

文章编号:1001-5078(2024)07-1028-09

· 激光应用技术 ·

基于激光雷达的室内机器人路径规划方法研究

韩慧妍^{1,2,3}, 韩方正^{1,2,3}, 韩 燮^{1,2,3}, 况立群^{1,2,3}, 曹亚明^{1,2,3}

(1. 中北大学, 计算机科学与技术学院, 太原 030051; 2. 机器视觉与虚拟现实山西省重点实验室, 太原 030051;
3. 山西省视觉信息处理及智能机器人工程研究中心, 太原 030051)

摘要: 自主移动机器人在室内环境中的导航取得了重大进展, 然而地图构建精度较差且路径规划欠佳, 限制了这类机器人的实际应用。为了解决这些问题, 提出了一种基于引导搜索的路径规划算法, 即引力双向快速搜索随机树算法 (GBI-RRT), 该算法采用目标偏差抽样, 有效地引导节点走向目标, 减少无效搜索。为了进一步提高导航效率, 又提出了一种消除低质量节点, 提高路径曲率的路径重组策略, 为了验证上述方法的有效性, 将其集成到一个基于 ROS 系统的移动机器人中, 并在仿真和真实环境实验中进行了评估。结果表明, GBI-RRT 在各种室内环境下的性能均优于现有算法。

关键词: 自主移动; 地图构建; 路径规划; 偏差抽样

中图分类号: TN249; TP24 文献标识码: A DOI: 10.3969/j.issn.1001-5078.2024.07.004

Research on indoor robot path planning method based on LiDAR

HAN Hui-yan^{1,2,3}, HAN Fang-zheng^{1,2,3}, HAN Xie^{1,2,3}, KUANG Li-qun^{1,2,3}, CAO Ya-ming^{1,2,3}

(1. School of Computer Science and Technology, North University of China, Taiyuan 030051, China;
2. Shanxi Provincial Key Laboratory of Machine Vision and Virtual Reality, Taiyuan 030051, China;
3. Shanxi Province Visual Information Processing and Intelligent Robot Engineering Research Center, Taiyuan 030051, China)

Abstract: Significant progress has been made in the navigation of autonomous mobile robots in indoor environments; however, poor map construction accuracy and poor path planning limit the practical applications of such robots. To solve these problems, a path planning algorithm based on guided search, the Gravitational Bidirectional Rapid Search Randomized Tree Algorithm (GBI-RRT) is proposed, which employs target bias sampling to efficiently guide nodes towards the target and reduce ineffective search. In order to further improve the navigation efficiency, another path reorganization strategy that eliminates low-quality nodes and improves the path curvature is proposed. It is integrated into a mobile robot based on a ROS system and evaluated in simulation and real environment experiments to verify the effectiveness of the above method. The results show that GBI-RRT outperforms the existing algorithms in various indoor environments.

Keywords: autonomous mobility; map construction; path planning; bias sampling

1 引言

近年来, 移动机器人被广泛应用于各类场景, 如救援行动^[1]、家庭清洁^[2]和食品服务^[3], 为了满足

这些应用的需求, 移动机器人需要从激光雷达传感器中获取位姿, 并构建地图, 然后使用路径规划算法来确定运行轨迹。随着应用场景的多样化和复杂

基金项目: 国家自然科学基金项目 (No. 62272426); 国家自然科学基金青年项目 (No. 62106238); 山西省自然科学基金项目 (No. 202303021211153); 山西省研究生教育创新项目 (No. 2022Y632) 资助。

作者简介: 韩慧妍 (1980 -), 女, 博士, 副教授, 主要研究方向为人工智能、机器视觉等。E-mail: hhy980344@163.com

收稿日期: 2023-09-22; 修订日期: 2023-10-31

性,对机器人自主导航^[4]性能的要求越来越高。

自主导航的关键在于机器人路径规划,即在已映射的环境中规划一个从起点到目标点的最佳无碰撞路径。根据环境信息是否提前已知,路径规划可分为全局路径规划和局部路径规划。Dijkstra 等人^[5]提出了最短路径规划算法,它采用宽度优先方式来搜索路径。Hart 等人^[6]提出了 A* 算法,它利用启发式评估函数减少了搜索节点,提高了路径搜索的效率。Fox 等人^[7]提出了 DWA,根据机器人的运动学模型和当前的运动参数,对机器人采样空间中的速度进行动态采样,并选择最佳轨迹。为了解决 DWA 评价功能不足的问题,Chang 等人^[8]提出了一种基于 Q-learning 的改进 DWA 算法,修改和扩展了评价功能,增加了两个评价功能,以提高导航性能,在复杂的未知环境中实现了更高的导航效率和成功率。Rosmann 等人^[9]提出了基于多目标优化算法,保证机器人在满足运动约束的前提下输出平滑轨迹。

最新的路径规划算法是基于抽样方法寻找最优路径,该算法通过在空间中随机选择一个点作为起点或终点来寻找最优路径,当扩展到达目标点时,即可确定最优路径。快速探索随机树(RRT)^[10]算法即基于抽样方法来搜索空间,可以有效地避免局部最优问题,但 RRT 需要对整个图进行抽样和搜索,会在每个节点附近产生大量冗余的随机节点,增加了相应的搜索时间,导致收敛速度较慢。双向快速探索随机树(Bi-RRT)^[11-12]算法与 RRT 算法相比效率高,该算法同时在两个方向上展开搜索,空间可以减少一半,可以显著减少搜索时间,但当 Bi-RRT 算法在复杂的高维状态空间中,由于所依赖的随机抽样方法不能有效地覆盖高维状态空间的所有部分,导致无法及时找到最优解。针对上述问题,Xu 等人^[13]提出了一种结合了 PRM 和 P-Bi-RRT 算法的后处理融合算法。与 RRT、Bi-RRT 和 P-Bi-RRT 算法相比,该算法在规划时间、路径长度和路径节点数等方面都取得了更好的效果。Yi 等人^[14]提出了 I-OBg-RRT 算法,该算法使用 1 和 0 变化的概率构造树,与传统 RRT 算法相比,计算时间和路径更短。王等人^[15]提出了一种具有高效分支算法的动态约束 Bi-RRT。该算法扩展了 Bi-RRT 方法,结合了 kino 动态约束,从而提高了路径规划性能。Grothe 等人^[16]提出了 ST-RRT* 算法,可以有效地处理无界

