

基于 DBSCAN 算法的改进 RRT-Connect 路径规划研究

李刘洋[†], 王可庆, 焦思韬, 周奇

(南京信息工程大学自动化学院, 江苏 南京 210044)

摘要:针对双向快速扩展随机树(RRT-Connect)算法收敛速度慢、路径搜索效率低、路径曲折的问题,提出了一种基于 DBSCAN 算法的改进 RRT-Connect 算法。在 RRT-Connect 算法基础上加入节点引力场引导新节点产生方向,使收敛速度变快;同时通过引入椭圆采样方法,缩小采样范围,提高路径规划效率;最后在改进 RRT-Connect 的基础上引入 DBSCAN 聚类算法,使得到的规划路径更加平滑可靠,增强算法的鲁棒性。为了验证改进后的算法优化效果,分别在不同环境中与 RRT 算法、RRT-Connect 算法进行仿真比较。仿真实验表明,改进后的 RRT-Connect 算法路径规划效果均要优于其他两种算法,不仅加快了路径规划速度,而且得到的路径接近最优解,具有普遍适用性、鲁棒性高等特点。

关键词:RRT-Connect 算法;DBSCAN 算法;椭圆采样;引力场;路径规划

中图分类号:TP39

文献标识码:A

Research on Improved RRT-Connect Path Planning Based on DBSCAN Algorithm

LI Liuyang[†], WANG Keqing, JIAO Sitao, ZHOU Qi

(School of Automation, Nanjing University of Information Science and Technology, Nanjing, Jiangsu 210044, China)

Abstract: Aiming at the problems of slow convergence speed, low path search efficiency and tortuous path of RRT-Connect algorithm, an improved RRT-Connect algorithm based on DBSCAN algorithm is proposed. On the basis of the RRT-Connect algorithm, the node gravitational field is added to guide the direction of the new node, which makes the convergence speed faster. At the same time, by introducing the elliptical sampling method, the sampling range is reduced to improve the efficiency of path planning. Finally, based on the improved RRT-Connect, the DBSCAN clustering algorithm is introduced to make the obtained planning path smoother and more reliable, and enhance the robustness of the algorithm. In order to verify the optimization effect of the improved algorithm, it is simulated and compared with the RRT algorithm and the RRT-Connect algorithm in different environments. The simulation results show that the improved RRT-Connect algorithm has better path planning effect than the other two algorithms. It not only accelerates the path planning speed, but also obtains the path close to the optimal solution, which has the characteristics of universal applicability and high robustness.

Key words: RRT-Connect algorithm; DBSCAN algorithm; ellipse sampling; gravitational field; path planning

路径规划是机器人技术和自动化领域中的一个核心问题,它涉及在给定环境中,避免与障碍物碰撞的前提下,找到最优路径的任务^[1]。路径规划在众多实际应用中发挥着重要作用,包括自动驾驶^[2]、机器人导航^[3]、无人机航行^[4]和运输^[5]等领域。

随着技术的不断发展和应用的广泛需求,路径规划算法的研究变得越来越关键。常见的路径规划算法有:A*算法^[6]、蚁群算法^[7]、人工势场算法^[8]、概率图算法(PRM)^[9]、D*算法^[10]、快速探索随机树(RRT)。

