

doi: 10.11835/j.issn.1000-582X.2022.107

改进 RRT 算法的机器人路径规划

谭波, 罗均, 罗雨松, 胡春晖, 卓俊康, 白泽鑫, 田金涛
(重庆大学机械传动国家重点实验室, 重庆 400044)

摘要: 机器人已被广泛应用于日常生活之中。路径规划作为机器人的主要技术之一, 优秀的路径规划算法能提升机器人的工作效率、降低其使用成本, 并为研究机器人的导航打下良好的基础。RRT (rapidly-exploring random trees) 算法具有扩展性强的优点, 但存在路径长度并非最优、光滑性差等不足, 为此提出反向寻优和三次样条曲线插值以改进算法, 并在 MATLAB 和 ROS (robot operating system) 系统中仿真。结果表明: 改进后的 RRT 算法能降低路径长度, 减少节点数目, 提高光滑性, 实现了算法的有效性。

关键词: RRT 算法; 路径规划; 反向寻优; 三次样条曲线插值; ROS 系统

中图分类号: TP242

文献标志码: A

文章编号: 1000-582X(2023)09-013-10

Robot path planning based on improved RRT algorithm

TAN Bo, LUO Jun, LUO Yusong, HU Chunhui, ZHUO Junkang, BAI Zexin, TIAN Jintao
(State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, P. R. China)

Abstract: Robots have been widely used in daily life. Path planning is one of the main technologies of robots. An outstanding path planning algorithm can improve the work efficiency of robots, reduce cost, and lay a good foundation for the research of robot navigation. The RRT (rapidly-exploring random trees) algorithm, which is well known for its strong scalability, has some shortcomings, such as suboptimal path length and poor smoothness. An improved algorithm of reverse optimization and cubic spline interpolation is proposed, and then simulated in different scenarios in MATLAB. Taking the restaurant environment as an example in ROS (robot operating system), the experimental results show that the improved RRT algorithm can decrease the path length and the number of nodes, and improve the smoothness, verifying the effectiveness of the algorithm.

Keywords: RRT algorithm; path planning; reverse optimization; cubic spline interpolation curve; robot operating system

进入 21 世纪后, 随着前沿科学技术的发展, 机器人技术被不少国家列为重点发展的一项高新技术, 且相关产业被用作衡量综合国力强弱的标志^[1]。现实生活中广泛为人们服务的机器人, 其相关的路径规划问题面临着前所未有的机遇, 同时也充满着挑战, 使其成为机器人的核心技术之一^[2-3]。路径规划算法的好坏, 直接决定事先设定的任务能否圆满完成并达到理想的效果。优秀的路径规划技术不仅能减少资金投入, 降低运行时间, 而且还能提高机器人的搜索效率, 为机器人导航的研究奠定良好的基础^[4-6]。

收稿日期: 2021-10-14 网络出版日期: 2022-03-02

作者简介: 谭波 (1999—), 男, 硕士研究生, 主要从事机器人路径规划研究, (E-mail)tanbo@cqu.edu.cn。

通信作者: 罗均, 男, 教授, 博士生导师, 机械传动国家重点实验室主任, 主要从事以控制为基础的机器人运动载体对环境力载荷的抗干扰技术研究, (E-mail)luojun@shu.edu.cn。

快速扩展随机树 (rapidly-exploring random trees, RRT) 算法作为一种基于采样的路径规划算法, 地图不需要做预处理, 且可以迅速搜索。但是, RRT 也有不足之处, 譬如, 搜索时较为盲目, 在复杂或者动态环境里往往会出现计算极其复杂、所需的时间过长、易于陷入死区等情况。针对以上问题, 康亮等^[7]将滚动窗口应用于 RRT 算法, 保留了渐进最优的特点, 并且减少了运行时间, 增强了算法对未知区域的探索能力; 贾李红等^[8]提出一种融合算法, 即将人工势场法与 RRT 算法配合使用, 该算法有效地解决了 RRT 算法局部避障的难题, 而且使规划效率大幅度提升; 孙钦鹏等^[9]提出了一种适应变步长的调整方法, 对新目标点设计一种新的选择方式来改进 RRT 算法, 通过仿真实验证明, 改进的 RRT 算法在复杂和狭窄环境中的平均路径长度更短。

以上对 RRT 算法的改进, 使搜索能力进一步增强, 搜索具有目标性, 路径长度缩短或节点数目减少, 但路径光滑性依旧很差, 路径长度还是很长且不是最优, 算法有待进一步改进。因此, 笔者提出反向寻优和三次样条曲线插值改进算法, 在 MATLAB 中仿真验证并进行实例分析, 并在 ROS 系统中以餐厅为例做进一步仿真分析, 验证了改进算法的有效性。

1 RRT 算法简介

1.1 RRT 算法基本原理

LaValle^[10]在 1998 年首次提出的 RRT 算法是一种高效的路径规划算法。RRT 算法以初始的一个根节点, 通过随机采样的方法在空间搜索, 然后添加一个又一个的叶节点来不断扩展随机树。当目标点进入随机树里面后, 随机树扩展立即停止, 此时能找到一条从起始点到目标点的路径。扩展树的具体结构如图 1 所示。