空间,优化了到达目标点的时间。赵等人^[17]提出了一种基于高斯抽样的改进 Bi-RRT 算法,在起始点附近选取高斯分布的样本点和目标点,以减少盲搜索,提高搜索效率。马等人^[18]提出了一种新的路径规划算法,即概率平滑 Bi-RRT (PSBi-RRT)。该算法利用 θ -cut 机制来优化全局最优解的路径,减少了陷入局部最优状态的可能性。与传统的 Bi-RRT 算法相比,PSBi-RRT 显著减少了运行时长,并提高了性能。

在此基础上,本文将对路径规划算法进行改进。Bi-RRT 算法由于只对随机节点进行选取而忽略了冗余计算。本文改进的 Bi-RRT 算法通过使用目标偏差抽样来减少无效搜索,并结合路径重组策略来最小化冗余的路径点,生成平滑的轨迹。综上所述,本文的贡献如下:

1. 将算法嵌入到 ROS 系统^[19]中,以验证该算法的有效性;
2. 提出了 GBI-RRT 算法,该算法采用目标偏差抽样来减少随机抽样对路径质量的负面影响,然后通过路径重组策略对初始路径进行优化,以消除冗余路径;
3. 对改进后的算法进行了大量的仿真评估,并将所提出的算法移植到移动机器人上进行真实场景实验。实验结果表明,该方法具有良好的应用性能,与其他算法相比,在仿真和真实场景中都具有较好的效果。

2 机器人组件和系统框架

本文搭建的地面机器人采用四轮差速底盘,可以四轮驱动,底盘最,转弯半径为 0 m,爬坡角度接近 30°。选用额定功率为 300 W 的 12 寸轮毂电机作为驱动电机,采用无刷直流驱动器驱动车轮转动。机器人分为三层:顶层包括 D435 相机、激光雷达;中间层包括各类传感器和工控机;底层包含电源、Jetson Nano 和显示器。其中,Jetson Nano 可以进行视觉测距、传感器融合、定位与地图绘制、障碍物检测等工作,拥有 32 万亿次运算/s(TOPS)的峰值计算能力和 750 Gbps 的高速 I/O 性能,如图 1、2 所示。

该机器人的系统控制结构如图 3 所示。

PC 端:PC 终端使用笔记本电脑,并通过 SSH 连接到同一局域网上的工控机。命令可以直接从 PC 上发送到移动机器人工控机上,实现 SLAM 和导航功能;

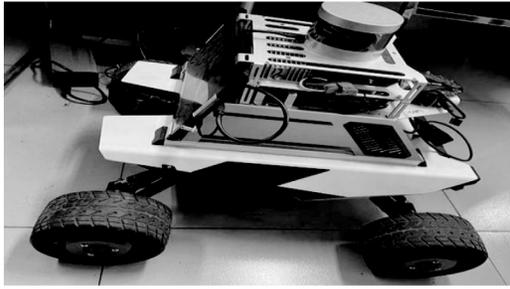


图1 机器人实物图

Fig. 1 Physical drawing of the robot

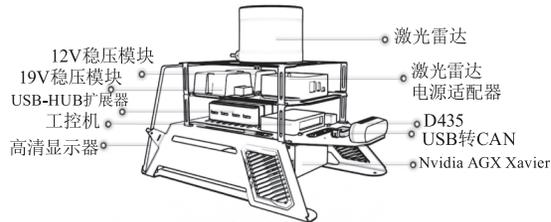


图2 机器人配件图

Fig. 2 Robot accessory diagram

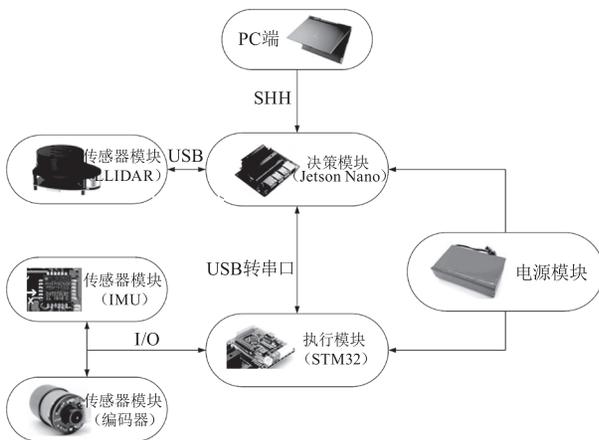


图3 机器人的系统控制结构图

Fig. 3 System control structure of the robot

决策模块:决策模块是机器人所配备的 NVIDIA Jetson Nano,可以接收来自 PC 的命令并运行算法;通过 USB 接口接收激光雷达数据,与机器人工控机进行 I/O 通信,并获取传感器数据;

执行模块:执行模块是以 STM32F4 为核心的控制器,接收来自决策模块的命令,从 IMU 和编码器获取数据,控制电机驱动操作;

