

基于多次采样启发式策略的改进 RRT 路径规划算法

左国玉^{1,2}, 关海山^{1,2}, 郑榜贵^{1,2}

(1. 北京工业大学 信息学部, 北京 100124;

2. 北京计算智能与智能系统重点实验室, 北京 100124)

摘要: 在路径规划领域, 快速探索随机树 (RRT) 算法是机械臂解决复杂环境中的路径规划问题的重要工具, 然而其纯随机的采样环节导致大量的无效或低效尝试, 浪费了计算资源; 为解决此问题, 提出一种基于多次采样启发式策略的改进 RRT 算法 (MH-RRT); 利用启发式函数评估多个采样点的代价值, 选择代价值最低的采样点, 引导路径树更快地向目标点生长; 将启发式函数策略类似地有效改进了 RRT* 算法和双向 RRT* 算法; 深入探讨不同参数对改进算法性能的影响, 并确定最优参数组合; 实验结果表明, 改进算法在路径搜索时间、路径长度以及采样点数量等方面均能取得显著提升, 提高了路径规划的有效性。

关键词: 路径规划; 启发式函数; RRT 算法; RRT* 算法; 双向 RRT* 算法

Improved RRT Path Planning Algorithm Based on Multiple Sampling Heuristic Strategy

ZUO Guoyu^{1,2}, GUAN Haishan^{1,2}, ZHENG Banggui^{1,2}

(1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;

2. Beijing Key Laboratory of Computing Intelligence and Intelligent Systems, Beijing 100124, China)

Abstract: In the field of path planning, rapid exploration random tree (RRT) algorithms are important tools for robotic arms to solve path planning in complex environments. However, their purely random sampling process results in a large number of invalid or inefficient attempts, wasting computational resources. To solve this problem, an improved RRT algorithm based on multiple sampling heuristic strategy (MH-RRT) is proposed. Firstly, the heuristic function strategy is used to evaluate the proxy value of multiple sampling points, select the sampling point with the lowest proxy value, and intruduce the path tree to grow faster towards the target point; Then, the heuristic function strategy is effectively improved similarly to the RRT* algorithm and the bidirectional RRT* algorithm; Finally, the impact of different parameters on the performance of the improved algorithm is thoroughly explored, and the optimal parameter combination is determined. Experimental results show that the improved algorithm can significantly improve the time efficiency, path length, and number of sampling points in path search, thereby enhancing the effectiveness of path planning.

Keywords: path planning; heuristic function; RRT; RRT*; bidirectional RRT*

0 引言

机械臂作为智能机器人技术的重要组成部分, 被广泛应用于工业自动化^[1]、医疗手术^[2]、空间探索^[3]等领域, 展现出其灵活性和高度可编程性。路径规划作为机械臂的一个基础性研究课题, 受到广泛的关注。

机械臂的路径规划主要是指在给定起点、终点及环境条件下, 如何找到满足各种约束条件的最佳路径。该路径需要满足诸如规划时间短、路径长度短、计算成本低、路径无碰撞等要求。路径规划算法的目的就是制定出一条既安全又高效的路径, 并满足特定的要求。

针对机器人领域的路径规划研究中, 研究者提出了多

种路径规划方法。一是基于图搜索^[4]的规划方法, 如 A* 算法^[5]、多目标粒子群算法^[6]。该方法在具有明确障碍物的环境中表现优异, 能快速找到有效路径, 但难以应对复杂或动态的环境, 且需要预先知道所有障碍物的信息。二是局部路径规划方法, 如人工势场法^[7]、遗传算法^[8]等。该方法针对特定问题能优化解决方案, 灵活性强, 但适用范围有限。三是机器学习方法, 如深度学习、强化学习方法^[9-11]。该方法结合了深度学习的感知能力和强化学习的决策能力, 使机械臂等设备更加智能化。该方法与视觉系统相结合, 识别障碍物, 规划无干涉路径, 抓取目标物体, 但训练时间长且需要大量数据。四是基于采样的路径规划方法^[12],

收稿日期: 2024-03-04; 修回日期: 2024-04-23。

基金项目: 国家自然科学基金项目(62373016); 多模态人工智能系统全国重点实验室开放课题(MAIS-2023-22)。

作者简介: 左国玉(1971-), 男, 博士, 教授。

通讯作者: 郑榜贵(1972-), 男, 博士, 讲师。

引用格式: 左国玉, 关海山, 郑榜贵. 基于多次采样启发式策略的改进 RRT 路径规划算法[J]. 计算机测量与控制, 2024, 32(7): 280-287.

如概率路线图 (PRM, probabilistic roadmap) 算法^[13-14]、快速探索随机树 (RRT, rapidly-exploring random tree)^[15-16] 算法等。该方法通过在连续空间中随机采样离散样本, 并利用这些样本构建树或图结构来表征连续空间。近年来, 此方法在机械臂路径规划中逐渐受到重视并占据了主导地位^[17]。

RRT 算法^[18-19] 通过不断随机采样新的节点, 并将其连接到最近的已有节点, 逐渐生成一棵覆盖整个环境的树。这种随机采样和快速探索的策略使得 RRT 算法能够在较短的时间内生成可行的路径。然而, 由于随机采样的特点, 其生成的路径质量不稳定, 且不能保证最优或最短。为此, 研究者又提出了 RRT* 算法^[20-22] 和双向 RRT*^[23-25] 算法等。RRT* 算法通过重新选择父节点和重新布线, 提高了路径的优化程度, 但需要一定的运算时间。双向 RRT* 算法从起点和终点同时开始构建随机搜索树, 加快了随机搜索树规划路径的速度。

