

DOI:10.19651/j.cnki.emt.2210854

改进型 A* 算法的可重构机器人路径规划研究*

段会龙¹ 武炎明² 刘金国³(1. 沈阳化工大学信息工程学院 沈阳 110142; 2. 沈阳航空航天大学自动化学院 沈阳 110136;
3. 中国科学院沈阳自动化研究所 沈阳 110016)

摘要: 路径规划是保证可重构机器人快速完成任务的关键技术之一。为提高可重构机器人的行驶效率,缩短行驶路径,首先,提出一种基于 Bresenham 直线算法思想的改进型 A* 路径规划算法,实现可重构机器人路径点数消减、拐点消除,提高路径平滑度。在此基础上,考虑可重构机器人本身体积以及机器人可重构的特性,建立可重构机器人构型库,讨论了可重构机器人体积与周围障碍物的关系,减少机器人行走过程中与障碍物的碰撞几率。利用 MATLAB 仿真平台对改进型 A* 路径规划算法进行仿真实验,验证了算法的有效性,可应用于复杂环境的机器人路径规划;分析了机器人重构后路径规划问题,利用可重构特性可缩短机器人运行路程,体现了可重构机器人的优越性。

关键词: 可重构机器人; A*; Bresenham 直线算法; 路径规划; 构型库; MATLAB

中图分类号: TP242 **文献标识码:** A **国家标准学科分类代码:** 520.6

Path planning of reconfigurable robot based on improved A* algorithm

Duan Huilong¹ Wu Yanming² Liu Jinguo³(1. School of Information Engineering, Shenyang University of Chemical Technology, Shenyang 110142, China;
2. College of Automation, Shenyang Aerospace University, Shenyang 110136, China;
3. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China)

Abstract: Path planning is one of the key technologies for reconfigurable robots to accomplish tasks quickly. In order to improve the driving efficiency of the reconfigurable robot and shorten the driving path, an improved A* path planning algorithm based on the idea of Bresenham straight line algorithm is proposed to achieve path point reduction and inflection point elimination of the reconfigurable robot and improve the smoothness of the path firstly. On this basis, considering the volume of the reconfigurable robot itself and the reconfigurable characteristics of the robot, the reconfigurable robot configuration library is established, and the relationship between the volume of the reconfigurable robot and the surrounding obstacles is discussed to reduce the probability of collision between the robot and obstacles in the process of walking. The improved A* path planning algorithm is simulated by using MATLAB simulation platform to verify the effectiveness of the algorithm, which can be applied to robot path planning in complex environment. The path planning problem after robot reconfiguration is analyzed, and the running distance of robot can be shortened by using reconfigurable characteristics, which reflects the superiority of reconfigurable robot.

Keywords: reconfigurable robot; A*; Bresenham's line drawing algorithm; path planning; configuration library; MATLAB

0 引言

可重构机器人在复杂环境中可以依靠模块间的相互通信和自主运动重组来适应新环境、新任务,如灾难救援、反恐防爆等环境。与传统机器人相比,可重构机器人具有鲁

棒性高、多功能性、适应性强的独特性质^[1]。由于机器人工作环境具有不确定性与复杂性,机器人行驶路径中会遇到一些特殊障碍物,固定尺寸的机器人总是因为过小或者过大而无法通过一些特殊障碍物,并且固定尺寸的机器人在非结构化环境中运行时容易产生倾翻问题,极大地限制了

收稿日期:2022-07-27

* 基金项目:国家重点研发计划(2018YFB1304600)、辽宁省博士科研启动基金(2020-BS-026)、沈阳航空航天大学博士科研启动基金(22YB15)项目资助

机器人的作业环境,可重构机器人能够根据具体的环境状况和不同的任务要求做到构型切换,最终完成难度大、危险性高或不可预知的作业任务^[2]。

可重构机器人最终目标是能够顺利自主地完成指定任务,所以可重构机器人的路径规划显得尤为重要。路径规划主要是让目标对象在规定范围内快速找到一条从起始点到目标点的无碰撞安全路径^[3-6]。目前可重构机器人较为广泛使用的路径规划算法有A*算法^[7]、蚁群算法^[8-9]、视觉图法^[10]、人工势场法^[11]、遗传算法^[12]、神经网络^[13]等。其中,A*算法是一种很常用的全局路径查找和图形遍历算法,相较于蚁群算法、人工势场法、遗传算法等具有良好的通用性和拓展性,在单点对单点的寻径情况下有着较好的性能和准确度,能做到快速求解最优路径。但是传统A*算法规划的路径仍旧存在过多冗余点、转折点以及部分路径带有尖角不够平滑等问题。Cao等^[14]提出了Any-Angle A*算法,该算法在起点到目标点的连线附近进行搜索,减少了搜索节点,缩短了计算时间,然而它尚未考虑到移动机器人的体积实现防撞处理。赵丽华等^[15]提出一种将搜索邻域拓展为多个的方法使得生成路径长度变短,拐点个数变少,但搜索效率严重降低,而且没有考虑机器人与障碍物边缘碰撞可能。卫珊等^[16]改进A*算法中的启发函数的估计代价,对搜寻节点的数目减少起到作用,但是对于拐点问题没解决,同时也没有考虑机器人实际大小。孔继利等^[17]提出双向搜索路径,对于搜索时间有所改进,但没有解决拐点以及路径最优问题。

