基于PG-RRT算法的移动机器人路径规划

郗枫飞[1] 曾 晰[1,2] 计时鸣[1] 陈国达[1] 蔡超鹏[1]
(浙江工业大学特种装备制造与先进加工技术教育部重点实验室 杭州 310023)[1]
(浙江大学流体动力与机电系统国家重点实验室 杭州 310027)[2]

摘 要 路径规划问题是移动机器人领域的重点问题,也是发展移动式机器人智能工厂的基础。快速扩展随机树算法(RRT算法)由于其良好的求解性,广泛应用于移动机器人路径规划。针对RRT算法面对复杂地图时随机采样效率低、路径重复性差的问题,提出一种基于模拟植物生长引导的RRT移动机器人路径规划算法(PG-RRT算法),提升了路径寻优的稳定性和效率。利用植物生长遵循的三大原则(向光性原则、遮挡物影响原则、负向地性原则),结合变步长技术、膨胀技术快速得到用于RRT算法采样的PG膨胀引导域,并得到最终路径。多组不同障碍物地图的仿真实验表明:相比于传统RRT算法和单一PG算法,PG-RRT算法减少了迭代次数,获得了更优的路径距离,而相比于A*算法,该算法则大大缩短了计算时间。最后通过基于ROS系统机器人平台的实车测试,验证了PG-RRT算法的实用性。

关键词 路径规划,植物生长,快速扩展随机树算法
中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.04.039

Path Planning of Mobile Robot Based on PG-RRT Algorithm

XI Feng-fei[1] ZENG Xi[1,2] JI Shi-ming[1] CHEN Guo-da[1] CAI Chao-peng[1]
(Key Laboratory of Special Purpose Equipment and Advanced Processing Technology of the Ministry of Education, Zhejiang University of Technology, Hangzhou 310023, China)[1]
(The State Key Lab of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China)[2]

Abstract The path planning problem is a vital problem in the field of mobile robots and is the basis for the development of smart factories. Rapidly-expanding random tree algorithm (RRT algorithm) is widely used in path planning because of its excellent solving performance. Aiming at the problem of low execution efficiency and poor path repeatability of the RRT algorithm when faced with complex maps, a plant growth guidance based RRT path planning algorithm (PG-RRT algorithm) for mobile robot was proposed to improve the stability and efficiency of path optimization. By using three principles followed by plant growth (Phototropism, Obstacle influence characteristics and Negative geotropism), combing variable step technique and inflation technique, the PG dilation guide field used for RRT algorithm can be obtained. Finally, the ideal path is obtained by using the RRT algorithm with random sampling characteristics. Abundant simulations show that the PG-RRT algorithm reduces the number of iterations and obtains better path distance compared to the traditional RRT algorithm and the single PG algorithm. It is noteworthy that the search efficiency of the presented path planning algorithm is improved compared with the A* algorithm. Moreover, the actual vehicle test of robot verifies the practicability of the PG-RRT algorithm.

Keywords Path planning, Plant growth, Rapidly-expanding random tree algorithm

1 引言

在未来智能工厂,移动平台的准确、高效、稳定运行是机器人与人类协作完成复杂工程任务的基础[1-2]。而路径规划技术作为实现这一任务的核心工作,近年来越来越受到众多学者的关注。路径规划要求机器人或移动平台可以在感知障碍物的同时自主避开障碍物,并通过某些性能(如时间、距离、能量)规划出机器人无碰撞的最终路径[3-4]。

当前路径规划算法在移动机器人领域得到了广泛研究,人工势场法、可视图法、A*算法、D*算法、Dijkstra算法等[5-7]经典路径规划算法被陆续提出。这些算法经过多次改进后已得到了广泛应用。但上述算法都不可避免地存在计算效率

到稿日期:2018-03-31 返修日期:2018-07-26 本文受国家自然科学基金(51875526),浙江省自然科学基金(LY18E050023),浙江省大学生科技创新活动计划(新苗人才计划)(2017R403079),2018年度浙江省科协育才工程项目(2018YCGC016)资助。
郗枫飞(1993-),男,硕士生,主要研究方向为工业机器人研发、机器人算法研究等,E-mail:kealxff@163.com;曾 晰(1984-),男,博士后,副教授,硕士生导师,主要研究方向为工业特种机器人研发、智能化装备设计与制造等,E-mail:zengxi@zjut.edu.cn(通信作者);计时鸣(1957-),男,博士,教授,博士生导师,主要研究方向为工业机器人研发、数字图像处理、超精密加工等;陈国达(1986-),男,博士,讲师,硕士生导师,主要研究方向为机器人智能装备与精密制造技术;蔡超鹏(1994-),男,硕士生,主要研究方向为机器学习、深度学习、机器视觉、数据挖掘等。

低、容易陷入局部最小的问题，面对大型复杂环境时实用性大打折扣。由此，LaValle[8] 提出了快速扩展随机树算法（Rapidly-exploring Random Tree，RRT），该算法在经典的机器人路径规划中表现出了良好的求解性。但基本 RRT 算法缺乏引导信息，在工业领域复杂的障碍物分布环境下，搜寻效率低且可执行性差[9-10]。