传感器模块:激光雷达检测周围环境、IMU 估计机器人运动姿态和编码器估计机器人运动距离和旋转角度。

在机器人导航系统的设计中,涉及了多个关键步骤,包括数据转换、SLAM 映射和路径规划。本文设计了一个综合性的机器人导航系统框架来实现导航能力。该框架通过 ROS 系统实现了分布式通信,

从而实现了 SLAM 映射和导航路径规划之间的协作,并允许节点的发布和订阅,进一步提高了系统的效率和可靠性。机器人导航系统框架的流程如图 4 所示。机器人的导航可分为以下四个步骤:

1. 在机器人工控机和 PC 端安装 Ubuntu 操作系统和 ROS 系统,使用 SSH 远程控制工具实现 PC 与机器人之间的连接,并通过 PC 输入命令对机器人进行控制;

2. 接收到 PC 命令后,机器人利用激光雷达的数据定位并构建地图,映射完成后,地图保存在机器人工控机中;

3. 启动导航命令后,在 ROS 的 Rviz 可视化工具上选择起点和终点,机器人使用来自激光雷达的数据自动规划导航路径。

4. 当机器人到达目的地时,导航结束。

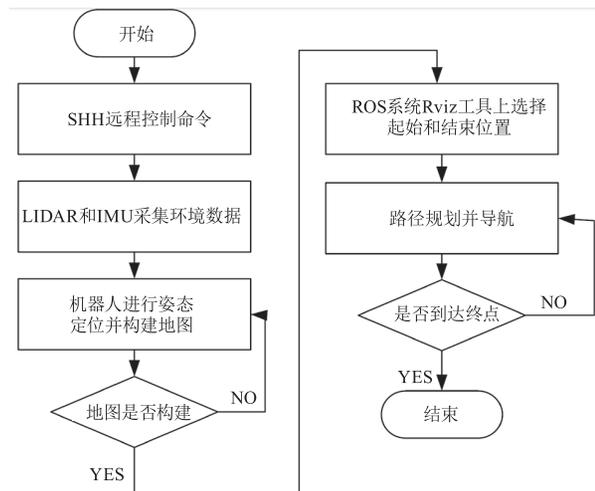


图4 机器人导航系统的框架流程

Fig. 4 Framework flow of the robot navigation system

3 算法改进

3.1 RRT 算法

图 5 显示了 RRT 算法的规划过程,其中 q_{init} 和 q_{goal} 分别表示随机树的起始节点和目标节点; q_{rand} 为每个抽样点生成的随机节点; q_{near} 为树上最接近 q_{rand} 的节点; q_{new} 为碰撞检测后得到的新节点,由 q_{near} 以步长 ϵ 从 q_{new} 生长到 q_{rand} 。

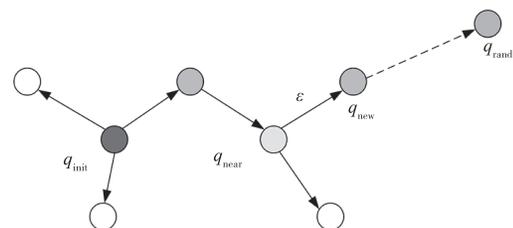


图5 RRT 原理图

Fig. 5 RRT schematic diagram

RRT 算法首先选择 q_{init} 作为随机树生长的根节点,然后在空间内生成 q_{rand} 。算法搜索最接近 q_{rand} 的节点 q_{near} , q_{near} 初始设置为 q_{init} 。从 q_{near} 开始,随机树向 q_{rand} 的方向移动 ε 步,得到一个新的节点 q_{new} 。重复这个过程,直到 q_{init} 和 q_{goal} 之间的欧氏距离小于预定的阈值,此时搜索被终止。结果路径是从初始节点 q_{init} 到目标节点 q_{goal} 的扩展树路径。RRT 算法中新节点的展开式规则用方程(1)表示:

$$q_{new} = q_{near} + \varepsilon \frac{q_{rand} - q_{near}}{\|q_{rand} - q_{near}\|} \quad (1)$$

其中, $q_{rand} - q_{near}$ 表示两个向量的归一化; $\|q_{rand} - q_{near}\|$ 表示两点之间的欧氏距离。

虽然 RRT 算法在复杂的环境中比传统的算法更好,但它的单向搜索方法意味着需要更长的时间才能到达端点,而 Bi-RRT 算法可以实现双向搜索。Bi-RRT 算法如图 6 所示。

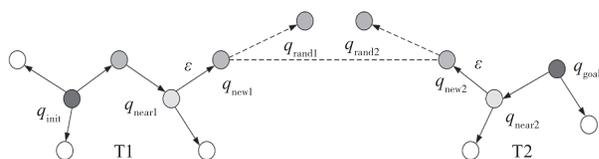


图 6 Bi-RRT 原理图

Fig. 6 Bi-RRT schematic diagram

Bi-RRT 算法在环境状态空间中构造了两个随机树 T1 和 T2,使用与基本 RRT 算法相同的节点生成方法。T1 以根节点为初始节点, T2 以目标点为初始节点。Bi-RRT 算法的处理过程见算法 1。

算法 1 给出了基本的 Bi-RRT 算法。首先,该算法使用 q_{init} 初始化随机树 T1,然后使用 q_{goal} 初始化随机树 T2。为了向外扩展随机树, $Sample()$ 函数返回一个样本点 q_{rand} , $extend()$ 函数在随机树中搜索最近的节点,并向节点 q_{rand} 方向生长,生成一个新的节点 q_{new} 。随后,如果 q_{new} 通过碰撞检测,则将 q_{new1} , q_{new2} 分别添加到随机树 T1 和 T2 中,如果两个随机树的 q_{new} 相同,则循环终止。

