

复杂环境下的改进 RRT 算法路径规划^{*}

谢春圆¹ 王欣¹ 吴迪² 王殿龙¹

(1. 大连理工大学机械工程学院 大连 116024; 2. 大连理工大学计算机科学与技术学院 大连 116024)

摘 要:针对快速扩展随机树算法(rapidly-exploring trees, RRT)在一些复杂环境中存在搜索效率低、收敛速度慢、生成的路径冗余节点多等问题,提出一种改进的 RRT 算法。首先引入自适应目标概率策略,实时调整对目标点的采样概率;其次引入节点转向策略,提高单次采样的成功率;最后对生成的路径进行冗余节点裁剪,使路径更符合实际应用需求。在 MATLAB 中进行仿真实验,并与 RRT 算法、RRTGoalBias 算法进行对比。实验结果表明,改进算法在多种不同环境下具有较好的适应性,在寻路时间、采样次数和采样成功率 3 个方面均有较大提升,最终平均路径长度降低了 21.1%,平均节点数降低了 75.3%,证明了改进算法的优越性和实用性。

关键词:复杂环境;路径规划;快速扩展随机树;自适应性;节点转向

中图分类号: TP242 **文献标识码:** A **国家标准学科分类代码:** 510.8050

Improved RRT algorithm path planning in complex environment

Xie Chunyuan¹ Wang Xin¹ Wu Di² Wang Dianlong¹

(1. School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China;

2. School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China)

Abstract: Aiming at the problems of rapidly exploring trees (RRT) in some complex environments, such as low search efficiency, slow convergence rate and many redundant nodes, an improved RRT algorithm is proposed. Firstly, the adaptive target probability strategy is introduced to adjust the sampling probability of the target point in real time. Secondly, node turning strategy is introduced to improve the success rate of single sampling. Finally, redundant nodes are clipped to the generated path to make the path more in line with the actual application requirements. Simulation experiments are carried out in MATLAB and compared with RRT algorithm and RRTGoalBias algorithm. The experimental results show that the improved algorithm has good adaptability in a variety of different environments, and the running time, sampling times and sampling success rate are greatly improved. The final average path length is reduced by 21.1%, and the average node number is reduced by 75.3%, which proves the superiority and practicability of the improved algorithm.

Keywords: complex environment; path planning; fast-exploring random tree; self-adaptability; node steering

0 引言

路径规划的目的是在规定范围的区域内找到一条从起点到终点的无碰撞安全路径^[1]。常用的路径规划算法有迪杰斯特拉算法^[2]、A* 算法^[3]、跳点算法^[4]、人工势场法^[5]、蚁群算法^[6]、快速扩展随机树(rapidly-exploring trees, RRT)算法^[7]等。其中 RRT 算法由于地图无需栅格

化、适应性强、具有概率完备性等优点,在各个领域都得到了广泛的应用^[8]。但是由于采样的随机性,RRT 算法存在盲目搜索、搜索时间长、路径转折多、难以通过狭窄区域等问题。

针对 RRT 算法存在的问题,国内外许多学者对其进行了改进。文献[9]提出了 RRT-Connect 算法,该算法同时从起点和终点进行扩展,加快了收敛速度,但在复杂的

收稿日期:2023-08-16

^{*} 基金项目:中央高校基本科研业务费(DUT22LAB507)资助

环境下效果不佳。文献[10]提出了 RRT* 算法,引入了父节点重选与区域内节点重连的策略,使所规划出的路径为渐进最优,但收敛时间却大大增加。文献[11]提出了 informed-RRT* 算法,通过建立椭圆来限制随机扩展树的采样区域,从而加快算法的收敛速度,但在起始点与目标点相距较远的大规模地图中效果不明显。文献[12]提出了引入目标偏执和自适应步长策略的改进 RRT-Connect 算法,优化了寻路时间和路径长度,但其目标偏置策略的参数不能实时调整,且仅适用于果树修剪机械手,具有一定的局限性。文献[13]提出一种改进 RRT 算法,引入自适应目标偏向策略和区域采样调整策略,提高了算法在多障碍环境中的寻路效率。文献[14]提出了一种改进 RRT-Connect 算法,引入了目标偏置和最近邻节点选择优化机制,缩短了路径长度,并进行了路径平滑处理。文献[15]提出了一种改进 BI-RRT 算法,引入可变权重系数实现目标导向,并对路径进行剪枝优化和平滑处理,但其对可变权重系数的选取缺少详细的策略。文献[16]提出了 GS-RRT* 算法,通过在目标附近取更多的样本,解决了 RRT* 算法收敛缓慢的问题,但其对复杂环境的适应性较差。文献[17]提出一种改进 RRT*-DR 算法,引入半目标导向实现快速搜索,随后采用动态区域采样和变步长机制提高算法效率,但该算法中目标导向参数为提前设置的定值,并不能根据地图情况动态调节。文献[18]通过实验统计分析了步长和目标采样概率对算法性能的影响,提出了参数选择原则和冗余节点删除策略,提高了算法的运行效率,但其参数仍然为提前设置的定值,并不能根据环境进行改变。