收稿日期:2023-10-17

基金项目:江苏高校“青蓝工程”资助项目

作者简介:李刘洋(1998—),男,河南周口人,硕士研究生,研究方向:机械臂目标识别抓取。

[†]通信联系人, E-mail: 1195193743@qq.com

其中,最常用的算法是基于采样的方法。RRT 算法凭借其搜索路径效率高、适用于高维空间的优点,在路径规划算法中得到快速发展和应用。传统的 RRT 算法采用随机采样的方法进行搜索,具有较强的搜索能力,但 RRT 的随机性也导致其搜索效率有提升的空间,同时规划路径的质量方面也不能达到最优,特别是在复杂环境和高维空间中。为了提高 RRT 算法的搜索效率和路径质量问题,很多学者对其进行了改进。Kuffner 等^[11]在 2000 年提出 RRT-Connect 算法,通过起点和目标点同时生成随机扩展树来提高算法的搜索效率,但是规划的路径比较曲折,在应用于空间机械臂避障路径规划时,存在盲采样点大、无效点多等问题^[12]。Karaman 等^[13]在 2011 年提出 RRT * 算法,该算法随着样本数量的增加而不断优化随机树,由于其渐进优化的特性,RRT * 算法理论上可以得到最优解,但是收敛速度过慢。赵超力等^[14]将引力场思想引入 RRT-Connect 算法,引导随机树生长,并且在路径起点和终点之间设置第三节点,由两棵随机树变为四棵随机树,大大加快搜索效率,但存在获取路径曲折问题。游达章等^[15]提出一种基于椭球子集采样的 RRT-Connect 算法,缩小采样空间,同时引入目标偏执采样策略,提高路径规划效率,最后基于三角不等式的路径修剪策略来优化路径质量。Zhu 等^[16]提出将 Dijkstra 算法与 RRT-Connect 算法相结合,去除冗余路径节点,缩短路径长度,引用人工势场思想,减少路径盲目搜索,最后利用 B 样条曲线对规划路径进行平滑处理,获得连续轨迹。

通过对上面的文献总结得知,RRT-Connect 算法优点还是比较突出的,相较于 RRT 算法和 RRT * 算法,它采用两棵随机扩展树来搜索路径,使搜索效率大大提升,但是 RRT-Connect 算法的缺点也很明显,比如扩展路径曲折、随机盲目扩展等问题。为了解决以上问题,以 RRT-Connect 算法为基础,在路径起点以及目标点上叠加引力场引导随机树扩展方向,使收敛速度变快,实现路径平滑的效果;同时引入椭圆采样策略,将采样空间限制在一个椭圆空间内,以减少路径搜索的盲目性,提高路径规划效率;最后针对路径曲折问题,在改进 RRT-Connect 算法的基础上采用 DBSCAN 聚类算法,对以往的仿真数据进行聚类分析,就可以直接、有效地通过已有的历史数据提取出重要且有意义的路径点,使得到的路径更接近最优解,算法普适性和鲁棒性更高。通过仿真实验,验证改进算法的有效性和可行性。

1 传统算法

1.1 RRT 算法

RRT 算法作为一种随机性算法,在搜索空间中随机生成节点,并将新节点连接到树中已有的节点,从而构建一棵随机生长的树,直到找到目标节点为止。RRT 算法凭借其快速、简单且易于实现的特点在路径规划问题中得到广泛的研究和应用。

RRT 算法通过在起始点构建一棵只包含根节点的树,定义起始节点为 n_{init} ,见图 1,在可搜索空间中随机采样一个点 n_{rand} ,再根据定义的搜索步长 λ 在树中搜索最近的节点 n_{near} ,如果 n_{rand} 和 n_{near} 之间的直线上有障碍物,则舍弃该采样点并重新生成一个新的采样点;如果没有障碍物,则根据搜索步长 λ 逐步连接 n_{rand} 和 n_{near} 。重复上面的步骤直到到达目标点 n_{goal} 或设定好的迭代次数。最后通过回溯随机树在上述过程中生成的节点,获取从起始点到终点的路径。

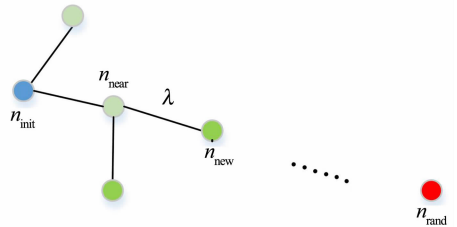


图 1 RRT 算法扩展图

1.2 RRT-Connect 算法

RRT-Connect 算法在 RRT 算法的基础上将起始点和目标点构建为两棵随机树,分别向彼此的方向扩展,同时引入节点贪婪扩展策略,在每次迭代生成节点时,尽量将另一棵树的最近节点作为该树的扩展方向,使两棵树快速相交。另外,如果在扩展过程中没有遇到障碍物,算法继续在该方向上扩展节点;如遇到障碍物,该算法使用交换函数,让另一个随机树扩展,这样避免算法陷入局部最优困境。相较于 RRT 算法,搜索路径效率得到了极大提升。RRT-Connect 算法扩展过程如图 2 所示。