与 RRT 算法相比, Bi-RRT 算法在保留 RRT 算法优点的同时减少了搜索时间。但这两种算法都有共同的缺点:都随机生成展开点,导致搜索路径质量较差^[20]。在此基础上,本文提出了一种改进的 Bi-RRT 算法,通过引入目标抽样,产生向目标点有更高概率的随机点,减少算法在节点扩展阶段的盲目性。

算法 1 BI-RRT 算法

Alg. 1 BI-RRT algorithm

算法 1: BI-RRT(q_{init}, q_{goal})	
1	$T_1.add(q_{init}); T_2.add(q_{goal}); i=0;$
2	while($i < N$)
3	$q_{rand1} = Sample();$
4	$q_{rand2} = Sample(); i++;$
5	$q_{new1} = Extend(T_1, q_{rand1})$
6	$q_{new2} = Extend(T_2, q_{rand2})$
7	if $q_{new1} = q_{new2}$ then
8	return Path(T_i, T_2)
9	Swap(T_1, T_2)

3.2 GBI-RRT 算法

Bi-RRT 算法的随机抽样过程采用全局随机搜索策略,产生大量的冗余随机点且增加机器人运动路径的长度。只有当随机树向目标点增长时,才能加速路径规划过程,因此可以将目标点视为抽样点。但如果选择目标点作为唯一的采样点,生成的随机树可能会被困在有障碍物死循环中。为了解决这个问题,本文提出了一种结合了随机搜索和目标导向搜索的目标偏差抽样方法(GBI-RRT 算法)。该方法有效地引导随机树以更高的概率向目标点生长,同时避免了来自障碍物的干扰。

图 7 说明了 GBI-RRT 算法,首先选择一个初始点 q_{init} ,在每次迭代过程中,系统生成一个随机数 p_{rand} 。如果 p_{rand} 小于给定的阈值 p_{bias} ,该算法在空间生成一个随机点 $SampleFree()$ 。否则,该随机点将被设置为目标点坐标。为了实现目标偏差抽样,如公式(2)所示,其有效地引导随机树在避免障碍物的同时以更高概率向目标方向延展:

$$q_{rand} = \begin{cases} q_{goal}, & p_{rand} > p_{bias} \\ q_{SampleFree()}, & \text{else} \end{cases} \quad (2)$$

在上述等式(2)中, p_{bias} 表示目标偏差阈值, p_{rand} 表示随机采样概率范围为(0, 1), $SampleFree()$ 表示空间生成的随机点。

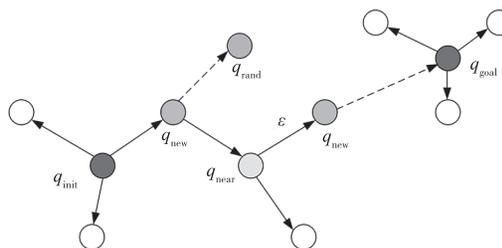


图 7 添加目标偏差抽样构建的 Bi-RRT 随机树

Fig. 7 Bi-RRT random tree constructed by adding target bias sampling

当得到随机节点 q_{rand} 时,使用目标偏差抽样来引导随机树向目标点的扩展,其增长步长为 ε 。该算法保留了 RRT 算法的全局扩展特性,并允许节点在整个状态空间中扩展。此外,目标偏差抽样可以在全局扩展属性之上保留局部节点属性,增加了随机树向目标点扩展的可能性。但阈值 p_{bias} 太大会导致向目标点的扩展概率变小,对扩展速度没有显著影响,而阈值 p_{bias} 太小会导致向目标点的扩展概率过大,在有较多障碍物环境中容易出现局部最小值。经过实验分析后,本文将 p_{bias} 设为 0.5。随机树向目标方向展开的随机生长函数如式(3)所示:

$$X(n) = \varepsilon \frac{q_{near} - q_{goal}}{\|q_{near} - q_{goal}\|} \quad (3)$$

式中, ε 为向目标点展开时的步长; $\|q_{near} - q_{goal}\|$ 表示 q_{near} 与 q_{goal} 之间的欧氏距离。

另外,随机树在空间中展开和避免障碍的随机增长函数 $Y(n)$ 为:

$$Y(n) = \varepsilon \frac{q_{SampleFree()} - q_{near}}{\|q_{SampleFree()} - q_{near}\|} \quad (4)$$

因此,结合式(2)~(4),可以得到利用目标偏差抽样生成新节点的公式,如下:

$$q_{new} = \begin{cases} q_{near} + X(n), & p_{rand} > p_{bias} \\ q_{near} + Y(n), & \text{else} \end{cases} \quad (5)$$

当生成的随机抽样点接近障碍物时,可能会导致新生成的节点与障碍物发生碰撞,导致扩展失败,陷入死回路。如果 p_{rand} 大于 p_{bias} ,则所选的随机点 p_{rand} 满足向目标点扩展的要求,能够更快地接近目标点。然而当 p_{rand} 小于 p_{bias} 时,所选的随机点 q_{rand} 不再满足直接向目标点扩展的要求,将生成随机抽样点进行展开,即扩展的树可以绕过障碍,更有效地到达终点。

3.3 GBI-RRT 算法路径重组策略

在 Bi-RRT 算法中,最近的树节点是通过计算从一个随机点到一个树节点的欧氏距离来确定的。这种方法会导致连接树节点呈锯齿状节点路径,这种不光滑路径对于移动机器人运行不是最优的,因为增加了不必要的转向时间^[21]。即使加入了目标偏差采样,由 Bi-RRT 算法生成的路径仍然可能包含大量冗余节点。因此,需要采用路径重组策略来优化生成的路径,获得更高质量的路径。