针对以上问题,本文设计了一种基于Bresenham直线算法思想的改进型A*路径规划算法。通过Bresenham直线算法思想,对A*规划出来的路径进行路径点数消减、拐点消除,提高路径平滑度;考虑机器人与周围障碍物的关系,分析障碍物顶点与机器人之间的距离,减小机器人行走过程中与障碍物碰撞几率;建立可重构机器人构型库,利用机器人可重构的特性,使得机器人行驶路径大幅度减少。

1 可重构机器人平台

1.1 可重构机器人结构

根据可重构机器人具有鲁棒性高、多功能性、适应性强等优点,设计了一款可重构机器人的整体拓扑结构方案,图1所示为可重构机器人外形,图2为可重构机器人内部机械结构,该机器人采用了模块化结构,4个连接臂将一个主模块与4个副模块连接在一起,每个副模块均具有独立驱动能力,可根据不同环境进行变形,增强机器人对环境的适应性。在图1中a、b为可重构机器人主履带,主要是实现机器人的前进后退;c、d、e、f是4个副模块,主要实现俯仰和偏航功能;g、h、i、j分别为c、d、e、f 4个副模块的俯仰关节;k、l、m、n分别为c、d、e、f 4个副模块的偏航关节。机器人主要参数:自然状态下机器人长度为93 cm;宽度为93 cm;其中副模块长为43 cm,宽为15 cm。

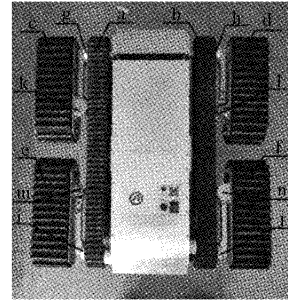


图1 可重构机器人

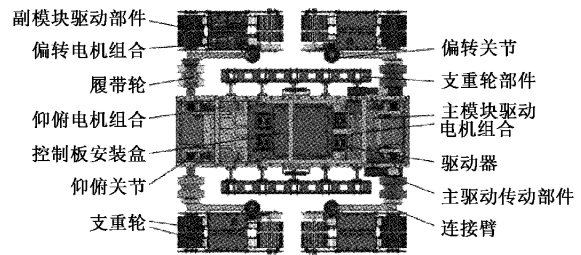


图2 机械结构

1.2 可重构机器人构型库

通过对可重构机器人的构型枚举计数并剔除对称的重复构形后,可得五模块可重构机器人共有27种非同构型变换^[18]。每种构型对地形以及任务的适应性能各不相同,可重构机器人在平地上移动常用构型如图3所示,参数如表1所示。

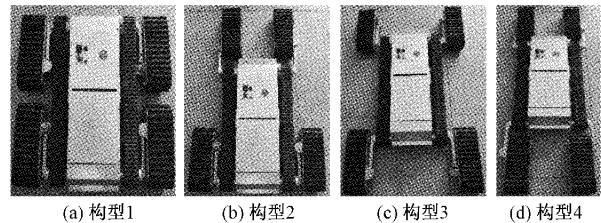


图3 常用构型

表1 常用变形结构参数

构型	长	宽	使用场景
构型1	93 cm	93 cm	自然状态
构型2	140 cm	93 cm	过沟
构型3	160 cm	93 cm	过沟
构型4	187 cm	55 cm	窄道、过沟

2 路径规划

2.1 环境建模

环境地图构建方法采用栅格法。栅格法是将机器人工作环境分成一系列具有二值信息的网格单元,模型建立具有简单,易于维护的优点^[19]。在复杂环境情况下,将机器人自身的体积问题考虑在内,建立环境栅格地图时选取比

可重构机器人自身体积稍大的栅格。考虑到可重构机器人自然状态下自身长为 93 cm、宽为 93 cm,忽略变形机器人的高度,建立环境的栅格地图方格为 100 cm。