针对上述不足，各种改进的 RRT 算法被提出。Karaman 等提出了 RRT* 算法[11]，保证了算法的近似最优性，但是该算法的搜寻效率低。Qureshi 等[12] 将新的随机点加入树中，一定程度上解决了生成路径迂回曲折的问题。Ma 等[13] 针对城市道路环境提出了一种快速的 RRT 算法，但不能解决非结构化环境中的路径规划。Kuffner 等[14] 提出了 RRT-connect 算法用于提高节点扩展的效率，但是 RRT-connect 算法规划出的路径仍存在迂回曲折的缺点。Melchior 等[15] 结合粒子滤波算法，提出了 PRRT 算法用于局部航迹规划。Lindemann 等[16] 将泰森多边形（Voronoi）引入树的生长中，提高了 RRT 找到可行解的速度。Moon 等[17] 提出了基于对偶树的 RRT 算法，提高了算法的收敛率并实现了更加灵活的拓展。Hidalgo-Paniagua 等[18] 提出了用四棵树生长的方式，进一步提高了传统 RRT 算法的路径搜索能力并用于现代 GPU 计算。Li 等[19] 提出了一种基于活性的 RRT 算法（Li-RRT），并将其应用于自主水下航行器（AUV）的运动问题。Wei 等[20] 提出的 Smoothly RRT（S-RRT）以定向节点为目标进行 RRT 拓展，显著提高了 RRT 采样速度和效率。诚然，RRT 算法已经得到了众多改进，但其收敛效率低和所规划路径迂回曲折的问题依旧存在[21]。如何将 RRT 算法应用于室内移动机器人平台并实现稳定、高效的路径规划是当前的研究重点。

针对以上分析，文中提出一种模拟植物生长引导的 RRT 算法（以下简称 PG-RRT 算法）。该算法通过特殊设计的植物生长算法（PG 算法）快速建立 PG 引导域，RRT 算法在 PG 引导域中进一步规划得到最终路径。最后得出结论及未来研究方向。

2 引导域 PG 算法原理

本节介绍 PG 算法原理，该算法利用植物生长过程中对各种外界信息的处理机制，以植物顶芽作为基本生长单元来计算和寻找最佳路径。PG 算法中顶芽的生长主要遵循向光性原则与负向地性原则、遮挡物影响原则、变步长生长基本规则。

向光性和负向地性是植物生长的主要因素。向光性抽象为光源始终对当前顶芽的一个吸引作用（如图 1 中的 G_L），吸引力大小与当前顶芽位置的欧氏距离大小成正比。负向地性是指植物背离地面生长的特性，抽象为一个笔直指向目标点的作用，作为向光性吸引作用的补充（如图 1 中 G_G），防止顶芽在接近目标位置时，向光性作用力过小，陷入局部最小值。向光性与负向地性共同作为引导植物生长的主要引导因素。

遮挡物影响是植物生长的次要因素。遮挡物的影响主要是遮挡光源，根据植物生长的规律，植物在发芽生长遇到障碍物时，会沿障碍物“爬行”，绕过障碍物后继续向目标点生长。本算法中，该特性抽象为障碍物对当前顶芽的一个阻碍生长的效果，该效果作为植物生长的次要引导因素，与向光性原则和负向地性原则结合，共同引导顶芽的生长，生成路径。

变步长生长为快速建立引导域而设计。本算法中为提高算法的收敛速度，为 RRT 算法快速建立引导域，在前期寻径过程中，会以较大生长步长进行生长，使顶芽快速接近光源点；在接近光源点时，实行“奖惩式变步长”策略，防止顶芽在接近光源时剧烈抖动，从而提高收敛效率。

光源、负向地性、遮挡物影响等对应到二维地图上的效果如图 1 所示。

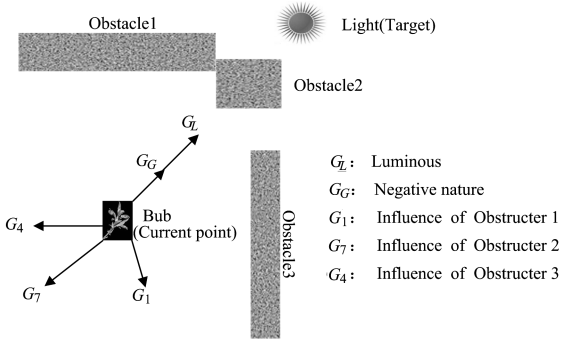


图 1 PG 算法的原理

Fig. 1 Principle of PG algorithm

3 基于植物生长引导的 RRT 算法

3.1 基本 RRT 算法

基于 RRT 算法的航迹规划以状态空间中的规划起始点为根节点，通过随机采样增加叶节点的方式生成随机扩展树。当随机树节点中包含了目标点或者目标区域的点时，随机树扩展停止，便可在随机树中找到由根节点组成的从起始点到目标点的路径。RRT 算法的扩展方式如图 2 所示。

