

改进 RRT* 的室内机器人路径规划算法

杨也, 倪建军, 陈一楠, 陈颜

(河海大学 物联网工程学院, 江苏 常州 213022)

摘要: 路径规划在室内机器人的应用中有无可比拟的作用; 为了提高路径规划算法收敛的速度, 综合时间消耗和路径质量方面考虑, 针对 RRT* (渐进最优快速扩展随机树) 算法的局限性, 提出一种改进的 RRT* 算法; 该算法采用目的性的设置采样点代替原算法中的随机高斯采样和引入人工势场与避障策略结合的思想, 设置目标偏向性, 引导随机树生长方向, 然后利用曼哈顿距离代替欧几里得距离作为代价估值函数, 防止陷入极小值以及一定程度上减小算法时间损耗; 实验表明, 该方法可有效平衡算法收敛时间与最佳路径的可靠性。

关键词: RRT*; 人工势场; 目标偏向性; 曼哈顿距离

Improved Indoor Robot Path Planning Algorithm of RRT*

Yang Ye, Ni Jianjun, Chen Yinan, Chen Yan

(College of Internet of Things Engineering, Hohai University, Changzhou 213022, China)

Abstract: path planning plays an incomparable role in the application of indoor robot. In order to improve the convergence speed of path planning algorithm, and considering the time consumption and path quality, an improved RRT* (Rapidly Exploring Random Tree Star) algorithm is proposed, aiming at the limitations of the algorithm. The algorithm adopts the purpose of setting instead of the original algorithm of random sampling gaussian sampling and the introduction of artificial potential field and obstacle avoidance strategy of combination of ideas, set up the target bias, guide the random tree growth direction, and then use the Manhattan distance to replace the Euclidean distance as the price valuation function, prevent a minimum loss of time and to a certain extent reduce algorithm. Experiments show that this method can effectively balance the convergence time and the reliability of the optimal path.

Keywords: RRT*; potential field; target bias; Manhattan distance

0 引言

在过去的几十年中, 室内机器人已经越发的和人类生活息息相关^[1]。它们在各种企业、家庭中都日益占着较大的比重且有着不同的任务, 例如目标的探测、移物、清洁等。若想要完成这些任务, 首先需要考虑的是如何成功到达目标节点, 这也是室内机器人路径规划的核心问题^[2]。它是确保室内机器人可以充分利用先验信息, 同时安全有效地完成任务的基础。为了保证移动机器人正确执行各种任务, 高效且实用的路径规划算法研究十分必要。

在室内机器人的应用领域, 通常避障包含于路径规划问题。最佳路径是起始点和目标点之间的合理路径^[3-4]。为了获得合理的路径, 算法需要考虑多方面因素。考虑多方面因素的目的是找到一种平衡各方因素的方法, 这也是优化算法可以实现的。优化算法不仅可以确保任务的高效完成, 还可以确保消耗的成本尽可能的少。因此, 路径规划

问题也可以转化为具有约束条件的数学优化问题。在这些优化问题中, 一些具有启发式的算法, 如 RRT* 算法, 启发式 A* 搜索法, PRM 算法等, 往往起着至关重要的作用。这些算法在一些较为简单的环境下都能较好的完成路径规划且结果都是比较优良的路径, 但在复杂环境情况下, 算法的收敛速度都急剧变差, 更不用说能否找寻到高质量的路径。

在上述算法中, 很多路径规划算法如栅格法, 可视图法, A*, D* 等算法虽然有各自的优势, 但是难以考虑移动机器人非完整约束限制。快速随机搜索树算法可以有效地考虑到移动机器人的非完整约束限制并且可以有效的搜索整个解空间, 从而快速得到路径。RRT* 算法是主要算法之一, 也是完成度较高的一种算法。其作为一种基于 RRT 算法改进的算法, RRT* 算法保证了计算复杂度和渐进最优解, 但 RRT* 的缺点任然很明显: 在快速找出初始路径后, 伴随着采样点的不断增加, 算法需要不停地进行优化, 需要花费大量的时间在算法收敛上。因此, 它的算法收敛时间任然是一个较为突出的研究点, 仍有优化的空间。

