

DOI: 10.11830/ISSN.1000-5013.202404033



融合距离引导式 A^* 算法与动态窗口法的 的移动机器人路径规划

黄昱航^{1,2}, 李国刚^{1,2}, 焦启^{1,2}, 曹冬平^{1,2}

(1. 华侨大学 信息科学与工程学院, 福建 厦门 361021;

2. 华侨大学 厦门市专用集成电路系统重点实验室, 福建 厦门 361021)

摘要: 为解决移动机器人路径规划中效率低下等问题,提出一种距离引导式 A^* 算法与动态窗口法的融合算法。在改进 A^* 算法中,引入双向搜索策略,采用综合距离启发函数,利用全局路径筛选策略对路径进行优化。在得到全局最优路径的基础上,与动态窗口法结合,实现移动机器人的动态避障。结果表明:距离引导式 A^* 算法和文中融合算法在路径长度、遍历节点数目和运行时间方面实现了显著提升,能更好地满足移动机器人对路径规划的要求。

关键词: 移动机器人; 路径规划; A^* 算法; 动态窗口法; 动态避障

中图分类号: TP 242

文献标志码: A

文章编号: 1000-5013(2025)01-0087-08

Path Planning for Mobile Robots by Integrating Distance-Guided A^* Algorithm and Dynamic Window Approach

HUANG Yuhang^{1,2}, LI Guogang^{1,2}, JIAO Qi^{1,2}, CAO Dongping^{1,2}

(1. College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China;

2. Xiamen City Key Laboratory of Application Specific Integrated Circuit System,
Huaqiao University, Xiamen 361021, China)

Abstract: To solve the problem of low efficiency in mobile robot path planning, a fusion algorithm combining a distance-guided A^* algorithm with the dynamic window approach is proposed. In the improved A^* algorithm, a bidirectional search strategy is introduced, and a comprehensive distance heuristic function is adopted, utilizing a global path filtering strategy to optimize the path. On the basis of obtaining the global optimal path, it is combined with the dynamic window approach to achieve dynamic obstacle avoidance for the mobile robot. The results show that the distance-guided A^* algorithm and the proposed fusion algorithm achieve significant improvements in terms of path length, the number of traversed nodes, and running time, which better satisfies the requirements of mobile robot for path planning.

Keywords: mobile robot; path planning; A^* algorithm; dynamic window approach; dynamic obstacle avoidance

随着智能科技和自动化技术的不断进步,移动机器人的路径规划问题逐渐成为备受关注的研究热点。机器人路径规划中,Dijkstra 算法^[1]虽然被广泛使用,但在实际应用中可能会导致非最优路径的问题,同时也会出现计算复杂度较高等问题。D* 算法^[2]主要用于动态环境下的路径规划,但在静态环境

收稿日期: 2024-04-11

通信作者: 李国刚(1973-),男,副教授,博士,主要从事电子信息方面的研究。E-mail:lgg@hqu.edu.cn。

中,其效率略逊于其他算法,且计算复杂度较高。快速扩展随机树(RRT)算法^[3]是一种基于采样的算法,其主要目标是快速探索,并找到一个近似的解决方案,但并不能保证找到最短或最优路径。相较而言,A*算法^[4]通过选择合适的启发函数,能够在室内环境下高效地搜索并找到最佳路径,使其成为移动机器人全局路径规划的首选算法之一。动态窗口法(DWA)是一种常见的局部路径规划算法,根据机器人的运动学模型,对各个轨迹进行评分,从而选择最优路径,实现动态避障。

赵晓等^[5]巧妙地加入跳点搜索机制,实现长距离的跳跃,避免 A* 算法寻路过程中对冗余节点的计算,显著降低了计算的复杂度,使寻路更加快速和高效。Wang 等^[6]提出一种指数加权方法,该方法充分考虑了路况的影响因素,并通过对启发函数的改进,成功找到全局最优的路径。田海波等^[7]提出一种考虑转弯成本的估价评估搜索效率,得到相对较短的路径长度,但这种方法会导致节点扩展较多,消耗大量内存。李炯逸等^[8]提出一种针对 A* 算法在室内环境路径规划中存在问题的改进方法。辛煜等^[9]通过将 A* 算法的可搜索邻域扩展为无限个,缩短了路径长度,减少了拐点数量。王中玉等^[10]通过检测当前节点和其祖父节点之间的连线是否存在障碍物,消除路径中的不必要的节点,从而优化路径的长度和转折角度。姬鹏等^[11]通过优化搜索点、评价函数和拐点,提高检索速度和路径平滑度。程传奇等^[12]通过优化启发函数,选择关键点,减少路径规划的长度,并与动态窗口法结合,提高路径平滑性和避障能力。唐嘉宁等^[13]提出一种基于双向机制的改进 A* 算法,显著提高了路径规划的效率。基于此,本文提出一种距离引导式 A* 算法与动态窗口法相结合的融合算法^[14-16]。