如图 8 所示,在多节点的路径规划中,通过路径 $q_{init} \rightarrow b$ 的距离小于通过 $q_{init} \rightarrow a \rightarrow b$ 的距离,通

过 $b \rightarrow d$ 的距离小于通过 $b \rightarrow c \rightarrow d$ 的距离,通过 $d \rightarrow q_{goal}$ 的距离小于通过 $d \rightarrow e \rightarrow f \rightarrow q_{goal}$ 的距离。因此,在最终的路径规划过程中,可以删除冗余的节点 a、c、e 和 f。节点 $q_{init} \rightarrow b \rightarrow d \rightarrow q_{goal}$ 形成最优路径,从而提高了路径的平滑性,缩短了移动机器人的移动时间。

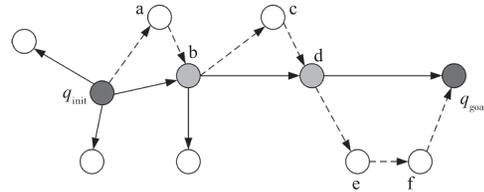


图 8 路径重组策略树

Fig. 8 Path reorganization strategy tree

路径重组策略的过程如算法 2 所示,其中 Key-Points 表示关键点的集合。从初始节点 q_{init} 开始,然后遍历它的子节点用于碰撞检测,只有最接近 q_{goal} 的节点 q_{temp} 被保留并添加到关键点中, q_{temp} 被当作下一次遍历的初始节点。

算法 2 BI-RRT 算法

Alg. 2 BI-RRT algorithm

算法 2: GetKey Points(path)

```

1   $q_{temp} = q_{init}$ ;
2  while( $q_{temp} \neq q_{goal}$ )
3    for( $x = q_{temp}$ ;  $q \neq q_{init}$ ;  $x = q_{temp} \cdot \text{child}$ )
4      if CheckLine( $x, q_{temp}$ )
5         $q_{temp} = x$ ;
6    key points. add( $q_{temp}$ );
```

4 机器人路径规划

4.1 模拟平台

为了评估 GBI-RRT 算法在精度方面的有效性,将在机器人工控机搭载 Ubuntu20.04 系统,配备 Intel i7-11800H 处理器和 16 GB DDR4 3200 MHz 内存,Ubuntu 系统内安装的机器人操作系统(Robot Operating System, ROS)作为机器人程序的运行环境建模平台,如图 9 所示。

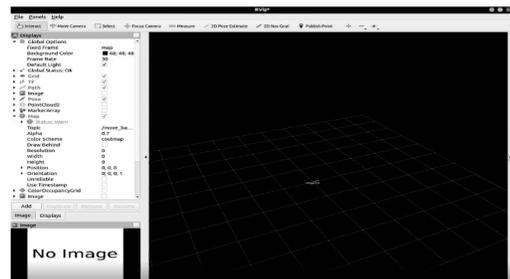


图 9 环境建模平台

Fig. 9 Environmental modeling platform

4.2 GBI-RRT 算法的仿真实验

为了验证所提出的 GBI-RRT 算法的有效性和搜索效率,在虚拟仿真实验中,将对 RRT、Bi-RRT 和 GBI-RRT 算法进行可视化性能比较。本实验使用 Rviz 构造虚拟地图,模拟地图中的黑色点表示障碍物,白色表示安全空间,且包括固定的步长和相同的起始点和结束点位置。机器人将优先前往未搜索的区域,所有的边界都将被机器人搜索,通过在同一时间段内运行机器人,分别评估了各算法的导航效率。实验

结束后,记录各算法在运行过程中的运行时间、路径长度、拐角数。

如 10 图显示了 RRT、Bi-RRT 和 GBI-RRT 三种算法在模拟环境下获得的路径规划结果,从图 10 可以看出,RRT 和 Bi-RRT 算法产生了大量不必要的节点,分散在整个模拟地图中,且存在大量拐角,即角不够光滑,对目标点搜索的效率下降。然而 GBI-RRT 算法只有较少的转折点,并生成了一个较平稳的规划路径。