2.2 传统 A* 算法

A* 算法是一种启发式搜索算法,将 Dijkstra 和 BFS 两种算法各自的优势信息进行结合。其基本思想是包含启发函数的估价函数来对节点进行评估,选择估价函数最小的节点进行扩展,拓展到终点时停止搜索,完成路径规划任务^[20]。A* 算法的常用扩展方式采用 8 邻域扩展,如图 4 所示,圆代表可重构机器人当前位置,箭头表示可重构机器人移动过程中可搜索的方向,分别是前、后、左、右、左前、左后、右前、右后。估价函数的形式如式(1)所示。

$$f(n) = g(n) + h(n) \tag{1}$$

式中: $f(n)$ 为当前节点 n 的总代价值, $g(n)$ 表示从起始节点到当前节点 n 的实际代价, $h(n)$ 表示从当前节点 n 到目标点的代价估计值。常见的计算代价的方法有曼哈顿距离、欧氏距离和切比雪夫距离,本文算法采取曼哈顿度量节点之间的代价,曼哈顿计算如式(2)所示。

$$d = |x_i - x_j| + |y_i - y_j| \tag{2}$$

式中: (x_i, y_i) 为起点 i 的坐标, (x_j, y_j) 为终点 j 的坐标。

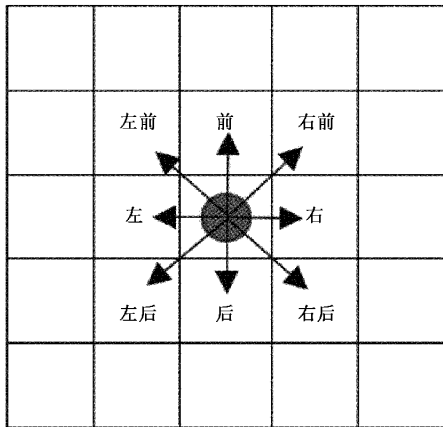


图 4 8 个运动方向

2.3 Bresenham 直线算法

Bresenham 直线算法^[21]是计算机图形学领域使用最广泛的直线扫描转换算法,该算法巧妙地利用误差项符号来判断下一像素点的位置,其优点是只需利用整数的加法和乘法即可完成待生成像素的计算,该算法不仅效率高、大大降低需要的计算资源而且能直接在硬件上实现^[22]。

将一个平面坐标系分成 8 个区域,区域划分如图 5 所示,Bresenham 算法绘制直线思路:

- 1) 水平直线、竖直直线单独处理;
- 2) 从最简单的 $x_0 < x_1, 0 < slope < 1$ 入手,剩下的 7 部分稍作修改即可,其中 $slope$ 为斜率。

以 $x_0 < x_1, 0 < slope < 1$ 为例说明 Bresenham 算法绘制直线方法。假设起始点为 (x_0, y_0) , 终点为 (x_1, y_1) 。

每次判断需要遍历横坐标: $x = x_0$ 到 x_1 , 根据直线方程,找出每个横坐标对应的纵坐标 y (注意:由于斜率 $0 < slope < 1, x$ 的值每增长 1, y 的值会在 y_0 和 $y_0 + 1$ 之间选择)。

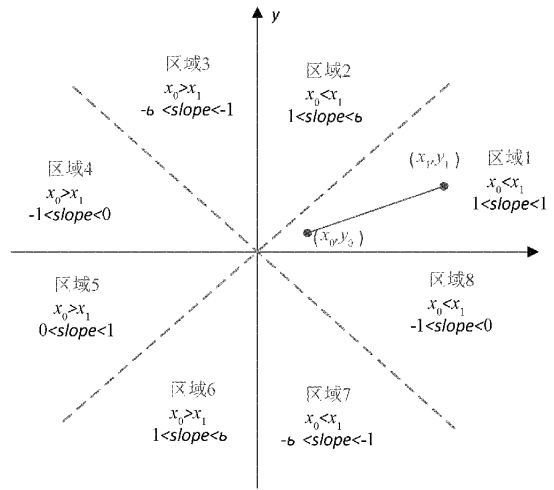


图 5 区域划分

图 5 中,可求得直线斜率为式(3):

$$slope = (y_1 - y_0) / (x_1 - x_0) = dy / dx \tag{3}$$

定义变量及初值如下:

$$d = slope = 2 \times dy - dx$$

$$incrNE = 2 \times (dy - dx)$$

$$incrE = 2 \times dy$$

分析多个点可得出结论:若前一像素是自东北方衍生的,则下一点与 $d = d + incrNE$ 进行比较;若前一像素是自东方衍生的,则下一点与 $d = d + incrE$ 进行比较,对 d 进行叠加,方便为下一点的判断。