1 A* 算法概述

1.1 A* 算法

A* 算法结合贪心策略和 Dijkstra 算法的特点,使用一个评价函数,通过构建和更新开放列表及关闭列表,不断选择代价最低的节点作为当前节点,直到找到目标节点或满足停止条件为止。A* 算法的代价函数是其核心要素,它能够有效地指导搜索过程朝着目标节点进行。A* 算法的代价函数 $f(n)$ 可表示为

$$f(n)=g(n)+h(n)。(1)$$

式(1)中: n 为当前节点; $g(n)$ 为起点到当前节点 n 的实际代价; $h(n)$ 为启发函数,表示当前节点 n 到目标节点的估计代价。

1.2 改进 A* 算法

为了克服 A* 算法在搜索过程中出现的遍历节点过多、路径较长、效率较低等问题,采用双向搜索策略、综合距离启发函数及全局路径筛选策略来提升效率。

1.2.1 双向搜索策略 同时从起始节点和目标节点进行搜索,加速路径规划过程。双向搜索策略具体有以下 7 个步骤。

步骤 1 在双向搜索策略中,需要初始化两个开放列表 openlist1,openlist2,以及两个关闭列表 closelist1,closelist2,分别用于存储待搜索的节点和已经搜索过的节点。

步骤 2 将起始节点加入 openlist1 中,目标节点加入 openlist2 中。同时从起始节点和目标节点开始搜索,每次从两个开放列表中选择代价最小的节点进行正向搜索和反向搜索。在正向搜索中,需要更新正向搜索节点的代价函数,并判断最优节点是否在反向搜索的路径上。若存在,则找到路径;若不存在,则将节点加入 closelist1。

步骤 3 对当前最优节点进行处理。找出当前节点的连接点 m ,若节点 m 在 closelist1 中,则搜索下一个节点 m ;若节点 m 不在 closelist1 中,则判断节点 m 是否在正向搜索的 openlist1 中,若节点 m 不在 openlist1 中,则计算该点的实际代价 $g(m)$ 、估计代价 $h(m)$ 和代价函数 $f(m)$ 。

步骤 4 节点 m 的父节点设为 n ,并将节点 m 按代价函数 $f(m)$ 从小到大的顺序添加到正向搜索的 openlist1 中。

步骤 5 判断实际代价 $g(n)$ 加上父节点 n 到节点 m 的距离是否大于 $g(m)$ 。若大于 $g(m)$,则进一步搜索父节点 n 的下一连接点;若小于 $g(m)$,则重新计算 $g(m),h(m),f(m)$ 。直至与父节点 n 的所有连接点都已搜索完毕。

步骤 6 进行反向搜索, 步骤与正向搜索一致。

步骤 7 判断步骤 4, 5 的返回值, 若返回 openlist1 与 openlist2 列表均为空, 仍未到达目标节点, 则表示寻路失败; 若当起始节点和目标节点搜索队列中的某个节点在地图上相遇, 则找到一条从起始节点到目标节点的路径。此时, 可以根据相遇节点及其前驱节点, 回溯构造出完整的路径。

1.2.2 综合距离启发函数 首先, 综合距离启发函数在计算中进行对角线方向距离的调整, 通过引入一个额外的因子平衡直线和对角线移动的代价, 确保搜索路径更加均匀和平滑, 能够更好地引导搜索方向。然后, 引入动态权重来控制搜索速度。最后, 加入当前节点和目标节点的距离与起始节点和目标节点距离的比值, 进一步优化搜索的准确性。通过对启发函数的改进, 算法可以在不同情况下灵活地调整搜索速度和寻优性能, 从而找到最优解。

综合距离启发函数公式为

$$h(n) = 1.2 \times w(n) \times d(n) - u(n), \quad (2)$$

$$d(n) = [x(n) + y(n) + (\sqrt{2} - 2) \times \min_{x,y}], \quad (3)$$

$$x(n) = |x_g - x_n|, \quad (4)$$

$$y(n) = |y_g - y_n|, \quad (5)$$

$$\min_{x,y} = \min(|x_g - x_n|, |y_g - y_n|), \quad (6)$$

$$u(n) = \frac{\sqrt{(x_g - x_n)^2 + (y_g - y_n)^2}}{\sqrt{(x_g - x_s)^2 + (y_g - y_s)^2}}. \quad (7)$$

式(2)~(7)中: (x_s, y_s) 为起始节点坐标, (x_g, y_g) 为目标节点坐标; (x_n, y_n) 为当前节点坐标; 当 $d(n) > 5$ 时, $w(n) = 2.8$, 当 $d(n) \leq 5$ 时, $w(n) = 0.9$ 。