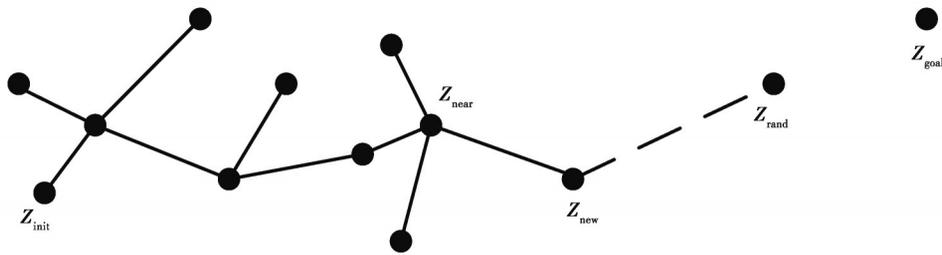


图 1 RRT 算法扩展树构造过程

Fig. 1 Construction process of RRT algorithm extension tree

算法伪代码如表 1 所示, 描述如下。起始节点 Z_{init} 作为一个根节点, 把它加入到随机树 T_k 中 (步骤 2)。从起始节点开始设置迭代次数 N , 接着随机采样得到节点 Z_{rand} , 找到与 Z_{rand} 距离最近的节点 Z_{near} (步骤 5); 然后根据节点 Z_{rand} 和 Z_{near} 判断新节点 Z_{new} 与障碍物是否发生碰撞 (步骤 6); 把新节点 Z_{new} 加入随机树中迭代, 并将 Z_{new} 和 Z_{near} 两点之间构成的路径加入随机树中, 如果新节点 Z_{near} 到达目标点或目标区域, 则完成了随机树的构建。

表 1 RRT 算法伪代码

Table 1 RRT algorithm pseudo code

步骤	伪代码
1	procedure RRT
2	$V \leftarrow \{Z_{init}\}; E \leftarrow \phi; T_k \leftarrow (V, E)$
3	for 0 : N do
4	$Z_{rand} \leftarrow \text{Sample}(i)$
5	$Z_{near} \leftarrow \text{NN}(Z_{rand}, T_k)$
6	$Z_{new} \leftarrow \text{New_config}(Z_{near}, Z_{rand})$
7	if $Z_{new} \neq Z_{near}$ then
8	$T_k.\text{add_node}(Z_{new})$
9	$T_k.\text{add_arc}(Z_{near}, Z_{new})$
10	if $Z_{new} = Z_{goal}$ then
11	return Reached

1.2 RRT算法缺点

1)稳定性差。由于扩展树节点的生长方向是随机的,所以在相同的任务下重复规划,也会出现不同的路径,导致每次规划的结果都不同,甚至偏离最优路径。

2)随机性强。因为扩展树是在整个区域内随机搜索,不受目标点的引导和启发,因此,存在大量无效的扩展节点,使算法具有很强的随机性。

3)光滑性差。RRT算法生成的路径是很多折线段相连,并不是一条光滑的曲线。

4)不适合动态环境。RRT算法规划出一条可行路径后,当环境中出现动态障碍物时,RRT算法不能有效地检测,因此,不适合动态环境下的路径规划问题^[11-12]。

1.3 RRT算法改进思路

针对RRT算法存在的缺点,总结各类改进方法,得到如下几点常见的改进思路。

1)偏向性方面:为了提高收敛速度和搜索效率,扩展树生长方向是目标点还是随机点是依靠随机概率来确定的,与原始的RRT算法相比,改进后的算法能够加速收敛。

2)节点选取方面:通过改进父节点选取的方式来改进RRT算法,改进后的算法能够提高收敛速度,减少节点数目,改善路径质量。

3)光滑性方面:通过数学中的曲率约束,使生成规划路径光滑无节点,使路径更具有可行性^[13-14]。

在上述改进思路的启发下,在节点选取方面本研究中提出了反向寻优改进RRT算法,在光滑性方面用三次样条曲线插值优化。

2 RRT算法改进

2.1 反向寻优

机器人的路径若不是最优,就会增加工作时间,降低运作效率。为了改善RRT算法的路径长度,通过反向寻优的方法来改进RRT算法,如图2所示。反向寻优方法充分应用两点之间线段最短这一数学思想,达到缩短路径长度的目的。图中A点为起点,B点为终点,原始的RRT算法扩展产生的节点为 $X_i(i=1,2,\dots,6)$,障碍物用黑色矩形表示,原始的RRT算法初步生成的路径为A与B之间的细实线,图中A与B之间的粗实线为改进后生成的路径。

