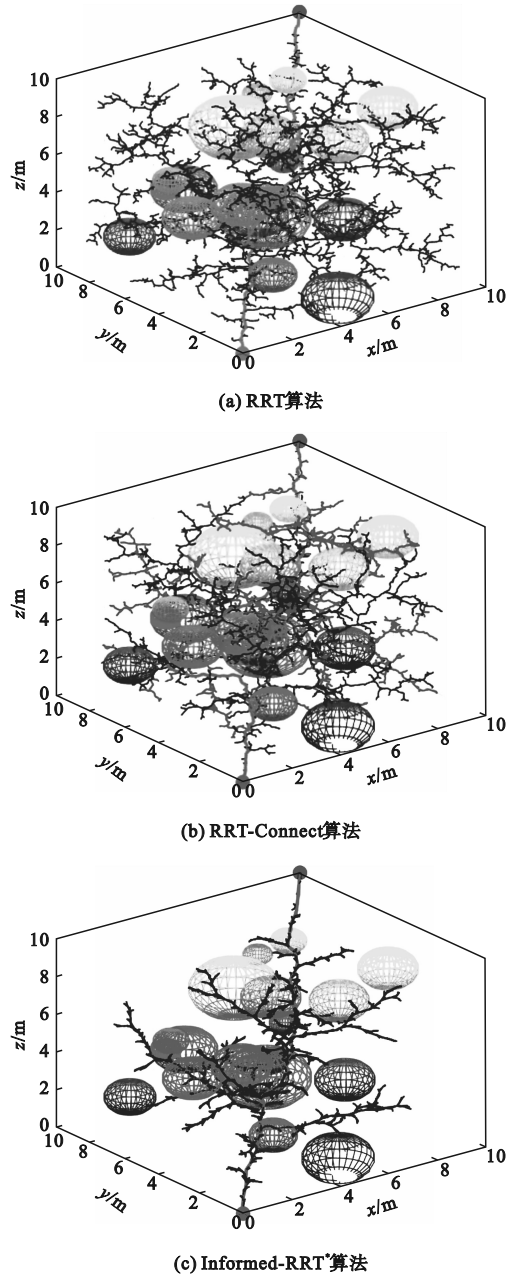
算法	平均规划时间/s	平均路径长度/m	平均节点数量/个
RRT	9.76	19.37	2 151
RRT-Connect	6.52	21.74	2 843
Informed-RRT*	3.41	16.89	1 953
TR-RRT	2.83	14.91	1 083

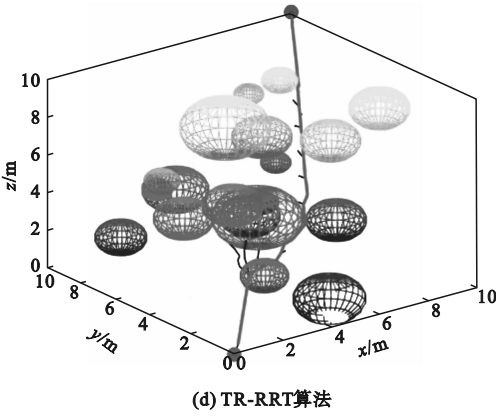
由表 1 可知, 与 RRT 算法、RRT-Connect 算法、Informed-RRT* 算法相比, 本文提出的 TR-RRT 算法在二维环境地图上规划路径所需时间分

别减少了 71.00%、56.60%、17.01%, 节点数量分别减少了 49.65%、61.90%、44.54%, 路径长度分别减少了 23.02%、31.41%、11.72%, 表现出明显的优势。

3.2 三维空间仿真

设置 10 m × 10 m × 10 m 的三维工作空间, 起始点为 (0, 0, 0), 目标点为 (10, 10, 10), TR-RRT 算法的目标偏置阈值 Q 仍设置为 0.3。分别采用 RRT 算法、RRT-Connect 算法、Informed-RRT* 算法、TR-RRT 算法进行仿真实验, 结果如图 2 所示, 图中圆球代表障碍物, 细实线与粗实线的含义与图 1 相同。可见, TR-RRT 算法的路径长度明显短于三个对比算法且更加平滑、分支更少。





(d) TR-RRT算法

图2 不同算法在三维空间的仿真结果

Fig.2 Simulation results of different algorithms in 3D space

进行50次重复实验,记录各算法的规划时间、节点数目和路径长度,并取平均值,结果如表2所示。

表2 不同算法在三维空间仿真的平均规划时间、平均路径长度及平均节点数量

Table 2 Average planning time, path length and node number of different algorithms in 3D space

算法	平均规划时间/s	平均路径长度/m	平均节点数量/个
RRT	13.71	26.82	5 164
RRT-Connect	9.65	28.73	7 041
Informed-RRT*	5.13	19.47	3 076
TR-RRT	4.22	18.46	1 984