2.4 改进 A* 算法

传统的 A* 算法应用范围广、适应性强,但是其规划的路径往往存在拐点多、路径不平滑等不足,不利于机器人的运行。本文提出一种基于改进型 A* 算法,利用 Bresenham 直线算法思想改善此问题,实现路径长度、平滑度和安全性等多重优化。

路径优化处理过程:将 A* 算法规划的路径点按顺序命名为节点 1、节点 2、节点 3 等。将节点 1 与节点 3 相连(记为 $L(1,3)$),若所连路径安全且无碰撞,则可以省略中间节点 2,并将节点 1 与节点 4 相连(记为 $L(1,4)$)继续判断,以此类推。若不可达,将前一节点作为新的起始点与下一节点相连,判断是否可达。以此类推,直到搜寻到终点位置,此时所得路径为优化后的路径。

假设 A* 算法规划的路径为 $Path$, 路径点数为 N 。每次优化的路径点数可以根据实际情况进行设定修改,在流程图中 $j > M$ 说明每次优化的路径点数不超过 M 。优化处理过程流程如图 6 所示。

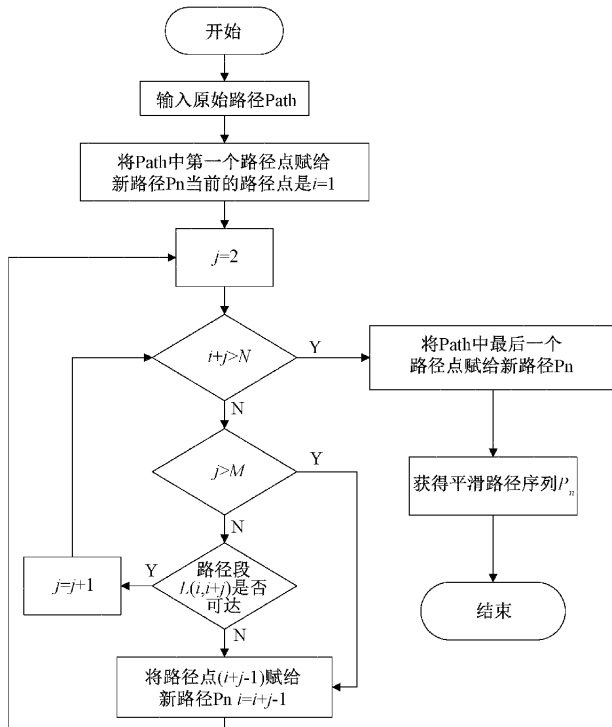


图 6 优化处理过程

使用 Bresenham 直线算法的思想判断路径点 $L(i, i+j)$ 是否可达, 通过计算直线经过的栅格点和障碍物点进行比较, 栅格上有障碍物, 则不可达, 若栅格上没有障碍物, 则进行下一步讨论。以区域 1 内为例说明一下如何判断路径是否可达: 假设要遍历横坐标为 x_0 到 x_1 。判断第一个点 (x_0+1) 时, 首先要判断 y_0 和 (y_0+1) 是否是障碍物, 同时计算出这个点衍生方向并使标志位置位, 接下来判断 (x_0+2) , 若上一点 (x_0+1) 是由东北方向衍生出来的, 则下一点必定是两个坐标点与障碍物点比较; 若上一点是由东方衍生出来的, 则下一点是一个坐标点与障碍物点比较。判断路径点 $L(i, i+j)$ 是否可达伪代码如表 2 所示。

表 2 判断路径点是否可达

Algorithm 1 Determine process

```

1:  $dx \leftarrow x_1 - x_0, dy \leftarrow y_1 - y_0$ 
2:  $incrNE \leftarrow 2 \times (dy - dx), incrE \leftarrow 2 \times dy$ 
3:  $d \leftarrow 2 \times dy - dx, k \leftarrow 0 \times dy$ 
4:  $flag0 \leftarrow flag1 \leftarrow false$ 
5: for  $x_0+1$  to  $x_1$  by 1 do
6:   if  $flag1 == true$  then
7:     if  $d < 0$  then
8:        $d \leftarrow d + incrE$ 
9:        $flag0 \leftarrow true$ 
10:    else
11:       $d \leftarrow d + incrNE$ 
12:       $flag1 \leftarrow true$ 