1.2.3 全局路径筛选策略 为了删除全局路径中产生的冗余拐点, 引入全局路径筛选策略, 可有效地去除冗余拐点, 从而提升路径规划的结果。优化后的路径, 如图 1 所示。图 1 中: A~K 为算法遍历的最优节点。

原始路径为 A-B-C-D-E-F-G-H-I-J-K, 优化后的路径为 A-E-F-I-K。全局路径筛选策略具体有以下 4 个步骤。

步骤 1 获取双向 A* 算法规划路径中的每个拐点坐标 $\{p_1, p_2, p_3, \dots, p_n\}$, 将起始节点 $p_s(x_s, y_s)$ 加入拐点坐标集的首位, 把目标节点 $p_g(x_g, y_g)$ 加入拐点坐标集的末位, 得到所有关键节点, 并将其放入集合 $A\{p_s, p_1, p_2, p_3, \dots, p_n, p_g\}$ 。

步骤 2 建立一个空的集合 $B\{\text{Set}B\}$, 用于存放经过优化后的坐标点。

步骤 3 首先, 判断集合 A 中的首位坐标 p_s 与集合 A 中的末位坐标 p_g 形成的线段是否与障碍物相交; 若不相交, 则判断集合 A 中的首位坐标 p_s 与集合 A 中的倒数第 2 位坐标 p_n 形成的线段是否与障碍物相交; 若不相交, 则判断集合 A 中的首位坐标 p_s 与集合 A 中的倒数第 3 位坐标 p_{n-1} 形成的线段是否与障碍物相交。直到找到集合 A 中的首位坐标 p_s 与集合 A 中的某个坐标 p_x 形成的线段与障碍物不相交, 再将 p_s 加入集合 B 中, 并将集合 A 更新为 $\{p_x, \dots, p_n, p_g\}$ 。

步骤 4 将更新后的集合 A 重复步骤 3, 再次进行判断, 并持续更新集合 A, B。当集合 A 中的坐标数量为两个时, 将最后两个坐标加入集合 B。最后, 集合 B 为优化后的坐标点, 即优化后的路径。

路径优化策略中的线段相交检测采用向量叉积的性质进行判断, p_1, p_2 是拐点集中的两个坐标, p_3, p_4 是障碍物坐标集中的两个坐标。

判断线段是否相交的过程具体有以下 2 个步骤。

步骤 1 cp_1 是向量 $p_1 p_2$ 和向量 $p_1 p_3$ 的叉积; cp_2 是向量 $p_1 p_2$ 和向量 $p_1 p_4$ 的叉积; cp_3 是向量 $p_3 p_4$ 和向量 $p_3 p_1$ 的叉积; cp_4 是向量 $p_3 p_4$ 和向量 $p_3 p_2$ 的叉积。

步骤 2 若 $cp_1 \times cp_2 \leq 0$, 且 $cp_3 \times cp_4 \leq 0$, 说明拐点间形成的线段与障碍物点间形成的线段相交。

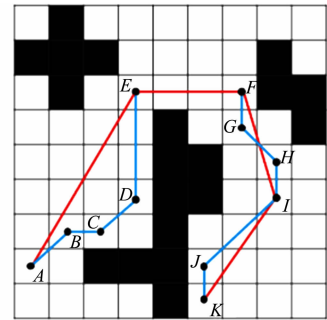


图 1 优化后的路径

Fig. 1 Optimized path

否则,说明拐点间形成的线段与障碍物点间形成的线段不相交。

2 动态窗口法

2.1 机器人运动学模型

机器人的运动学模型为

$$x(t+1)=x(t)+v_t\Delta t\cos\theta_t,$$

$$y(t+1)=y(t)+v_t\Delta t\sin\theta_t,$$

$$\theta(t+1)=\theta(t)+\omega_t\Delta t.$$

式(8)~(10)中: $x(t+1),y(t+1)$ 为 $t+1$ 时刻的坐标; $\theta(t+1)$ 为 $t+1$ 时刻的偏航角; Δt 为时间间隔。

2.2 速度采样

速度采样依赖于机器人当前所处的状态,并需要考虑机器人自身约束及环境约束等参数。机器人的速度限制 v_m 为

$$v_m=\{(v,\omega)|v\in[v_{\min},v_{\max}],\omega\in[\omega_{\min},\omega_{\max}]\}.$$

式(11)中: v 为机器人的线速度; ω 为机器人的角速度; v_{\min},v_{\max} 分别为机器人的最小线速度和最大线速度; $\omega_{\min},\omega_{\max}$ 分别为机器人的最小角速度和最大角速度。