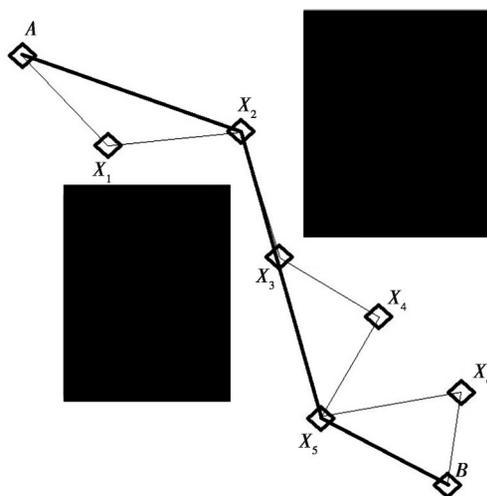


图2 反向寻优示意图

Fig. 2 Reverse optimization schematic

改进后的算法伪代码如表2所示,具体优化过程如下。首先直接连接起点A和终点B,再判断A与B连线是否被障碍物覆盖,若没有被障碍物覆盖,则此次搜索最优路径就是A与B两点间的连线;若被障碍物覆盖,则按照节点顺序找到B的父节点 X_6 ,判断A与 X_6 连线之间是否被障碍物覆盖,若A与 X_6 之间没有被障碍物覆盖,则直接把 X_6 当作A的子节点,依次连接A- X_6 -B,此路径作为搜索的最优路径;若A与 X_6 连线被障碍物覆

盖,则继续寻找 X_6 的父节点,即图中的 X_5 ,判断起点 A 与 X_5 连线是否被障碍物覆盖;按照上述方法一直进行下去,优化后得到图中 X_2 作为 A 的子节点,然后再把子节点 X_2 当作下一步算法改进的起点,不断地重复以上步骤,当整个路径优化完成时,得到如图2所示的最优路径,即算法结束。

表2 改进RRT算法伪代码

Table 2 Improved RRT algorithm pseudo code

步骤	伪代码
1	procedure Improved RRT
2	find RRT
3	for $X_1:X_N (N=6)$ do
4	link: A and B
5	if: obstacle then
6	Yes: return AB
7	No: find $X_{\text{近}}(X_i) \leftarrow B$
8	link: i and A
9	if: obstacle then
10	Yes: return AX_i
11	No: find $X_{\text{近}} \leftarrow X_i$
12	……
13	return Reached

2.2 三次样条曲线插值

RRT算法规划出的路径往往不光滑,机器人在节点处需要先减速,然后再加速前行,增加了机器人的能量消耗和工作时间。为了让路径更加光滑,使机器人平稳移动,用三次样条曲线插值优化反向寻优改进算法。三次样条曲线是一种常用的插值平滑算法,通过一系列的控制点得到一条连续平滑的曲线。

2.2.1 数学基本原理

三次样条曲线插值是将原始长序列分成若干段,每一段都是一个三次函数,在分段处左右两边函数值相等,一阶导数和二阶导数都连续^[15]。

假设有 $n+1$ 个节点,把 $n+1$ 个节点划分成以下 n 个区间:

$$[(x_0, x_1), (x_1, x_2), (x_2, x_3), \dots, (x_{n-2}, x_{n-1}), (x_{n-1}, x_n)]。$$

每个分段区间的三次多项式构造形式如下:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad i = 0, 1, 2, \dots, n-1。$$

所以, $S(x)$ 表达式是一个分段函数:

$$S(x) = \begin{cases} S_0(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3, & x_0 \leq x \leq x_1; \\ S_1(x) = a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3, & x_1 \leq x \leq x_2; \\ \vdots \\ S_{n-1}(x) = a_{n-1} + b_{n-1}(x - x_{n-1}) + c_{n-1}(x - x_{n-1})^2 + d_{n-1}(x - x_{n-1})^3, & x_{n-1} \leq x \leq x_n。 \end{cases} \quad (1)$$

总共 n 个区间,每个区间都有 a_i 、 b_i 、 c_i 、 d_i 4个未知数,所以整个三次样条曲线共有 $4n$ 个未知数,需要 $4n$ 个方程才能求解。

分段函数满足以下4个条件。

条件1:函数穿过所有已知节点,可得 $n+1$ 个方程。

条件2:节点处0阶连续,即前一段方程在节点处的函数值和后一段方程在相同节点处的函数值都相等,可得 $n-1$ 个方程。

条件3:在所有节点(除了第1个节点和最后的节点)处1阶连续,保证节点处有相同的斜率,原函数曲线上没有剧烈的跳变,可得 $n-1$ 个方程。

条件4:在所有节点(除了第1个节点和最后的节点)处2阶连续,保证节点处有相同的曲率,意味着每个点的曲率半径有定义,共 $n-1$ 个方程。

以上总共 $4n-2$ 个方程,由端点条件可以增加2个约束,增加2个方程,总共 $4n$ 个方程,由数学软件可以求解线性方程组,得到 $S(x)$ 的解^[16]。

2.2.2 端点条件

以下方程组中: $m_i = 2c_i$, $h_i = x_{i+1} - x_i$ 。

1) 自由边界

首尾两端没有任何让曲线弯曲的力,则要求解线性方程组:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & 0 \\ 0 & 0 & h_2 & 2(h_2+h_3) & h_3 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_n \end{bmatrix} = 6 \begin{bmatrix} (y_2-y_1)/h_1 - (y_1-y_0)/h_0 \\ (y_3-y_2)/h_2 - (y_2-y_1)/h_1 \\ (y_4-y_3)/h_3 - (y_3-y_2)/h_2 \\ \vdots \\ (y_n-y_{n-1})/h_{n-1} - (y_{n-1}-y_{n-2})/h_{n-2} \\ 0 \end{bmatrix} \quad (2)$$