基于随机采样和碰撞检测的 RRT 算法及其变体算法, 能够处理未知环境, 为机械臂在多维运动空间中的路径规划问题提供了有效的解决方案。然而, 这些算法还存在一些不足, 以 RRT 算法为例, 其在采样环节中采用纯随机的方式得到采样点, 该采样点未能考虑其自身与起点和终点的距离关系, 导致算法缺乏明确的目标导向, 不仅延长了算法的计算时间, 还导致生成的路径质量欠佳, 出现绕行现象。针对此问题, 本文提出一种基于多次采样启发式策略的改进 RRT 算法。在 RRT 算法的采样阶段, 该算法引入基于多次采样的启发式函数策略, 对每个采样点的成本进行评估和比较, 选择成本最低的采样点作为新节点, 进行后续的路径探索和生长操作, 显著提高了算法的效率。该算法还充分考虑了采样点与起点和目标点之间的位置关系, 从而更精准地选择更具优势的采样点, 解决了传统 RRT 算法中缺乏目标性的问题。同时, 将这种策略成功应用于其他的 RRT 衍生算法中, 进一步扩大了其应用领域。实验结果表明, 改进后的算法相对于原始算法能够在较短的时间内找到高质量的路径, 且路径长度更短、采样点个数更少, 充分验证了该算法的有效性和可行性, 为路径规划领域带来了有益的改进和全新的思路。

1 快速探索随机树的相关算法

1.1 问题定义

设机械臂的自由度为 n , 则机械臂的状态可以表示为一个 n 维向量 $\vec{q} = [q_1, q_2, \dots, q_n]$ 。同时, 假设机械臂的状态空间为 $X \in R^n$, 可以将机械臂的状态 \vec{q} 映射到状态空间中的 x 。障碍物空间和自由空间分别用 X_{obs} 和 $X_{\text{free}} = X/X_{\text{obs}}$ 表示。设初始状态为 x_{init} 、目标状态为 x_{goal} 。路径规划的目标是寻找到一条无碰撞的可行路径 $\sigma: [0, T] \rightarrow X_{\text{free}}$, 使得路径起点 $\sigma(0) = x_{\text{init}}$ 且路径终点 $\sigma(T) = x_{\text{goal}}$ 。

1.2 RRT 算法

RRT 算法是基于采样的方法代表, 如图 1 所示, 它能够快速探索状态空间中的无障碍区域 X_{free} , 生成一棵描述可行解集合的随机树 $T = (V, E)$, 并不断扩展树的分支,

直到找到一条连接起始点和目标点的路径, 其中 V 表示从 X_{free} 中采样得到的状态顶点, E 为连接这些顶点的边。

RRT 算法的具体步骤如下。

- 1) 初始化: 确定起始节点 q_{init} 和目标节点 q_{goal} , 并将起始节点 q_{init} 作为根节点插入树 T 中。
 - 2) 采样环节: 随机生成一个无碰撞节点 q_{rand} 。
 - 3) 最近邻搜索: 从 T 中找到离 q_{rand} 最近的节点 q_{near} 。
 - 4) 扩展树: 从 q_{near} 向着 q_{rand} 的方向延伸一段距离, 得到一个新节点 q_{new} 。
 - 5) 连接节点: 如果 q_{new} 与 q_{near} 之间的形成的边 E , 不与障碍物相交, 则将 q_{new} 和 E 加入到 T 中, 并标记 q_{near} 为 q_{new} 的父节点。
 - 6) 重复: 重复步骤 2) ~ 5) 进行迭代搜索, 并以一定的概率检查是否可以直接将目标节点 q_{goal} 直接添加到 T 中, 或者达到了最大迭代次数。
 - 7) 回溯路径: 从目标节点 q_{goal} 依次回溯父节点, 找到一条连接起始节点 q_{init} 到目标节点 q_{goal} 的路径。
- 相比其他算法, RRT 算法能够有效地将非完整约束纳入算法中, 避免了复杂运动学约束的考虑, 使得路径规划问题变得更为简单。

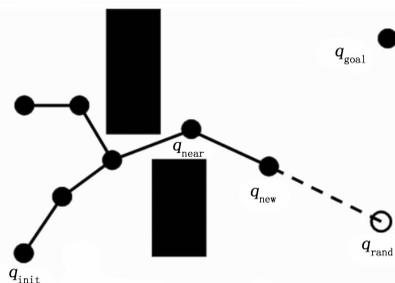


图 1 RRT 算法示意图

RRT 算法伪代码如下。

```

初始化根节点  $q_{\text{init}}$ ;
 $T \leftarrow \{q_{\text{init}}\}$ ;
for  $k=1$  to  $K$  do:
     $q_{\text{rand}} \leftarrow$  采样一个随机节点;
     $q_{\text{near}} \leftarrow$  在  $T$  中找到最近的节点;
     $q_{\text{new}} \leftarrow$  从  $q_{\text{near}}$  向着  $q_{\text{rand}}$  的方向延伸一段距离;
    if  $q_{\text{new}}$  是无碰撞节点 Then:
        将  $q_{\text{new}}$  加入到  $T$  中;
         $E \leftarrow$  连接  $q_{\text{near}}$  向着  $q_{\text{new}}$ ;
        if  $q_{\text{goal}}$  可以连接  $T$  Then:
            return 从  $q_{\text{init}}$  到  $q_{\text{goal}}$  的路径;
        end if;
    end if;
end for;

```

1.3 RRT* 算法和双向 RRT* 算法

RRT* 也是基于采样的算法, 是在 RRT 算法的基础上进行优化产生的变体算法, 它引入了重新选择父节点和重新布线的过程, 可不断调整随机树的结构, 使路径得到渐进优化。重新选择父节点可以确保新生成的节点路径代价