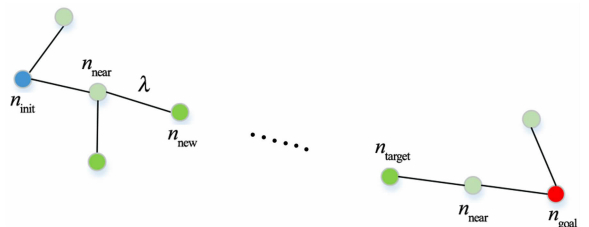


图 2 RRT-Connect 算法扩展图

2 改进算法

RRT-Connect 算法相较于 RRT 算法在搜索效率以及路径规划质量方面虽然有所提升,但是本身的盲目搜索特性还有路径曲折问题依然存在,所以我们就根据 RRT-Connect 算法所存在的问题,提出针对性的算法优化。针对盲目搜索问题,引入节点引力场概念,给起始点和目标点各叠加一个引力场,引导随机树向目标节点生长;同时引入椭圆采样策略,将随机采样空间限制在一个椭圆空间内,进一步降低路径盲目搜索性;针对规划路径曲折问题,采用 DBSCAN 聚类算法去除路径冗余点,优化路径质量,提高算法适用性和鲁棒性。

2.1 节点引力场

将目标节点引力思想^[17,18]引入 RRT-Connect 算法中,分别对双向随机树中的每个节点 n 都叠加一个目标引力函数 $G(n)$,这个节点是由起始节点(或目标接点)向外扩展的第 n 个节点,表达式为:

$$F(n) = R(n) + G(n) \quad (1)$$

方程(1)表示节点 n 到目标点的生长指导函数 $F(n)$ 是由随机生长函数 $R(n)$ 和目标引力函数 $G(n)$ 组成的。生长指导函数 $F(n)$ 是指导节点 n 向目标点生长的总力量。随机生长函数 $R(n)$ 表示从初始节点到节点 n 的随机生长力量,它模拟了随机的生长过程。目标引力函数 $G(n)$ 则代表了目标点对节点 n 的引力,它使节点 n 被引导向目标点。

新节点的随机生长函数 $R(n)$ 的计算公式为:

$$R(n) = \lambda * \frac{n_{rand} - n_{near}}{\|n_{rand} - n_{near}\|} \quad (2)$$

其中 λ 为步长。目标引力函数 $G(n)$ 为:

$$G(n) = \lambda * \gamma * \|n_{goal} - n_{near}\|^2 \quad (3)$$

其中 γ 为引力系数, $\|n_{goal} - n_{near}\|^2$ 为目标点到随机生长树上最近点的欧氏距离的平方,这样离目标点越远,引力越大。

将式(2)和式(3)代入式(1)得:

$$F(n) = \lambda * \frac{n_{rand} - n_{near}}{\|n_{rand} - n_{near}\|} + \lambda * \gamma * \|n_{goal} - n_{near}\|^2 \quad (4)$$

根据式(4)可得到叠加引力场后的新节点产生方式为:

$$n_{new} = n_{near} + \lambda * \left(\frac{n_{rand} - n_{near}}{\|n_{rand} - n_{near}\|} + \gamma * \|n_{goal} - n_{near}\|^2 \right) \quad (5)$$

在叠加引力场之后,每个节点在双向随机树中都使用相同的生长指导函数 $F(n)$,该函数综合考虑了引力分量和自由空间的因子。引力分量使节

点朝向目标方向生长,而自由空间的概念确保节点在生长过程中避免碰撞或穿越障碍物。这种一致性的生长指导函数确保了树的协调和一致性,使得双向随机树能够高效地搜索路径并找到目标点。引力场下影响节点扩展示意图如图 3 所示。