为了在一定程度上克服此缺点, 优化该算法。本文算法替换了原算法的随机采样, 使采样点具有目的性,

收稿日期: 2019-11-22; 修回日期: 2019-12-13。

作者简介: 杨也(1996-), 男, 江苏宿迁人, 硕士研究生, 主要从事机器人控制、视觉 SLAM 方向的研究。

倪建军(1978-), 男, 教授, 主要从事机器人、智能感知与信息处理方向的研究。

紧接着引进人工势场思想以及避障策略相结合引导随机树快速扩展和生长方向，同时我们采用曼哈顿距离代替欧几里得距离作为代价估值函数。通过仿真实验进行验证，本文算法比原算法拥有着更小的时间代价和更高的路径质量。

1 传统 RRT* 算法

RRT* 算法是 MIT 的 Sertac 和 Emilio 在 2010 年提出的一种基于随机采样渐进最优的快速路径规划方法，其也是 RRT（快速扩展随机树）算法的改进版^[5-7]。尽管 RRT 算法是一个相对高效率，同时可以较好的处理带有非完整约束的路径规划问题的算法，并且在很多方面有很大的优势，但是 RRT 算法并不能保证所得出的可行路径是相对优化的。因此许多关于 RRT 算法的改进也致力于解决路径优化的问题，RRT* 算法就是其中一个。RRT* 算法的主要特征是能快速的找出初始路径，之后随着采样点的增加，不断地进行优化直到找到目标点或者达到设定的最大循环次数。RRT* 算法是渐进最优化的，也就是随着迭代次数的增加，得出的路径是越来越优化的，而且永远不可能在有限的时间中得出最优的路径。而 RRT* 算法与 RRT 算法相比贡献主要在于新节点的父节点重选过程以及对新节点领域内节点重布线随机树的过程。

RRT* 的主要思想是通过随机采样生成一个随机扩展树，在新产生的节点 n_{new} 附近以规定的范围内寻找“邻居”，这些“邻居”作为替换 n_{new} 的候选节点。依次计算候选节点到起点的路径代价，同时加上 n_{new} 到每个候选节点的路径代价。选出候选节点中作为 n_{new} 父节点后，路径代价相对最小的节点，更换 n_{new} 的父节点。

重选父节点的具体步骤如下文所示，如下图 1 (a)、(b) 所示，分别为新节点 c 重选父节点和新节点邻域内节点 {a, b, e} 重布线过程，其中 n_{init} 为起始起点， n_{near} 为距离随机点最近的随机树节点。计算新节点 c 的原来路径代价，即新节点 c 通过节点 d 再到初始节点的路径代价，由图 1 (a) 可以得到原来路径代价为 14。再分别计算 c 节点以邻域内其它节点 a, b, e 为父节点到初始节点的路径，由图 1 (a) 回溯法可以计算出 c 节点通过 a 节点再到达初始节点的路径代价为 11；c 节点通过 b 节点回溯到初始节点的路径代价为 12；c 节点通过 e 节点再回溯到初始节点的路径代价为 16。所以新节点 c 将和 d 节点断开连接并以 a 节点为新的父节点进行连接。新节点 c 重选父节点后的连接图如图 1 (b) 所示。每当生成新节点后，通过重选父节点，是否能使得某些节点的路径代价变小，其意义便是减小路径代价。