尽可能小,重新布线则有助于减少生成新节点后的随机树中的冗余通路。

RRT * 算法伪代码如下:

```

初始化根节点  $q_{init}$ ;
 $T \leftarrow \{q_{init}\}$ ;
for  $k=1$  to  $K$  do:
     $q_{rand} \leftarrow$  采样一个随机节点;
     $q_{near} \leftarrow$  在  $T$  中找到最近的节点;
     $q_{new} \leftarrow$  从  $q_{near}$  向着  $q_{rand}$  的方向延伸一段距离;
    if  $q_{new}$  是无碰撞节点 Then:
         $q_{min}$  连接到最近的顶点 ( $T, q_{new}$ );
        将  $q_{new}$  加入到  $T$  中;
        重连 ( $T, q_{new}, q_{min}()$ );
        if  $q_{goal}$  可以连接  $T$  Then:
            return 从  $q_{init}$  到  $q_{goal}$  的路径;
        end if;
    end if;
end for;
```

双向 RRT * 算法是在 RRT * 算法之上,引入一个反向树,从终点开始进行搜索,同时与正向树进行交替搜索,可以在两个方向上同时探索空间,提高搜索效率和速度。

双向 RRT * 算法伪代码如下:

```

初始化根节点  $q_{init}$  和  $q_{goal}$ ;
 $T_A \leftarrow \{q_{init}\}$  和  $T_B \leftarrow \{q_{goal}\}$ ;
for  $k=1$  to  $K$  do:
     $q_{rand} \leftarrow$  采样一个随机节点;
     $q_{near} \leftarrow$  在  $T_{A/B}$  中找到最近的节点;
     $q_{new} \leftarrow$  从  $q_{near}$  向着  $q_{rand}$  的方向延伸一段距离;
    if  $q_{new}$  是无碰撞节点 Then:
         $q_{min}$  连接到最近的顶点 ( $T_{A/B}, q_{new}$ );
        将  $q_{new}$  加入到  $T_{A/B}$  中;
        重连 ( $T_{A/B}, q_{new}, q_{min}()$ );
        if  $T_A$  可以连接  $T_B$  Then:
            return 从  $q_{init}$  到  $q_{goal}$  的路径;
        end if;
    end if;
    交换 ( $T_A, T_B$ );
end for;
```

2 基于多次采样启发式策略的改进 RRT 算法

2.1 启发式函数策略

传统 RRT 算法在处理多维运动空间问题时,尽管展现出一定优势,但仍然存在着一些显著缺陷,如图 2 所示。首先,由于 RRT 算法采用随机采样方式寻找路径,导致采样点质量不高,会使算法在寻找目标点过程中可能会在非目标点方向进行大量的无效探索,消耗大量的计算资源和时间,导致路径规划效率低下。其次,由于 RRT 算法的随机性,生成的路径可能存在绕行现象,即路径不是最优路径,会多走一些不必要的路程。这不仅增加了路径的长度,还可能使机器人或车辆在实际运行中需要消耗更多的能源或时间。因此,RRT 算法在生成高质量路径方面也存在着一定的问题。

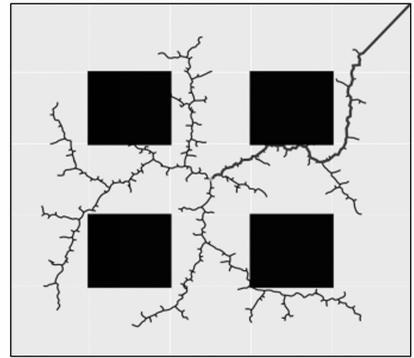


图 2 RRT 算法的路径生成过程

为进一步优化 RRT 算法的性能并提升其生成的路径质量,本文深入分析研究 RRT 算法的运行过程后发现采样环节是影响 RRT 算法整体效率和路径质量的关键环节。在传统 RRT 算法中,采样环节主要依赖于纯随机采样来完成搜索任务。虽然该随机采样方式可以实现全覆盖的搜索,但其效率往往不尽人意,因为随机采样可能会在非目标方向上浪费过多的时间。如图 2 所示,RRT 算法从图像中央的起始点出发,向右上方的目标点搜索路径。图中深色线条展示了算法最终生成的路径,而浅色线条则代表了探索过程中所遍历的路径。观察发现,由于 RRT 算法的随机采样特性,在非目标点方向上花费了大量时间进行路径探索,最终得到的路径质量相对较低,出现了明显的绕行现象。

为解决此问题,本文借鉴了启发式函数的思想,并将其引入到 RRT 算法的采样环节中。启发式函数的核心思想在于通过提供一个预估值来引导搜索过程向目标状态迈进,而非单纯依赖于数学模型的精确计算。在 RRT 算法中,本文为每个采样点赋予了与其位置相对应的成本值,为算法提供明确的导向,使其能够更有目标性地探寻潜在的目标方向。

首先,为采样点分配成本值,根据成本的大小来评估采样点的优劣,进而选择更有希望的搜索方向。然后,在选取启发式函数时,综合考虑采样点与起点、终点的距离等多个因素。经过反复试验和调整,最终确定了如下形式的启发式函数:

$$h(q_{rand}) = r \cdot d^2(q_{rand}, q_{goal}) + (1 - r) \cdot d^2(q_{rand}, q_{init}) \quad (1)$$

其中: $h(q_{rand})$ 是采样点 q_{rand} 的成本值; $d(q_{rand}, q_{goal})$ 是采样节点 q_{rand} 与目标节点 q_{goal} 的欧式距离; $d(q_{rand}, q_{init})$ 是采样节点 q_{rand} 与起始节点 q_{init} 的欧式距离; r 是权重参数。权重参数 r 决定了启发式函数对采样点与起始节点和目标节点距离的重视程度, r 的值越大,意味着启发式函数对采样节点与起始节点距离的重视程度越高。当 r 的值较大时,算法将更倾向于选择靠近目标节点的采样点,以尽快构建从起始节点到目标节点的路径。这有助于减少不必要的搜索方向,提高算法的效率。

如果启发式函数仅仅考虑采样节点与目标节点之间的距离,会导致采样节点过分集中在目标节点的附近,从而使得从采样节点到起始节点构建路径变得困难,甚至是失败。为了解决该问题,本文在设计启发式函数时,加入了采样节点