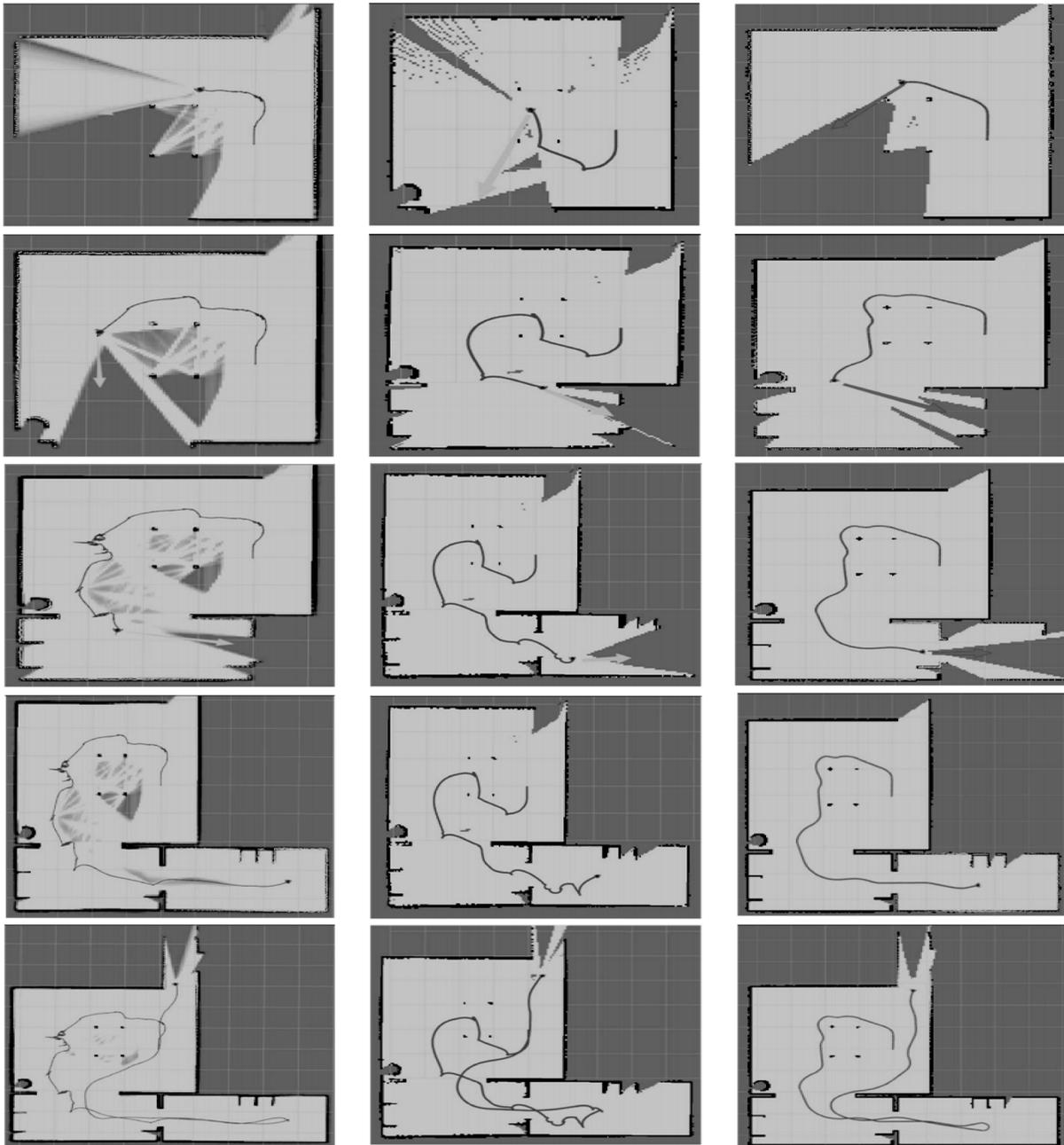


图 10 路径规划

Fig. 10 Route planning

表 1 显示 RRT、Bi-RRT 和 GBI-RRT 算法获得的路径运行时间,长度和拐角数。结果表明,在仿真环境中,RRT 算法比其他两种方法规划路径的时间要长,这主要是由于 RRT 算法扩展的盲目性。Bi-RRT 采用双向搜索来进行速度优化,在一定程度上减少了规划时间,然而使用与 RRT 相同的随机扩张策略并没有明显改善最终的路径长度,只提高了约 26.55 m。GBI-RRT 算法通过所提出的目标偏差采样,可以有效地向目标点搜索,与前两种算法相比,规划时间明显减少,规划时间为 4.84 s。再对已经通过 GBI-RRT 算法规划的路径采用路径重组策略,进一步优化了路径长度,比前两种路径分别减少了 209.76 m 和 183.21 m。上述实验数据表明,GBI-RRT 算法可以在短时间内完成环境遍历搜索,实现了高效搜索未知环境和自主导航。

表 1 三种路径规划仿真实验结果

Tab. 1 Experimental results of three path planning simulations

项目	RRT	Bi-RRT	GBI-RRT
运行时间/s	11.05	5.28	4.84
路径长度/m	803.90	777.35	594.14
拐角数/次	9	7	3

5 机器人真实场景实验

5.1 真实场景实验设置

使用 ROS 平台的分布式框架来执行机器人任务。该框架通过松耦合的方式实现节点之间的通讯,并且能够在不同的计算机上运行。机器人和计算机需要在同一局域网,才能通过 SSH 命令远程控制机器人。此外,提供可视化的界面,通过在计算机终端上操作机器人,使机器人的控制更加直观,这种配置提高了机器人在执行导航任务的灵活性和可操作性。

具体步骤如下:

第一步:主控制器 Jetson Nano 和笔记本电脑连接到同一个网络,利用手机热点网络覆盖机器人的运动区域。

第二步:查看 Jetson Nano 和笔记本的 IP 地址。

第三步:在笔记本的 Ubuntu 系统中,环境变量“ROS_MASTER_URI”和“ROS_HOSTNAME_URI”被添加到“bashrc”文件中,“ROS_MASTER_URI”指向 Jetson Nano 的 IP 地址,而“ROS_HOSTNAME_URI”指向笔记本电脑上 Ubuntu 系统的 IP 地址。

第四步:在 Ubuntu 系统终端中使用 SSH 命令远程访问机器人,实现可视化远程控制。

5.2 路径规划算法的实验

本实验比较了 RRT、Bi-RRT 和 GBI-RRT 算法在三种不同的真实场景下的路径规划结果。

如图 11 所示,设置三个实验地图,标注移动机器人的起点和终点。表 2 显示了 RRT、Bi-RRT 和 GBI-RRT 三种算法的路径规划实验数据。

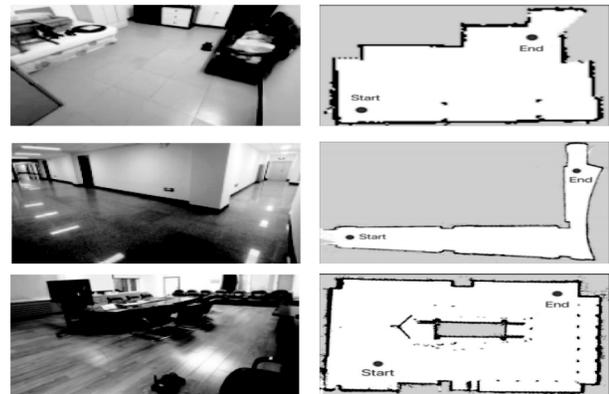


图 11 三种环境地图

Fig. 11 Three environmental maps

表 2 三种不同环境下路径规划算法的质量分析

Tab. 2 Quality analysis of path planning algorithms in three different environments