机器人电机加减速限制 v_d 为

$$v_d=\{(v,\omega)|v\in[v_n-a_{\max}\Delta t,v_n+a_{\max}\Delta t],\omega\in[\omega_n-\alpha_{\max}\Delta t,\omega_n+\alpha_{\max}\Delta t]\}.$$

式(12)中: a_{\max},α_{\max} 分别为机器人的最大线加速度和最大角加速度; v_n,ω_n 分别为机器人当前的线速度和角速度。

机器人安全距离限制 v_s 为

$$v_s=\{(v,\omega)|v\in[v_{\min},\sqrt{2\text{dist}(v,\omega)a_{\max}}],\omega\in[\omega_{\min},\sqrt{2\text{dist}(v,\omega)\alpha_{\max}}]\}.$$

式(13)中: $\text{dist}(v,\omega)$ 为当前状态和障碍物之间的距离。

动态窗口速度限制 v_w 为

$$v_w=v_m\cap v_d\cap v_s.$$

2.3 评价函数

评价函数起到决策机器人运动路径的作用。改进后的评价函数 $G(v,\omega)$ 为

$$G(v,\omega)=\delta\text{head}(v,\omega)+\beta\text{dist}(v,\omega)+\gamma\text{vel}(v,\omega)+\eta\text{dist}_s(v,\omega).$$

式(15)中: $\text{head}(v,\omega)$ 为方位角的评价函数,是当前采样速度下产生的轨迹终点位置方向与目标点连线的夹角; $\text{dist}(v,\omega)$ 为障碍物距离的评价函数,是当前速度下对应轨迹与障碍物之间的距离; $\text{vel}(v,\omega)$ 为速度的评价函数,是当前轨迹的速度大小; $\text{dist}_s(v,\omega)$ 为当前轨迹与目标节点距离的评价函数; δ,β,γ,η 均为加权系数。

3 距离引导式A*算法与动态窗口法的融合

为了保证机器人路径规划的安全和效率,提出一种融合路径规划算法(文中融合算法),其流程图如图2所示。即将距离引导式A*算法与动态窗口法结合,使用距离引导式A*算法在已知的环境地图中规划出一条初始路径,在机器人沿着这条路径移动的过程中,动态窗口法会根据实时的传感器数据动态地调整机器人的速度和方向,以避免突然出现的未知障碍物。这种结合使机器人能够在保证路径规划效率的同时,具备一定的安全性,能够在复杂

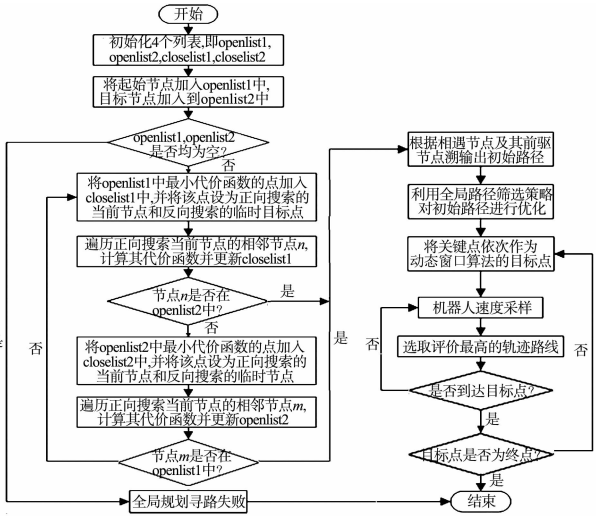


Fig. 2 Flowchart of proposed fusion algorithm

多变的环境中灵活调整路径,从而提高机器人的效率。

4 仿真结果与分析

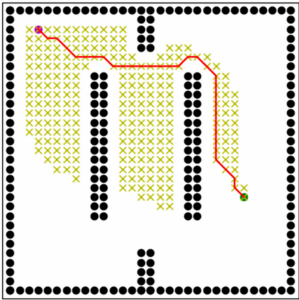
为了验证文中融合算法的有效性,使用 i7-13700H,16 GB 运行内存的笔记本电脑作为实验硬件设备,并采用 Python 3.10 作为实验的软件运行环境。A* 算法采用 8 邻域搜索方法,使用欧几里得距离作为启发函数来计算估计代价 $h(n)$ 。

为更好地体现距离引导式 A* 算法和动态窗口法的效果及地图适应性,分别设计 4 种实验的地图环境,并与 A* 算法和文献[12]A* 算法进行对比。融合算法评价函数的加权系数 $\delta=0.15,\beta=1.0,\gamma=1.0,\eta=2.0$ 。融合算法的仿真机器人运动学参数,如表 1 所示。表 1 中: d_v 为速度分辨率; d_ω 为转速分辨率。不同的地图环境下全局路径规划仿真结果,如图 3~6 所示。全局路径规划实验结果对比,如表 2 所示。表 2 中: t 为运行时间; l 为路径长度; N 为遍历节点数。

