

DOI:10.19651/j.cnki.emt.2314983

# 融合 A\* 和 DWA 算法的移动机器人路径规划方法

袁新亚<sup>1,2,3</sup> 戴娟<sup>1,2,3</sup> 孙胜强<sup>1,2,3</sup> 刘经纬<sup>1,2,3</sup>

(1. 北京信息科技大学高动态导航技术北京市重点实验室 北京 100192; 2. 现代测控技术教育部重点实验室 北京 100192; 3. 北京信息科技大学自动化学院 北京 100192)

**摘要:** 本研究旨在解决自主移动机器人在点到点路径规划中面临的搜索效率低下、易陷入局部最优解以及对未知动态障碍物处理不够实时的问题。为此,将改进 A\* 算法与改进 DWA 进行了有效融合。在改进的 A\* 算法中,我们引入了基于障碍率的权重因子和双向优化策略,以提升搜索效率并生成更加平滑的路径。同时,改进的 DWA 算法融入了两种新的障碍物评价函数,并通过调整权重系数有效地避免了局部最优解问题。通过将改进的 DWA 算法与改进的 A\* 算法结合,实现了对未知动态障碍物的高效实时避障。仿真实验结果显示,提出的改进 A 算法与传统 A 算法以及文献[23]的改进算法相比,在四种环境下的表现表明:路径转弯次数分别平均减少了 30.14% 和 18.16%,搜索空间分别减少了 35.09% 和 15.21%,规划时间分别降低了 82.36% 和 38.26%。进一步地,结合改进的 DWA 算法后,路径规划时间、路径长度和平均运动速度相比融合传统 DWA 算法和文献[23]的融合算法分别平均减少了 37.46% 和 9.82%,减少了 4.59% 和 3.63%,提高了 53.49% 和 7.09%。

**关键词:** 自主移动机器人;改进 A\* 算法;改进 DWA 算法;点到点路径规划

**中图分类号:** TP242 **文献标识码:** A **国家标准学科分类代码:** 510.8050

## Integration of A\* and DWA algorithms for mobile robot path planning

Yuan Xinya<sup>1,2,3</sup> Dai Juan<sup>1,2,3</sup> Sun Shengqiang<sup>1,2,3</sup> Liu Jingwei<sup>1,2,3</sup>

(1. University of Beijing Information Science & Technology Beijing Key Laboratory of High Dynamic Navigation Technology, Beijing 100192, China; 2. Key Laboratory of Modern Measurement & Control Technology, Ministry of Education, Beijing 100192, China; 3. School of Automation, Beijing Information Science & Technology University, Beijing 100192, China)

**Abstract:** This study aims to tackle the challenges encountered by autonomous mobile robots in point-to-point path planning, encompassing issues such as low search efficiency, susceptibility to local optima, and inadequate real-time handling of unknown dynamic and static obstacles. To this end, we have carried out an effective integration of the enhanced A\* algorithm with the improved DWA. Within the enhanced A\* algorithm, we have introduced obstacle-rate-based weighting factors and a bidirectional optimization strategy, aiming to bolster search efficiency and facilitate the generation of smoother paths. Furthermore, the refined DWA algorithm integrates two novel obstacle evaluation functions and adeptly addresses the local optima issue through the adjustment of weight coefficients. By unifying the enhanced DWA algorithm with the improved A\* algorithm, we have enabled proficient real-time obstacle avoidance for unknown dynamic and static obstacles. Simulation results indicate that the improved A\* algorithm proposed in this paper, compared with the traditional A\* algorithm and the enhanced algorithm from reference [23], demonstrates significant performance improvements in four different environments. Specifically, the number of path turns decreased by an average of 30.14% and 18.16%, the search space was reduced by 35.09% and 15.21%, and the planning time was shortened by 82.36% and 38.26%, respectively. Furthermore, when integrated with the improved DWA algorithm, the time required for path planning, the length of the planned path, and the average motion speed were optimized compared to combining with the traditional DWA algorithm and the fusion algorithm from reference [23], showing an average reduction of 37.46% and 9.82% in planning time, a decrease of 4.59% and 3.63% in path length, and an increase of 53.49% and 7.09% in average motion speed.

**Keywords:** autonomous mobile robots; improved A\* algorithm; improved DWA algorithm; point-to-point path planning

## 0 引言

近年来,新冠疫情的爆发和人口老龄化的加剧对传统

服务业带来了巨大挑战,推动各行业积极追求智能化和自动化解决方案<sup>[1]</sup>。自主移动机器人作为服务业升级和创新发展的主要产品之一,显示出巨大潜力和优势。其中,路径

规划<sup>[2]</sup>在自主移动机器人应用中至关重要。无论是在新能源汽车领域、智能家居领域还是商场和餐饮业,通过路径规划,移动机器人能够高效、安全地完成各种任务。因此,深入研究和探索自主移动机器人的路径规划问题对未来服务业和机器人技术的应用具有重要意义和深远影响<sup>[3]</sup>。