与起始节点之间的距离信息。通过平衡这两类距离的比重, 确保它们之和为 1, 这样能够促使采样更为分散且均匀, 从而增加找到有效路径的可能性。此调整有助于避免因过度集中而导致的搜索局限, 进而提升路径构建的成功率。

值得注意的是, 本文在实际计算时采用了欧氏距离的平方, 这种方式有效避免了欧氏距离中繁琐的开方运算, 从而使得启发式函数的运行更加高效。欧氏距离的平方计算如下所示:

$$d^2(A, B) = (a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2 \quad (2)$$

其中: $A = (a_1, a_2, \dots, a_n)$ 以及 $B = (b_1, b_2, \dots, b_n)$ 代表 n 维的机械臂状态的向量。

2.2 多次采样启发式策略的改进 RRT 算法

在引入启发式函数策略后, 本文针对算法的采样环节进行深度的优化, 形成了多次采样启发式策略的改进 RRT (MH-RRT, multi-sampling heuristic strategy RRT) 算法, 显著提高传统 RRT 算法的性能。以相同的策略对 RRT * 算法和双向 RRT * 算法的采样环节进行优化, 得到 MH-RRT * 算法和 MH-双向 RRT * 算法, 同样提升算法的整体效果。

为更直观地展示此改进方法, 本文以 MH-RRT 算法为例进行详细的阐述, 如图 3 所示。MH-RRT 算法的伪代码如下:

```

初始化根节点  $q_{init}$ ;
 $T \leftarrow \{q_{init}\}$ ;
for  $k=1$  to  $K$  do;
     $q_{rand} \leftarrow$  采样一个成本最低的随机节点;
     $q_{near} \leftarrow$  在  $T$  中找到最近的节点;
     $q_{new} \leftarrow$  从  $q_{near}$  向着  $q_{rand}$  的方向延伸一段距离;
    if  $q_{new}$  是无碰撞节点 Then;
         $q_{min}$  连接到最近的顶点 ( $T, q_{new}$ );
        将  $q_{new}$  加入到  $T$  中;
        重连 ( $T, q_{new}, q_{min}()$ );
        if  $q_{goal}$  可以连接  $T$  Then;
            return 从  $q_{init}$  到  $q_{goal}$  的路径;
        end if;
    end if;
end for;
```

针对 RRT 算法的采样环节, 本文摒弃了原先随机生成一个无碰撞节点的方式, 将原先的“采样一个随机节点”替换为“采样一个成本最低的随机节点”, 即先生成多个无碰撞节点, 利用启发式函数计算每个节点的成本值, 然后挑选出成本值最低的节点作为本次采样环节的样本。此改进方法可显著提高采样环节的效率 and 准确性, 从而提升 RRT 算法的整体性能。如图 3 所示, 空心虚线节点表示多次采样的候选点, 从中选出启发式函数最优的节点作为本次采样环节的采样节点, 并用空心实线节点标示。由图可知, 采用改进方法获取的采样节点位置会更加靠近目标节点, 进而促使 RRT 树向目标节点生长。

“采样一个最低成本的随机节点”的伪代码如下:

```

初始化采样次数  $N$ ;
```

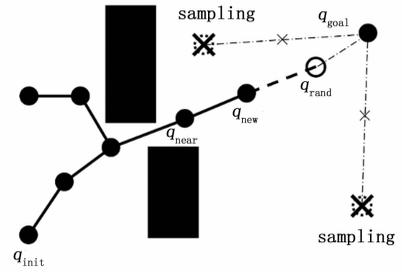


图 3 MH-RRT 算法示意图

```

for  $n=1$  to  $N$  do;
```

```
    采样一个无碰撞的随机节点;
```

```
    启发式函数  $h$  计算随机节点的成本值;
```

```
end for;
```

```
return 成本值最小的随机节点;
```

同样, 使用类似的方法改进 RRT * 算法和双向 RRT * 算法的采样环节。MH-RRT * 算法伪代码如下:

```

初始化根节点  $q_{init}$ ;
 $T \leftarrow \{q_{init}\}$ ;
for  $k=1$  to  $K$  do;
     $q_{rand} \leftarrow$  采样一个成本最低的随机节点;
     $q_{near} \leftarrow$  在  $T$  中找到最近的节点;
     $q_{new} \leftarrow$  从  $q_{near}$  向着  $q_{rand}$  的方向延伸一段距离;
    if  $q_{new}$  是无碰撞节点 Then;
         $q_{min}$  连接到最近的顶点 ( $T, q_{new}$ );
        将  $q_{new}$  加入到  $T$  中;
        重连 ( $T, q_{new}, q_{min}()$ );
        if  $q_{goal}$  可以连接  $T$  Then;
            return 从  $q_{init}$  到  $q_{goal}$  的路径;
        end if;
    end if;
end for;
```

MH-双向 RRT * 算法伪代码如下:

```

初始化根节点  $q_{init}$  和  $q_{goal}$ ;
 $T_A \leftarrow \{q_{init}\}$  和  $T_B \leftarrow \{q_{goal}\}$ ;
for  $k=1$  to  $K$  do;
     $q_{rand} \leftarrow$  采样一个成本最低的随机节点;
     $q_{near} \leftarrow$  在  $T_{A/B}$  中找到最近的节点;
     $q_{new} \leftarrow$  从  $q_{near}$  向着  $q_{rand}$  的方向延伸一段距离;
    if  $q_{new}$  是无碰撞节点 Then;
         $q_{min}$  连接到最近的顶点 ( $T_{A/B}, q_{new}$ );
        将  $q_{new}$  加入到  $T_{A/B}$  中;
        重连 ( $T_{A/B}, q_{new}, q_{min}()$ );
        if  $T_A$  可以连接  $T_B$  Then;
            return 从  $q_{init}$  到  $q_{goal}$  的路径;
        end if;
    end if;
end for;
交换 ( $T_A, T_B$ );
```