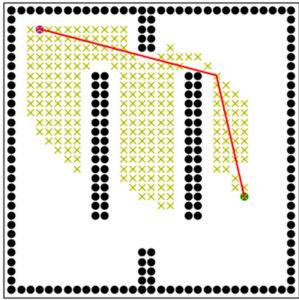
表 1 融合算法的仿真机器人运动学参数

Tab. 1 Simulation robots kinematic parameters of fusion algorithm

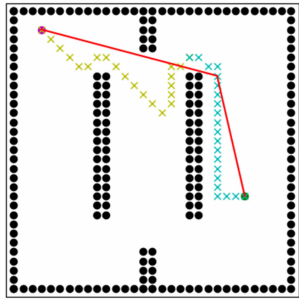
参数	取值	参数	取值	参数	取值
$v_{\max}/\text{m}\cdot\text{s}^{-1}$	5.00	$\alpha_{\max}/\text{rad}\cdot\text{s}^{-2}$	0.70	$d_v/\text{m}\cdot\text{s}^{-1}$	0.01
$\omega_{\max}/\text{rad}\cdot\text{s}^{-1}$	0.70	$a_{\max}/\text{m}\cdot\text{s}^{-2}$	0.20	$d_\omega/\text{rad}\cdot\text{s}^{-1}$	0.02



(a) A* 算法



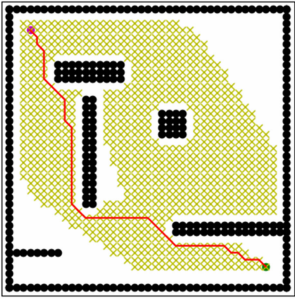
(b) 文献[12]A* 算法



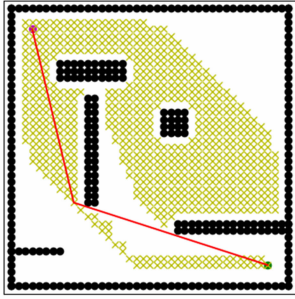
(c) 距离引导式 A* 算法

图 3 30×30 地图环境下全局路径规划仿真结果

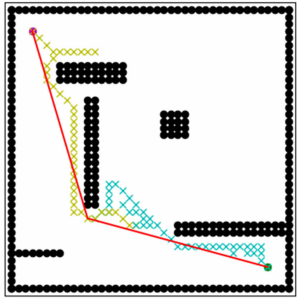
Fig. 3 Simulation results of global path planning in 30×30 map environment



(a) A* 算法



(b) 文献[12]A* 算法



(c) 距离引导式 A* 算法

图 4 40×40 地图环境下全局路径规划仿真结果

Fig. 4 Simulation results of global path planning in 40×40 map environment

表 2 全局路径规划实验结果对比

Tab. 2 Comparison of global path planning experimental results

地图大小	起点与终点坐标	A* 算法			文献[12]A* 算法			距离引导式 A* 算法		
		t/s	l/m	$N/\text{个}$	t/s	l/m	$N/\text{个}$	t/s	l/m	$N/\text{个}$
30×30	(3,28),(25,10)	1.21	36.14	248	0.97	32.99	239	0.21	32.99	40
40×40	(3,37),(37,3)	3.98	59.21	867	3.29	55.12	700	0.26	55.09	82
50×50	(3,47),(47,3)	2.43	66.33	475	1.33	63.08	257	0.27	63.08	82
60×60	(2,52),(57,4)	9.88	80.15	1 206	5.15	77.73	815	0.23	75.85	92