路径规划可分为全局路径规划和局部路径规划这两个关键的方面。全局路径规划是在已知的环境地图中规划出一条无碰撞的路径,全局路径规划算法包括 Dijkstra<sup>[4]</sup>算法、A\* 算法<sup>[5]</sup>、D\* 算法<sup>[6]</sup>和蚁群算法<sup>[7]</sup>等;局部路径规划方法包括人工势场法<sup>[8]</sup>、快速扩展随机树算法<sup>[9]</sup>(rapidly-exploring random trees, RRT)和 DWA 算法(dynamic windows approach, DWA)<sup>[10]</sup>等。目前,许多学者在自主移动机器人的路径规划领域开展了大量研究工作,并对传统路径规划算法进行了改进和融合,以提升算法性能。例如,迟旭等<sup>[11]</sup>提出一种结合 A\* 算法和动态窗口算法,通过当前点到目标点的距离与起点到终点的距离的比值作为估计值的权重,优化 A\* 算法的启发函数,以提高运算效率。然而,这种方法会导致路径具有较多的转弯次数,可能会陷入局部最优解,且极端时导致无法搜索到可行路径。赵晓等<sup>[12]</sup>针对 A 算法的优化,通过跳点搜索策略显著提升了算法在大场景中的搜索效率,减少了内存占用和计算时间。Wang 等<sup>[13]</sup>探讨了 DWA 算法的改进,通过参数自适应机制,根据机器人与障碍物的距离和障碍物密度来调整权重因子,提高了穿越稠密障碍物时的路径规划效率。袁千贺等<sup>[14]</sup>引入了障碍率自适应调节 A\* 算法的评价函数,以提高全局搜索效率,并结合关键点信息的 DWA 算法评价函数,实现了全局路径优化和未知障碍物的躲避。封硕等<sup>[15]</sup>着眼于动态障碍物的路径规划问题,提出了一种融合动态障碍物运动信息的算法,实现了在动态环境中的路径规划。然而,该算法仍存在当面对动态障碍物时易陷入局部最优的问题。

鉴于前述研究存在的问题,例如低搜索效率、转弯次数多、对未知动态障碍物处理不及时以及容易陷入局部最优解等,本文提出了一种融合改进 A\* 算法和改进 DWA 算法的路径规划方法。对于 A\* 算法,在估计代价函数中引入了基于障碍率的权重因子和双向优化方式提高了搜索效率并且使路径更平滑。对于 DWA 算法,我们引入了两种不同类别的障碍物评价函数,并通过调整系数的方式,成功解决了陷入局部最优的问题。最后,将改进后的 DWA 算法与改进后的 A\* 算法融合,以解决动态障碍物处理实时性差的问题。并通过仿真实验充分验证了其效果和效率的提升。

## 1 A\* 算法的改进

A\* 算法是将地图简化成一个一个栅格,计算路径时是将每个栅格的中心点作为单位进行计算<sup>[16]</sup>。它是从起点(start)开始,根据估价函数搜索周围节点,得到当前代价最

小的节点作为子节点继续搜索,直至搜索到目标点。估价函数  $f(n)$  的公式为:

$$f(n) = g(n) + h(n) \quad (1)$$

其中,  $g(n)$  表示当前节点  $n$  与起点的实际代价,并且常用的代价子函数为欧几里得距离,  $h(n)$  表示当前节点  $n$  与终点的估计代价。

### 1.1 量化地图信息

栅格地图法<sup>[17]</sup>是一种将实际地图大小表示为栅格的方法,其中每个栅格单元代表地图上的二维信息。在栅格地图中,障碍物的格子数量越多,可选的路径就越多,而最短的路径长度也会超过起点到终点的直线距离。为了估计最小路径的距离,引入了障碍率  $N$ ,这个概念抽象地描述了地图的复杂性。障碍率定义为在起点和终点构成的矩形区域内,障碍物栅格数与总栅格数的比值。假设在局部区域内的障碍物栅格数为  $O$ 。起始点的坐标是  $(x_s, y_s)$ ,而终点的坐标是  $(x_g, y_g)$ ,该表达式如下:

$$N = \frac{O}{(|x_s - x_g| + 1)(|y_s - y_g| + 1)}, N \in (0, 1) \quad (2)$$

### 1.2 评价函数的改进

在 A\* 算法中,式(1)表明估计代价函数  $h(n)$  对搜索性能起主导作用。针对不同地图环境,估计代价函数的取值情况如下:

1)当障碍物较少时,障碍率较低,我们希望估计代价函数值以增加终点方向性。

2)当障碍物增多时,简单地增大估计代价函数值和仅仅朝终点方向搜索可能导致局部最优和多次转弯的路径,使得最优路径难以搜索。因此,随着障碍率增大,我们减小估计代价函数值,扩大搜索空间,提高搜索精度以获得全局最优路径。

根据式(2),障碍率  $N$  由障碍栅格个数  $O$  以及起点和终点的位置共同确定, A\* 算法的搜索空间随三者情况的改变而变化。为增加 A\* 算法的灵活性,本文将障碍率纳入评价函数中式(1),通过调整实际代价  $g(n)$  和估计代价  $h(n)$  的权重,实现评价函数  $f(n)$  的自适应调整。这样可以产生不同的有效搜索空间,同时保证路径搜索效率。

综上所述,优化后总评价函数可以表示为:

$$f(n) = g(n) + e^{-(N-1)} \times h(n) \quad (3)$$

### 1.3 路径平滑度优化

在栅格地图中,路径是由栅格的中心点形成的,这导致路径因多次转折而不平滑。为解决此问题,本文提出了双向优化策略,并为节点设置了安全距离防止碰撞。如图 1 展示的安全距离示意图。给定路径节点 A 和 B 的坐标  $(x_a, y_a)$  和  $(x_b, y_b)$ ,以及障碍物 C 的坐标  $(x_c, y_c)$ ,我们可以计算出 C 到 AB 线段的垂直距离  $d$  和竖直距离  $l$ ,以及 AB 线段与横轴的夹角  $\alpha$ 。同时,我们还定义了单元障碍栅格的边长  $d_{cell}$  和外接圆的半径  $r$ 。通过 A、B、C 的