2) 固定边界

在端点有固定的斜率,则要求解线性方程组:

$$\begin{bmatrix} 2h_0 & h_0 & 0 & 0 & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & 0 \\ 0 & 0 & h_2 & 2(h_2+h_3) & h_3 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & 0 & h_{n-2} & 2h_{n-1} \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_n \end{bmatrix} = 6 \begin{bmatrix} (y_1-y_0)/h_0 - A \\ (y_2-y_1)/h_1 - (y_1-y_0)/h_0 \\ (y_3-y_2)/h_2 - (y_2-y_1)/h_1 \\ \vdots \\ (y_n-y_{n-1})/h_{n-1} - (y_{n-1}-y_{n-2})/h_{n-2} \\ B - (y_n-y_{n-1})/h_{n-1} \end{bmatrix} \quad (3)$$

3) 非节点边界

指定样条曲线的三次微分相等,则要求解线性方程组:

$$\begin{bmatrix} -h_1 & h_0+h_1 & -h_0 & 0 & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & 0 \\ 0 & 0 & h_2 & 2(h_2+h_3) & h_3 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & -h_{n-1} & h_{n-2}+h_{n-1} & -h_{n-2} \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_n \end{bmatrix} = 6 \begin{bmatrix} 0 \\ (y_2-y_1)/h_1 - (y_1-y_0)/h_0 \\ (y_3-y_2)/h_2 - (y_2-y_1)/h_1 \\ \vdots \\ (y_n-y_{n-1})/h_{n-1} - (y_{n-1}-y_{n-2})/h_{n-2} \\ 0 \end{bmatrix} \quad (4)$$

3 基于MATLAB仿真分析

将原始的RRT算法、反向寻优改进的RRT算法、三次样条曲线插值(自由边界)优化后的RRT算法在MATLAB(R2018a)中仿真分析。实验地图为二维场景图,仿真分为无障碍物、单个障碍物、复杂环境、狭窄通道4个不同的场景,这样可以使仿真结果更具有可比性,更能验证算法的有效性。起点坐标(0,0),终点坐标(750,750),搜索步长80。在下面的路径规划图中,白色区域表示算法可覆盖的区域。

3.1 无障碍物情况

在无障碍情况下,3种算法对应的结果如图3所示。没有障碍物时,在安全区域中,随机扩展树需要扩展

大量的节点数,迅速规划出了一条可行的路径,但路径长度不是最优,算法的收敛速度较缓慢,节点较多;反向寻优改进的RRT算法大大减少了随机扩展树节点的数量,找到了一条最短路径,即起点与终点间的连线段,实验结果表明,改进的RRT算法路径质量显著提高;在反向寻优的基础上,三次样条曲线插值优化的RRT算法路径长度变化不大,但光滑性变好。

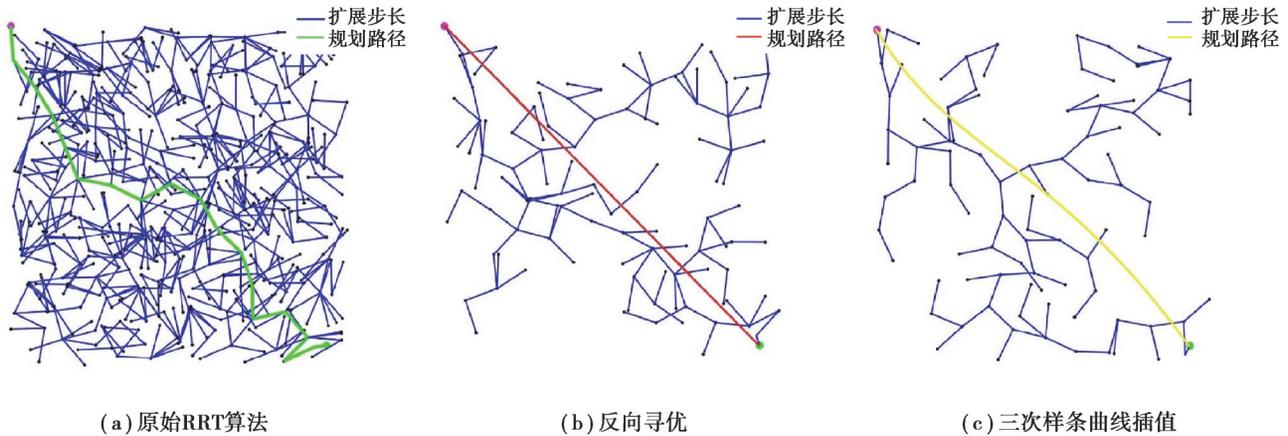


图3 无障碍物情况路径规划图

Fig. 3 Path planning for a scenario without obstacles

3.2 单个障碍物情况

在单个障碍物情况下,3种算法对应的结果如图4所示。由图可知,当环境地图中存在障碍物时,原始RRT算法能很快规划出一条路径,但是路径节点较多,光滑性很差,路径长度不是最优;反向寻优改进的RRT算法大大减少了节点数量,路径长度缩短,路径质量显著提高;在反向寻优的基础上,三次样条曲线插值优化后的RRT算法路径长度相差不大,但是节点数目进一步减少,光滑性变好。