3 实验结果与分析

为确保验证改进算法性能的公正性和实验的一致性,

实验均在同一台设备上进行。该设备搭载了 Intel (R) Core (TM) i7-12700H 2.70 GHz 处理器, 运行 Windows 11 操作系统, 并使用 PyCharm 2022. 2. 3 软件。

本文设计了二维网格地图和三维网格地图等两种实验场景, 分别在不同场景下进行路径规划测试, 评估所用算法的实际应用效果和性能。另外, 为更准确地评估改进算法中某些参数对算法性能的影响, 还设计了两个实验。第一个实验关注最低成本采样环节中不同的权重参数 r 对算法性能的影响。第二个实验着重评估最低成本采样环节中不同的采样个数 N 对算法性能的影响。

3.1 二维网格地图实验场景

在二维网格地图仿真实验场景中, 如图 4 所示, 设定了两个具有不同复杂度的场景。一个地图较为简单, 只包含 4 个矩形障碍物; 另一个地图则较为复杂, 有很多条状的障碍物, 但仅有两个狭窄的通道可以通行。这两个场景都是 100×100 像素的区域, 其中黑色区域代表障碍物空间, 灰色区域则代表自由空间。在两个场景的实验中, 初始状态都设定在地图的左下角, 坐标为 $(0, 0)$ 。目标状态都设定在地图的右上角, 坐标为 $(100, 100)$ 。实验目标是寻找一条从初始状态到目标状态的无碰撞路径。

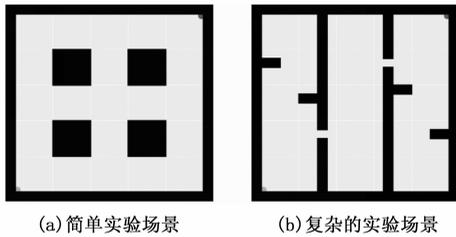


图 4 二维网格地图仿真实验场景

3.2 三维网格地图实验场景

在三维网格地图仿真实验中, 如图 5 所示, 创建了一个 $100 \times 100 \times 100$ 像素的实验场景。场景里放置了 8 个立方体障碍物, 深色区域代表障碍物空间, 浅色区域代表自由空间。实验的初始状态设定在地图的左下角, 坐标为 $(0, 0, 0)$ 。目标状态设定在地图的右上角, 坐标为 $(100, 100, 100)$ 。实验目标是寻找一条从初始状态到目标状态的无碰撞路径。

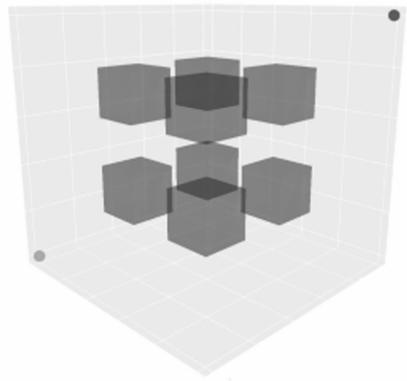


图 5 三维网格地图实验场景

包括整棵路径树上的所有采样点。该指标直接关联到算法的计算成本和路径树的存储成本, 较少的采样点意味着算法的效率更高, 能够以更低成本实现路径规划。经过对这些指标的综合比较, 可较为全面评估各种算法的性能差异及其优缺点。图 6、图 7 和图 8 展示了每个场景中所有算法的部分运行结果。而表 1、表 2 和表 3 详细列出了各项评价指标的具体数据。

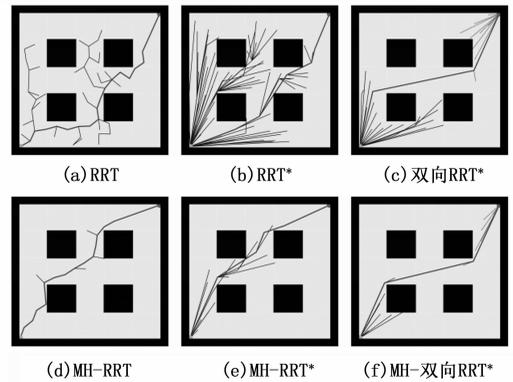


图 6 二维网格地图简单场景的实验结果

表 1 二维网格地图简单场景 100 次实验的平均结果

算法	规划时间/ms	路径长度/px	采样点数量
RRT	21.93	177.10	117.65
MH-RRT	16.56	169.38	58.27
RRT*	141.35	149.21	100.83
MH-RRT*	115.77	150.12	66.28
双向 RRT*	97.70	151.63	56.04
MH-双向 RRT*	93.89	149.19	55.72

3.3 实验结果

在实验中, 分别运行了 RRT 算法、RRT* 算法、双向 RRT* 算法及其使用我们的策略改进后的算法。在本文提出的改进策略中, 所有的算法均取权重参数 r 为 0.9, 采样次数 N 为 3。关于这些参数对改进效果的具体影响, 将在后续部分进行深入探讨。为确保实验的严谨性, 针对每个场景都进行了 100 次实验, 并取其平均值作为最终的实验结果。评估指标包括规划时间、路径长度和采样点数量。规划时间指算法从开始运行至找到可行路径的总耗时, 用于衡量算法的效率和时间复杂度。路径长度指可行路径从起点到终点的总距离, 希望找到既短又安全的无碰撞路径。采样点数量反映了在生成路径过程中所需的总采样点数量,

图 6 展示了 RRT 算法、RRT* 算法、双向 RRT* 算法及对应改进算法的路径规划结果。深色路径为算法的最终结果路径, 浅色路径为所有搜索过的路径。通过观察图 6 上方的 (a)、(b)、(c) 和下方的 (d)、(e)、(f), 可明显看出, 改进的算法在路径长度和采样点数量上均有所提升。表 1 的实验结果进一步证实了本文启发式函数策略显著改进了每种算法。改进算法的规划时间平均提升了 15.5%,