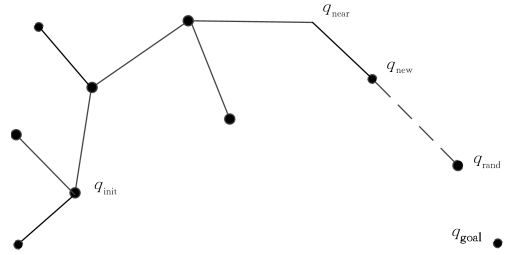


图 2 RRT 算法的扩展过程

Fig. 2 Expansion process of RRT algorithm

q_{rand} 为规划可行区域内随机选取的一个点。选择树中与 q_{rand} 距离最近的节点 q_{near}，将 q_{near} 以规定步长朝 q_{rand} 扩展，产生新节点 q_{new}。连接 q_{new} 和 q_{near} 的局部路径，加入扩展树。继续迭代计算直到 q_{new} 到达目标点或者目标点 q_{goal} 区域，则算法结束，此时可以在扩展树 T 中找到一条从起点 q_{init} 到目标点 q_{goal} 的路径。

由原理可知，RRT 算法具有其他算法所不具备的自由性和随机性，能够在复杂、狭小的地图信息中找到路径，且不易陷入局部最小值。但 RRT 算法的随机性也造成其在同一条件下的规划过程缺乏可重复性，轨迹的长度往往不可控；且在地图信息过于庞大时计算效率较低。因此，本文将前文所述的植物生长算法（PG 算法）引入到基本的 RRT 算法中，为 RRT 算法提供采样域，为移动机器人领域提出一套 PG-RRT 算法。

3.2 PG-RRT 算法

该算法的主要思想是先根据模拟植物生长算法进行路径规划，再根据得到的规划结果生成引导域；然后利用 RRT 算法在引导域中进行采样，并在最后的结果中选择最优路径。PG-RRT 算法的总流程如图 3 所示。

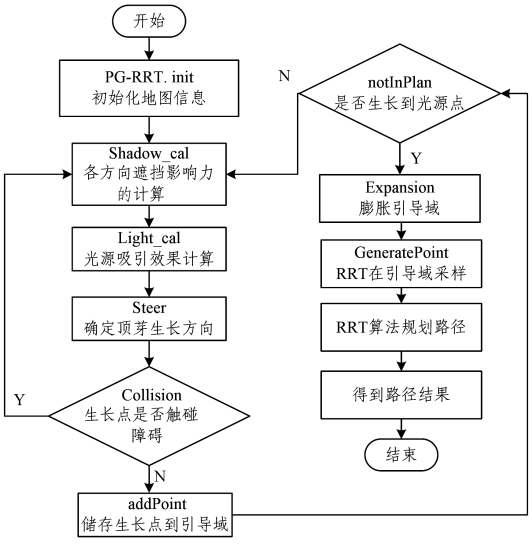


图 3　PG-RRT 算法流程图

Fig. 3　Flow chart of PG-RRT algorithm

图 3 中 Shadow_cal，light_cal，Steer 为 PG 算法的核心计算部分；步骤 Expansion 为采样域膨胀部分，是两个算法的衔接部分；步骤 GeneratePoint 为 RRT 采样过程。下面将对以上 3 段内容进行重点介绍。计算过程在基于二维栅格的地图上进行，假定每个网格具有相同属性。

3.2.1　植物生长算法的核心计算部分

(1)各方向遮挡影响力 *Shadow_cal* 的计算

首先，在栅格地图上以顶芽所在位置为出发点，考虑其可以生长的方向，如图 4 所示，以顶芽位置为中心，可供生长的栅格选择有上、下、左、右、左上、右上、左下、右下 8 种，代表了其可选择的运动方向有 8 个。在计算力影响的过程中，将分别计算这 8 个方向障碍物的影响，以获得总的影响力。

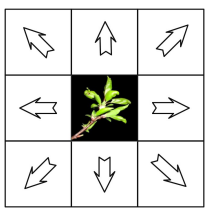


图 4　顶芽可生长的 8 个方向

Fig. 4　Eight directions top bud can grow

以单一方向(Y 轴正方向)为例：当前点位置 $Point_{\mathrm{cur}}$ 向 Y 轴正方向进行试探，记录试探长度 l_1，接触到遮挡物或地图边界时停止，并更新 l_1 的值。则可计算 Y 轴负方向的遮挡影响力为：

$$G_1 = 1.0/l_1 \tag{1}$$

若接触到的是光源点，则表明可以在这个方向直接到达光源点，则自动补足路径上的点，并跳出植物生长算法循环，得到路径。

依照此思路，依次计算出其他 7 个方向的影响效果 G_2，G_3，G_4，G_5，G_6，G_7，G_8。

将前文所述的多个方向障碍物影响力进行综合处理，合成适合后续方向计算的 X 分量和 Y 分量：

$$G_{O\_x} = \sum_{i=1}^{8}(G_i * \cos\theta_i) \tag{2}$$

$$G_{O\_y} = \sum_{i=1}^{8}(G_i * \sin\theta_i) \tag{3}$$

其中，G_{O_x} 为 8 个方向对植物生长的 X 方向的影响力之和；G_{O_y} 为 8 个方向对植物生长的 Y 方向的影响力之和；$\cos\theta_i$，$\sin\theta_i$ 为 8 个方向对 X 方向和 Y 方向的投影角度。

(2)光源吸引效果 *light_cal* 的计算

根据第 2 节所述，本算法抽象出的光强效果与距离成反比，这样在远离目标点时，目标点有强大的吸引力使顶芽快速生长；在接近目标点时，可以兼顾障碍物的影响，下面给出具体的计算过程。