项目	RRT	Bi-RRT	GBI-RRT
运行时间/s	17.95	17.70	14.55
路径长度/m	11.29	11.05	4.70
拐角数/次	14.55	8.70	2.50

如表 2 所示,对 3 种真实场景进行了 20 次实验,然后取其平均值进行路径规划质量分析。RRT 和 Bi-RRT 算法平均分别需要 14.55 和 8.70 次拐角,而 GBI-RRT 算法平均只需要 2.50 次拐角,且在路径规划时间和长度方面优于其他两种算法,与 RRT 相比,GBI-RRT 的路径规划时间减少了 3.4 s,路径规划长度减少了 6.59 m,拐角数减少了 12.05 次;再与 Bi-RRT 相比,GBI-RRT 的路径规划时间减少了 3.15 s,路径规划长度减少了 6.35 m,拐角数减少了 6.2 次。这些结果表明,GBI-RRT 算法可以快速生成从原点到终点的平滑最优路径。

5.3 机器人导航

使用 SSH 命令将计算机连接到机器人端,如图 12 所示在 PC 端操作启动可视化工具 Rviz,控制机器人进行实时地图构建,当扫描完周围环境后,将地

图进行保存,然后运行导航命令。

如图 12 所示,描述了机器人的自主导航过程,在界面的左下角显示机器人在真实环境的位姿,右边为机器人扫描出的点云地图。如图 13 所示,为机器人在运动过程中的初始姿态和规划信息。图中长红线段表示全局路径规划,实验表明 GBI-RRT 算法得出的路径线路转折点较少,并生成较平滑的规划路径。