由表2可知,与RRT算法、RRT-Connect算法、Informed-RRT*算法相比,本文提出TR-RRT算法在三维环境地图上规划路径所需时间分别减少了69.22%、56.27%、17.73%,节点数量分别减少了61.58%、71.82%、35.53%,路径长度分别减少了31.17%、35.75%、5.19%。TR-RRT算法在三维地图环境下规划路径的长度、规划时间和节点数量均大幅减少,生成的避障路径也更加顺畅。

3.3 Gazebo 环境仿真

为验证本文提出算法在机械臂上实际应用的效率及稳定性,将UR5机械臂作为实验对象,在Gazebo环境中应用四种算法进行仿真实验,UR5机械臂3D模型如图3所示,图中Joint1~Joint6表示机械臂各关节。首先将图3中的UR5模型导入SolidWork软件,并使用插件将其转化为机器人专用的urdf描述文件,然后将其与表3所示数据(UR5机械臂DH参数)一起导入机器人控制系统(ROS)的moveit-setup-assistant功能包,并转化为ROS专用的moveit(ROS的控制核心)配置包。

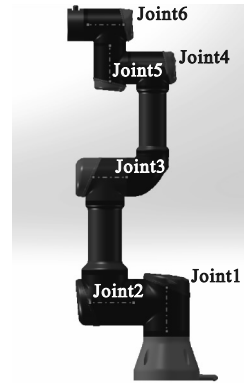


图3 UR5机械臂3D模型

Fig.3 3D model of UR5 robotic arm

表3 UR5机械臂DH参数

Table 3 DH parameters of UR5 robot arm

关节	关节角度/rad	连杆长度/m	连杆偏移/m	扭转角/rad
Joint1	0	0	0.089 15	$\pi/2$
Joint2	$-\pi/2$	-0.425	0	0
Joint3	0	-0.392 25	0	0
Joint4	$-\pi/2$	0	0.109 15	$\pi/2$
Joint5	2π	0	0.094 65	$-\pi/2$
Joint6	0	0	0.082 37	0

通过ROS框架将算法集成至配置包,并在ROS提供的Gazebo仿真空间中进行仿真实验,仿真环境设置如图4所示,图中长方体为障碍物。

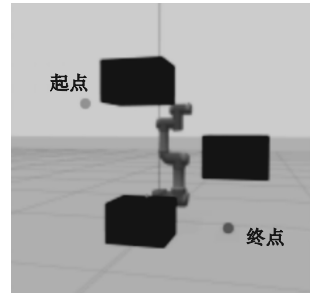


图4 Gazebo仿真环境设置

Fig.4 Gazebo simulation environment settings

将四种算法在Gazebo环境下分别进行20次仿真实验,计算平均规划时间,结果如表4所示。

表4 不同算法在Gazebo环境下仿真的平均规划时间
Table 4 Average planning time of different algorithms in Gazebo environment

算法	平均规划时间/s
RRT	33.36
RRT-Connect	24.97
Informed-RRT*	15.47
TR-RRT	13.32

由表4可知,与RRT算法、RRT-Connect算法、Informed-RRT*算法相比,本文提出TR-RRT

算法在 Gazebo 环境中规划路径所需时间分别减少了 60.08%、46.66%、13.59%。

4 结论

针对 RRT 及其衍生算法在复杂环境中存在规划效率低、路径曲折等问题,提出了一种基于约束点引导的目标约束 RRT 算法。通过约束点引导策略优化了搜索方向,使算法在复杂环境中能够更快找到可行路径;通过目标偏置策略进一步引导搜索树向目标区域扩展,减少了不必要的探索和冗余计算;引入动态步长策略,根据环境复杂度实时调整步长,平衡了探索与扩展能力,提升了路径规划的效率和稳定性;通过基于 Dijkstra 算法的冗余点移除策略减少了冗余点,利用三次 B 样条插值优化了路径,进一步提高了路径的质量。

仿真实验结果表明,与 RRT、RRT-Connect 以及 Informed-RRT* 算法相比,本文提出的 TR-RRT 算法在二维、三维以及 Gazebo 仿真环境中均有显著的性能提升,在规划时间、路径长度和节点数量等方面均具有明显优势。

参考文献(References):

- [1] KUMAR A, TIWARI S, MAJUMDER A. A* -VG algorithm: a hybrid algorithm for the path planning of inspection robots [J]. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 2023, 45(7): 386.
- [2] ZHANG W, WANG N X, WU W H. A hybrid path planning algorithm considering AUV dynamic constraints based on improved A* algorithm and APF algorithm [J]. Ocean Engineering, 2023, 285: 115333.
- [3] LI B L, TAN C Y, LIAN Y F, et al. Mobile robot global planning based on improved A* algorithm path planning research [C]//Proceedings of the 2023 International Conference on Advances in Artificial Intelligence and Applications. Wuhan, China: ACM, 2023: 305 - 311.