光源在 X 方向和 Y 方向对顶芽的吸引力为：

$$G_{L\_x} = 1/\mathrm{ABS}(Point_{light\_x} - Point_{cur\_x}) \tag{4}$$

$$G_{L\_y} = 1/\mathrm{ABS}(Point_{light\_y} - Point_{cur\_y}) \tag{5}$$

其中，ABS()为取绝对值函数，$Point_{light\_x}$ 和 $Point_{light\_y}$ 为目标点的 x，y 坐标，$Point_{cur\_x}$ 和 $Point_{cur\_y}$ 为当前顶芽位置的 x，y 坐标。

(3)顶芽生长点 *Steer* 的确定

为避免在迭代后期计算陷入最小值，引入 X，Y 方向负向地性参数 G_{G_x}，G_{G_y}，当顶芽接近目标点时，负向地性限制光源吸引力最小值，避免陷入局部最小值，同时避免在障碍物比较复杂时出现倒退、打转、停滞不前的局面。

联合前文所述得到的障碍物遮挡效果和光吸引效果，得到 X，Y 方向的综合影响力为：

$$G_x = K_1 * G_{L\_x} + K_2 * G_{O\_x} + K_3 * G_{G\_x} \tag{6}$$

$$G_y = K_1 * G_{L\_y} + K_2 * G_{O\_y} + K_3 * G_{G\_y} \tag{7}$$

进一步得到综合影响角：

$$\varphi = \arctan\left(\frac{G_x}{G_y}\right) \tag{8}$$

根据 φ 的值可确定顶芽下一步生长的方向，以步长 p 进行生长。

为了在提高规划效率并兼顾接近光源时路径的稳定性，提出了“变步长 p”策略：远离光源点时，以较大步长寻径，尽快得到路径；接近光源点时，实行“奖惩式变步长”策略。具体计算过程如下。首先，判断生长点 $Point_{\mathrm{cur}}$ 和 $Point_{\mathrm{new}}$ 是否已经“跨越”目标点 $Point_{\mathrm{light}}$：

$$\tan\alpha = \frac{Point_{cur\_y} - Point_{light\_y}}{Point_{cur\_x} - Point_{light\_x}} \tag{9}$$

$$\tan\beta = \frac{Point_{new\_y} - Point_{light\_y}}{Point_{new\_x} - Point_{light\_x}} \tag{10}$$

计算“跨越标识符”$CrossJudgment$：

$$CrossJudgment = \tan\alpha - \tan\beta \tag{11}$$

如果 $CrossJudgment < 0$，则代表已“跨越目标点”，需要对步长实施“惩罚”措施：

$$p_{\mathrm{new}} = p - p_1 \tag{12}$$

其中，p 为原生长步长，p_1 为惩罚步长精度，p_{new} 为惩罚后的步长。

如果 $CrossJudgment \geqslant 0$，则代表“未跨越目标点”，需要进行对步长的“奖励”措施：

$$p_{\mathrm{new}} = p + p_2 \tag{13}$$

其中，p_2 为奖励步长精度。

然后再次计算 $\tan\alpha$，$\tan\beta$ 的值并计算 $CrossJudgment$，判断是否满足：

$$CrossJudgment\in(-\sigma,+\sigma) \tag{14}$$

其中，σ 为判断精度。

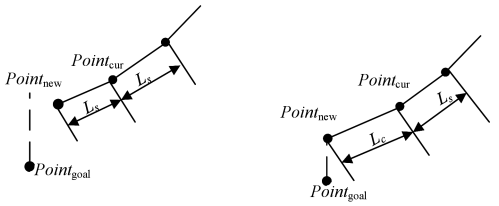
若式(14)成立，则确定新生长点 $Point_{new}$，否则继续“惩罚”或“奖励”步长，直到确定新生长点 $Point_{new}$。

奖惩式变步长算法的效果如图 5 和图 6 所示。

(a)固定步长生长 (b)变步长生长

图 5 “惩罚”效果图

Fig. 5 Renderings of “Penalty”



(a)固定步长生长 (b)变步长生长

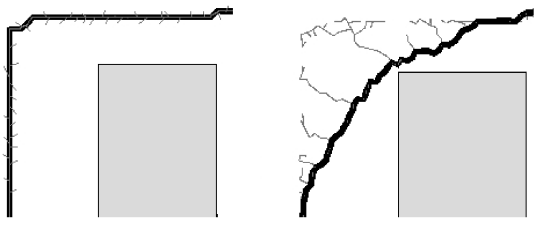
图 6 “奖励”效果图

Fig. 6 Renderings of “Rewards”

其中，L_S 为原生长步长，L_c 为通过奖惩规则修改后的步长。通过改变步长，在接近光源点时，杜绝了“翻越”光源点的情况，每次都尽可能地靠近光源点，提高后续计算的效率。这减少了路径靠近终点时的抖动问题，同时也提高了算法的收敛速度，有效提高了工业使用移动机器人进行路径规划的效率。