坐标,可以得到相关参数。

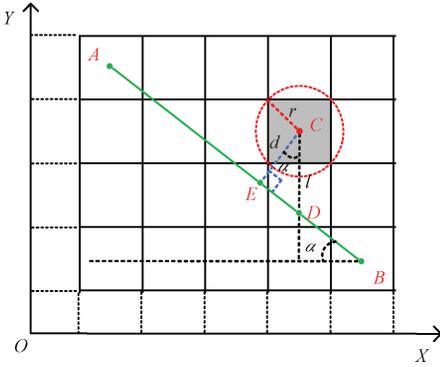


图 1 安全距离示意图

$$l = \left| y_c - \left( \frac{y_a - y_b}{x_a - x_b} (x_c - x_a) + y_b \right) \right| \quad (4)$$

$$d = l \cos \alpha \quad (5)$$

$$r = d_{cell} / \sqrt{2} \quad (6)$$

$$\alpha = \left| \arctan \frac{y_a - y_b}{x_a - x_b} \right| \quad (7)$$

自主移动机器人在转弯时需要加减速,因此转弯比直线行驶耗时更长。大角度转弯还会带来各种损耗,如物理磨损、电池消耗和时间成本,使路径不平滑且增加转弯代价。为解决这一问题,本文采用了双向优化方法来提高路径的平滑度。具体步骤如下:

- 1) 删除路径上同一直线上的中间点,只保留起点、拐点和终点,则新路径为  $S \rightarrow n_7 \rightarrow n_8 \rightarrow n_9 \rightarrow G$ 。
- 2) 从起点 S 开始,固定步长选择节点,并与前一个路径节点进行比较。判断两个节点之间是否有障碍物,并计算障碍物与路径的距离。如果判断距离  $d$  超过了安全距离  $D$ ,将当前节点选为路径节点,则新路径为  $S \rightarrow n_{11} \rightarrow G$ 。
- 3) 从终点 G 开始,反向进行节点判断,使用步骤 2 的方法选择路径节点,则新路径为  $S \rightarrow n_{12} \rightarrow G$ 。
- 4) 输出优化后的路径,并结束算法。具体过程如图 2 所示。

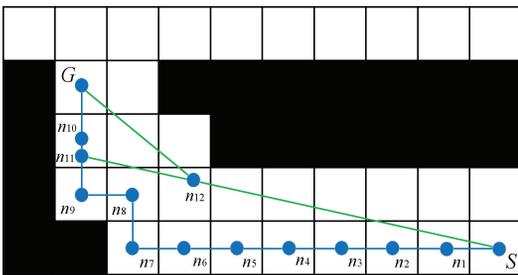


图 2 双向平滑优化过程

## 2 DWA 算法的改进

传统 DWA 算法是在速度空间中采样多组速度,模拟机器人在这些速度下一段时间的轨迹,并评价这些轨迹,最

后以选取最优评价速度来驱动机器人运动的指令<sup>[18]</sup>。因此,在模拟机器人运动前需要对其建立运动模型<sup>[19]</sup>。

### 2.1 自主移动机器人运动模型

在动态窗口技术中,模拟机器人的行动路径是必要的,因此必须了解机器人的运动模型。机器人的移动可以通过线速度和角速度的变动来展现,一组  $(v, \omega)$  可以描述一段轨迹。如果机器人在时间间隔  $\Delta t$  进行匀速直线运动,其运动模型可以表示为:

$$\begin{cases} x(t+1) = x(t) + v\Delta t \cos(\theta_t) - v\Delta t \sin(\theta_t) \\ y(t+1) = y(t) + v\Delta t \sin(\theta_t) + v\Delta t \cos(\theta_t) \\ \theta_{t+1} = \theta_t + \omega\Delta t \end{cases} \quad (8)$$

### 2.2 自主移动机器人的速度采样

速度空间需要满足的 3 个约束条件:首先是自主移动机器人的速度上下限制;其次是自主移动机器人的动态性能约束;最后是自主移动机器人的安全性约束。

自主移动机器人的最大和最小速度约束为:

$$v_m = \{(v, \omega) \mid v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (9)$$

自主移动机器人动态性能约束:机器人在模拟前进时间间隔内,存在由于加速度带来的最大、最小速度:

$$\begin{aligned} v_d = \{(v, \omega) \mid v \in [v_c - \dot{v}\Delta t, v_c + \dot{v}\Delta t], \\ \omega \in [\omega_c - \dot{\omega}\Delta t, \omega_c + \dot{\omega}\Delta t]\} \end{aligned} \quad (10)$$

其中,  $(v_c, \omega_c)$  为当前线速度和角速度,  $\dot{v}$  为最大的线加速度;  $\dot{\omega}$  为最大角加速度。

自主移动机器人的安全约束:保证机器人在最大减速条件下,当前速度能在撞击障碍物之前减速为 0,则:

$$\begin{aligned} v_a = \{(v, \omega) \mid v \leq \sqrt{2 \text{dist}(v, \omega) \dot{v}}, \\ \omega \leq \sqrt{2 \text{dist}(v, \omega) \dot{\omega}}\} \end{aligned} \quad (11)$$

其中,  $\text{dist}(v, \omega)$  为当前轨迹距最近障碍物的距离。

自主移动机器人的活动范围受到其最大和最小速度的约束、电机的加减速约束以及安全约束的共同影响。结合式(9)~(11),设自主移动机器人的速度集合为  $v_r$ , 机器人的运动速度范围可以表示为:

$$v_r = v_m \cap v_d \cap v_a \quad (12)$$

### 2.3 评价函数

自主移动机器人在运行过程中不断进行速度采样,可以确定速度范围。根据自主移动机器人的运动模型和运动速度范围,可以模拟并推演出多条轨迹,这些轨迹的优劣可以通过评价函数进行评价。评价函数的定义如下:

$$G(v, \omega) = \alpha \text{heading}(v, \omega) + \beta \text{dist}(v, \omega) + \gamma \text{vel}(v, \omega) \quad (13)$$

其中,  $\text{heading}(v, \omega)$  是方位角评价函数,如图 3 所示,它评估轨迹末端与目标点的角度差,确保自主移动机器人不偏离终点<sup>[20]</sup>;  $\text{dist}(v, \omega)$  表示机器人与最近障碍物的距离;自主移动机器人当前轨迹上离最近障碍物的距离;  $\text{vel}(v, \omega)$  为当前速度大小的评价函数;  $\alpha, \beta, \gamma$  为三项的加权系数。