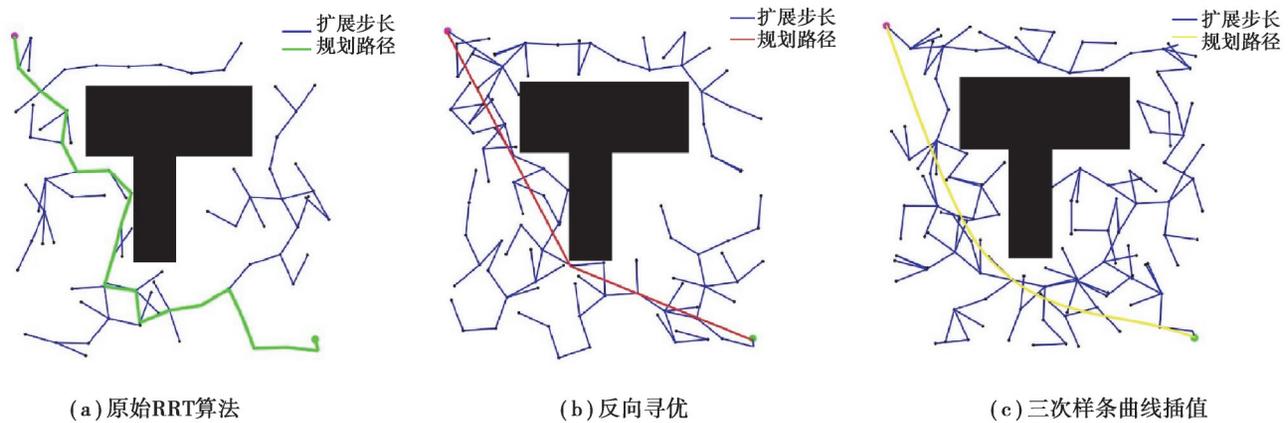


图4 单个障碍物情况路径规划图

Fig. 4 Path planning for a scenario with a single obstacle

3.3 复杂环境情况

在复杂环境情况下,即存在很多障碍物,3种算法对应的结果如图5所示。由图可知,当障碍物很多时,原始RRT算法搜索时间较长,规划出的路径节点很多,光滑性很差,路径长度不是最优;反向寻优改进的RRT算法搜索时间没有太大变化,但是路径节点数量明显减少,路径长度缩短,显著提高了路径的质量;在反向寻优的基础上,三次样条曲线插值优化后的RRT算法路径长度相差不大,但是节点数目进一步减少,光滑性变好。

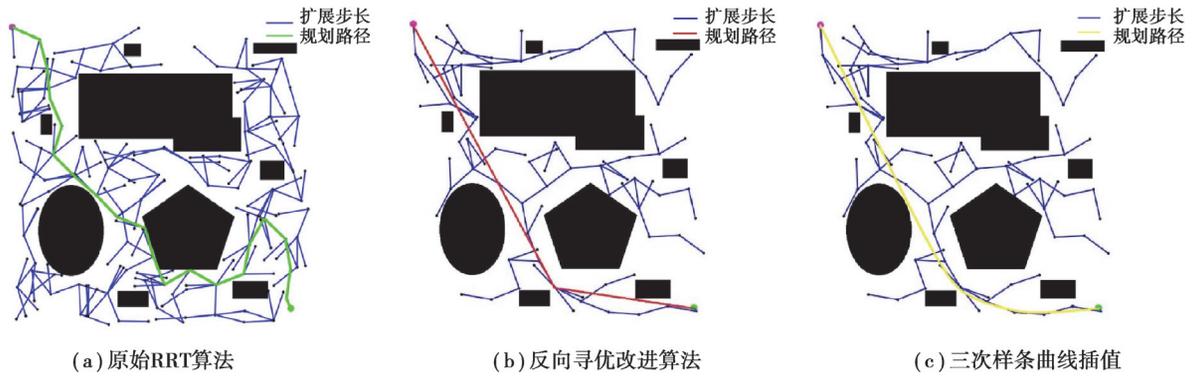


图 5 复杂环境路径规划图

Fig. 5 Path planning for a scenario of complex environment

3.4 狭窄通道情况

在狭窄通道情况下,即可搜索区域和障碍物之间构成很多的狭窄通道,3种算法对应结果的如图6所示。3种算法规划出的路径中,由于通道狭窄,扩展树会在障碍物和可搜索区域之间来回不停地碰撞,导致搜索时间较长。RRT算法扩展节点太多,光滑性非常差,路径并非最优;而反向寻优改进后的RRT算法由于改变了父节点的选取,规划出的路径长度降低,节点数目减少,光滑性相对变好;在反向寻优的基础上,三次样条曲线插值优化的RRT算法节点数目进一步减少,光滑性变好,且路径长度相差甚小。