很多改进的算法都引入了与目标点相关的导向策略,但对导向参数的选取通常为一个定值或粗略调整,缺乏较为精确的调整策略。本文提出自适应目标概率,可以根据算法运行过程中的信息对目标采样概率值进行实时调整;针对 RRT 算法在多障碍、狭长通道和迷宫等复杂环境中存在的搜索效率低、搜索时间长、采样成功率低等问题,提出节点转向策略,即在障碍物碰撞检测失败的新节点附近引入多个备选新节点,提高对障碍物附近的探索性和通过狭长通道的概率;最后针对 RRT 算法生成的路径转折过多的问题,提出冗余节点裁剪策略,使生成的路径更符合实际应用的需求。

1 RRT 算法原理

常规 RRT 算法的搜索过程类似于一棵树不断生长的过程,如图 1 所示,其搜索步骤如下。

- 1) 初始化扩展树 T 、步长 Δ 、起始点 X_{init} 和目标点 X_{goal} 等参数,将起始点 X_{init} 加入到扩展树中,作为树的根节点。
- 2) 在机器人的工作空间内随机采样一个点 X_{rand} ,并在扩展树 T 中寻找距离 X_{rand} 最近的节点 X_{near} 。
- 3) 按照预先设定的步长 Δ 将邻近节点 X_{near} 向随

机采样点 X_{rand} 的方向进行偏移,得到新节点 X_{new} 。

4) 对新节点 X_{new} 与邻近节点 X_{near} 间的连线进行障碍物碰撞检测,若检测失败到步骤 5),检测成功到步骤 6)。

5) 舍弃当前生成的新节点,当前循环采样失败,转到步骤 2),进行下一轮采样。

6) 将新节点 X_{new} 添加到扩展树中,并将邻近节点 X_{near} 作为新节点的父节点。

7) 判断新节点 X_{new} 是否到达目标点 X_{goal} 处,若未到达,转到步骤 2),进行下一轮采样,若已到达,则路径规划成功,寻路结束。

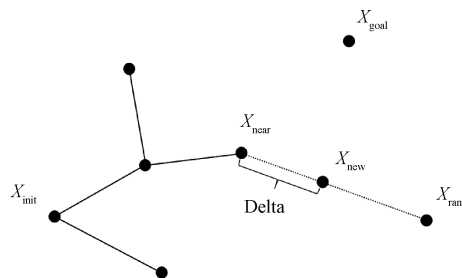


图 1 RRT 算法扩展过程

常规 RRT 算法无需对地图栅格化且具有概率完备性,适应性较强。但在上述的运行步骤中可以看出,采样是完全随机的,因此扩展过程比较盲目,会产生很多低质量扩展浪费时间;当新节点障碍物碰撞检测失败后被直接舍弃,这在障碍物占比较大的环境中,采样成功率会明显降低,从而导致扩展速度变慢;由于 RRT 算法并没有对路径进行优化的步骤,因此最终生成的路径会存在较多的非必要转折点。

2 改进 RRT 算法

针对 RRT 算法存在的问题,本文提出自适应目标概率策略、节点转向策略和冗余节点裁剪策略。

2.1 自适应目标概率

针对 RRT 算法盲目搜索、收敛速度慢的缺点,基于 RRTGoalBias 算法,提出自适应目标概率策略,可以根据地图环境对目标采样概率进行实时调整。

1) RRTGoalBias 算法

RRTGoalBias 算法是由 Lavalley 教授提出的,该算法对 RRT 算法的采样策略进行改进,即以一定的概率把目标点 X_{goal} 作为随机采样点 X_{rand} 。RRTGoalBias 算法的整体流程与常规 RRT 算法基本一致,只是在采样阶段有所不同。RRTGoalBias 算法的采样步骤如下:预先设定一个概率值 k ,在进行采样前先由计算机生成一个 $(0,1)$ 内的随机数,若该随机数大于设定的概率值 k ,则在工作空间内进行随机采样;若小于 k ,则直接将目标点 X_{goal} 作为本轮扩展的采样点 X_{rand} 。这样对扩展树的生长方向就有了引导性,能够在一定程度上减少搜索时间,加快算法的收敛。

2) 自适应目标概率

RRTGoalBias 算法中目标点的采样概率设置为一个固定值 k , 这在简单的环境下对路径规划的速度会有一定的提升。但在较为复杂的环境中, 往往需要根据经验来设定对目标点的采样概率。每次切换环境都需要调整目标概率, 这是非常不方便的, 并且在复杂的环境中, 目标概率设置不合理可能会增加寻路时间甚至导致无解的情况。