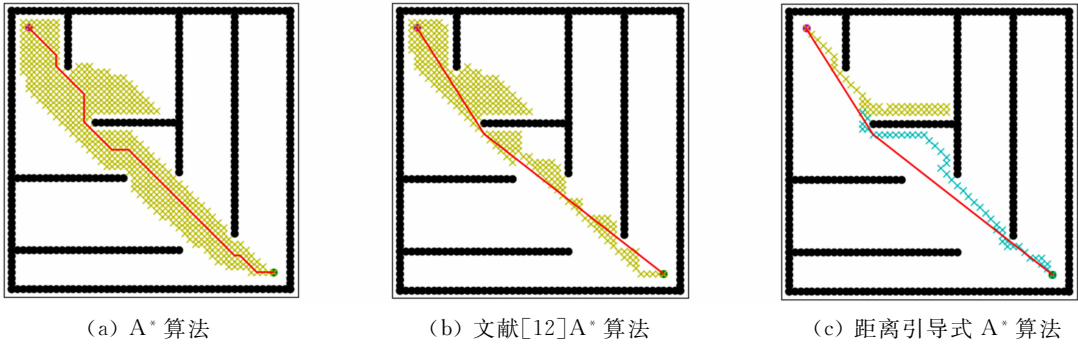


图 5 50×50 地图环境下全局路径规划仿真结果

Fig. 5 Simulation results of global path planning in 50×50 map environment

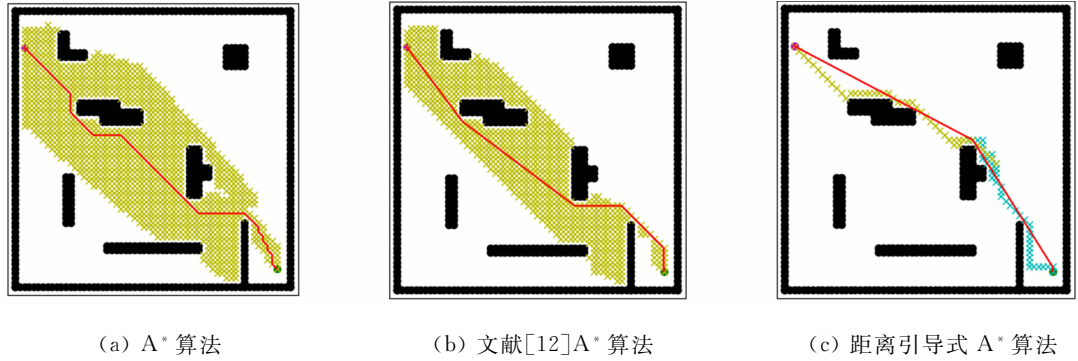


图 6 60×60 地图环境下全局路径规划仿真结果

Fig. 6 Simulation results of global path planning in 60×60 map environment

不同的地图环境下融合算法仿真结果,如图 7~10 所示。图 7~10 中:黄色区域表示正向遍历节点;青色区域表示反向遍历节点;红色线段表示规划的路径;黑色区域表示障碍物;紫色区域表示临时障碍物。

融合算法实验结果对比,如表 3 所示。由表 2,3 可知:距离引导式 A* 算法和文中融合算法在路径长度、遍历节点数目和运行时间方面均有提升,能更好地完成路径规划任务。

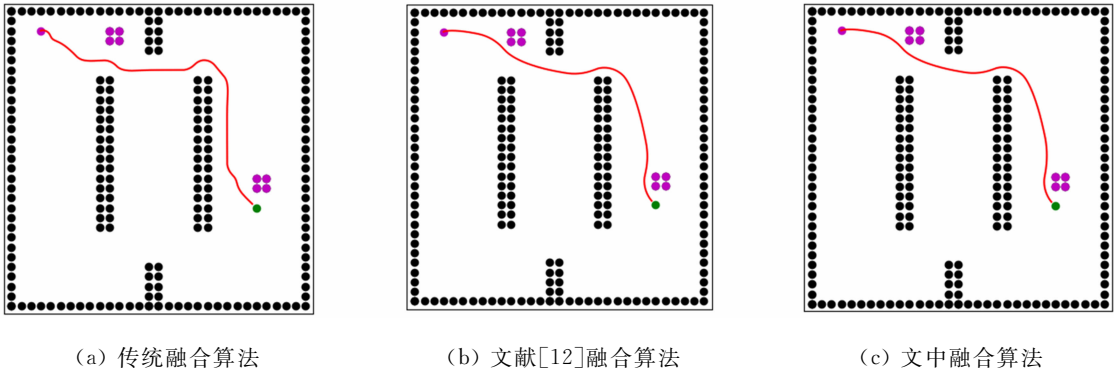


图 7 30×30 地图环境下融合算法仿真结果

Fig. 7 Simulation results of fusion algorithm in 30×30 map environment

表 3 融合算法实验结果对比

Tab. 3 Comparison of fusion algorithm experimental results

地图大小	起点与终点坐标	传统融合算法		文献[12]融合算法		文中融合算法	
		t/s	l/m	t/s	l/m	t/s	l/m
30×30	(3,28),(25,10)	130.73	34.95	40.37	33.74	34.70	33.68
40×40	(3,37),(37,3)	169.44	57.70	61.68	55.99	56.78	55.87
50×50	(3,47),(47,3)	190.65	65.16	75.23	64.25	65.45	64.22
60×60	(2,52),(57,4)	263.09	78.61	145.20	78.89	86.36	76.24