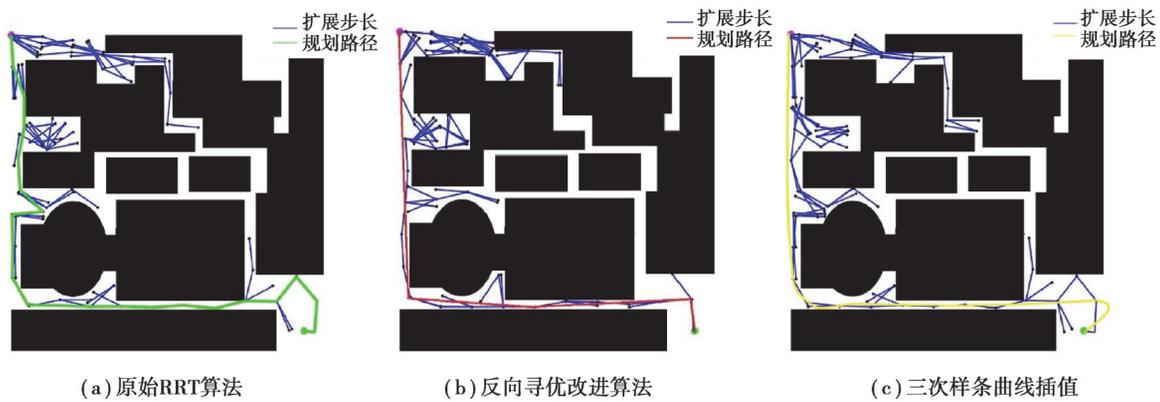


图 6 狭窄通道路径规划图

Fig. 6 Path planning for a scenario with narrow channels

为了便于进一步和原始RRT算法的相关数据进行比较,验证反向寻优改进和三次样条曲线插值优化后的算法的有效性和准确性,在仿真条件下对以上4个不同的场景进行了200次路径规划,计算对应算法的平均路径长度和平均节点数目,结果如表3和4所示。

表 3 不同算法的平均路径长度

Table 3 Average path lengths by different methods

算法	平均路径长度/mm			
	无障碍物	单个障碍物	复杂环境	狭窄通道
原始 RRT 算法	1 430.1	1 644.1	1 822.3	1 798.7
反向寻优改进	1 057.7	1 287.6	1 410.5	1 450.1
三次样条曲线	1 068.9	1 307.3	1 453.9	1 477.1

表 4 不同算法的平均节点数

Table 4 Average number of nodes by different method

算法	平均节点数			
	无障碍物	单个障碍物	复杂环境	狭窄通道
原始 RRT 算法	17.9	20.3	23.1	22.1
反向寻优改进	0	1.1	2.4	2.7
三次样条曲线	0	0	0	0

计算结果显示原始RRT算法路径长度较大,且节点数多;反向寻优改进后的算法路径长度明显减小,节点数急剧减少,说明光滑性变好,算法改进效果显著;三次样条曲线插值在反向寻优的基础上进行优化后,虽然路径长度略微增加,但是节点数归零,光滑性非常好。

结合以上改进算法验证结果分析,可得以下结论:移动机器人在路径规划中如果更偏重于路径长度的要求,可以选择反向寻优改进后的算法;如果更偏重于光滑性的要求,使移动机器人行走更加平稳,可以选择三次样条曲线插值优化后的算法。