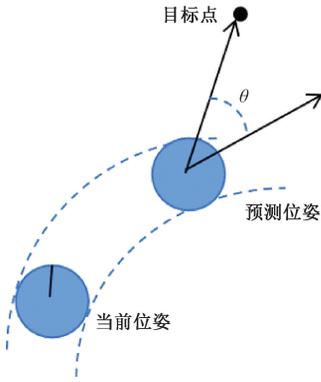


图 3 自主移动机器人方位角

2.4 改进 DWA 算法

评价函数是选择路径的重要指标,其权重影响路径选择<sup>[21]</sup>。传统 DWA 算法只考虑终点方向,缺少全局引导,主要依赖实时避障规划。 $dist(v, \omega)$  的权重过大导致机器人避开障碍,可能产生局部最优路径并增加总移动距离。减少其权重可能减弱避障能力,增加与障碍物的碰撞风险。为此,在原函数中加入全局路径方位角评价函数  $Heading(v, \omega)$ , 并设计了两类障碍物评价函数  $dist\_sta(v, \omega)$  和  $dist\_dyn(v, \omega)$ , 则改进的评价函数为:

$$G(v, \omega) = \alpha Heading(v, \omega) + \beta vel(v, \omega) + \gamma dist\_sta(v, \omega) + \lambda dist\_dyn(v, \omega) \quad (14)$$

其中,  $\alpha, \beta, \gamma, \lambda$  为 4 项评价函数的加权系数。 $Heading(v, \omega)$  是评价子目标点的方位偏角评价函数,表示预测轨迹方向与全局路径节点的角度偏差  $\Delta\varphi$ 。 $dist\_sta(v, \omega)$  用于评估全局已知静态障碍物,有助于接近全局最优路径并减少绕过障碍物的距离。 $dist\_dyn(v, \omega)$  评估局部未知动态障碍物,增强了对未知障碍物的避障能力。

2.5 算法融合

通过将改进的 A\* 算法与改进的 DWA 算法结合,自主移动机器人能够实现全局最优路径规划并对动态障碍物进行避障<sup>[22]</sup>。该融合算法首先使用改进的 A\* 算法进行全局路径规划,生成全局最优节点序列,并在自主移动机器人工作过程中使用改进的 DWA 算法在相邻节点之间进行局部路径规划。路径规划流程如图 4 所示。

3 仿真分析

为验证改进和融合算法的有效性,我们在室内地图场景中二维栅格地图进行了一系列仿真实验,运行环境为 Windows 11 和 MATLAB R2019a。

3.1 改进 A\* 算法在静态环境下的仿真分析

为了比较路径规划性能,在 4 种  $30 \times 30$  的栅格环境中进行了仿真实验,栅格边长为 1 m,对传统 A\* 算法、本文改进 A\* 算法和文献[23]的改进 A\* 算法进行了比较,以验证其性能的优越性。起点和终点分别位于 (3, 27) 和 (27, 4),

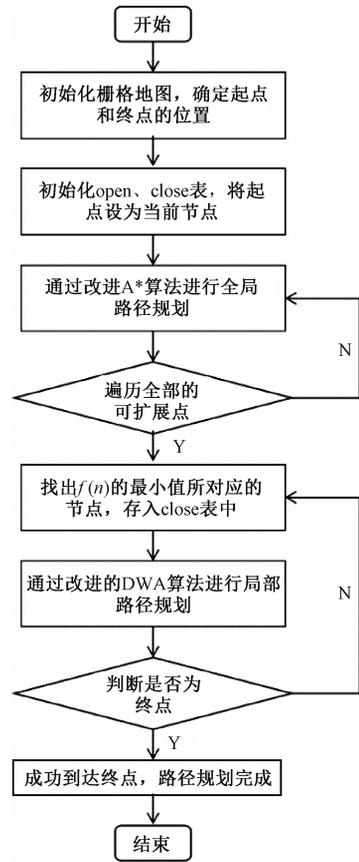


图 4 改进 A\* 融合改进 DWA 路径规划流程图

用三角形和圆形来标记。图 5 展示了 3 种算法在不同环境下的规划路径,其中用不同类型线段分别表示传统 A\* 算法、本文改进 A\* 算法和文献[23]改进算法的路径规划曲线,表 1 和 2 列出了 3 种算法对应的性能数据。

由图 5 可知,通过改进 A\* 算法并融入地图信息,本文改进算法选择了障碍较少的区域进行规划,降低了路径转折次数,并保持了与障碍物的安全距离,增强了安全性。由表 1 可知,本文改进 A\* 算法比传统 A\* 算法在四种环境下的路径转弯次数平均下降了 30.14%,与文献[23]的改进 A\* 算法相比,本文改进 A\* 算法的路径转弯次数平均减少了 18.16%。

根据图 6,传统 A\* 算法冗余节点多,搜索效率低。本文改进后的算法提高了效率,路径更远离障碍,更安全且规划的路径长度更优。从表 1 的数据可知,相比传统 A\* 算法,本文改进 A\* 算法搜索空间减少了 35.09%,规划时间降低了 82.36%;相比文献[23]的改进 A\* 算法,本文改进 A\* 算法搜索空间减少了 15.21%,规划时间降低了 38.26%。

由表 2 数据可知本文改进的 A\* 算法对应的规划路径长度是这 3 种算法中最短的,与传统 A\* 算法相比,本文改进 A\* 算法规划的路径长度平均减少 2.52%;与文献[23]的改进 A\* 算法,本文改进 A\* 算法本文改进 A\* 算法规划的路径长度平均减少 2.80%。这证明了本文的改进算法