```

```

13:   end if
14:    $e \leftarrow (x, y), f \leftarrow (x, y+1)$ 
15:   else if  $flag0 == true$  then
16:     if  $d < 0$  then
17:        $d \leftarrow d + incrE$ 
18:        $flag0 \leftarrow true$ 
19:     else
20:        $d \leftarrow d + incrNE$ 
21:        $flag1 \leftarrow true$ 
22:     end if
23:      $e \leftarrow (x, y), f \leftarrow (x, y)$ 
24:   else
25:     if  $k == 0$  then
26:       if  $d < 0$  then
27:          $d \leftarrow d + incrE$ 
28:          $flag0 \leftarrow true, flag1 \leftarrow false$ 
29:       else
30:          $d \leftarrow d + incrNE$ 
31:          $flag1 \leftarrow true, flag0 \leftarrow false$ 
32:       end if
33:     end if
34:      $k \leftarrow -1, e \leftarrow (x, y), f \leftarrow (x, y+1)$ 
35:   end if
36:   if  $isObstacle(e, obstacle) \vee isObstacle(f, obstacle)$  then
37:     break
38:   end if
39: end for

```

在路径优化时, 除了对线段经过的栅格进行判断外, 还要进一步考虑可重构机器人本身体积的情况下, 判断机器人是否与障碍物发生碰撞问题。判断依据: 栅格顶点到规划路径线段的距离, 若距离小于 30 cm (根据构型 4 中车宽为 55 cm 求得车中心到车边缘为 27.5 cm), 则说明机器人行驶过程中会碰到障碍物, 则只优化到前一节点即可; 若距离大于 30 cm, 说明不会碰到障碍物, 继续运行。

判断方法: 根据 $(x_0, y_0), (x_1, y_1)$ 点将要规划的线段方程求出, 然后沿 y 轴方向进行上下平移一定距离, 比较障碍物的顶点是否存在于上下两条直线围成的矩形之中, 若是, 则说明会碰到障碍物; 若不是, 则此路径可行。上下平移的距离 l 按式(4)计算出, 其中 $slope$ 表示斜率。

$$l = \sqrt{0.3^2 + (0.3 \times slope)^2} \quad (4)$$

在图 7、8 中, 设 $(1, 1)$ 为优化路径的起始坐标, $(4, 3)$ 为优化路径的终点坐标。线段 1 为两坐标点构成的路径, 线段 2 和线段 3 分别是线段 1 向上平移 36 cm 和向下平移 36 cm 得到的。如图 7 所示, 障碍点坐标为 $(3, 1)$, 离线段最近的顶点 $(2.5, 1.5)$ 不在线段 2 线段 3 围成的区域内, 说

明可以安全通过;如图 8 所示,障碍点为(4,2),障碍物顶点坐标(3.5,2.5)在线段 2 线段 3 围成的区域内,说明行驶时

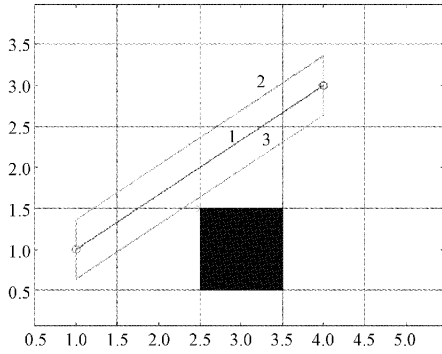


图 7 安全通过

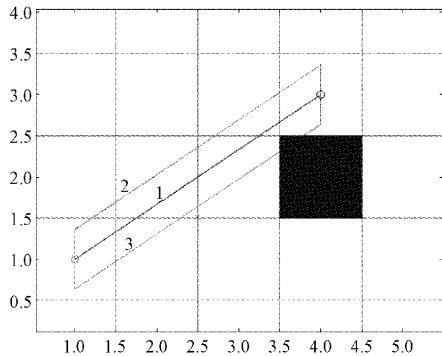
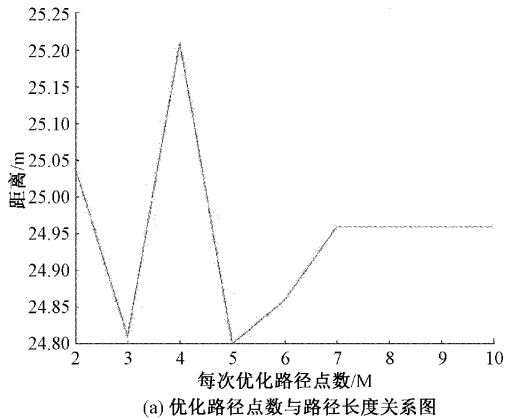


图 8 不能通过

3 实验演示

为验证改进 A* 算法的有效性,本文分别在 3 个不同尺寸的二维栅格地图中进行了仿真对比实验。在区域中分布着一些障碍物,设置机器人行走的起点与终点。在实验环境下,分析每次优化路径点数的改变对路径长度产生的影响,并将最优改进 A* 算法路径规划方法设置为实验方法,将本文提出的改进 A* 算法与传统的 A* 算法、基于双向搜索改进的 A* 算法、基于 16 邻域搜索改进的 A* 算法进行比较,运行结果图如图 9 所示。