运行速度更快, 能够更高效地完成路径规划任务; 改进算法的路径长度平均缩短了 1.78%, 能够找到更优的路径, 提高了路径规划的质量; 改进算法的采样点数量平均减少了 28.44%, 进一步证明了新算法的优化效果, 减少了不必要的计算和采样, 提高了算法的效率。

图 7 同样展示了 RRT 算法、RRT* 算法、双向 RRT* 算法以及对应改进算法的路径规划结果。可以明显看出, 改进算法在路径长度和采样点数量上均有所提升。表 2 实验结果显示, 改进算法的规划时间平均提升了 39.21%, 路径长度平均缩短了 1.99%, 采样点数量平均减少了 29.63%, 这表明改进算法能够找到更优的路径, 提高路径规划的质量。

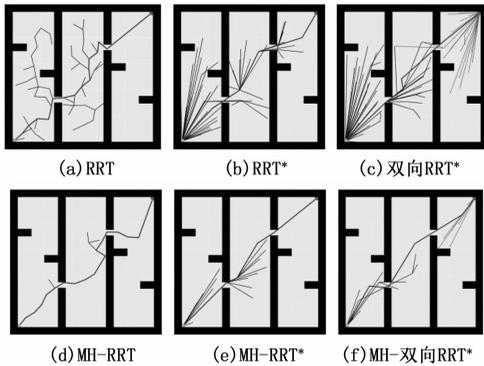


图 7 二维网格地图复杂场景的实验结果

表 2 二维网格地图复杂场景 100 次实验的平均结果

算法	规划时间/ms	路径长度/px	采样点数量
RRT	139.68	178.50	407.03
MH-RRT	124.05	171.22	401.89
RRT*	530.82	144.04	194.06
MH-RRT*	316.17	143.57	96.72
双向 RRT*	365.42	144.72	92.93
MH-双向 RRT*	124.24	142.44	58.11

图 8 同样展示了各个原算法以及对应改进算法的路径规划结果。可以明显看出, 改进算法在路径长度和采样点数量上均有所提升。表 3 实验结果显示, 改进算法的规划时间平均提升了 40.82%, 路径长度平均缩短了 0.75%, 采样点数量平均减少了 24.00%, 这表明改进算法能够找到更优的路径, 提高了路径规划的质量。

表 3 三维网格地图 100 次实验的平均结果

算法	规划时间/ms	路径长度/px	采样点数量
RRT	29.97	225.69	147.15
MH-RRT	23.58	219.32	91.40
RRT*	382.78	183.01	141.06
MH-RRT*	169.29	184.02	87.34
双向 RRT*	348.19	182.00	171.16
MH-双向 RRT*	190.20	182.07	177.94

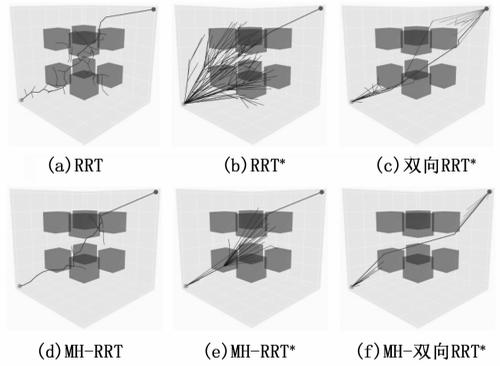


图 8 三维网格地图的实验结果

实验结果表明, 通过引入启发式函数策略, 可成功优化 RRT 算法、RRT* 算法和双向 RRT* 算法, 显著提升了路径规划的性能。规划时间的优化使算法的运行速度得到提升, 能够更高效地完成路径规划。路径长度的缩短表明改进算法能够找到更优的路径, 提高路径规划的质量。采样点数量的减少进一步证实改进算法的优化效果, 减少不必要的计算和采样, 提高算法的效率。

3.4 权重参数和采样次数对轨迹规划的影响

为验证启发式函数策略参数设置的合理性, 本文特别设计了实验, 从权重参数和采样次数两个维度来分析研究其对轨迹规划的影响。实验环境设定为二维网格地图的简单场景, 采用经过优化的 RRT 算法。针对不同的权重参数和采样次数, 分别进行 50 次独立的实验。图 9、图 10 和表 4 展示了不同权重参数 r 的实验结果。图 11、图 12 和表 5 展示了不同采样次数 N 的实验结果。

图 9、图 10 和表 4 展示了 MH-RRT 算法在不同权重参数下的路径规划实验结果。实验结果表明, 不同的权重参数 r 对算法性能的影响明显。本实验选取 $r=0.9$ 对算法的提升效果最好。相对地, 当 $r=0.3$ 时, 路径规划时间长, 路径长度较大, 且采样点数量较多, 算法的路径规划结果并不理想。随着 r 值的逐渐增加, 我们发现规划时间明显缩短, 路径长度得到优化, 采样点数量也相应减少。然而, 需要注意的是, 当 $r=1$ 时, 算法的路径规划结果再次出现下滑。综上所述, 为了实现最佳的路径规划和算法性能, 本文将权重参数 r 设置为 0.9。