在 n_{new} 节点更换新的父节点后，为进一步使得随机树节点之间的代价小，需要为随机树重新布线。即若 n_{new} 节点附

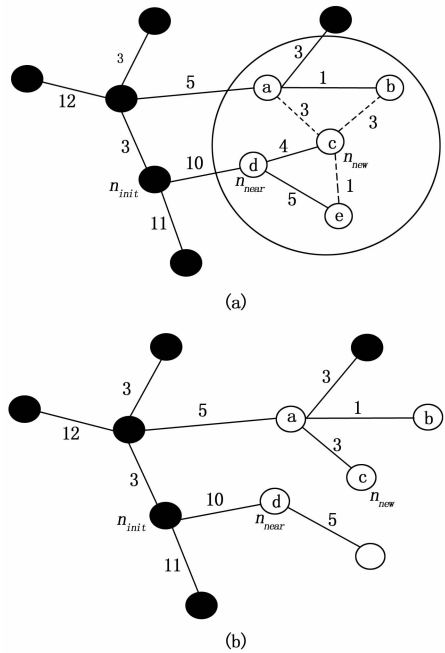


图 1 n_{new} 节点重选父节点过程

近的“邻居”节点以新节点为父节点可以减小路径代价，则进行重布线。重布线过程如下图 2 (a)、(b) 所示。

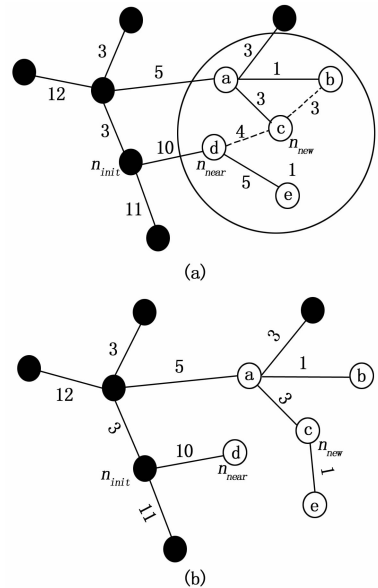


图 2 n_{new} 节点附近重布线过程

如图 2 (a) 中所示，c 节点为新节点，{d, e, b} 为新节点 c 邻域内节点。初始节点到达 d 节点的路径代价为 10；初始节点到达 b 节点的路径代价为 9；初始节点到达 e 节点的路径代价为 15。分别计算 d, e, b 节点以 c 节点为父节点后的路径代价，d 节点通过 c 节点回溯到初始节点的路径代价为 15；b 节点通过 c 节点回溯到初始节点的路径代价为 14；e 节点通过 c 节点回溯到初始节点的路径代

价为 12。通过上述计算可知存在 e 节点以 c 节点为父节点可以减小路径代价, 对 e 节点进行重布线。重布线结果如图 2 (b) 所示。

重新选则父节点作用是使新生成的节点路径代价尽可能的小, 重布线是使生成新节点后减少随机树的冗余分支, 减小搜索路径代价。如果以整体的眼光看, 并不是每一个重新布线的节点都会出现在最终生成的路径中, 但在生成随机树的过程中, 每一次的重布线都尽可能的为最终路径代价减小创造机会。

2 改进 RRT* 算法

虽然 RRT* 渐进最优且带有启发式搜索, 但该算法在收敛时间上仍然代价较大^[8]。针对这一突出问题, 本文改变了原算法中的随机采样方法, 并将人工势场思想和避障且不影响随机树扩展的策略相结合, 使随机树扩展速度加快且具有方向性, 减少算法收敛时间。

2.1 采样点的选取

RRT* 中采样点的生成一般依赖于随机高斯采样, 仍然具有很大的随机性, 这也是该算法可以优化的一点。本文算法将更具目的性的选取采样点。因为在地图中, 起始点和目标点都属于先验知识^[9]。因此在此基础上, 本文算法直接定义起始点和目标点的中间点为我们所需要的临时随机点。紧接着需要判断该临时节点处是否恰好存在障碍物, 即碰撞检测。若不存在障碍物, 则该临时随机点就设为随机点 n_{rand} 。若存在障碍物, 则需要重新选取采样点。通过上述临时采样点做垂直于起点和终点连接线的直线 L , 在直线上两端分别取距离上述临时点为 $m * s$ 的点为新的采样点, 其中 s 为步长, m 为整数 (超出障碍物的最小整数), 如图 3 所示。