(a) 优化路径点数与路径长度关系图

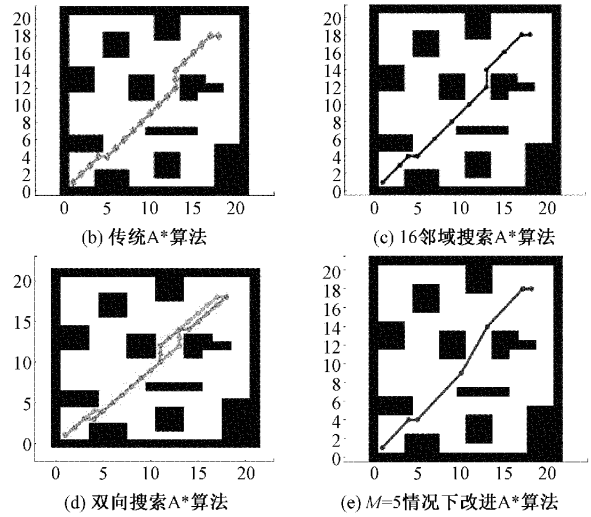
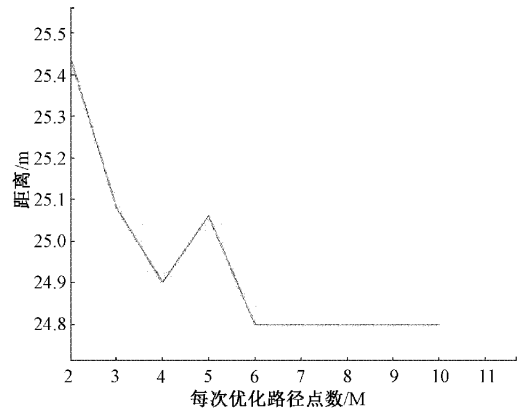


图 9 20×20 栅格地图路径规划对比



(a) 优化路径点数与路径长度关系图

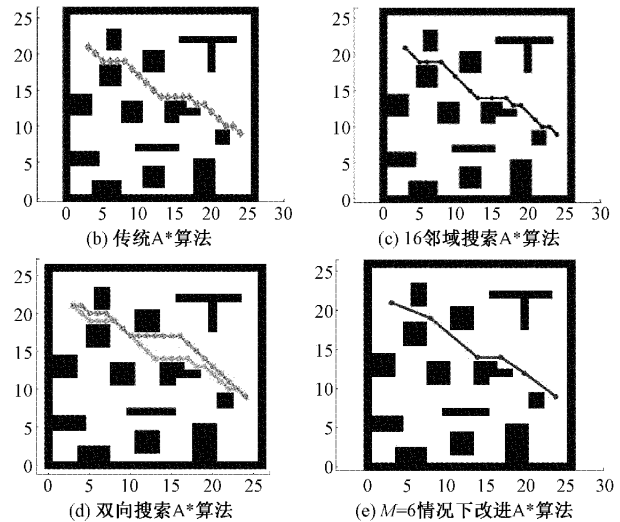
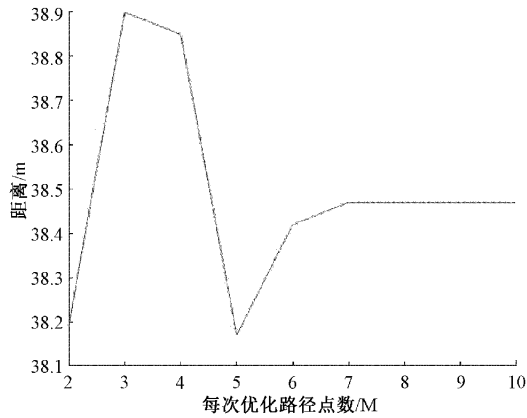