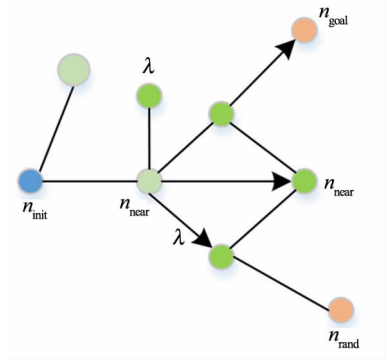


图 3 引力场下影响节点扩展示意图

2.2 椭圆采样

在规划路径时,由于起始点和目标点都是已知的,两点连线是成本最低的路径。将随机点限制在最小路径周围生成的椭圆区域内,而不是在自由空间中任意搜索,这样既能提高收敛速度,又能克服一定的随机性。

为了将采样空间限制在椭圆子集中,引用以起始点和目标点为焦点的椭圆。 C_{min} 是起始点和目标点之间的欧氏距离,椭球的短轴为 $\sqrt{C_{best}^2 - C_{min}^2}$,椭球的长轴为 C_{best} ,见图 4。

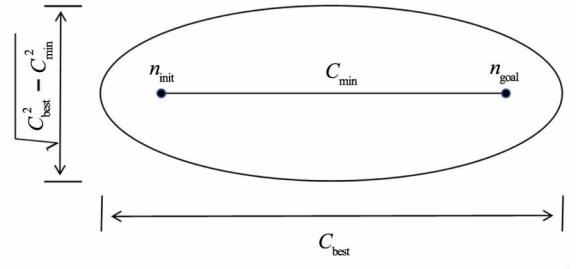


图 4 椭圆采样子集

该采样过程需要在子集 $X_{\text{ellipse}} \sim U(X_{\text{ellipse}})$ 内均匀地分布样本。通过将样本均匀分布在半径为 n 的球上,然后将它们转移到椭圆子集 $X_{\text{ball}} \sim U(X_{\text{ball}})$ 来实现。

$$x_{\text{ellipse}} = L x_{\text{ball}} + x_{\text{center}} \quad (6)$$

$$x_{\text{center}} = \frac{x_{f1} + x_{f2}}{2} \quad (7)$$

x_{center} 为椭球的中心, x_{f1}, x_{f2} 为椭圆子集的焦点, X_{ball} 是单位 r 球内的样本。

$$X_{\text{ball}} = \{x \in X \mid \|x\|_2 \leq 1\} \quad (8)$$

使用超椭球内迭代采样方法,初始路径长度与

后续迭代生成的路径长度相比较,后者较短。随着迭代次数的增加,超椭球的体积逐渐缩小,导致路径长度不断减小。最终,算法能够找到最优路径。椭圆采样在高维空间中展现出更显著的优势,是因为它具有更快的收敛速度和更高的搜索效率。

2.3 DBSCAN 聚类算法

经过 RRT-Connect 算法的多次规划,虽然可以避免障碍物并到达目的地,但每个规划路径过程都是相互独立的,虽然有些路径能够安全到达目标点,但它们远不是最优路径,如果将规划结果直接应用到真实的场景中,可能会付出高昂的代价。引进的 DBSCAN 算法不需要事先确定簇数,对野值不敏感,操作简单,计算速度快。DBSCAN 算法是一种典型的基于密度分布的聚类算法。该算法将聚类定义为密度连接的最大点集,可以将密度足够大的区域划分为聚类,也可以在噪声空间数据库中

找到任意形状的聚类。该算法需要预先定义半径区域和最小点。

半径区域:DBSCAN 算法用来判断数据点的相似度、接近度以及是否属于同一类别的标准距离。

最小点:标准用于选择聚类中心。如果附近点的数量小于 N ,则该点被标记为噪声点。如果附近点的数量大于或等于 N ,则将该点和附近点组合以形成聚类。然后标记点,采用递归算法不断扩展聚类。

DBSCAN 算法根据每个数据点的空间分布,将点集划分为核心点、边界点和离群点。核心点均在半径范围内,其他数据点包含的点数超过最小值;边界点在半径内,其他数据点包含少于最小数量的点;离群点既不是核心点,也不是边界点。