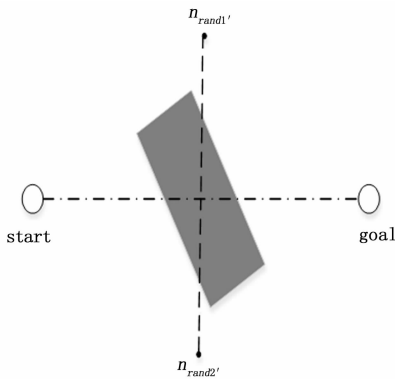


图 3 采样点的选取

采样点选取的具体策略为, 起始点坐标为 (x_{init}, y_{init}) , 终点坐标为 (x_{goal}, y_{goal}) , 垂直于起点和终点连线的直线 L 公式为:

$$y = k(x - \frac{x_{init} + x_{goal}}{2}) + \frac{y_{init} + y_{goal}}{2} \quad (1)$$

其中: 斜率 $k = -\frac{(x_{goal} - x_{init})}{(y_{goal} - y_{init})}$ 。取直线 L 上距离中点

$m * s$ 远的点为暂时的新的随机点, m 初始为 1, 取到的随机点再与障碍物进行碰撞检测。若两个候选随机点都仍处于障碍物中, 将 m 的值累加 1 后重新取随机点后, 继续进行碰撞检测, 直到寻找到处于障碍物外的候选随机点停止, 此时的随机点不再是临时随机点而是本文算法所需的随机点 n_{rand} 。若两个候选随机点进行碰撞检测后都不处于障碍物里, 随机选择一个即可。

本文算法中新的采样点的选取比原算法更具有目标性, 也多了约束性, 可以更容易的确定随机点, 减少了算法的随机性, 同时也加快了算法的收敛。

2.2 新节点的生成

人工势场算法的基本思想是, 移动机器人在虚拟的环境力场中朝着目标运动^[10]。目标点对移动机器人有着吸引力的作用, 而障碍物对其有着排斥力作用, 在二者的合力作用下使机器人向着目标点方向移动^[11]。

本文算法将人工势场与避障策略的思想融入新节点的生成策略, 引导随机树往目标点方向扩展^[12]。以此来减少 RRT* 算法随机树扩展、搜索的随机性, 减少不必要的冗余计算和路径规划所需要的时间代价。

当采样点为起始点和目标点中点时, 即所取采样点处不存在障碍物, 随机树直接往采样点方向以步长 s 扩展; 当采样点不在起始点与目标点的中点时, 引入目标点具有吸引力的思想后, 随机树的生长方向将不再仅仅由随机点决定, 而是由最近随机树节点 n_{near} 到目标点的方向以及其到随机点方向共同决定, 以此来引导随机树往目标点扩展。如图 4 所示, 其中 n_{rand} 为随机点, n_{goal} 为目标点, s 为随机树扩展步长, g 是大于零的引力系数, n_{new} 为目标点和随机点共同作用下生成的新节点。新节点明显具有向目标点靠近的偏向性。

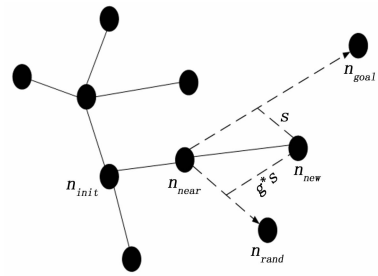


图 4 带目标引力的新节点生长

若随机点 n_{rand} 为起始点 n_{init} 与目标点 n_{goal} 连线的中点, 即采样点所处位置不存在障碍物, 则新节点 n_{new} 节点的坐标公式仅由随机点确定, 其沿随机点方向分布的新节点公式如下所示:

$$n_{new} = s * \frac{n_{rand} - n_{near}}{\|n_{rand} - n_{near}\|} \quad (2)$$

若随机点 n_{rand} 不在起始点 n_{init} 与目标点 n_{goal} 的连线上, 对于本文算法引入目标点对新节点引力作用后, 可以得到新节点的公式如下所示:

$$n_{new} = s * \left(\frac{n_{rand} - n_{near}}{\|n_{rand} - n_{near}\|} + g * \frac{n_{goal} - n_{near}}{\|n_{goal} - n_{near}\|} \right) \quad (3)$$