4 基于ROS系统的仿真验证

为进一步验证改进算法的有效性,以餐厅环境为例在ROS平台进行仿真,使用gmapping实现机器人的SLAM建图,如图7所示。

启动相应的launch文件,启动gazebo并加载机器人URDF模型的餐厅环境,启动rviz并加载了gmapping建立的地图^[17-18],如图8和9所示。

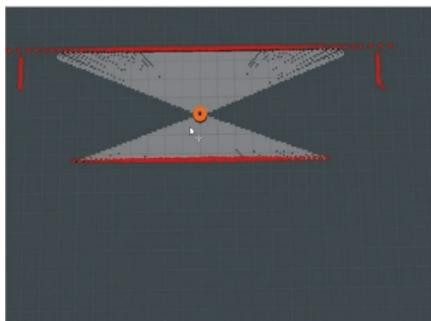


图7 rviz中的机器人状态

Fig. 7 The state of robot in rviz

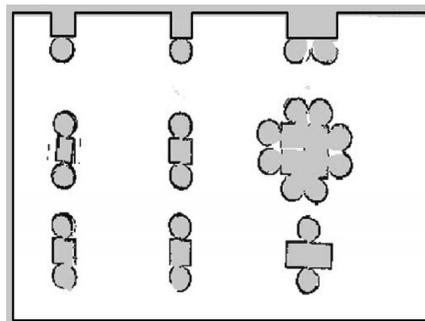


图8 基于激光雷达的gmapping建图结果

Fig. 8 Mapping results of gmapping based on lidar

根据现实餐厅障碍物情况,分为空餐厅情况、障碍物较少、障碍物很多3个不同的场景进行仿真,设置相同的起点和终点,并对它们的路径长度和运行时间进行对比分析。

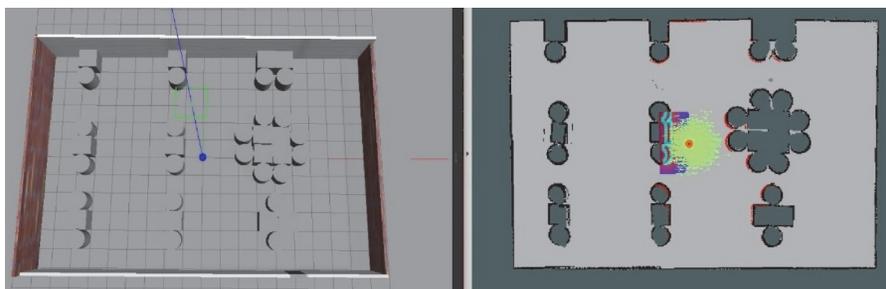


图9 机器人导航启动界面

Fig. 9 Robot navigation startup interface

4.1 空餐厅环境

餐厅内部全空时,机器人规划出的从当前位置到目标位置的有效路径如图10所示。从图中可以看出,机器人在运动过程中的路径长度较短且光滑。

4.2 障碍物较少

在障碍物较少的情况下,机器人迅速规划出一条路径,如图11所示。与空餐厅场景相比,路径仍然保持光滑,路径长度几乎不变,这是由于障碍很少,对路径长度没有影响。

4.3 障碍物很多

在障碍物很多的情况下,机器人规划出的路径如图12所示。此路径并非最优,相对于障碍物较少的情况路径长度明显增加,其原因在于障碍物太多,机器人需要不停地避开障碍物,因此路径长度增加。

为了更好地对比以上3种场景,统计路径长度和运行时间,如表5所示。

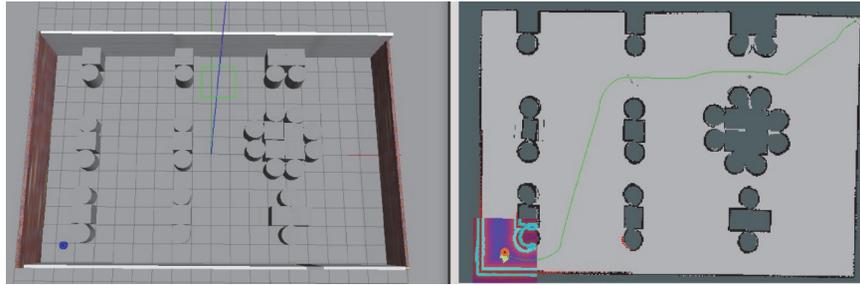


图 10 空餐厅环境下的路径规划

Fig. 10 Path planning in empty restaurant environment

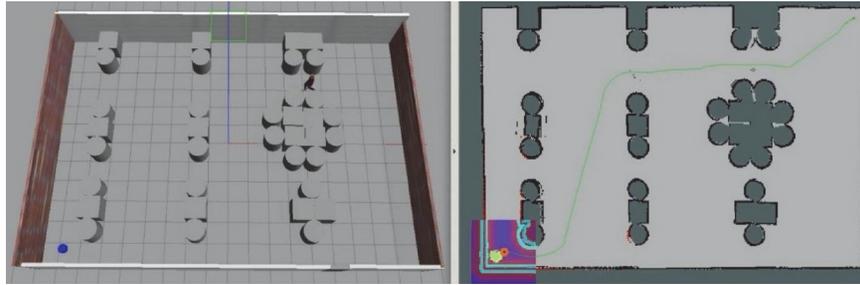


图 11 障碍物较少情况下的路径规划

Fig. 11 Path planning with a few obstacles

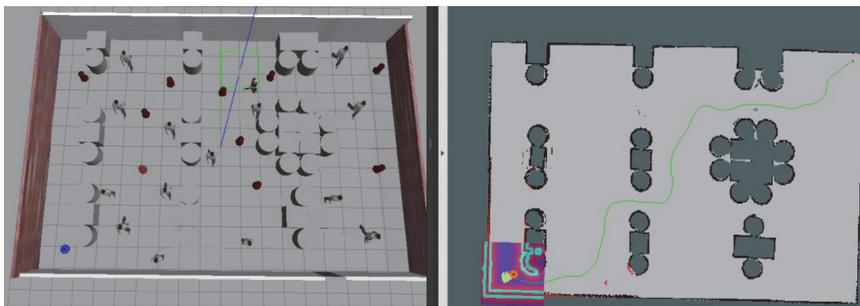


图 12 障碍物很多情况下的路径规划

Fig. 12 Path planning with many obstacles

表 5 不同场景的路径长度和运行时间

Table 5 Path length and running schedule under three scenarios

场景	路径长度/m	运行时间/s
空餐厅情况	21.58	43.98
障碍物较少	21.71	47.96
障碍物很多	24.89	60.01

分析以上图可知,障碍物较少的情况相对于空餐厅情况,路径长度和运行时间略微增加,总体变化不大;在障碍物很多的情况下,路径长度和运行时间明显增加,原因在于机器人为躲避较多的障碍物,规划的路径并非最优,导致路径长度和运行时间明显增加。

5 结 论

- 1)针对 RRT 算法存在的缺点提出了反向寻优和三次样条曲线插值改进方法。
- 2)将改进的 RRT 算法在 MATLAB 中不同的场景下进行多次仿真,结果表明,改进后的算法可以缩短路径长度,减少节点数目,提高光滑性,验证了改进算法的有效性。
- 3)以餐厅环境为例在 ROS 系统中进一步仿真,结果表明,机器人能实现路径规划功能,验证了改进算法