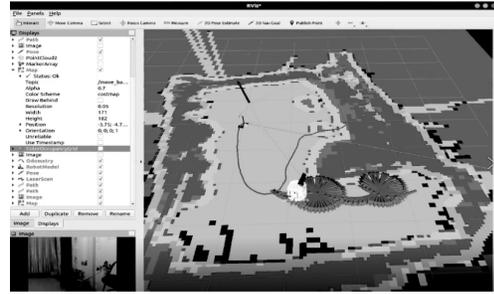


图 12 可视化工具 Rviz

Fig. 12 Visualization tool Rviz

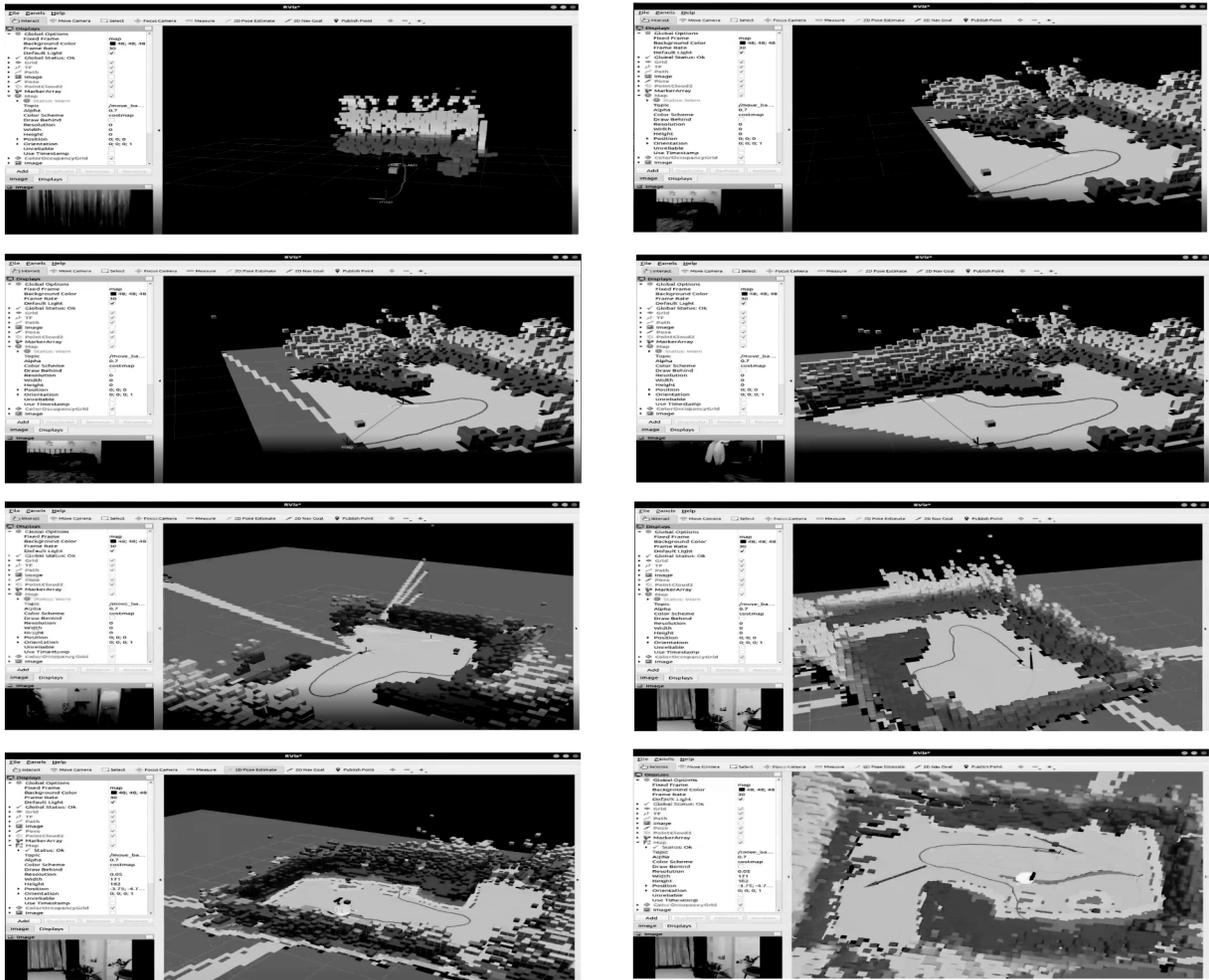


图 13 机器人的自主导航过程

Fig. 13 Autonomous navigation process of the robot

6 结 语

本研究提出了一种 GBI-RRT 路径规划算法,以改进自主移动机器人在室内环境中的导航能力。GBI-RRT 结合了目标偏差抽样,有效地引导节点走向目标,减少了无效的搜索。路径重组策略通过消除低质量节点和提高路径曲率,进一步提高了导航效率。通过仿真和真实场景实验进行了验证,结果表明,该方法比现有算法具有更好的性能。未来的研究可以集中于将目前提出的方法应用于更复杂的

环境中。研究人员还可以考虑如何更有效地提高模型的速度和准确性,并将这些算法应用于其他领域。

参考文献:

- [1] Py F, Robbiani G, Marafioti G, et al. SMURF software architecture for low power mobile robots: experience in search and rescue operations [C]//2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2022: 264 - 269.

- [2] Sui L, Lin L. Design of household cleaning robot based on low-cost 2D LiDAR SLAM [C]//2020 International Symposium on Autonomous Systems (ISAS). IEEE, 2020: 223 – 227.
- [3] Farooq M U, Eizad A, Bae H K. Power solutions for autonomous mobile robots; a survey [J]. Robotics and Autonomous Systems, 2023, 159: 104285.
- [4] Ismail H, Roy R, Sheu L J, et al. Exploration-based SLAM (e-SLAM) for the indoor mobile robot using lidar [J]. Sensors, 2022, 22(4): 1689.
- [5] Noto M, Sato H. A method for the shortest path search by extended dijkstra algorithm [C]//2000 IEEE International Conference on Systems, IEEE, 2000.
- [6] Seet B C, Liu G, Lee B S, et al. A-STAR; a mobile ad hoc routing strategy for metropolis vehicular communications [C]//Networking 2004: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Third International IFIP-TC6 Networking Conference Athens, Greece, May 9 – 14, 2004.
- [7] Ogren P, Leonard N E. A convergent dynamic window approach to obstacle avoidance [J]. IEEE Transactions on Robotics, 2005, 21(2): 188 – 195.
- [8] Chang L, Shan L, Jiang C, et al. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment [J]. Autonomous Robots, 2021, 45: 51 – 76.
- [9] Rösmann C, Hoffmann F, Bertram T. Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control [C]//2015 European Control Conference (ECC). IEEE, 2015: 3352 – 3357.
- [10] Liu B, Liu C. Path planning of mobile robots based on improved RRT algorithm [J]. Journal of Physics: Conference Series, 2022, 2216(1): 012020.
- [11] Li Z, Li L, Zhang W, et al. Research on unmanned ship path planning based on RRT algorithm [J]. Journal of Physics: Conference Series. IOP Publishing, 2022, 2281(1): 012004.
- [12] 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.
- [13] Xu J, Tian Z, He W, et al. A fast path planning algorithm fusing PRM and P-BI-RRT [C]//2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan). IEEE, 2020: 503 – 508.
- [14] Gan Y, Zhang B, Ke C, et al. Research on robot motion planning based on RRT algorithm with nonholonomic constraints [J]. Neural Processing Letters, 2021, 53: 3011 – 3029.
- [15] Wang J, Li B, Meng M Q H. Kinematic Constrained bi-directional RRT with efficient branch pruning for robot path planning [J]. Expert Systems with Applications, 2021, 170: 114541.
- [16] Grothe F, Hartmann V N, Orthey A, et al. ST-RRT*: asymptotically-optimal bidirectional motion planning through space-time [C]//2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022: 3314 – 3320.
- [17] Zhao H. Path Planning of mobile robots based on improved BI-RRT algorithm [C]//2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE). IEEE, 2022: 1043 – 1050.
- [18] Ma G, Duan Y, Li M, et al. A probability smoothing Bi-RRT path planning algorithm for indoor robot [J]. Future Generation Computer Systems, 2023, 143: 349 – 360.
- [19] Choi J, Jeong B, Theotokatos G, et al. Approach an autonomous vessel as a single robot with robot operating system in virtual environment [J]. Journal of International Maritime Safety, Environmental Affairs, and Shipping, 2022, 6(1): 50 – 66.
- [20] Kang J G, Lim D W, Choi Y S, et al. Improved RRT-connect algorithm based on triangular inequality for robot path planning [J]. Sensors, 2021, 21(2): 333.
- [21] Zhang Y, Wang H, Yin M, et al. Bi-AM-RRT*: a fast and efficient sampling-based motion planning algorithm in dynamic environments [J/OL]. arXiv preprint arXiv: 2301.11816, 2023.