由上述图 4 和式 (3) 能看出快速扩展随机树的生长具有着一定的方向性, 从而加快随机树的生长与搜索。

紧接着再将避障策略融入新节点的生成中。在原算法中, 将随机点 n_{near} 与新节点 n_{new} 的连线进行碰撞检测。若不能规避障碍物, 则进行重新选择新的随机点, 再进行新的节点生成; 若可以成功规避障碍物, 则随机树进行扩展。这就导致随机树的新节点和随机树的扩展速度缓慢。本文算法对原算法新节点的选取的加入了避障策略, 如图 5 所示, 其中 s 为步长。

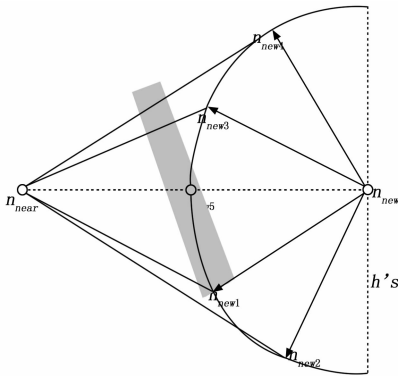


图 5 新节点的生成与避障策略

当新节点 n_{new} 与最邻近节点 n_{near} 的连线与障碍物进行碰撞检测后, 若检测结果为不存在障碍物, 直接按照文章上述方法, 将最近邻节点 n_{near} 作为新节点 n_{new} 的父节点直接进行随机树扩展; 若检测结果为存在障碍物, 则做以临时新节点 n_{new} 为圆心, $h * s$ 为半径的半圆 (h 为顺利通过碰撞检测的最小正整数), 并且半圆的直径垂直于最近邻节点与临时新节点的连线。将半圆平均划分为 6 等份, 分别选取圆周上五个采样点 n_{new1} 、 n_{new2} 、 n_{new3} 、 n_{new4} 、 n_{new5} 作为候选新节点。分别将这五个候选新节点与最近邻节点 n_{near} 的连线与障碍物进行碰撞检测, 选取候选新节点中无碰撞且最靠近最近邻节点 n_{near} 的采样点为新的新节点 n_{new} 。若这五个候选新节点在与障碍物进行碰撞检测后, 任然处于会碰撞状态, 则将初始值为 1 的 h 自动加 1, 直到这五个候选节点中出现可以顺利通过碰撞检测的节点, 此时将该候选新节点设置为新的新节点 n_{new} 。紧接着便按照原算法中的方法将最近邻节点作为新节点的父节点进行随机树扩展。

此后, 新节点的选取就按上述方法进行, 比原算法中先返回重选随机点阶段再选取新节点要更加的快速性。

2.3 改进估价函数

RRT* 算法中将启发式函数用作估值函数减少不必要的采样。本文算法在这将曼哈顿距离代替欧氏距离作为代价函数。由式 (4)、(5) 分别可知点 $p(x_1, y_1)$ 与点 $q(x_2, y_2)$ 欧氏距离表示该两点之间的直线距离, 曼哈顿距离表示该两点横纵坐标之差绝对值的和, 由公式的计算方法即可看出曼哈顿距离在计算方便方面比欧式距离更加出众, 也可以避免浮点运算, 从而加快运算速度。

$$dis = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4)$$

$$dis = |x_1 - x_2| + |y_1 - y_2| \quad (5)$$

将上述 2.1 节、2.2 节、2.3 节的改进方法带到 RRT* 算法中, 具体凡人算法步骤如下所示:

- 1) 初始化起点、终点以及障碍物。
- 2) 按本文算法中采样点选取方法选取采样点 n_{rand} 。
- 3) 找到距离采样点最近的节点 n_{near} 。
- 4) 按文中 2.2 节中节点生成的方法生成新节点 n_{new} 。
- 5) 通过最小路径代价判断新节点是否需要重选父节点 (规定范围内)。

6) 同理, 通过最小路径代价判断, 新节点邻域内的“邻居”节点是否需要重布线。

7) 判断新节点与目标点之间的距离是否小于步长 s 。是, 路径规划到此结束; 否, 重新回到步骤 2)。

3 实验结果与分析

为了验证本文算法对 RRT* 改进的有效性, 本文仿真实验在 Windows10 环境中使用 8G RAM 和 i7-7700HQ 2.80 GHz CPU 64 位操作系统的笔记本电脑上进行了实验。文中算法采用 MATLAB 编程实现。

3.1 仿真结果图分析

为了验证本文对算法的改进是否有着较为直观的体现。本文以简单环境和复杂环境两种布局结构作为仿真地图, 方形区域为机器人不可行走的障碍物区域, 空白区域为机器人的可行区域。由图 6 的 (a) 图、(b) 图可以明显的看出, 在较为简单的环境背景下, 本文算法显然比原算法在冗余树枝方面明显减少, 且树枝具有一定的方向扩展性; 而且从路径质量上能看出, 本文算法更加平滑更加最优。

同理, 由图 7 中的 (a) 图、(b) 图, 可以较明显的发现改进后的算法比原算法的随机树扩展更具有目的性, 冗余树枝的数量明显较少, 这在一定程度上可以减少搜索时间, 从而加快算法的收敛速度, 并且路径的质量也有明显的提高。

3.2 算法定量比较分析

下面统计上述改进后算法的实际运行时间以及最终规划的路径长度, 以此来做一个定量的比较。两种环境下,

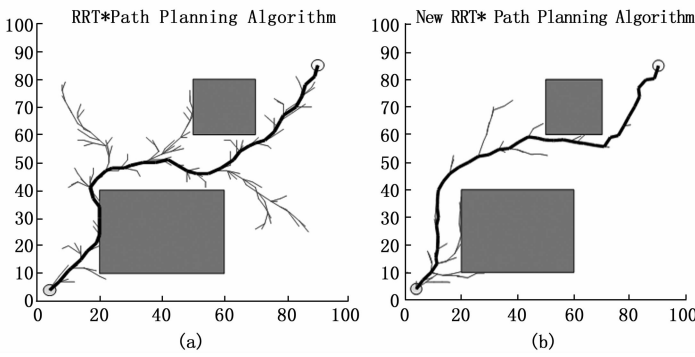


图 6 简单环境下算法改进前后对比

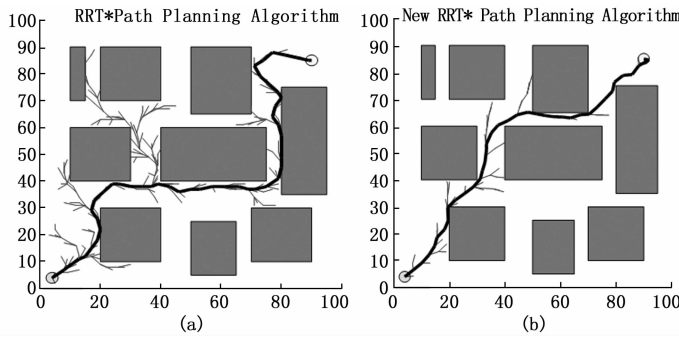


图 7 复杂环境下算法改进前后对比

每个算法运行 5 次进行试验。由表 1 在简单的背景环境下的对比可以看出本文算法所需要的时间比原算法所需时间平均减少了 60%；在路径长度方面，本文算法也比原算法要短。