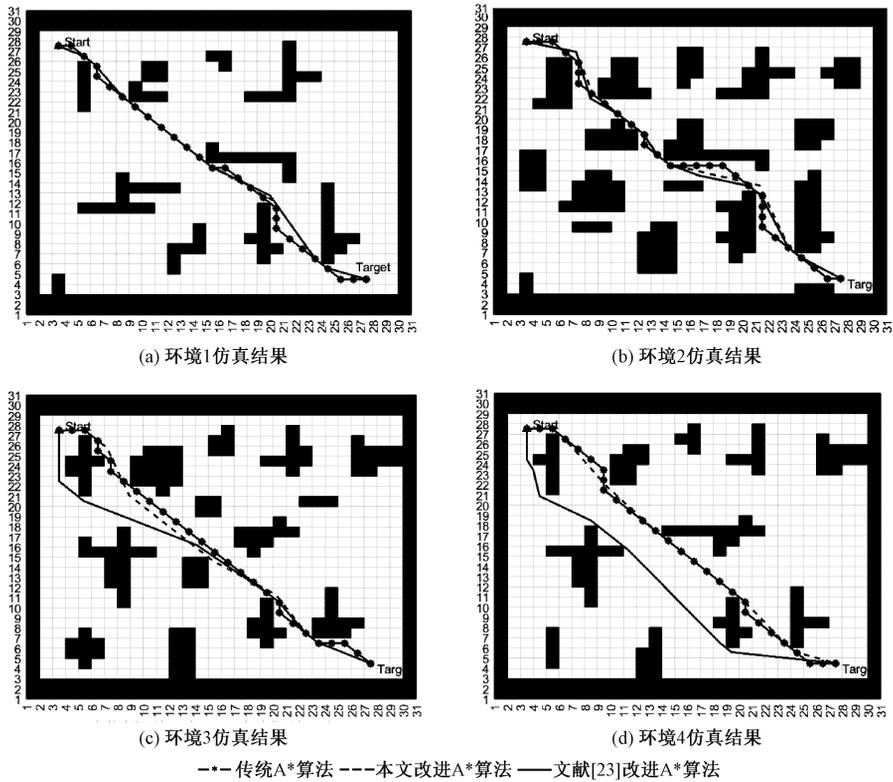


图 5 传统 A\* 算法和改进 A\* 算的实验结果

表 1 算法性能对比

算法	环境 1			环境 2			环境 3			环境 4		
	转弯次数	搜索空间/个节点	时间/s									
传统 A* 算法	8	274	0.131 7	10	272	0.130 9	9	187	0.116 0	6	247	0.101 9
本文改进 A* 算法	6	173	0.013 3	6	148	0.022 1	6	140	0.023 6	4	166	0.023 7
文献[23]改进算法	6	200	0.016 1	7	186	0.038 5	8	166	0.080 0	6	187	0.030 6

有效地提高了搜索效率和安全性,减少了规划时间和路径长度。

### 3.2 融合算法在动态环境下的仿真分析

为验证融合算法的可行性和有效性,在两种环境地图中,将改进 A\* 算法融合传统 DWA 算法、本文融合算法、文献[23]的融合算法进行仿真实验。选择  $20 \times 20$  的栅格化环境地图,环境 1 的自主移动机器人起点坐标为(18, 2),终点坐标为(2, 19),环境 2 的自主移动机器人起点坐标为(2, 19),终点坐标为(19, 2)。在静态路径规划基础上,添加未知动、静态障碍物,以评估本文融合算法避开障碍物的能力。设置速度分辨率 0.01 m/s,角速度分辨率 1 rad/s,我们设置了相应的参数  $\alpha = 0.2, \beta = 0.4, \gamma = 0.2, \lambda = 0.4$ 。

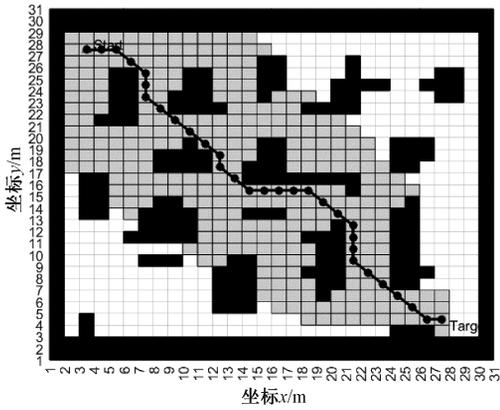
仿真实验结果如图 7~12 所示。地图大小为  $20 \times 20$ ,每个小栅格边长为 1 m。灰色栅格表示未知静态障碍物。

表 2 算法规划路径长度对比

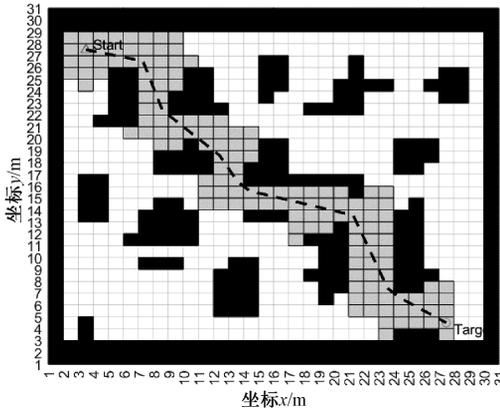
算法	环境 1	环境 2	环境 3	环境 4
	路径长度/m	路径长度/m	路径长度/m	路径长度/m
传统 A* 算法	36.284 3	36.041 6	35.284 3	35.284 3
本文改进 A* 算法	35.017 5	34.881 9	35.040 5	34.343 9
文献[23]改进算法	35.301 2	35.399 2	35.823 7	36.829 1