3.2.2 膨胀引导域 *Expansion*

根据 PG-RRT 算法原理，RRT 算法的采样范围将被缩小到采用 PG 算法获得的采样域上。考虑到 PG 算法得到路径的单一性问题，如果结果欠佳，将直接影响 RRT 采样之后的结果，也限制了 RRT 算法“随机采样”的优势，因此提出采样域膨胀策略。增加原采样域附近的自由栅格到采样域，扩大采样范围。如图 7 所示，未膨胀引导域时，由于植物生长算法所得路径距离障碍物较远，RRT 采样得到的路径欠佳(见图 7(a))；膨胀引导域后，RRT 采样范围更广，能够弥补植物生长算法规划欠佳的问题(见图 7(b))。



(a)未膨胀引导域 (b)膨胀引导域

图 7 膨胀引导域效果图

Fig. 7 Renderings of expanded guide area

3.2.3 引导域中的 RRT 采样 *GeneratePoint*

基本 RRT 算法对整个无障碍空间进行均值采样，这样虽然增加了搜索到可行路径的可能性，但是采样区域过于分散导致了很多不必要的采样，减缓了算法的收敛速度。

因此，从 3.2.2 节得到膨胀后的采样域 $Path\_Group$ 后，RRT 进行引导域偏向采样，既保证了采样的随机性，又加快了算法的收敛速度。

4 PG-RRT 算法仿真及实验结果

4.1 仿真实验

本部分对 PG-RRT 算法进行仿真实验。首先，在大小为 100×100 的简单二维地图中进行与传统 RRT 算法的采样对比实验；其次，在多个 100×100 复杂二维地图中与传统 RRT 算法、单一 PG 算法、A* 算法进行对比，地图类型包括矩形障碍物地图、圆形障碍物地图、矩形＋圆形障碍物地图。

二维环境仿真实验在 Ubuntu 的 64 位系统下进行，利用 ROS(机器人操作系统)平台中的 Rviz 仿真软件进行可视化演示，硬件及软件配置如表 1 所列。

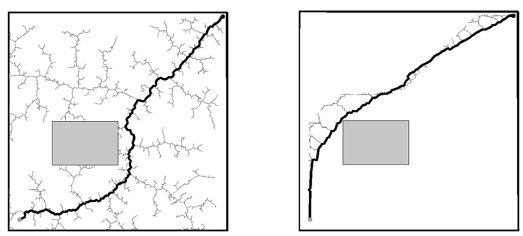
表 1 实验硬件及软件配置

Table 1 Hardware and software configuration

硬件软件配置	
处理器	Intel(R) Core(TM) i3-2350M CPU @ 2.30GHz
内存	2GB
系统版本	Ubuntu 14.04 64Bit
ROS 版本	ROS Indigo

4.1.1 采样测试

此节将对传统 RRT 算法和 PG-RRT 算法的采样进行对比实验。障碍物设置：一个矩形障碍物的长为 30 单位长度，宽为 20 单位长度，其中心在点(65,60)上。传统 RRT 算法和 PG-RRT 算法的采样结果如图 8 所示。图 8(a)代表传统 RRT 算法的寻径结果，图 8(b)代表 PG-RRT 算法的寻径结果。其中，右上角为起始点，左下角为目标点，灰色区域代表障碍物，细线代表 RRT 算法的采样范围及可视化结果，粗线代表最终的路径结果。



(a)RRT 算法采样结果 (b)PG-RRT 算法采样结果

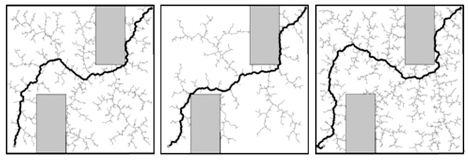
图 8 PG-RRT 与 RRT 算法的采样对比

Fig. 8 Sampling comparison between PG-RRT and RRT

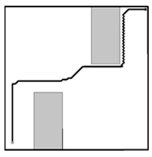
4.1.2 矩形障碍物

此节将验证 PG-RRT 算法在矩形障碍物中的可行性，并与传统 RRT 算法、单一 PG 算法、A* 算法进行对比。障碍物设置：两个矩形障碍物的长为 40 单位长度，宽为 20 单位长度，其中心分别在点(30,20)和点(70,80)上。不同算法的路径规划结果如图 9 所示。其中，图 9(a)代表传统 RRT 算法多次的寻径的结果，图 9(b)代表单一 PG 算法的寻径结果，图

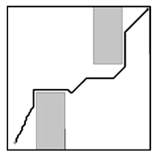
9(c)代表 A* 算法的寻径结果，图 9(d)代表 PG-RRT 算法的多次寻径结果。测试结果如表 2 所列。



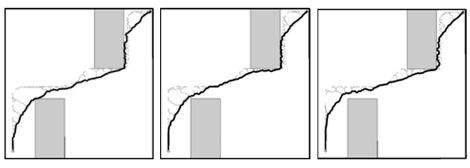
(a)RRT 算法多次寻径结果



(b)PG 算法寻径结果



(c)A* 算法寻径结果



(d)PG-RRT 算法多次寻径结果

图 9　矩形障碍物环境下路径规划结果

Fig. 9　Path planning result about rectangular obstacles

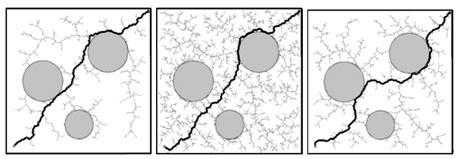
表 2　矩形障碍物环境下的测试结果

Table 2　Test results about rectangular obstacles

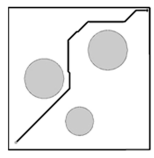
算法	迭代次数	搜寻时间/s	路径长度
RRT	2501	0.453	192
	1452	0.256	173
	3362	0.879	227
PG	130	0.000336	190.8
A*	1586	0.604	161.6
PG-RRT	944	0.206	162
	894	0.208	157
	992	0.163	159