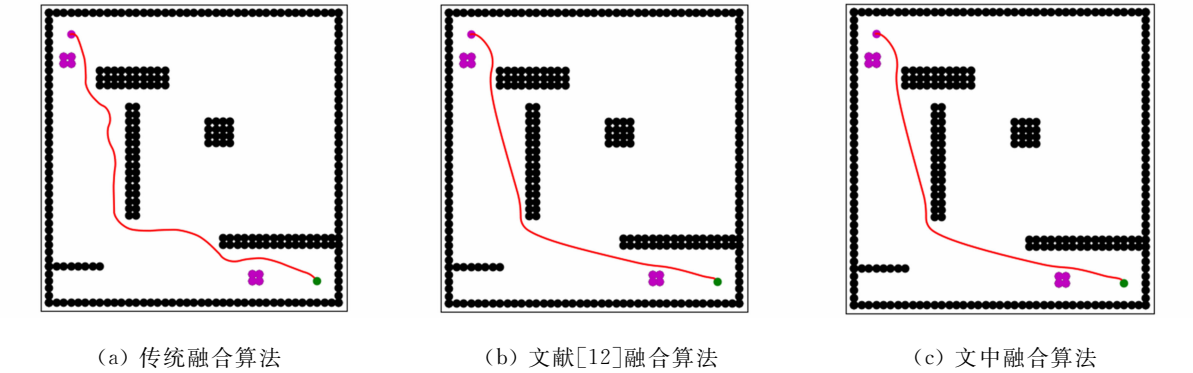


图 8 40×40 地图环境下融合算法仿真结果

Fig. 8 Simulation results of fusion algorithm in 40×40 map environment

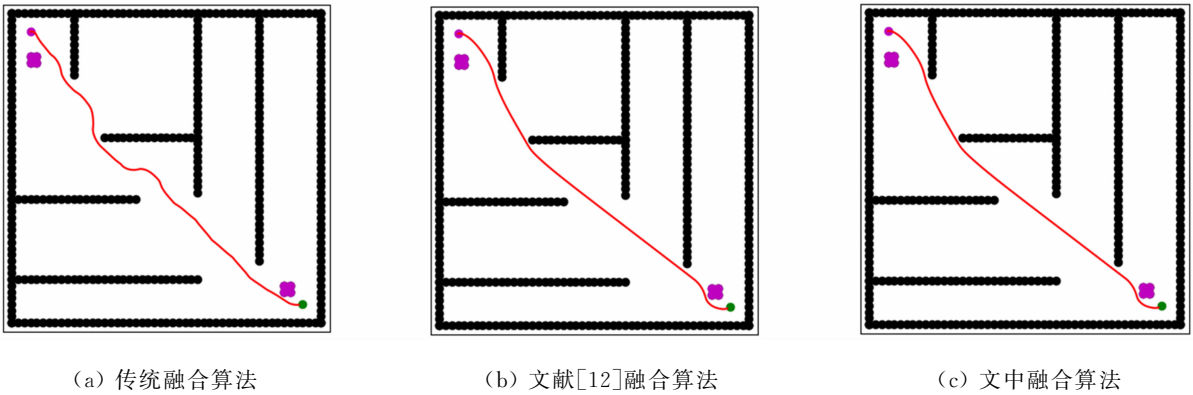


图 9 50×50 地图环境下融合算法仿真结果

Fig. 9 Simulation results of fusion algorithm in 50×50 map environment

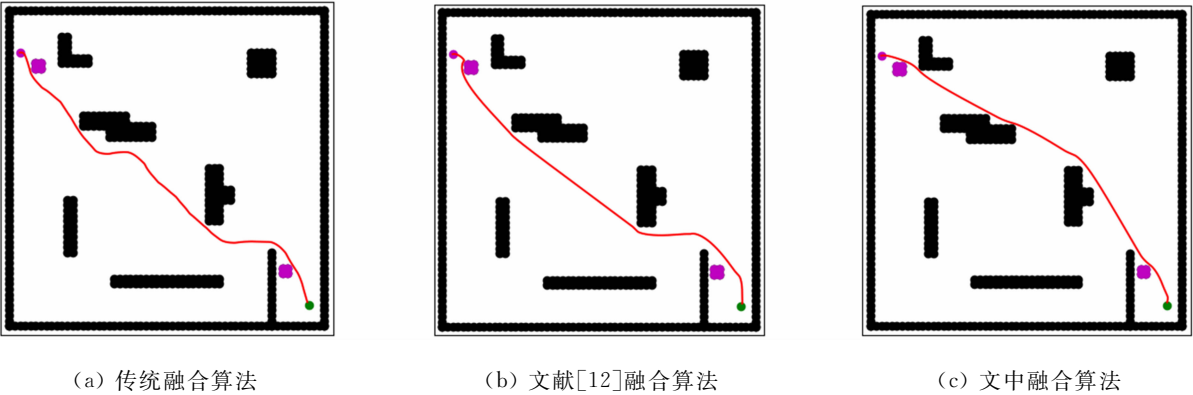


图 10 60×60 地图环境下融合算法仿真结果

Fig. 10 Simulation results of fusion algorithm in 60×60 map environment

5 结束语

路径规划技术在国家的无人系统领域具有重要应用价值,提升路径规划算法的性能对推动智能领域的发展具有关键意义。在全局路径规划中,采用双向搜索策略并提出一种综合距离启发函数,显著减少运行时间,利用全局路径筛选策略对初始路径进行优化,减少了拐点数和路径长度。在全局路径最优的基础上,采用动态窗口法进行局部路径规划,实现动态避障,并避免陷入局部最优解的情况。由实验结果可知,文中融合算法在路径长度、运行时间方面均有明显提升,能够更好地满足移动机器人对于路径规划的要求。

参考文献:

[1] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik,1959,1(1):269-

271, DOI:10. 1007/BF01386390.

[2] STENTZ A. Optimal and efficient path planning for partially known environments[C]//IEEE International Conference on Robotics and Automation. Piscataway:IEEE Press,2002:3310-3317. DOI:10. 1109/ROBOT. 1994. 351061.

[3] LAVALLE S M. Rapidly-exploring random trees: A new tool for path planning[J]. Research Report,1998(7): 3632-3648.

[4] HART P E,NILSSON N J,RAPHAEL B. A formal basis for the heuristic determination of minimum cost path[J]. IEEE Transactions on Systems Science and Cybernetics,1968,4(2):100-107. DOI:10. 1109/tssc. 1968. 300136.

[5] 赵晓,王铮,黄程侃,等. 基于改进 A* 算法的移动机器人路径规划[J]. 机器人,2018,40(6):903-910. DOI:10. 13973/j. cnki. robot. 170591.