三角形为起点,圆形为终点。实线是动态规划的路径,虚线是改进 A\* 算法静态规划的路径,图中的 Obstacle<sub>b</sub> 表示是动态障碍物。

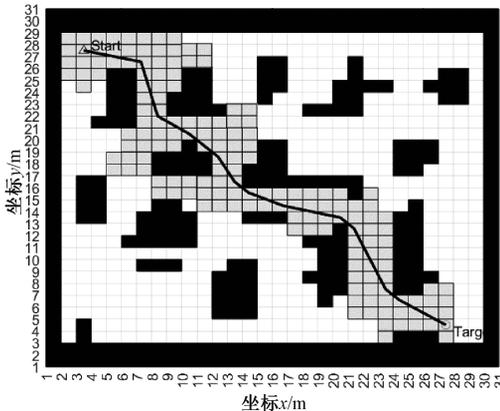
从图 7 (a)和图 10 (a)可以观察到,在改进 A\* 算法融合传统 DWA 算法的动态路径规划中,机器人在路径规划开始时由于方向不匹配需要进行圆弧转弯,增加了移动路径。在图 7 (b)和图 10 (b)中,当遇到动态障碍物时,融合的



(a) A\*算法的搜索空间



(b) 本文改进A\*算法的搜索空间

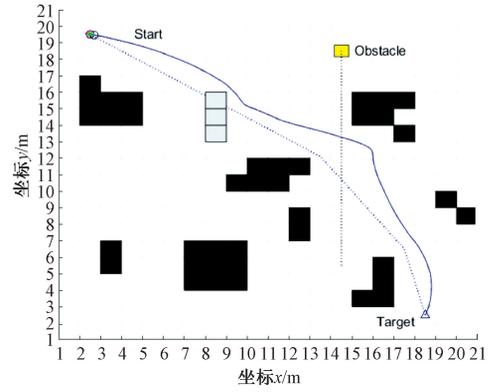


(c) 文献[23]改进算法的搜索空间

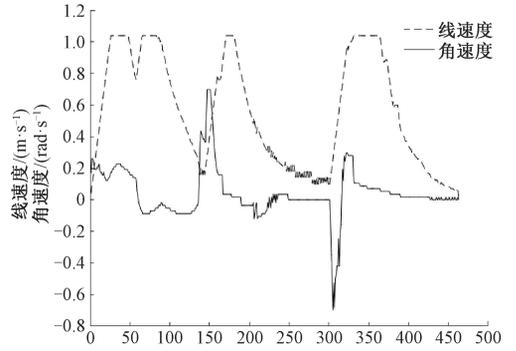
图 6 环境 2 下的两种算法搜索空间对比

表 3 仿真参数表

参数	数值
最大线速度	2 m/s
最小线速度	0 m/s
最大角速度	1 rad/s
最小角速度	-1 rad/s
最大线加速度	0.4 m/s <sup>2</sup>
最小线加速度	0 m/s <sup>2</sup>
最大角加速度	1.7 rad/s <sup>2</sup>
最小角加速度	-1.7 rad/s <sup>2</sup>
预测时间	3 s