4.1.3　圆形障碍物

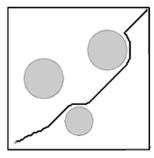
此节将验证 PG-RRT 算法在圆形障碍物中的可行性，并将其与其他 3 种算法进行对比。障碍物设置：两个圆形障碍物的半径均为 14 单位长度，其中心分别在点(30，30)和点(75，50)上；一个圆形障碍物的半径为 10 单位长度，其中心点在(50，80)上，结果如图 10 所示。



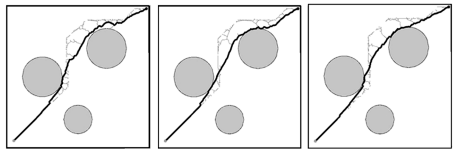
(a)RRT 算法多次寻径结果



(b)PG 算法寻径结果



(c)A* 算法寻径结果



(d)PG-RRT 算法多次寻径结果

图 10　圆形障碍物环境下路径规划结果

Fig. 10　Path planning results about circular obstacles

测试结果如表 3 所列。

表 3　圆形障碍物环境下的测试结果

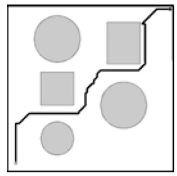
Table 3　Test results about circular obstacles

算法	迭代次数	搜寻时间/s	路径长度
RRT	1317	0.207	162
	4189	1.886	168
	2052	0.452	171
PG	91	0.000358	152
A*	1724	0.682	148.8
PG-RRT	597	0.191	142
	749	0.205	141
	696	0.196	140

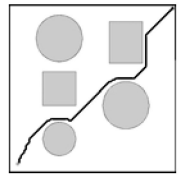
4.1.4　矩形＋圆形障碍物

此节将验证 PG-RRT 算法在矩形＋圆形障碍物中的可行性，并与其他 3 种算法进行对比。障碍物设置：两个圆形障碍物的半径均为 14，其中心分别在点(70，20)和点(30，60)上；一个圆形障碍物的半径均为 10，其中心在点(70，80)上；一个矩形障碍物的长为 25 单位长度，宽为 20 单位长度，其中心在点(30，22.5)上；一个矩形障碍物的长、宽为 20 单位长度，其中心在点(70，50)上，结果如图 11 所示。测试结果如表 4 所列。

(a)RRT 算法多次寻径结果



(b)PG 算法寻径结果



(c)A* 算法寻径结果

(d)PG-RRT 算法多次寻径结果

图 11　矩形＋圆形障碍物环境下的路径规划结果

Fig. 11　Path planning results about combination obstacles

表 4　矩形＋圆形障碍物环境下的测试结果

Table 4　Test results about combination obstacles

算法	迭代次数	搜寻时间/s	路径长度
RRT	3995	0.892	188
	2897	0.554	193
	2261	0.481	161
PG	136	0.001173	171.4
A*	1491	0.557	149
PG-RRT	854	0.237	154
	677	0.226	148
	702	0.244	149

4.2　实车测试

本节将介绍室内环境中的实车测试。使用基于 ROS 系统的移动机器人进行实车测试。参见图 12 中的一般工作空

间，本文首先建立了与上述矩形＋圆形障碍物等比例放大的地图，然后规定起始点和目标点，最后进行实车测试。同时，结合 KinectV1 深度相机，采集机器人与障碍物的安全距离，从而进行局部路径调整。

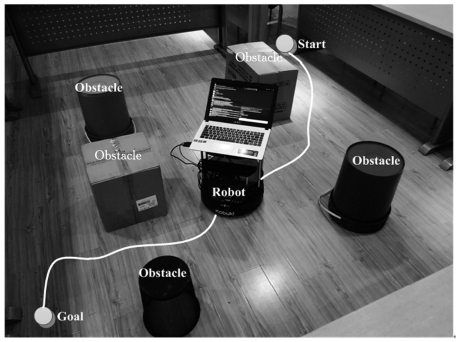
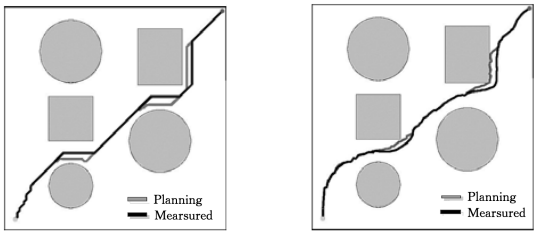


图 12 PG-RRT 算法的实车测试

Fig. 12 Real vehicle test of PG-RRT algorithm

分别利用 PG-RRT 算法和 A* 算法进行全局路径规划，同时记录机器人的实际运行路径，得到如图 13 所示的规划路径与实际测量路径对比图。



(a)A* 算法 (b)PG-RRT

图 13 规划路径与实际路径对比图

Fig. 13 Comparison of planning path and actual path

两种算法运行过程中的实际搜寻时间、实际路径长度以及路径契合度对比如表 5 所列(考虑到工厂环境下不可避免地存在局部路径规划，在实际搜寻时间上加上了局部路径调整时间)。