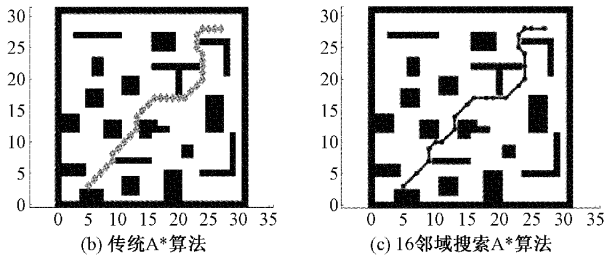
图 10 25×25 栅格地图路径规划对比

根据上述栅格地图下路径规划线路可以得到表 3 所示结果。

表 3 可以直观的看出行驶路径长度方面优劣,本文改进算法>16 邻域搜索>双向搜索>传统 A* 算法。将图 11

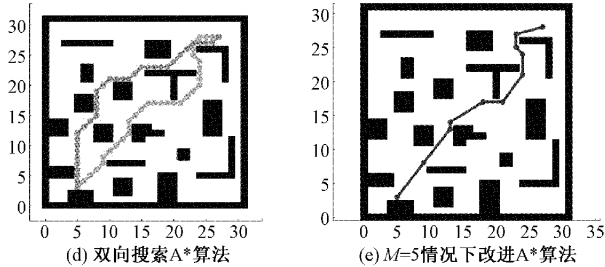


(a) 优化路径点数与路径长度关系图



(b) 传统A*算法

(c) 16邻域搜索A*算法



(d) 双向搜索A*算法

(e) M=5情况下改进A*算法

图 11 30×30 栅格地图路径规划对比

表 3 实验结果对比

图例	算法	路径点数	拐点个数	路径长度/m
图 9	传统 A* 算法	20	5	25.21
	16 邻域搜索	12	5	24.96
	双向搜索	20	5	25.13
图 10	改进 A* 算法	7	5	24.80
	传统 A* 算法	22	8	25.97
	16 邻域搜索	15	8	25.10
图 11	双向搜索	22	4	25.97
	改进 A* 算法	6	4	24.80
	传统 A* 算法	34	11	39.63
图 11	16 邻域搜索	22	13	39.39
	双向搜索	34	11	39.42
	改进 A* 算法	11	8	38.17

对比实验中本文改进 A* 算法与其他算法进行详细数据分析,在路径点数优化方面,本文算法相较于传统 A* 算法、双向搜索、16 邻域搜索优化率分别为 67.65%、67.65%、50%;在拐点个数优化方面,优化率分别为 27.27%、

27.27%,38.46%;在路径长度优化方面,优化率分别为 3.68%、3.17%、3.10%。由表格数据及上述分析可知,相同环境下本文改进 A* 算法规划出的路径远远优于传统 A* 算法、双向搜索 A* 算法和 16 邻域搜索 A* 算法。

为验证可重构机器人的优越性,在区域中添加变形可通过障碍物,添加 70 cm 宽度的窄道如图中粉色方格所示,添加 40 cm 沟壑如图中黄色方格所示,根据构型库可知构型 4 可以穿过该窄道,构型 2 或 4 可跨越沟壑。传统机器人路径规划如图 12 所示,可重构机器人路径规划如图 13 所示。

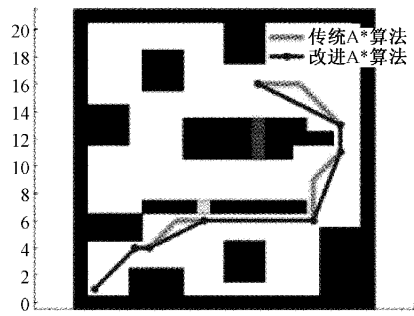


图 12 传统机器人路径规划

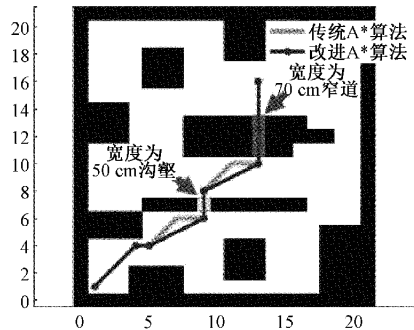


图 13 可重构机器人路径规划

根据两个路径规划的线路我们可以得到表 4 所示结果。

表 4 实验结果对比

图例	算法	路径点数	拐点个数	路径长度/m
图 12	传统 A* 算法	30	8	33.14
	改进 A* 算法	8	6	31.81
图 13	传统 A* 算法	21	7	22.90
	改进 A* 算法	7	5	22.19

从表 4 中可以得出结论:改进 A* 算法与传统 A* 算法相比,图 12 路径点数优化率为 73.33%,拐点个数优化率为 25%,路径长度优化率为 4.01%;图 13 路径点数优化率为 66.67%,拐点个数优化率为 28.57%,路径长度优化率为 3.10%。对图 13 和图 12 中两种改进 A* 算法进行比较,路径点数优化率为 12.50%,拐点个数优化率为 16.67%,路径

长度优化率为 30.24%。所以,可重构机器人在改进 A* 算法的基础上通过重构穿越特殊障碍物进行路径规划,更能大幅度缩短机器人运行路程,体现了可重构机器人的优越性。