本文引入随机采样成功率 p_r 的概念:

$$p_r = \frac{rs_suc}{rs} \quad (1)$$

式中: rs 为算法运行过程中随机采样的次数; rs_suc 为随机采样成功的次数, 即障碍物碰撞检测成功并生成新节点的采样。随机采样成功率可以在一定程度上表明当前环境的复杂程度, 环境中障碍物占比越多, 随机采样成功率就越低; 反之环境中障碍物占比越少, 随机采样成功率就越高。而不同的环境通常会对应一个运行时间最短的目标概率值, 因此通过随机采样成功率来将目标概率调节为最适合当前环境的值, 可以达到适应不同环境的目的。但随机采样成功率与最适合的目标概率之间并没有一个明确的函数关系, 如果盲目地建立两者的联系, 结果可能会适得其反, 因此本文采用实验收集数据并进行函数拟合的方式来建立两者之间的关系。

实验共设计 50 张不同的地图, 尺寸为 800×800 , 地图的复杂程度各不相同, 使采样成功率尽量均匀地分布在 $(0, 1)$ 。

首先将每张地图在常规 RRT 算法下均运行 500 遍, 相应的参数设置如表 1 所示。

表 1 RRT 算法参数设置

参数	值
起点	(50, 50)
终点	(750, 750)
步长	50
最大迭代次数	3 000

对这 500 遍运行下的随机采样成功率进行记录, 并取平均值 $\bar{p}_r$ 为:

$$\bar{p}_r = \frac{\sum_{i=1}^{500} p_{ri}}{500} \quad (2)$$

这样每张地图就有了一个对应的随机采样成功率。将 RRTGoalBias 算法中的目标概率从 $0 \sim 0.95$, 间隔 0.05 取值, 共 20 个概率。每张地图在这 20 个概率值下各运行 500 遍, 记录每遍的运行时间并计算每个概率值下的平均运行时间, 筛选出每个地图下平均运行时间最短的目标概率值 $k_{\min}$ 。将每张地图的随机采样成功率与最适合的目标概率进行对应, 得到实验结果如图 2 所示。由图 2 可以看出, 两者在整体上成正相关, 地图的随机采样成功率越高即地图越简单, 对应的最适合目标概率就越大; 地图的

随机采样成功率越低即地图越复杂, 对应的最适合目标概率就越小。

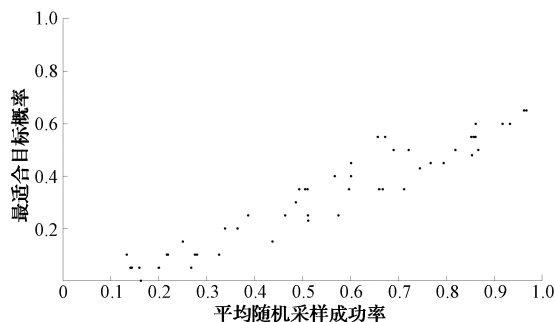
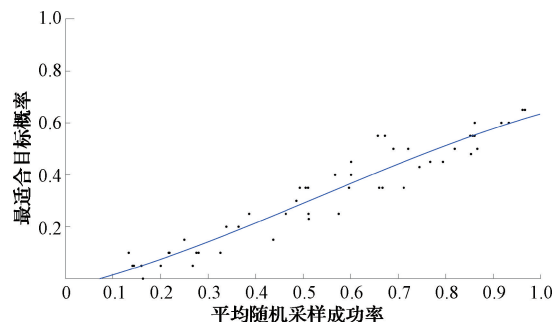


图 2 实验结果数据

利用 MATLAB 软件中的曲线拟合工具箱对上述实验数据进行拟合, 对比多种不同形式的函数, 最终确定拟合形式为三次多项式, 拟合曲线与结果如图 3 所示, 拟合得多项式为:

$$k = -0.38p_r^3 + 0.6p_r^2 + 0.44p_r - 0.035 \quad (3)$$

拟合的确定系数 $R^2 = 0.9169$, 说明曲线的拟合程度很高, 可以较好地表明随机采样成功率与最适合目标概率之间的关系。



(a) 拟合曲线

线性模型 Poly3:
 $f(x) = p1 \cdot x^3 + p2 \cdot x^2 + p3 \cdot x + p4$
 系数(置信边界为 95%):
 p1 = -0.3818 (-1.77, 1.007)
 p2 = 0.611 (-1.682, 2.904)
 p3 = 0.4398 (-0.6876, 1.567)
 p4 = -0.03463 (-0.1911, 0.1218)

拟合优度:
 SSE: 0.1515
 R 方: 0.9169
 调整 R 方: 0.9115
 RMSE: 0.05739

(b) 拟合结果

图 3 拟合曲线与结果

为了对随机采样成功率与最适合的目标概率之间的拟合关系进行验证, 新设计 6 张地图 (图 4), 经运算, 获得 6 张地图的随机采样成功率, 依次为 94%、76%、56%、36%、20% 和 11%, 如表 2 所示, 可以看出, 地图环境越简单, 障碍物数量和占比越少, 随机采样成功率越高。