[6] WANG Xingdong,ZHANG Haowei,LIU Shuo. *et al.* Path planning of scenic spots based on improved A* algorithm [J]. Scientific Reports,2022,12(1):1-7. DOI:10. 1038/s41598-022-05386-6.

[7] 田海波,李陆军,畅科剑,等. 用于无人车路径规划的改进 A* 算法[J]. 现代制造工程,2021(11):63-68,92. DOI:10. 16731/j. cnki. 1671-3133. 2021. 11. 009.

[8] 李炯逸,李强,张新闻,等. 移动机器人用改进的双向 A* 二次路径规划算法[J/OL]. 系统仿真学报:1-11(2023-12-19)[2024-03-15]. <https://www.cnki.com.cn/Article/CJFDTotat-XTFZ20231215007.htm>.

[9] 辛煜,梁华为,杜明博,等. 一种可搜索无限个邻域的改进 A* 算法[J]. 机器人,2014,36(5):627-633. DOI:10. 13973/j. cnki. robot. 2014. 0627.

[10] 王中玉,曾国辉,黄勃,等. 改进 A* 算法的机器人全局最优路径规划[J]. 计算机应用,2019,39(9):2517-2522. DOI:10. 11772/j. issn. 1001-9081. 2019020284.

[11] 姬鹏,张新元,高帅轩,等. 融合改进 A* 算法与动态窗口法的路径规划研究[J/OL]. 系统仿真学报:1-11(2023-07-09)[2024-03-15]. <https://doi.org/10.16182/j.issn1004731x.joss.23-0619>.

[12] 程传奇,郝向阳,李建胜,等. 融合改进 A* 算法和动态窗口法的全局动态路径规划[J]. 西安交通大学学报,2017, 51(11):137-143. DOI:10. 7652/xjtuxb201711019.

[13] 唐嘉宁,彭志祥,李孟霜,等. 基于改进 A* 算法的无人机路径规划研究[J]. 电子测量技术,2023,46(8):99-104. DOI: 10. 19651/j. cnki. emt. 2211107.

[14] 张旭,程传奇,郝向阳,等. 一种兼顾全局与局部特性的机器人动态路径规划算法[J]. 测绘科学技术学报,2018,35 (3):315-320. DOI: 10. 3969/j. issn. 1673-6338. 2018. 03. 018.

[15] 刘钰铭,黄海松,范青松,等. 基于改进 A* -DWA 算法的移动机器人路径规划[J]. 计算机集成制造系统,2024,30 (1):158-171. DOI: 10. 13196/j. cims. 2022. 0634.

[16] 邹文,韩丙辰,李鹏飞,等. 融合改进 A* 算法和优化动态窗口法的路径规划[J]. 计算机集成制造系统,2024,30 (1):184-195. DOI: 10. 13196/j. cims. 2021. 0498.

(责任编辑: 钱筠 英文审校: 陈婧)