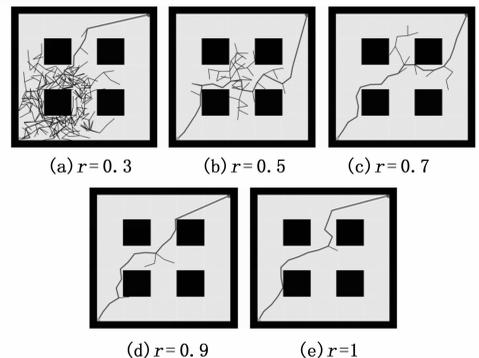


图 9 MH-RRT 算法不同 r 值的实验结果

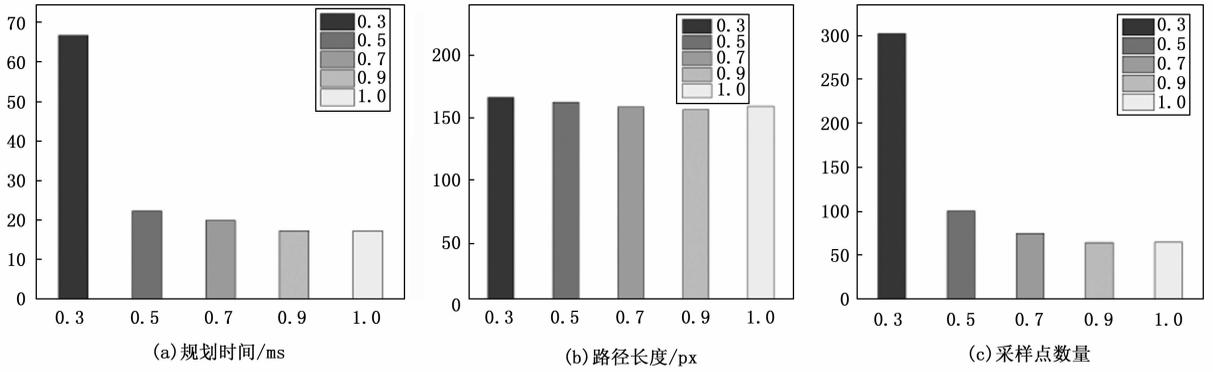


图 10 MH-RRT 算法不同 r 值的平均仿真数据对比

表 4 MH-RRT 算法不同 r 值的平均实验结果

r	规划时间/ms	路径长度/px	采样点数量
0.3	67.12	178.29	304.72
0.5	22.57	174.48	102.54
0.7	20.18	170.67	77.02
0.9	17.53	168.44	66.56
1	17.55	171.32	66.90

次数下的表现效果。实验结果显示，当采样次数 $N=3$ 时，算法的优越性得到了最大程度的体现。相对地，当 $N=5$ 时，尽管路径长度和采样点数量变化不大，但算法的规划时间明显增加，导致效率降低。而当 $N=10$ 时，虽然路径长度有所减少，但规划时间和采样点数量急剧上升，这不符合优化目标。因此，为实现最佳的路径规划和算法性能，应将采样次数 N 设置为 3。该设置可以确保算法在保持较短路径的同时，避免规划时间的无谓增加，从而达到最佳的效率和性能。

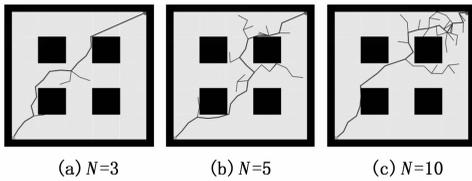


图 11 MH-RRT 算法不同 N 值的实验结果

表 5 MH-RRT 算法不同 N 值的平均实验结果

N	规划时间/ms	路径长度/px	采样点数量
3	17.53	168.44	66.56
5	372.10	167.09	70.34
10	1513.23	163.95	489.58

4 结束语

本文提出了一种基于多次采样启发式函数策略的改进 RRT 路径规划算法，利用启发式函数对多个采样点的代价值进行评估，从而引导算法优先选择代价值最低的采样点进行路径树的生长，使路径树更快地向目标点延伸。这一策略打破了传统 RRT 算法仅依赖随机采样过程的局限。同样，启发式函数策略也可有效改进 RRT* 算法和双向 RRT* 算法。实验结果表明，经过策略改进后的算法在路径搜索的时间效率、路径长度以及采样点数量等方面均取得了显著提升，充分证明了基于多次采样的启发式函数策略的有效性和优越性。此外，本文还深入探讨了不同参数对策略性能的影响，并确定了最优参数组合。实验结果分析表明，改进后的算法在面对复杂环境时仍能保持高效的搜索能力，为解决实际路径规划问题提供了有力的工具。

图 11、图 12 和表 5 展示了 MH-RRT 算法在不同采样

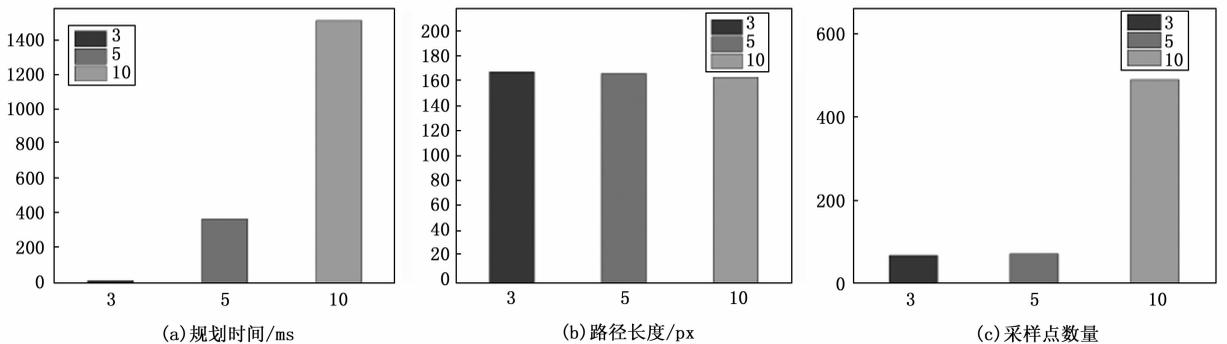


图 12 MH-RRT 算法不同 N 值的平均仿真数据对比

参考文献:

- [1] 马宏伟, 周文剑, 王 鹏, 等. 改进的 ORB-FLANN 煤矸石识别图像与分拣图像高效匹配方法 [J/OL]. 煤炭科学技术, 1-10 [2024-01-23]. <https://link.cnki.net/urlid/11.2402.td.20240118.1012.001>.
- [2] 杨晓涵, 孙 振, 齐岩松, 等. 图像引导下的遥操作脑室穿刺手术机器人 [J]. 中国生物医学工程学报, 2023, 42 (3): 311-320.
- [3] 陈 钢, 高贤渊, 赵治恺, 等. 空间机械臂智能规划与控制技术 [J]. 南京航空航天大学学报, 2022, 54 (1): 1-16.
- [4] DEBNATH S K, OMAR R, LATIP N B A, et al. A review on graph search algorithms for optimal energy efficient path planning for an unmanned air vehicle [J]. Indonesian Journal of Electrical Engineering and Computer Science, 2019, 15 (2): 743-749.
- [5] 周 伟, 潘金宝, 王林琳, 等. 基于改进鲸鱼算法和 A* 算法的地面放线机器人路径规划 [J]. 现代制造工程, 2023 (12): 68-75.
- [6] 左国玉, 李 宓, 郑榜贵. 基于改进自适应多目标粒子群算法的机械臂最优轨迹规划方法 [J]. 实验技术与管理, 2024, 41 (3): 184-191.
- [7] 时维国, 宁 宁, 宋存利, 等. 基于蚁群算法与人工势场法的移动机器人路径规划 [J]. 农业机械学报, 2023, 54 (12): 407-416.
- [8] WANG M, LUO J, ZHENG L, et al. Generate optimal grasping trajectories to the end-effector using an improved genetic algorithm [J]. Advances in Space Research, 2020, 66 (7): 1803-1817.
- [9] 曹 毅, 郭银辉, 李 磊, 等. 基于深度强化学习的机械臂避障轨迹规划研究 [J]. 机械传动, 2023, 47 (12): 40-46, 96.
- [10] 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.
- [11] 曹 毅, 李 磊, 张景涛. 基于深度强化学习的机械臂避障路径规划研究 [J]. 制造业自动化, 2023, 45 (6): 160-164.
- [12] SANCHEZ-IBANEZ J R, PEREZ-DEL-PULGAR C J, GARCIA-CEREZO A. Path planning for autonomous mobile robots: A review [J]. Sensors, 2021, 21 (23): 7898.
- [13] CHEN G, LUO N, LIU D, et al. Path planning for manipulators based on an improved probabilistic roadmap method [J]. Robotics and Computer-Integrated Manufacturing, 2021, 72: 102196.
- [14] YE L, CHEN J, ZHOU Y. Real-time path planning for robot using OP-PRM in complex dynamic environment [J]. Frontiers in Neurorobotics, 2022, 16: 910859.
- [15] 冯 磊, 周志峰, 沈亦纯, 等. 基于改进 RRT 算法的避障路径规划 [J]. 工程设计学报, 2023, 30 (6): 707-716.
- [16] 卢凌霄, 董乾鹏, 张天乐, 等. 机器人运动学与运动规划算法综述 [J]. 印刷与数字媒体技术研究, 2023 (5): 1-16.
- [17] 杨旭海, 周文皓, 李育峰, 等. 采摘机械臂路径规划算法研究现状综述 [J]. 中国农机化学报, 2023, 44 (5): 161-169.
- [18] YUAN C, LIU G, ZHANG W, et al. An efficient RRT cache method in dynamic environments for path planning [J]. Robotics and Autonomous Systems, 2020, 131: 103595.
- [19] 巩 浩, 谭向全, 李佳欣, 等. 基于改进 RRT 算法的移动机器人路径规划研究 [J]. 组合机床与自动化加工技术, 2024 (1): 19-24.
- [20] JI H, XIE H, WANG C, et al. E-RRT*: path planning for hyper-redundant manipulators [J]. IEEE Robotics and Automation Letters, 2023.
- [21] MOHAMMED H, ROMDHANE L, JARADAT M A. RRT*: N: an efficient approach to path planning in 3D for Static and Dynamic Environments [J]. Advanced Robotics, 2021, 35 (3-4): 168-180.
- [22] YU S, CHEN J, LIU G, et al. SOF-RRT*: an improved path planning algorithm using spatial offset sampling [J]. Engineering Applications of Artificial Intelligence, 2023, 126: 106875.
- [23] DAI J, ZHANG Y, DENG H. Novel potential guided bidirectional RRT* with direct connection strategy for path planning of redundant robot Manipulators in joint space [J]. IEEE Transactions on Industrial Electronics, 2023.
- [24] COVIC N, OSMANKOVIC D, LACEVIC B. Asymptotically optimal path planning for robotic manipulators: multi-directional, multi-tree approach [J]. Journal of Intelligent & Robotic Systems, 2023, 109 (1): 14.
- [25] FAN J, CHEN X, LIANG X. UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased [J]. Expert Systems with Applications, 2023, 213: 119137.
- [19] WANG R, CHEN C, LI J, et al. PBBFMM3D: a parallel black-box algorithm for kernel matrix-vector multiplication [J]. Journal of Parallel and Distributed Computing, 2021, 154 (1): 64-73.
- [20] NAZAR A, RAMO F M. A new parallel bat algorithm for musical note recognition [J]. International Journal of Electrical and Computer Engineering, 2021, 11 (1): 558-566.
- [21] WANG Z, WANG D, BAO X, et al. A parallel biological computing algorithm to solve the vertex coloring problem with polynomial time complexity [J]. Journal of Intelligent and Fuzzy Systems, 2021, 40 (3): 1-11.
- [22] YANG R, XU M, ZHOU J. Clone chaotic parallel evolutionary algorithm for low-energy clustering in high-density wireless sensor networks [J]. Scientific Programming, 2021, 2021 (5): 1-13.
- [23] WANG Z, DENG A, WANG D, et al. A parallel algorithm to solve the multiple travelling salesmen problem based on molecular computing model [J]. International Journal of Bio-Inspired Computation, 2022, 20 (3): 160-171.
- [24] ZHANG S, YUAN X, DOCHERTY P D, et al. An improved particle swarm optimization algorithm and its application in solving forward kinematics of a 3-DoF parallel manipulator: [J]. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2021, 235 (5): 896-907.

(上接第 279 页)