同理, 应用 RRTGoalBias 算法运算 6 张地图, 获得各地图运行时间最短的目标概率值, 并与随机采样成功率对应 (表 2)。

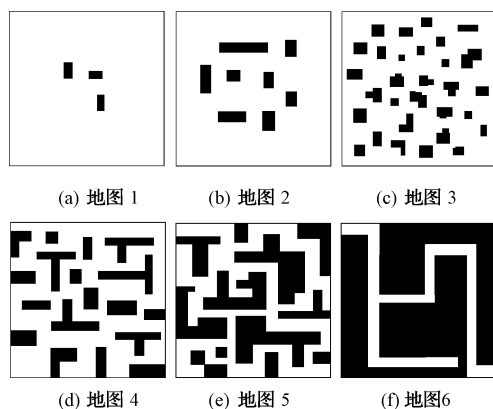


图4 不同随机采样成功率的地图

表2 各地图的采样成功率与目标概率

地图序号	随机采样成功率/%	最适合的目标概率
1	94	0.6
2	76	0.55
3	56	0.3
4	36	0.2
5	20	0.1
6	11	0.05

将以上6组数据与所拟合的函数曲线进行对比,如图5所示。可以看出6组数据均位于拟合曲线的附近,说明所拟合的函数在一定程度上能够正确地表明随机采样成功率与最适合目标概率之间的关系。

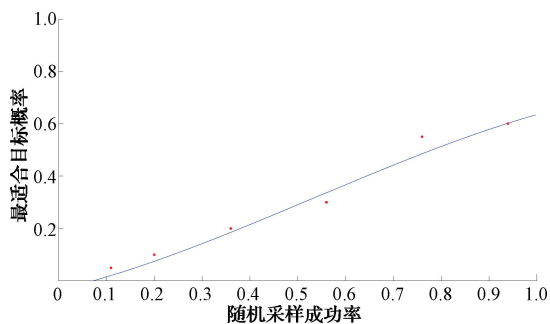


图5 数据验证

因此,根据采样成功率即可自适应的选取合适的目标概率。改进算法中随机采样成功率的计算为:

$$p_r = \frac{iter - gs - rs_f}{iter - gs} \quad (4)$$

式中: $iter$ 是当前迭代次数即总采样次数; gs 为对目标点的采样次数; rs_f 为随机采样失败的次数。首次采样时, gs 、 rs_f 均为0,所以随机采样成功率为1,由式(3)可得首次对目标点的采样概率为0.625。每次迭代后对随机采样成功率进行更新,并计算当前的目标概率。随着迭代的进行,随机采样成功率将逐渐符合当前地图环境复杂度,对应的目标概率值也逐渐接近当前地图的最适合目标

概率。

对目标点进行概率采样所需要面临的一个问题是当目标点被障碍物遮挡时,如果对目标点的采样概率偏高,就会导致在遮挡障碍物附近产生大量无效采样甚至出现无解的情况,路径规划效果反而不如常规RRT算法。因此改进算法中引入目标采样成功率,其计算方式如下:

$$suc_rate = \frac{gs - gs_f}{gs} \quad (5)$$

式中: suc_rate 为目标采样成功率; gs 为对目标点的采样次数; gs_f 为对目标点采样失败的次数。当目标采样成功率低于一定数值时,对目标概率进行削弱,以保证对未达空间的探索,避免陷入无解的情况。综合以上内容,最终自适应目标概率计算为:

$$k = \begin{cases} 0.2(-0.38p_r^3 + 0.6p_r^2 + 0.44p_r - 0.035), & suc_rate \leq 0.1 \\ 0.8(-0.38p_r^3 + 0.6p_r^2 + 0.44p_r - 0.035), & suc_rate \leq 0.5 \\ -0.38p_r^3 + 0.6p_r^2 + 0.44p_r - 0.035, & suc_rate > 0.5 \end{cases} \quad (6)$$

2.2 节点转向策略

自适应目标概率策略可以在一定程度上解决RRT算法盲目搜索、效率低下的问题,但在一些通道狭窄的复杂环境中,RRT算法会存在采样成功率低,难以通过的情况。为了增加采样成功率,加快通过狭窄区域的速度,提出节点转向策略,如图6(a)所示。当生成的新节点 X_{new} 障碍物碰撞检测失败时,并不直接舍弃该新节点,而是将新节点 X_{new} 绕 X_{near} 旋转一定角度生成备选新节点,继续进行碰撞检测,这样可以加强对障碍物附近的探索。共准备4个备选新节点,4个备选新节点的位置依次为原 X_{new} 绕 X_{near} 顺时针旋转 45° ,逆时针旋转 45° ,顺时针旋转 90° ,逆时针旋转 90° 。引入节点转向策略后,在狭窄通道和障碍物附近的采样成功率会得到有效提升,如图6(b)所示。