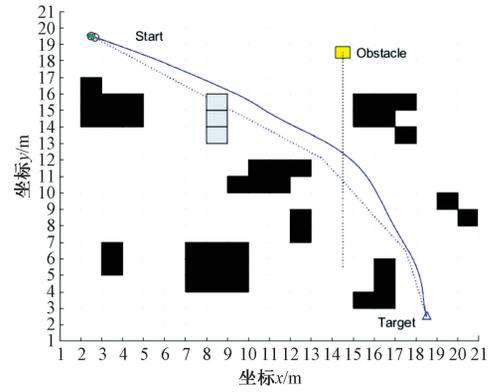


(a) 规划过程

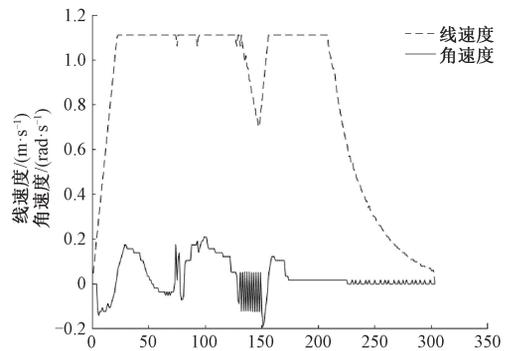


(b) 速度曲线

图 7 环境 1 融合传统 DWA 算法动态规划



(a) 规划过程



(b) 速度曲线

图 8 本文融合算法在环境 1 中的动态规划

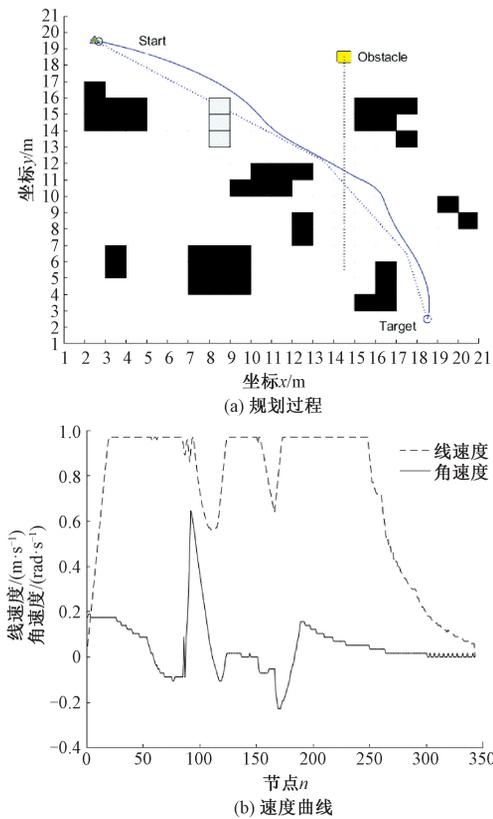


图 9 文献[23]融合算法在环境 1 中的动态规划

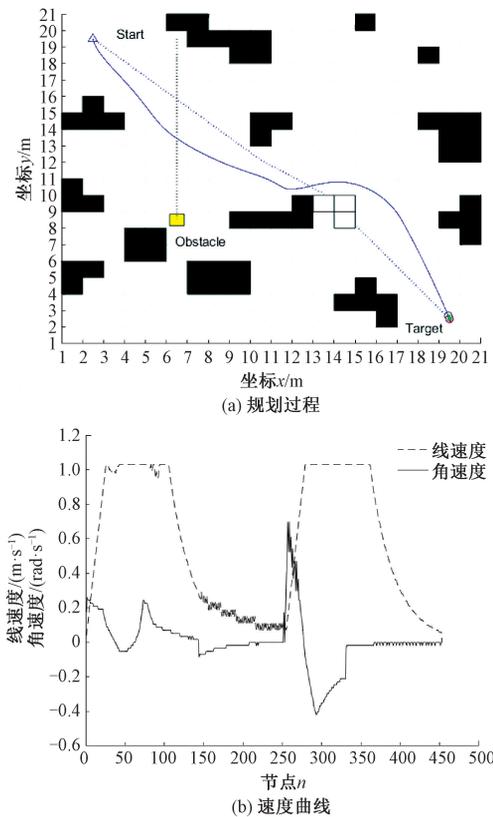


图 10 环境 2 融合传统 DWA 算法动态规划

传统 DWA 算法会迅速减少速度并进行刹车处理,然后重新规划下一步的移动方向,再逐渐提速。随后遇到未知静态障碍物,同样采取迅速减速的操作。从图 8(a)和图 11(a)可以看出,由本文融合算法结合了全局规划路径,并且对障碍物进行了区分,减少了已知静态障碍物对评价函数影响,所得规划路径转弯幅度小且更靠近全局规划的路径。从图 8(b)和图 11(b)可以观察到,在遇到动态障碍物时,本文融合算法的处理方式为轻微降低线速度,并增加给定角速度的值,进行转弯处理。尤其从图 11(b)可以看出,在路过静态未知障碍物时,线速度几乎没有波动,而是直接进行转弯处理,进而躲避掉障碍物。

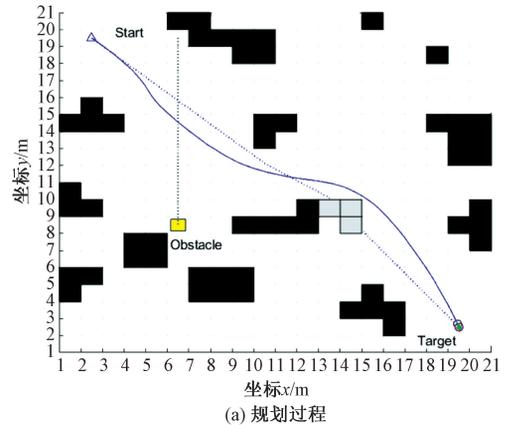


图 11 本文融合算法在环境 2 中的动态规划

由图 9(a)和图 12(a)可以看出,文献[23]对应的融合算法在处理动静障碍物时的总体效果比改进 A\* 融合传统传统 DWA 算法好,但是稍微比本文的融合算法差一些;由图 9(b)可得,在遇到动态障碍物时,该融合算法也会迅速减少线速度,再重新规划移动方向,之后迅速提高速度;尤其从图 12(b)中可以看到,当遇到动态障碍物时,线速度几乎没有变化,而是直接进行转弯处理,进而躲避掉障碍物;当遇到静态障碍物时,该融合算法的处理方式为略微降低线速度,并增加给定角速度,进行转弯处理。由表 4 可得,本文融合算法相比融合传统 DWA 算法,路径规划时间平均减少了 37.46%,规划路径长度平均减少了 4.59%,平均运动速度提高了 53.49%;与文献[23]的融合

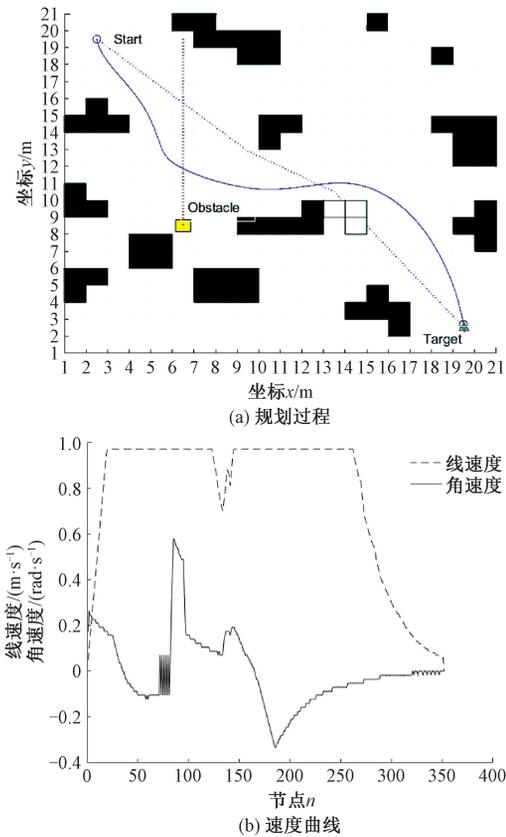


图 12 文献[23]融合算法在环境 2 中的动态规划

算法相比,本文融合算法的路径规划时间平均减少了 9.82%,规划路径长度平均减少了 3.63%,平均运动速度提高了 7.09%。

表 4 仿真实验数据对比

环境 地图	算法	规划 路径/m	规划 时间/s	平均速度/ m/s
环境 1	改进 A* 融合传统 DWA 算法	26.055	134.021	0.194
	本文融合算法	24.840	78.338	0.317
	文献[23]融合算法	25.674	91.565	0.280
	改进 A* 融合传统 DWA 算法	26.375	120.788	0.218
环境 2	本文融合算法	25.182	80.488	0.313
	文献[23]融合算法	26.235	84.895	0.310