使用最近邻的思想,Minkowski (闵可夫斯基)距离用于测量最近距离,其表示如下:

$$D(x,y) = \sqrt[p]{(|x_1 - y_1|)^p + (|x_2 - y_2|)^p + \cdots + (|x_m - y_m|)^p} = \sqrt[p]{\sum_{i=1}^m (|x_i - y_i|)^p} \tag{9}$$

本文采用曼哈顿距离来度量样本距离,其表示如下:

$$D(x,y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_m - y_m| = \sum_{i=1}^m |x_i - y_i| \tag{10}$$

每个聚类基于密度进行分离,用每个聚类的样本均值表示聚类的质心。

改进后的 RRT-Connect 算法伪代码如下:

表 1 改进后伪代码

改进的 RRT-Connect 算法:Improved RRT-Connect($n_{init}, n_{goal}, \epsilon, \epsilon_{min}, minpts$)	
1	Initialize trees T1 and T2 with n_{init} and n_{goal} as their respective roots
2	for $k = 1$ to K do:
3	sample_theta = random value between theta_min and theta_max
4	sample_r = random value between 0 and 1
5	sample_x = center.x + semi_major_axis * cos(sample_theta) * sqrt(sample_r)
6	sample_y = center.y + semi_minor_axis * sin(sample_theta) * sqrt(sample_r)
7	sample_point = (sample_x, sample_y)
8	samples.append(sample_point)
9	$n_{rand} = \text{EllipseSampling}(n_{goal}, \epsilon)$
10	$n_{near_T1} = \text{nearest_neighbor}(T1, n_{rand})$
11	$n_{new_T1} = \text{steer}(n_{near_T1}, n_{rand})$
12	if not collides(n_{near_T1}, n_{new_T1}):
13	add n_{new_T1} to T1
14	$n_{near_T2} = \text{nearest_neighbor}(T2, n_{new_T1})$
15	$n_{new_T2} = \text{steer}(n_{near_T2}, n_{new_T1})$
16	if not collides(n_{near_T2}, n_{new_T2}):
17	add n_{new_T2} to T2
18	if n_{new_T2} is in the vicinity of goal:
19	if not collides($n_{new_T2}, goal$):
20	path = DBSCANPath($n_{new_T1}, n_{new_T2}, \epsilon, minpts$)
21	if path is not empty:
22	return path from start to n_{new_T1} + path + path from n_{new_T2} to goal
23	swap(T1, T2)
24	return null
DBSCAN Path($n_{init}, n_{goal}, \epsilon, minpts$):	
1	Perform DBSCAN clustering on the points between n_{init} and n_{goal} with parameters ϵ and $minpts$
2	Identify the core points as the path waypoints
3	if there are core points:
4	return path from n_{init} to the first core point + core points + path from the last core point to n_{goal}
5	return empty path

其中,center 为椭圆的中心点;semi_major_axis 为椭圆的半长轴;semi_minor_axis 为椭圆的半短轴;theta_min 和 theta_max 为椭圆上采样点的角度范围;random value between 0 and 1 为在区间 $[0, 1]$ 内生成一个随机值;sqrt 为平方根函数;empty list 为空列表;append 为将元素添加到列表的末尾;eps 为领域半径,用于确定点的邻域;minpts 为邻域中所需的最小点数。

3 仿真结果与分析

相关仿真实验在 MATLAB R2020a 软件中完成,实验环境配置如下:11th Gen Intel(R) Core(TM) i7-11800H@2.3GHz, RAM16GB。

为了验证改进后的算法的适用性和有效性,进行仿真实验,与 RRT 算法、RRT-Connect 算法规划的路径结果相比较。并且应用于三种仿真环境,比较三种算法的生成节点数、路径长度以及规划时间三个参数。为了各算法实验参数的一致性,三种算法相关参数设置和步长设置一致。

3.1 少障碍物环境

本环境地图采用 15×15 ,起点位置为 $[3, 3]$,终点位置为 $[13, 13]$ 。三种算法在该环境下路径搜索及规划路径(红色路线为最终规划路径,灰色路线为搜索路径)见图 5。