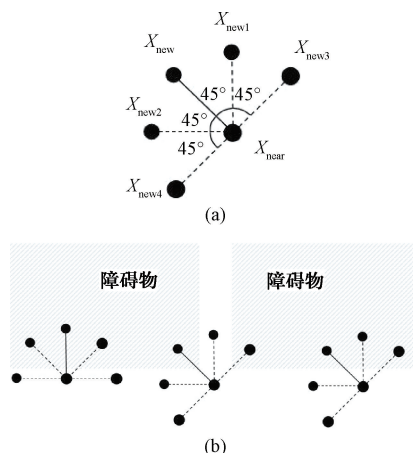


图6 节点转向策略

采用节点转向策略可能会导致在障碍物附近产生大量低质量节点,即多个父节点相同且相距很近的节点,因此需要对节点转向策略进行一定的约束。在扩展树的节点信息中添加节点转向策略可用次数这个变量,每个节点的初始值设置为 2。当一个节点被选为邻近节点并且生成的子节点采用了转向策略时,该邻近节点的转向策略可用次数减 1。当邻近节点的转向策略可用次数为 0 时,其子节点的障碍物碰撞检测失败后直接进入下一轮采样,不再启用节点转向策略。

引入节点转向策略后新节点生成流程如下。

1)对新节点 X_{new} 进行障碍物碰撞检测,若检测成功转到步骤 8),否则进行步骤 2)。

2) 判断被选中的邻近节点 X_{near} 的转向策略可用次数, 若为 0 转到步骤 7), 不为 0 则继续步骤 3)。

3)对 X_{newl} 进行障碍物碰撞检测,若检测成功转到步骤 8),否则进行步骤 4)。

4)对 $X_{\text{new}2}$ 进行障碍物碰撞检测,若检测成功转到步骤 8),否则进行步骤 5)。

5)对 $X_{\text{new}3}$ 进行障碍物碰撞检测,若检测成功转到步骤 8),否则进行步骤 6)。

6)对 $X_{\text{new}i}$ 进行障碍物碰撞检测,若检测成功转到步骤 8),否则进行步骤 7)。

7)本次扩展失败,舍弃当前新节点,更新参数,进入下一次扩展。

8)本次扩展成功,将新节点添加到搜索树中。

2.3 冗余节点裁剪

由于 RRT 算法中并不存在路径优化的步骤,搜索得到的路径通常具有较多转折点和冗余节点,不符合实际应用的需求,因此本文对生成的路径进行冗余节点裁剪,如图 7 所示。

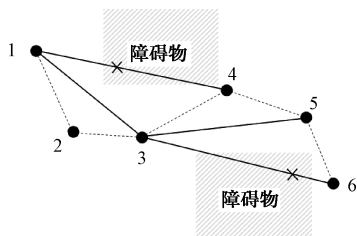


图 7 冗余节点裁剪

图 7 中,节点 1~6 为规划得到的初始路径,首先由节点 1 开始,依次与节点 2 之后的点进行障碍物碰撞检测,直到最后一个碰撞检测成功的节点,即节点 3,将节点 1 与节点 3 直接相连来替代路径段 1、2、3,再由节点 3 开始继续重复上述步骤直至整条路径迭代完毕。图 7 中的最终路径为 1、3、5、6,可以看出经过裁剪后的路径转折点有所减少,且由三角不等式可知裁剪后的路径代价在一定程度上也能降低。实际算法中的裁剪过程由目标点开始,到起

始点结束。

改进 RRT 算法的扩展过程如图 8 所示。

1) 初始化扩展树 T 以及起始点 X_{init} 、目标点 X_{goal} 、步长 Δ 等参数, 将起始点 X_{init} 作为根节点添加到扩展树 T 中。

2) 根据式(6)计算所得的目标采样概率进行采样(首次采样的目标概率为 0.625), 得到采样点 X_{rand} , 在扩展树中找到距离 X_{rand} 最近的节点 X_{near} 。

3)按照预先设定的步长 Δ 将邻近节点 X_{near} 向采样点 X_{rand} 偏移一个步长 Δ 得到新节点 X_{new} , 并对新节点进行障碍物碰撞检测, 若检测成功到步骤 7), 检测失败到步骤 4)。