表 1 简单环境下算法的定量比较

组别	RRT* (时间/路径)	本文算法(时间/路径)
1	22.248323s/142.1272	8.584845s/130.8751
2	45.086642s/152.7587	12.211196s/133.2398
3	25.762497s/149.8541	18.576491s/137.1317
4	44.622528s/146.1446	17.043284s/137.3330
5	35.780462s/140.6962	9.499641s/140.6305
平均	34.700090s/146.31616	13.1830914s/135.84202

表 2 复杂环境下算法的定量比较

组别	RRT* (时间/路径)	本文算法(时间/路径)
1	32.676581s/131.7396	49.503957s/137.0368
2	51.805314s/152.9709	13.152101s/135.8606
3	95.056019s/170.9324	19.919449s/138.4692
4	146.205977s/160.6049	28.621783s/138.5017
5	34.509431s/136.9671	63.943989s/141.5161
平均	72.0506644s/150.64298	35.0282558s/138.27688

由表 2 中在复杂环境背景下的对比数据可以看出本文算法的运行时间明显比原算法运行时间要小很多，粗略算

一下可得本文算法在复杂环境下所需时间比原算法所需时间要减小了 50% 左右；而在路径长度方面，本文算法最终规划的路径要比原算法规划得到的路径要短。

4 结束语

本文所研究的 RRT* 路径规划算法，针对 RRT* 算法存在的局限性，对该算法进行了改进。首先改变随机点的随机性，有目的性的设置采样点；在随机树新节点生成中，加入目标对其的吸引作用和新节点避障策略，减少因为随机性和随机树慢速生长而提高的时间成本；文中启发式代价估值函数采用曼哈顿距离可以加快运算。实验结果表明，本文算法不仅加快了算法的收敛时间，且规划的路径质量得到保障，是一种有效的路径规划算法。

参考文献:

- [1] 陈建平, 任 斌, 张会章. 人工智能在智能机器人领域中的研究与应用 [J]. 东莞理工学院学报, 2008, 15 (3): 33-37.
- [2] 王小伟. 室内自主导航轮式机器人路径规划策略研究 [D]. 合肥: 合肥工业大学, 2018.
- [3] Du M, Chen J, Pan Z, et al. An improved RRT-based motion planner for autonomous vehicle in cluttered environments [A]. IEEE International Conference on Robotics & Automation [C]. 2014.
- [4] Noreen I, Khan A, Ryu H, et al. Optimal path planning in cluttered environment using RRT* - AB [J]. Intelligent Service Robotics, 2017.
- [5] Gammell J D, Srinivasa S S, Barfoot T D. Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic [Z]. 2014.
- [6] 单宝明, 周培培. 基于改进人工势场法的机器人路径规划研究 [J]. 信息技术, 2014 (1): 170-173.
- [7] 梁献霞, 刘朝英, 宋雪玲, 等. 改进人工势场法的移动机器人路径规划研究 [J]. 计算机仿真, 2018, 35 (4): 291-294.
- [8] 朱 轶. 基于改进 RRT* 算法的路径规划 [J]. 武汉理工大学学报, 2017 (2): 72-76.
- [9] 潘思宇, 徐向荣. 基于改进 RRT* 的移动机器人运动规划算法 [J]. 山西大学学报 (自然科学版), 2017 (2): 244-254.
- [10] Gammell J D, Srinivasa S S, Barfoot T D. Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic [Z]. 2014.
- [11] Webb D J, Berg J V D. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics [A]. IEEE International Conference on Robotics & Automation [C]. 2013.
- [12] 朱云虹, 袁 一. 基于改进 A* 算法的最优路径搜索 [J]. 计算机技术与发展, 2018, 28 (4): 61-65.