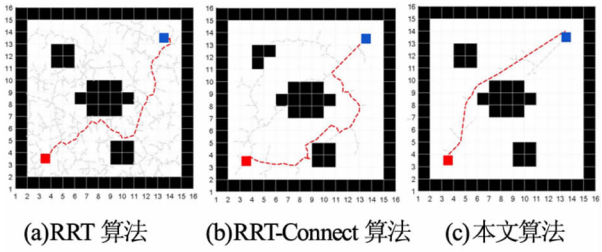


图 5 三种算法在少障碍物环境下路径搜索及规划路径

在该环境下,分别用三种算法进行 50 次仿真实验,对路径生成过程中的节点数、路径长度以及规划时间三个参数做记录取平均值,生成表 2。

表 2 仿真实验结果比较

算法	节点数	路径长度	规划时间/s
RRT 算法	438	76.2	3.12
RRT-Connect 算法	104	61.4	1.45
本文算法	78	55.4	1.26

由图 5 和表 2 可得,本文改进的算法相较于 RRT 算法和 RRT-Connect 算法,得到的规划路径更加平滑,搜索路径节点更少。在生成节点数方面,改进算法只有 RRT 算法的 17.8%,RRT-Con-

nect 算法的 75%;改进算法在路径长度方面比 RRT 算法缩短 27.3%,比 RRT-Connect 算法缩短 9.8%;规划时间也相应减短 59.6%、13.1%。在三个参数方面以及生成路径平滑度均有提升效果。

3.2 适度障碍物环境

本环境地图采用 20×20 ,起点位置为 $[2, 2]$,终点位置为 $[19, 19]$ 。三种算法在该环境下路径搜索及规划路径(红色路线为最终规划路径,灰色路线为搜索路径)见图 6。

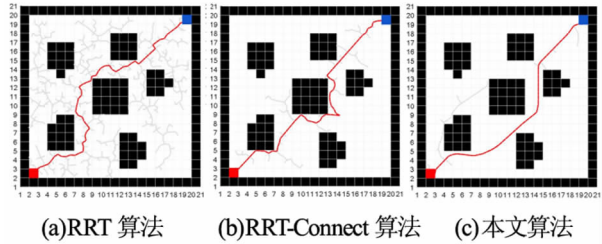


图 6 三种算法在适度障碍物环境下路径搜索及规划路径

在该环境下,分别用三种算法进行 50 次仿真实验,对路径生成过程中的节点数、路径长度以及规划时间三个参数做记录取平均值,生成表 3。

表 3 仿真实验结果比较

算法	节点数	路径长度	规划时间/s
RRT 算法	956	112.3	4.73
RRT-Connect 算法	196	86.5	2.05
本文算法	144	76.9	1.74

由图 6 和表 3 可得,在生成节点数方面,改进算法只有 RRT 算法的 15.1%,RRT-Connect 算法的 73.5%;改进算法在路径长度方面比 RRT 算法缩短 31.5%,比 RRT-Connect 算法缩短 11.1%;规划时间也相应减短 63.2%、15.1%。同样在三个参数方面以及生成路径平滑度均有提升效果。

3.3 多障碍物环境

本环境地图采用 25×25 ,起点位置为 $[2, 2]$,终点位置为 $[24, 24]$ 。三种算法在该环境下路径搜索及规划路径(红色路线为最终规划路径,灰色路线为搜索路径)见图 7。

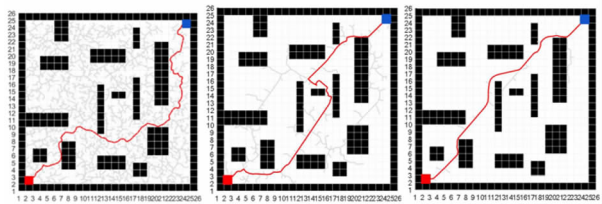


图 7 三种算法在多障碍物环境下路径搜索及规划路径

在该环境下,分别用三种算法进行 50 次仿真实验,对路径生成过程中的节点数、路径长度以及规划时间三个参数做记录取平均值,生成表 4。

表 4 仿真实验结果比较

算法	节点数	路径长度	规划时间/s
RRT 算法	1817	147.5	8.23
RRT-Connect 算法	386	121.9	3.16
本文算法	263	105.3	2.49