表 5 A* 和 PG-RRT 算法的实验测试结果

Table 5 Experimental test results for A* and PG-RRT algorithm

算法	实际搜寻时间/s	实际路径长度/增加量	路径契合度/%
A*	1.342	172/23	78
PG-RRT	1.284	164/17	82

5 讨论

针对采样测试：对比图 8 中 RRT 算法和 PG-RRT 算法对简单二维地图的采样结果，RRT 算法对整个地图的感知能力更强，采样覆盖面更广，但大量无用的采样会造成搜寻路径效率低且路径欠佳的问题。而 PG-RRT 算法由于 PG 算法引导域的引入，忽略了对地图其他无用信息的采集，而直接针对目标点和障碍物进行采样，大大缩短了采样时间，路径也更接近最优结果。

针对二维地图仿真实验：通过对比图 9(a)、图 10(a)、图 11(a)，传统 RRT 算法多次规划后路径差别明显，稳定性差，由图 10(a)可知，已经难以找到理想路径。通过对比图 9(b)、图 11(b)可知，PG 算法在遇到第一个障碍物时会明显抖动，这是因为“顶芽”距离光源较远，光吸引效果强，只有在靠近遮挡物时才会受到来自障碍物足够大的影响力与之平衡；图 10(b)中 PG 算法的路径结果趋于理想值，可见 PG 算法在针对圆形障碍物时比其他算法的表现更佳。A* 算法在 3 个地图上的结果较为稳定。通过图 9(d)、图 10(d)、图 11(d)，PG-RRT 算法在膨胀引导域的作用下，使 RRT 算法进行导向性采样，路径较为理想；RRT“随机采样”的特点使得 PG-RRT 算法最后的路径更趋于最优，与 A* 算法结果相比，路径结果也更加圆滑，更适合作为机器人移动路径。

图 14－图 16 为算法在 4.1.2 节－4.1.4 节中 3 个实验中找到初始可行解情况的对比柱状图。对于具有随机性的 RRT 算法和 PG-RRT 算法，取以上多次实验的平均值作为对比数据。

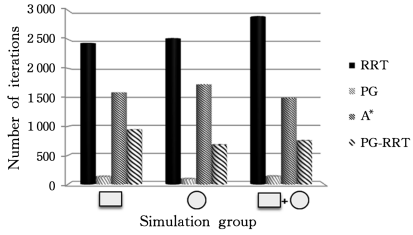


图 14 找到可行解的迭代次数对比图

Fig. 14 Comparison chart of number of iterations for finding feasible solution

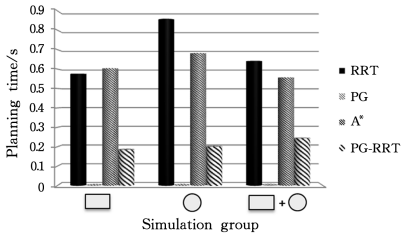


图 15 找到可行解的搜寻时间对比图

Fig. 15 Comparison chart of searchable time for finding feasible solution

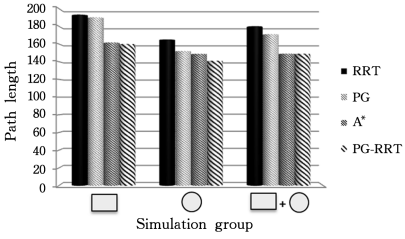


图 16 找到可行解的路径长度对比图

Fig. 16 Comparison chart of path length for finding feasible solution

RRT 算法的规划效果不可控，难以得到机器人进行路径规划的合适路径。PG 算法的计算效率最高，但与 RRT 算法均无法达到工业移动机器人的要求。而结合了 RRT 算法的 PG-RRT 算法，相比 PG 算法虽然牺牲了一部分时间优势，但但得到了更加理想的路径长度，且达到了当前广泛应用于移动机器人领域的 A* 算法水平(见图 16)，搜寻时间相比 A* 算法有显著提高(见图 15)。

针对实车测试，在利用 PG-RRT 和 A* 算法全局规划路

径后，都需要一定时间进行局部路径调整，实际测量长度略微增加。在此过程中，PG-RRT 算法展现出路径增加量较少、整体规划时间较短、路径契合度较高的特点，相对 A* 算法有更好的表现，适用于室内或工厂等封闭条件下的移动机器人路径规划工作。

结束语　文中提出一种基于模拟植物生长引导的 RRT 移动机器人算法(PG-RRT 算法)。特殊设计的基于植物生长机制的算法(PG 算法)，在向光性原则、遮挡物影响原则、负向地性原则以及变步长技术的联合作用下得到 PG 膨胀引导域；RRT 算法在引导域中进一步搜寻得到最终路径。其突出特点在于：