4) 判断当前邻近节点 X_{near} 的节点转向策略可用次数, 若等于 0 到步骤 6), 若不等于 0 到步骤 5)。

5)依次对4个备选新节点进行障碍物碰撞检测,若有备选新节点检测成功到步骤7),全部检测失败到步骤6)。

6) 本轮扩展失败,更新参数,转到步骤 2) 进入下一轮扩展。

7) 本轮扩展成功, 将新节点添加到扩展树 T 中, 判断新节点是否到达目标点处, 若到达步骤 8), 否则转到步骤 2) 进入下一轮扩展。

8) 路径规划成功,对生成的初始路径进行冗余节点裁剪处理,得到最终路径。

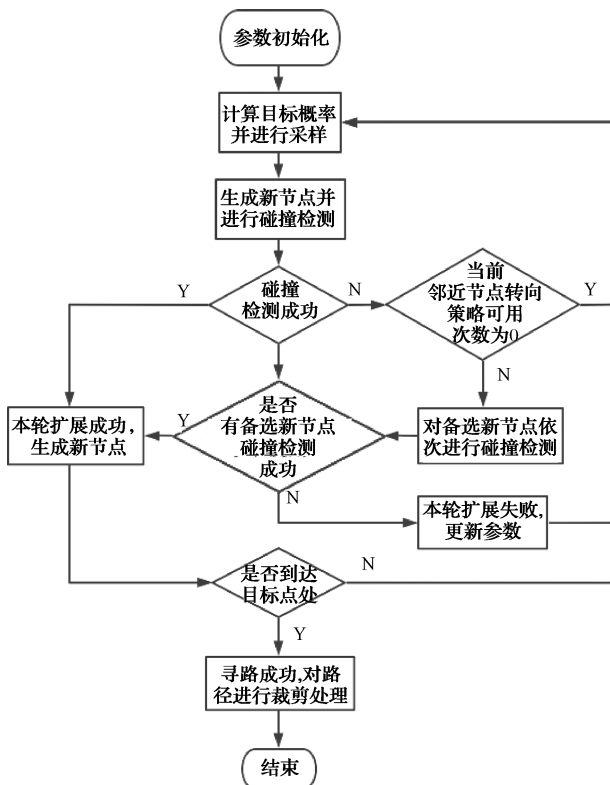


图 8 改进 RRT 算法流程

3 实验设计与分析

3.1 改进 RRT 算法

针对改进 RRT 算法,实验共设计 3 张地图来进行实验,并与常规 RRT 算法和 RRTGoalBias 算法进行结果对比。实验地图如图 9 所示,其中 map1 为普通环境, map2 为多障碍环境, map3 为迷宫环境。

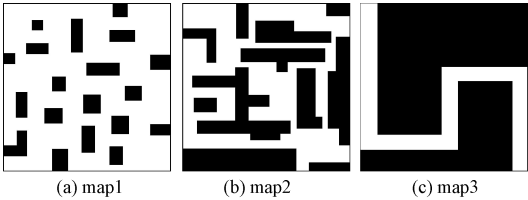


图 9 实验地图

地图的尺寸均为 800×800 ,坐标原点在左上角。在 MATLAB 平台上进行仿真实验,3 种算法的起始点为 $(50,50)$,目标点为 $(750,750)$,步长 Δ 为 50,最大搜索次数为 3 000。其中 RRTGoalBias 算法的目标点概率值 k 为一个定值,实验设置 $k=0.2$ 。每张地图在每个算法下各运行 200 遍,记录每遍运行的时间、总采样次数和采样成功次数,最后计算平均值。

普通地图的实验结果如图 10 所示,相关数据如表 3 所示。在障碍物较为稀疏的环境中,改进 RRT 算法的目标采样概率较高,可以将扩展树快速导向至目标点处。根据实验数据,在 map1 中改进 RRT 算法相较于常规 RRT 算法平均运行时间减少了 80.1%,平均采样次数降低了 85.8%,平均采样成功率提升了 33%。

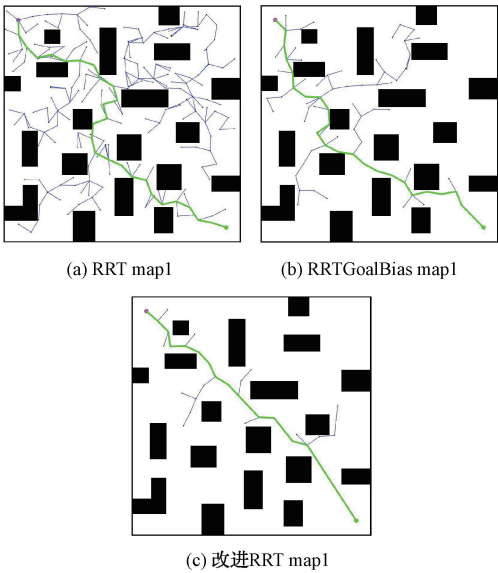


图 10 普通地图实验结果

多障碍和迷宫地图的实验结果如图 11 所示,实验数据如表 4、5 所示。在障碍物占比较高的环境中,随着搜索

表 3 map1 实验数据对比

算法	平均采样次数	平均成功采样次数	平均运行时间/ms
RRT	263.63	166.27	70.9
RRTGoalBias	136.95	74	29.7
改进 RRT	37.33	35.89	14