- [4] LIANG Y M, ZHAO H Y. CCPF-RRT*: an improved path planning algorithm with consideration of congestion [J]. Expert Systems with Applications, 2023, 228: 120403.
- [5] GANESAN S, RAMALINGAM B, MOHAN R E. A hybrid sampling-based RRT* path planning algorithm for autonomous mobile robot navigation [J]. Expert Systems with Applications, 2024, 258: 125206.
- [6] ZHANG X Q, WANG P Y, GUO Y R, et al. Path planning algorithm for manipulators in complex scenes based on improved RRT [J]. Sensors, 2025, 25(2): 328.
- [7] ZHOU X Y, LUO X D, ZHANG Y. A path planning algorithm based on improved RRT [C]//2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC). Chongqing, China: IEEE, 2022: 1135 - 1140.
- [8] LI Z Q, LI L B, ZHANG W J, et al. Research on unmanned ship path planning based on RRT algorithm [J]. Journal of Physics: Conference Series, 2022, 2281(1): 012004.
- [9] ZHANG X, ZHU T, DU L, et al. Local path planning of autonomous vehicle based on an improved heuristic Bi-RRT algorithm in dynamic obstacle avoidance environment [J]. Sensors, 2022, 22(20): 7968.
- [10] 姚凯文, 周锋, 李楠, 等. 基于改进 Informed-RRT* 的路径规划算法 [J]. 软件导刊, 2024, 23(7): 80 - 86.
- [11] YAO K W, ZHOU F, LI N, et al. Improved Informed-RRT* based path planning algorithm [J]. Software Guide, 2024, 23(7): 80 - 86. (in Chinese)
- [12] LI J, HUANG C W, PAN M Q. Path-planning algorithms for self-driving vehicles based on improved RRT-Connect [J]. Transportation Safety and Environment, 2023, 5(3): tda061.
- [13] CANDEMIR D, KÖK İ, SUAT Ö. Narrowed regions-based bidirectional path planning using RRT-Connect for single aircraft missions [J]. Procedia Computer Science, 2024, 231: 703 - 708.
- [14] MARISTANY DE LAS CASAS P, KRAUS L, SEDEÑO-NODA A, et al. Targeted multiobjective Dijkstra algorithm [J]. Networks, 2023, 82(3): 277 - 298.
- [15] LI X Y. Path planning of intelligent mobile robot based on Dijkstra algorithm [J]. Journal of Physics: Conference Series, 2021, 2083(4): 042034.
- [16] HE S T, SHEN L Y, WU Q, et al. A certified cubic B-spline interpolation method with tangential direction constraints [J]. Journal of Systems Science and Complexity, 2024, 37(3): 1271 - 1294.

(责任编辑:宋颖韬)

(上接第 16 页)

- [14] CAMPOS C, ELVIRA R, RODRÍGUEZ J J G, et al. ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM [J]. IEEE Transactions on Robotics, 2021, 37(6): 1874 - 1890.
- [15] 杨永刚, 武楚健, 杨正全. 基于融合改进 RANSAC 光流法的无人机视觉 SLAM 研究 [J]. 半导体光电, 2023, 44(2): 277 - 283.
- [16] YANG Y G, WU C J, YANG Z Q. Research on UAV visual SLAM based on fusing improved RANSAC optical flow method [J]. Semiconductor Optoelectronics, 2023, 44(2): 277 - 283. (in Chinese)
- [17] 黄友锐, 王照锋, 韩涛, 等. 结合轻量化 YOLOv5s 的动态视觉 SLAM 算法 [J]. 电子测量技术, 2024, 47(11): 59 - 68.
- [18] HUANG Y R, WANG Z F, HAN T, et al. Dynamic visual SLAM algorithm combined with lightweight YOLOv5s [J].

Electronic Measurement Technology, 2024, 47(11): 59 - 68. (in Chinese)

- [17] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems [C]//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vilamoura-Algarve, Portugal: IEEE, 2012: 573 - 580.
- [18] 刘砚菊, 晏佳华, 冯迎宾. 动态场景下基于视觉的 SLAM 技术研究 [J]. 半导体光电, 2024, 45(2): 327 - 335.
- [19] LIU Y J, YAN J H, FENG Y B. SLAM based on deep learning and optical flow constraints in dynamic scenes [J]. Semiconductor Optoelectronics, 2024, 45(2): 327 - 335. (in Chinese)
- [20] LIU Y B, MIURA J. RDS-SLAM: real-time dynamic SLAM using semantic segmentation methods [J]. IEEE Access, 2021, 9: 23772 - 23785.

(责任编辑:和晓军)