(1)特殊引入的负向地性原则和奖惩式变步长生长技术，在加快 PG 算法收敛速度的前提下，减缓了接近光源点时的震荡。

(2)PG 引导域的存在缩小了 RRT 采样的范围，避免了传统 RRT 算法大量的无用采样，显著提高了采样效率。

(3)膨胀 PG 引导域的应用，利用 RRT "随机采样"特点解决了 PG 算法靠近障碍物时的"爬行"现象，高概率得到接近最优解的结果。

与 RRT 算法、单一 PG 算法、A* 算法的比较表明，PG-RRT 算法能够在以高概率获得更优路径的前提下，保持稳定且理想的搜寻效率；通过基于 ROS 系统的实车测试，验证了 PG-RRT 算法对移动机器人路径规划的有效性。在未来工作中，为了将 PG-RRT 算法更好地应用于实际，将尝试研究在复杂形状障碍物环境中的移动机器人路径规划，同时研究其对动态障碍物的避障效果。该算法在机器人智能控制领域和智能工厂均会有较大应用前景。

参 考 文 献

[1] CUI M，LIU H，LIU W，et al. An Adaptive Unscented Kalman Filter-based Controller for Simultaneous Obstacle Avoidance and Tracking of Wheeled Mobile Robots with Unknown Slipping Parameters[J]. Journal of Intelligent & Robotic Systems，2018，92(3-4)：489-504.

[2] INDRI M，TRAPANI S. LAZZERO I. Development of a Virtual Collision Sensor for Industrial Robots：[J]. Sensors，2017，17(5)：1148.

[3] KIVELÄ T，MATTILA J，PUURA J，et al. Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance[C]// International Conference on Robotics in Alpe-Adria Danube Region. Springer，Cham，2017：208-216.

[4] KUWATA Y，TEO J，FIORE G，et al. Real-Time Motion Planning With Applications to Autonomous Urban Driving[J]. IEEE Transactions on Control Systems Technology，2009，17(5)：1105-1118.

[5] XU F. Research on Robot Obstacle Avoidance and Path Planning Based on Improved Artificial Potential Field Method[J]. Computer Science，2016，43(12)：293-296. (in Chinese)
徐飞. 基于改进人工势场法的机器人避障及路径规划研究[J]. 计算机科学，2016，43(12)：293-296.

[6] LV T，FENG M. A Smooth Local Path Planning Algorithm Based on Modified Visibility Graph[J]. Modern Physics Letters B，2017，31(19-21)：1-6.

[7] KRICHMAR J L. Path Planning using a Spiking Neuron Algorithm with Axonal Delays[J]. IEEE Transactions on Cognitive & Developmental Systems，2018，10(2)：126-137.

[8] LAVALLE S M. Rapidly-exploring Random Trees：A New Tool for Path Planning[R]. Ames：Computer Science Department of Iowa State University，1998.

[9] LAVALLE S，KUFFNER J. Randomized Kinodynamic Planning [J]. The International Journal of Robotics Research，2001，20(5)：378-400.

[10] DONG Y，CAMCI E，KAYACAN E. Faster RRT-based Nonholonomic Path Planning in 2D Building Environments Using Skeleton-constrained Path Biasing[J]. Journal of Intelligent & Robotic Systems，2018，89(3-4)：387-401.

[11] KARAMAN S，FRAZZOLI E. Sampling-Based Algorithms Foroptimal Motion Planning[J]. The International Journal of Robotics Research，2011，30(7)：846-894.

[12] QURESHI A H，MUMTAZ S，IQBAL K F，et al. Triangular geometry based optimal motion planning using RRT*-motion planner[C]// IEEE International Workshop on Advanced Motion Control. IEEE，2014：380-385.

[13] MA L，XUE J R，KAWABATA K，et al. A Fast RRT Algorithm for Motion Planning of Autonomous Road Vehicles[C]// 2014 IEEE 17th International Conference on Intelligent Transportation Systems. Qingdao，China，2014：1033-1038.

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

[15] MELCHIOR N A，SIMMONS R. Particle RRT for Path Planning with Uncertainty[C]// IEEE International Conference on Robotics and Automation. IEEE，2007：1617-1624.

[16] LINDEMANN S R，LAVALLE S. Incrementally Reducing Dispersion by Increasing Voronoi Bias In Rrts[C]// IEEE International Conference on Robotics and Automation，2004. ICRA，2004：3251-3257.

[17] MOON C B，CHUNG W. Kinodynamic Planner Dual-Tree RRT (DT-RRT) for Two-Wheeled Mobile Robots Using the Rapidly Exploring Random Tree[J]. IEEE Transactions on Industrial Electronics，2015，62(2)：1080-1090.

[18] HIDALGO-PANIAGUA A，BANDERA J P，RUIZ-DE-QUINTANILLA M，et al. Quad-RRT：A Real-Time GPU-Based Global Path Planner in Large-Scale Real Environments[J]. Expert Systems with Applications，2018，99：141-154.

[19] LI Y，ZHANG F，XU D，et al. Liveness-Based RRT Algorithm for Autonomous Underwater Vehicles Motion Planning [J]. Journal of Advanced Transportation，2017，2017(7)：1-10.

[20] WEI K，REN B. A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm. [J]. Sensors，2018，18(2)：1-15.

[21] CHEN J Y，SHI J，DU W Y，et al. Research on UAV Route Planning Algorithm Based on MB-RRT* [J]. Computer Science，2017，44(8)：198-206. (in Chinese)
陈晋音，施晋，杜文耀，等. 基于 MB-RRT* 的无人机航迹规划算法研究[J]. 计算机科学，2017，44(8)：198-206.