实验结果表明,自主移动机器在使用本文提出的改进 A\* 算法融合改进 DWA 算法时,可以有效地避开了未知静态障碍物。

## 4 结 论

本研究旨在解决自主移动机器人点到点路径规划中存在的搜索效率低、路径不平滑、处理障碍物的实时性差

和容易陷入局部最优解等问题。为此,本文提出了一种融合改进 A 算法和改进 DWA 算法的解决方案。对于 A\* 算法的改进,通过量化环境信息和改进评价函数来生成有效搜索空间;采用双向优化策略对路径节点进行优化,提高路径的平滑度并减少冗余节点。此外,为了解决自主移动机器人对未知静态障碍物处理实时性差和容易陷入局部最优的问题,改进的 DWA 算法通过引入不同类别障碍物的评价子函数并调节系数,与改进的 A\* 算法相融合,两个相邻最优节点序列节点间完成局部路径规划,解决了这些问题。仿真实验验证了该算法的有效性,能够使自主移动机器人搜索到最优路径并生成平滑轨迹,同时实现实时避障和安全运行。然而,本方法目前仅在仿真条件下测试,未来的工作需要在实际环境中进行更广泛的测试,以验证其适用性和可行性。

## 参考文献

- [1] 宋江一,李丹,陈文博. 融合 Dijkstra 和 PID 算法的室内移动机器人局部路径规划[J]. 安徽工业大学学报(自然科学版), 2023, 40(1): 59-64.
- [2] 周敬东,高伟周,杨文广,等. 基于改进蚁群算法的移动机器人路径规划[J]. 科学技术与工程, 2022, 22(28): 12484-12490.
- [3] 李晓旭,马兴录,王先鹏. 移动机器人路径规划算法综述[J]. 计算机测量与控制, 2022, 30(7): 9-19.
- [4] 翟文鹏,刘润南,朱承元. 基于改进 Dijkstra 算法的滑翔路径优化[J]. 中国民航大学学报, 2022, 40(1): 1-6.
- [5] 邓云峥,黄翼虎. 复杂动态环境下基于 A\* 的改进 DWA 算法研究[J]. 电子测量技术, 2023, 46(9): 69-76.
- [6] 何壮壮,丁德锐. 基于 D-star 和 DWA 的改进机器人导航方法[J]. 电子测量技术, 2019, 42(12): 122-128.
- [7] 张真诚. 机器人路径规划的改进粒子群-蚁群算法[J]. 电子测量技术, 2021, 44(8): 65-69.
- [8] 赵明,郑泽宇,么庆丰,等. 基于改进人工势场法的移动机器人路径规划方法[J]. 计算机应用研究, 2020, 37(S2): 66-68, 72.
- [9] 杨大庆,谷侃锋. 基于快速扩展随机树的机械臂路径优化算法研究[J]. 组合机床与自动化加工技术, 2021(5): 10-12, 17.
- [10] 严浙平,黄俊儒,吴迪. 基于 RRT 和 DWA 的欠驱动 UUV 路径规划[J]. 数字海洋与水下攻防, 2020, 3(3): 258-264.
- [11] 迟旭,李花,费继友. 基于改进 A\* 算法与动态窗口法融合的机器人随机避障方法研究[J]. 2021, 42(3): 132-140.
- [12] 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]. 2018, 40(6): 903-910.
- [13] WANG Y X, TIAN Y Y, LI X, et al. Adaptive DWA algorithm for crossing dense obstacles[J]. Control and Decision, 2019, 34(5): 927-936.
- [14] 袁千贺, 魏国亮, 田昕, 等. 改进 A\* 和 DWA 融合的移动机器人导航算法[J]. 小型微型计算机系统, 2023, 44(2): 334-339.
- [15] 封硕, 吉现友, 程博, 等. 融合动态障碍物运动信息的路径规划算法[J]. 计算机工程与应用, 2022, 58(21): 279-285.
- [16] 绳红强, 黄海英, 崔毅刚. 基于 A\* 蚁群融合算法的避障路径规划研究[J]. 机电工程技术, 2022, 51(7): 45-49.
- [17] Jinping Wang. A simulated annealing algorithm and grid map-based UAV coverage path planning method for 3D reconstruction[J]. Electronics, 2021, 10.
- [18] 付子聪, 王丰. 融合改进 A\* 与 DWA 算法的医用配送机器人路径规划[J]. 自动化应用, 2023, 64(11): 19-23.
- [19] 尹庆文. 基于 GB-APFB-RRT\* 算法的机械臂路径规划[J]. 组合机床与自动化加工技术, 2023, (10): 11-15.
- [20] 温杰茂. 基于深度学习的粒子滤波定位算法研究[D]. 南昌: 华东交通大学, 2021.
- [21] 徐宇真. 室内移动机器人路径规划[D]. 沈阳: 沈阳工业大学, 2022.
- [22] 王洪斌, 尹鹏衡, 郑维, 等. 基于改进的 A\* 算法与动态窗口法的移动机器人路径规划[J]. 机器人, 2020, 42(3): 346-353.
- [23] 彭斌, 王力, 杨思霖. 基于改进 A\* 算法和动态窗口算法的自动导引小车轨迹规划[J]. 计算机应用, 2022, 42(S1): 347-352.

### 作者简介

**袁新亚**, 硕士研究生, 主要研究方向为机器人路径规划与跟踪控制。

E-mail: yuanxy2896@qq.com

**戴娟**, 博士, 副研究员, 主要研究方向为自主导航制导与控制、深空探测器制导与控制、智能控制方法等。

E-mail: daijuan@bistu.edu.cn