由图 7 和表 4 可得,在生成节点数方面,改进算法只有 RRT 算法的 14.5%,RRT-Connect 算法的 68.1%;改进算法在路径长度方面比 RRT 算法缩短 28.6%,比 RRT-Connect 算法缩短 13.6%;规划时间也相应减短 69.7%、21.2%。在三个参数方面以及生成路径平滑度均有提升效果。

从以上仿真数据可以看出,改进后的算法相较于 RRT 算法、RRT-Connect 算法数值方面均有提升,尤其是在环境复杂度较高的时候,能够逐步提升规划时间,且最终规划路径均取得不错的效果。

4 结 论

通过对 RRT-Connect 算法的改进,解决了扩展路径曲折以及随机盲目扩展等问题。通过在路径起始点和目标点上引入引力场,以及椭圆采样策略,引导随机树在椭圆区域向目标点方向扩展,减少无用扩展,加快路径规划效率。通过 DBSCAN 聚类算法对历史采样点数据进行聚类分析,去除冗余点,增强算法鲁棒性,使得到的路径接近最优解。通过在三种环境下与 RRT 算法、RRT-Connect 算法的比较,验证了改进算法的有效性和适用性。

参考文献

[1] 汤云峰. 改进智能算法在移动机器人路径规划中的研究[D]. 南京: 南京邮电大学,2021.

[2] 李学鑫. 基于 UTMD 的汽车自动驾驶的路径规划寻优算法[J]. 汽车安全与节能学报,2018,9(4):449-455.

[3] 陈敏,李笑,武交峰. 基于改进 RRT 算法的差动机器人路径规划[J]. 计算机应用与软件,2019,36(9):276-280.

[4] 尹高扬,周绍磊,吴青坡. 基于改进 RRT 算法的无人机航迹规划[J]. 电子学报,2017,45(7):1764-1769.

[5] LIU C L, PAI T W, CHANG C T, et al. Path-planning algorithms for public transportation systems[C]//2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585). Oakland, CA, 2001:1061-1066.

[6] DING H, LI Y, CHAI Y, et al. Path planning for 2-DOF manipulator based on Bezier curve and A * algorithm[C]//2018 Chinese Automation. Chinese Automation Congress (CAC). IEEE,2018:670-674.

[7] LIU T, SONG C, JIANG J. Robotic path planning based on improved ant colony algorithm[C]//International Symposium on Neural Networks. Springer, Cham, 2019:351-358.

[8] BOUNINI F, GINGRAS D, POLLART H, et al. Modified artificial potential field method for online path planning applications [C]//2017 IEEE Intelligent Vehicles Symposium (IV), 2017:180-185.

[9] KAVRAKI L E, SVESTKA P, LATOMBE J C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces[J]. IEEE Transactions on Robotics and Automation 1996,12(4):566-580.

[10] KOENIG S, LIKHACHEV M. D * lite[J]. Journal of Artificial Intelligence Research,2002,17:1-5.

[11] KUFFNER J J, LAVALLE, S M. RRT-connect: An efficient approach to single-query path planning[C]//IEEE International Conference on Robotics and Automation. IEEE, 2000, 2: 995-1001.

[12] CAO Y, ZHANG Y B, ZHOU Y. Research on obstacle avoidance path planning of spatial manipulator based on improved RRT-connect [J]. Machine Tool & Hydraulics, 2020,48 (12):177-183.

[13] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research, 2011, 30(7):846-894.

[14] 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.

[15] 游达章,杨智杰,张业鹏. 基于改进 RRT-Connect 算法的机械臂运动规划[J]. 电子测量技术,2023,46(8):112-119.

[16] ZHU Y, TANG Y, ZHANG Y, et al. Path planning of manipulator based on improved RRT-Connect algorithm[C]//2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Zhuhai, China, 2021:44-47.

[17] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Improved RRT * : optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic [C]//Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2014: 2997-3004.

[18] LUCHI D, LOUREIROS R A, Miguel Varejão F. Sampling approaches for applying DBSCAN to large datasets[J]. Pattern Recogn Lett,2019,117:90-96.