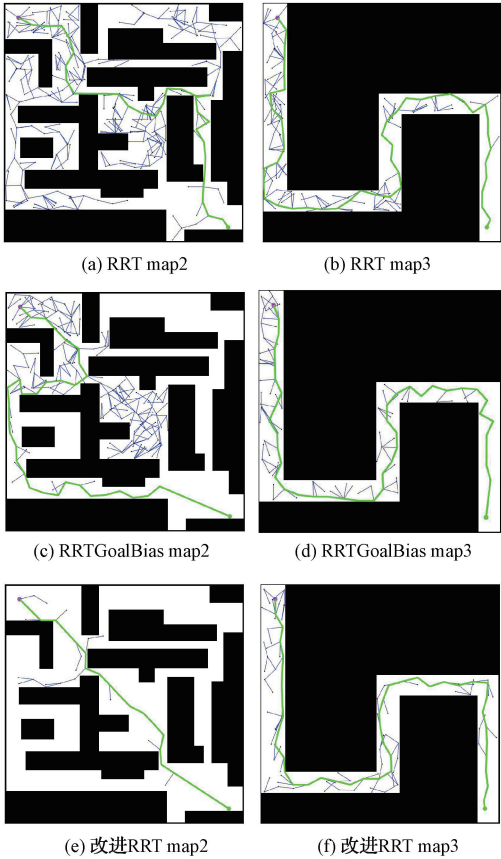


图 11 复杂地图实验结果

表 4 map2 实验数据对比

算法	平均采样次数	平均成功采样次数	平均运行时间/ms
RRT	520.97	129.54	96.5
RRTGoalBias	442.62	83.84	69.5
改进 RRT	143.1	78.98	33.5

表 5 map3 实验数据对比

算法	平均采样次数	平均成功采样次数	平均运行时间/ms
RRT	736.75	125.57	120.7
RRTGoalBias	665.88	96.96	90.8
改进 RRT	277.65	142.89	72.6

次数的增加,改进 RRT 算法的目标点采样概率逐渐降低为一个很小的值,避免算法陷入死区;节点转向策略可以

提高在障碍物附近的探索程度,帮助算法通过一些狭小区域,而对节点转向策略使用次数的限制可以避免在障碍物附近生成过多无效节点,浪费内存和时间。根据实验数据,在 map2 中改进 RRT 算法相较于常规 RRT 算法平均运行时间降低了 65.3%,平均采样次数降低了 72.5%,平均采样成功率提升了 30.3%;在 map3 中改进 RRT 算法相较于常规 RRT 算法平均运行时间降低了 39.9%,平均采样次数降低了 62.3%,平均采样成功率提升了 34.5%。

综上,本文所提出的改进 RRT 算法在运行时间、采样次数和采样成功率几方面相较于常规 RRT 算法均有较大的提升,并能够适应各种不同的环境,具有较好的稳定性。

3.2 消融实验

为了验证自适应目标概率与节点转向策略对算法的影响,额外设计一张地图 map4 进行消融实验,如图 12 所示。

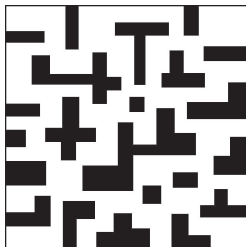


图 12 map4

对比的算法为改进 RRT 算法、只有自适应目标概率的改进 RRT 算法和只有节点转向策略的改进 RRT 算法。初始化参数与上一节相同,同样将每种算法在 map4 中运行 200 遍,实验数据如表 6 所示。

表 6 map4 实验数据对比

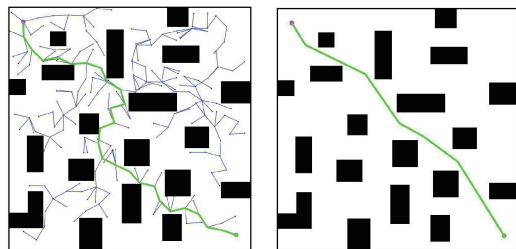
算法	平均采样次数	平均成功采样次数	平均运行时间/ms	寻路成功率/%
改进 RRT	406.43	209.16	134.9	100
自适应目标概率 RRT	1 869.4	182.25	—	81
节点转向 RRT	468.77	270.91	169.3	100

由实验结果可以看出,其中只有自适应目标概率的改进算法由于缺少节点转向策略,由平均成功采样次数与平均采样次数相比可知平均采样成功率由 51.5%降低到 9.7%,寻路成功所需的搜索次数大大增加,从而导致成功率仅为 81%。而只有节点转向策略的改进算法由于缺少自适应目标概率的引导,平均运行时间相较于改进 RRT 算法增加了 25.5%,平均采样次数也有一定的增加。综上,可以验证自适应目标概率与节点转向策略的必要性。

3.3 冗余节点裁剪

由实验结果可以看出,RRT 相关算法所规划出来的路径具有较多非必要的转折点,这不符合实际应用中的要

求。因此改进 RRT 算法采用冗余节点裁剪策略对生成的初始路径进行裁剪优化。以 map1 为例,将改进 RRT 算法运行 200 遍后的路径进行裁剪处理,计算最终路径的路径长度与节点数,并与常规 RRT 算法的结果进行对比,实验结果如图 13 所示,相关数据如表 7 所示。