的有效性,后续重点研究工作将搭建实物机器人验证,未来有望应用到送餐机器人中,提高工作效率,降低人工成本,产生经济效益。

参考文献

- [1] 金碚. 中国制造 2025[M]. 北京: 中信出版集团, 2015.
Jin B. Made in China 2025[M]. Beijing: Citic Publishing Group, 2015. (in Chinese)
- [2] 王淘淘. 中国机器人产业发展报告: 我国机器人市场进入高速增长长期[J]. 今日制造与升级, 2018(8): 22.
Wang T T. China robot industry development report: China's robot market has entered a period of rapid growth[J]. Manufacture & Upgrading Today, 2018(8): 22. (in Chinese)
- [3] 中国电子学会. 机器人简史[M]. 北京: 电子工业出版社, 2015.
Chinese Society of Electronics. A brief history of robot Brief history of robot[M]. Beijing: Publishing House of Electronics Industry, 2015. (in Chinese)
- [4] Yang L, Qi J T, Xiao J Z, et al. A literature review of UAV 3D path planning[C]//Proceeding of the 11th World Congress on Intelligent Control and Automation, June 29 - July 4, 2014, Shenyang. IEEE, 2014: 2376-2381.
- [5] Karasev V, Ayvaci A, Heisele B, et al. Intent-aware long-term prediction of pedestrian motion[C]//2016 IEEE International Conference on Robotics and Automation (ICRA), May 16-21, 2016, Stockholm, Sweden. IEEE, 2016: 2543-2549.
- [6] Bakdi A, Hentout A, Boutami H, et al. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control[J]. Robotics and Autonomous Systems, 2017, 89: 95-109.
- [7] 康亮, 赵春霞, 郭剑辉. 未知环境下改进的基于 RRT 算法的移动机器人路径规划[J]. 模式识别与人工智能, 2009, 22(3): 337-343.
Kang L, Zhao C X, Guo J H. Improved path planning based on rapidly-exploring random tree for mobile robot in unknown environment[J]. Pattern Recognition and Artificial Intelligence, 2009, 22(3): 337-343. (in Chinese)
- [8] 贾李红. 基于 GPS 的双向搜索路径的研究[D]. 安徽淮南: 安徽理工大学, 2017.
Jia L H. Research on two-way search path based on GPS[D]. Huainan, Anhui: Anhui University of Science & Technology, 2017. (in Chinese)
- [9] 孙钦鹏, 李猛, 王中华. 基于改进快速扩展随机树算法的移动机器人路径规划[J]. 济南大学学报(自然科学版), 2019, 33(5): 431-438.
Sun Q P, Li M, Wang Z H. Mobile robot path planning based on improved rapidly-exploring random tree algorithm[J]. Journal of University of Jinan (Science and Technology), 2019, 33(5): 431-438. (in Chinese)
- [10] Lavalley S M. Rapidly-exploring random trees: a new tool for path planning[R]. Ames, Iowa: Iowa State University, 1998.
- [11] Chen W B, Wu X B, Lu Y. An improved path planning method based on artificial potential field for a mobile robot[J]. Cybernetics and Information Technologies, 2015, 15(2): 181-191.
- [12] Zhang C. Path planning for robot based on chaotic artificial potential field method[J]. IOP Conference Series: Materials Science and Engineering, 2018, 317: 012056.
- [13] 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, 40(2): 244-254.
Pan S Y, Xu X R. Mobile robot motion planning algorithm based on improved RRT[J]. Journal of Shanxi University:(Natural Science Edition), 2017, 40(2): 244-254. (in Chinese)
- [14] 李金良, 舒翰儒, 刘德建, 等. 基于改进 RRT 路径规划算法[J]. 组合机床与自动化加工技术, 2021(2): 22-24,29.
Li J L, Shu H R, Liu D J, et al. Path planning algorithm based on improved RRT[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2021(2): 22-24,29. (in Chinese)
- [15] Saranrittichai P, Niparnan N, Sudsang A. Robust local obstacle avoidance for mobile robot based on dynamic window approach [C]//2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, May 15-17, 2013, Krabi, Thailand. IEEE, 2013: 1-4.
- [16] Choi B, Kim B, Kim E, et al. A modified dynamic window approach in crowded indoor environment for intelligent transport robot[C]//2012 12th International Conference on Control, Automation and Systems, October 17-21, 2012, Jeju, South Korea. IEEE, 2012: 1007-1009.
- [17] 于清晓. 轮式餐厅服务机器人移动定位技术研究[D]. 上海: 上海交通大学, 2013.
Yu Q X. Research on mobile localization techniques for wheeled restaurant service robots[D]. Shanghai: Shanghai Jiaotong University, 2013. (in Chinese)
- [18] 胡春旭. ROS 机器人开发实践[M]. 北京: 机械工业出版社, 2018.
Hu C X. ROS robot development practice[M]. Beijing: China Machine Press, 2018. (in Chinese)