(a) 常规RRT路径 (b) 改进RRT最终路径

图 13 冗余节点裁剪处理

表 7 map1 最终路径实验数据对比

算法	平均路径长度	平均节点数
RRT	1 319.8	27.6
改进 RRT	1 041.3	6.83

可以看出,经过裁剪后的路径转折点明显减少,转折角度更加平滑,平均路径长度相较于常规 RRT 算法降低了 21.1%,平均路径节点数降低了 75.3%。

4 结论

本文提出了一种改进的 RRT 算法。首先,针对当前较多研究引入基于目标点的导向策略,但却缺乏对导向策略进行实时调整的问题,提出了自适应目标概率策略。为了得到随机采样成功率与最适合目标概率之间的关系,本文设计大量地图进行实验,对实验数据进行拟合并验证,最终得到了较为准确的计算公式,使算法能够适应不同的地图环境。其次针对 RRT 算法在复杂环境中采样效率低、难以通过狭窄区域的问题,提出了节点转向策略,大大提升了算法在多障碍、狭窄通道等复杂环境中的采样效率。最后针对 RRT 算法生成的路径冗余节点过多的问题,采取冗余节点裁剪的方法,使最终路径转折点减少,路径代价降低。实验结果表明,在多种环境下,本文提出的改进 RRT 算法在运行时间、采样次数、采样成功率、路径长度和路径节点数几个方面相较于常规 RRT 算法均有较大提升,能够更好地满足实际应用中的需求。

参考文献

- [1] 王鹤静,王丽娜. 机器人路径规划算法综述 [J]. 桂林理工大学学报, 2023, 43(1): 137-147.
- [2] 巩慧,倪翠,王朋,等. 基于 Dijkstra 算法的平滑路径规划方法 [J/OL]. 北京航空航天大学学报, 1-10 [2024-01-05] <https://doi.org/10.13700/j.bh.1001-5965.2022.0377>.

- [3] 姜媛媛, 张阳阳. 改进8邻域节点搜索策略A*算法的路径规划[J]. 电子测量与仪器学报, 2022, 36(5): 234-241.
- [4] 侯宇翔, 高焕兵, 汪子健, 等. 基于改进跳点搜索法的移动机器人路径规划[J]. 组合机床与自动化加工技术, 2023(3): 54-58.
- [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] 肖金壮, 余雪乐, 周刚, 等. 一种面向室内AGV路径规划的改进蚁群算法[J]. 仪器仪表学报, 2022, 43(3): 277-285.
- [7] 许瑶, 刘晓峰. 基于改进RRT算法的无人车路径规划研究与测试[J]. 国外电子测量技术, 2023, 42(8): 132-138.
- [8] 杨大庆, 谷侃锋. 基于快速扩展随机树的机械臂路径优化算法研究[J]. 组合机床与自动化加工技术, 2021(5): 10-12, 17.
- [9] 赵超力, 马行, 张春涛, 等. 基于引力场引导的RRT-connect路径规划算法[J]. 电子测量技术, 2021, 44(22): 44-49.
- [10] 马智焕, 胡立坤, 陶兴华, 等. 基于分区思想的两阶段RRT*算法[J]. 国外电子测量技术, 2023, 42(8): 40-46.
- [11] UZUN G, OZDEMIR A, BOGOSYAN S, et al. n-Sliced informed RRT*: Intelligent sampling-based path planning in high eccentricity informed ellipsis[C]. Proceedings of the IEEE 31st International Symposium on Industrial Electronics (ISIE), 2022.
- [12] CHEN Y Y, FU Y X, ZHANG B, et al. Path planning of the fruit tree pruning manipulator based on improved RRT-Connect algorithm[J]. International Journal of Agricultural and Biological Engineering, 2022, 15(2): 177-188.
- [13] 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.
- [14] LI Y C, MA S C. Navigation of apple tree pruning robot based on improved RRT-connect algorithm[J]. Agriculture-Basel, 2023, 13(8): 1495.
- [15] 敖国鑫, 李林. 基于改进BI-RRT算法的AGV路径规划[J]. 计算机时代, 2023(5): 57-60, 65.
- [16] GANESAN S, NATARAJAN S K, THONDIYATH A. A novel goal-oriented sampling method for improving the convergence rate of sampling-based path planning for autonomous mobile robot navigation[J]. Defence Science Journal, 2023, 73(3): 322-331.
- [17] 商德勇, 汪俊杰, 樊虎, 等. 基于RRT*-DR算法的机械臂避障路径规划[J/OL]. 计算机集成制造系统, 1-14[2024-01-05] <http://kns.cnki.net/kcms/detail/11.5946.tp.20230322.1721.016.html>.
- [18] LIU Q, LUO W G, WANG Z T, et al. Research on improved RRT path planning algorithm[C]. Proceedings of the 32nd Chinese Control And Decision Conference (CCDC), 2020.

作者简介

谢春圆, 硕士研究生, 主要研究方向为机器人路径规划、运动控制。

E-mail: 2471218060@qq.com

王欣(通信作者), 副教授, 硕士生导师, 主要研究方向为工业机器人与动作规划。

E-mail: wangx@dlut.edu.cn