4 结 论

本文研究可重构机器人路径规划,提出一种改进型 A* 算法用来解决传统 A* 算法路径规划中存在路径点数多和转弯次数较多的问题。利用 Bresenham 直线算法思想对 A* 算法规划的路径点进行优化,使得路径平滑度得到大大提高。另外,将机器人实际体积纳入考虑范畴,讨论了机器人体积与周围障碍物的关系,减少机器人在行走过程中与障碍物的撞击率,使得不合理拐弯点以及撞击率明显减少,同时依靠机器人重构后的状态更能通过一些窄道等特殊障碍物,缩短了机器人运行路程。

参考文献

- [1] 刘永进,余旻婧,叶子鹏,等. 自重构模块化机器人路径规划方法综述[J]. 中国科学: 信息科学, 2018, 48(2): 143-176.
- [2] LIU J G, ZHANG X, HAO G B. Survey on Research and Development of Reconfigurable Modular Robots[J]. Advances in Mechanical Engineering, 2016, 8(8): 1-21.
- [3] ZENG M R, XI L, XIAO A M. The free step length ant colony algorithm in mobile robot path planning[J]. Advanced Robotics, 2016, 30(23): 1509-1514.
- [4] 李晓旭,马兴录,王先鹏. 基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.
- [5] CONTRERAS-CRUZ M A, AYALA-RAMIREZ V, BELMONTE U H. Mobile robot path planning using artificial bee colony and evolutionary programming[J]. Applied Soft Computing, 2015, 30: 319-328.
- [6] AJEIL F H, IBRAHEEM I K, AZAR A T, et al. Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments [J]. Sensors (Basel, Switzerland), 2020, 20(7): E1880.
- [7] PAN H, GUO C, WANG Z D. Research for path planning based on improved A start algorithm[C]. 2017 4th International Conference on Information, Cybernetics and Computational Social Systems(ICCSS), 2017.
- [8] 李志锟,黄宜庆,徐玉琼. 改进变步长蚁群算法的移动机器人路径规划[J]. 电子测量与仪器学报, 2020, 34(8): 15-21.
- [9] 王刚,张方,严大亮,等. 基于改进 A* 算法蚁群算法的机器人三维路径规划[J]. 国外电子测量技术, 2020, 39(11): 1-6.
- [10] LYU T Z, FENG M Y, A smooth local path planning algorithm based on modified visibility graph[J]. Modern Physics Letters B, 2017, 31(19-21): 1740091.
- [11] 朱颖,李元鹏,张亚婉,等. 基于改进人工势场法的搬运机器人路径规划[J]. 电子测量技术, 2020, 43(17): 101-104.
- [12] 段建民,陈强龙. 基于改进人工势场-遗传算法的路径规划算法研究[J]. 国外电子测量技术, 2019, 38(3): 19-24.
- [13] 李卫硕,孙剑,陈伟. 基于 BP 神经网络机器人实时避障算法[J]. 仪器仪表学报, 2019, 40(11): 204-211.
- [14] CAO P, FAN Z Y, ROBERT X, et al. A focal any-angle path-finding algorithm based on A* on visibility graphs[J]. ArXiv Preprint, 2017, ArXiv:1706.03144.
- [15] 赵丽华,万晓冬. 基于改进 A 算法的多无人机协同路径规划[J]. 电子测量技术, 2020, 43(7): 72-75,166.
- [16] 卫珊,王凌,王斌锐,等. A* 算法的改进及其在 AGV 路径规划中的应用[J]. 自动化仪表, 2017, 38(11): 51-54.
- [17] 孔继利,张鹏坤,刘晓平. 双向搜索机制的改进 A* 算法研究[J]. 计算机工程与应用, 2021, 57(8): 231-237.
- [18] LIU J G, WANG Y C, MA S G, et al. Enumeration of the non-isomorphic Configurations for a Reconfigurable Modular robot with square-cubic-cell modules[J]. International Journal of Advanced Robotic Systems, 2010, 7(4): 58-68.
- [19] 周超,谷玉海,任斌. 基于一种改进 A* 算法的移动机器人路径规划研究[J]. 重庆理工大学学报(自然科学), 2021, 35(5): 127-134.
- [20] 张建光,张方,陈良港,等. 基于改进 A* 算法的自动引导车的路径规划[J]. 国外电子测量技术, 2022, 41(1): 123-128.
- [21] 樊瑛. 模块化移动机器人路径规划的实验研究[D]. 湖南:中南大学, 2012.
- [22] 卫洪春. 直线的 Bresenham 并行绘制算法[J]. 计算机系统应用, 2017, 26(8): 180-183.

作者简介

段会龙, 硕士研究生, 主要研究方向可重构机器人路径规划

E-mail:1747511818@qq.com

武炎明, 副教授, 主要研究方向多机器人协同控制, 机器人自主控制。

E-mail:yanmingwu@yeah.net

刘金国(通信作者), 研究员, 主要研究方向空间机器人与空间智能装备、公共安全机器人与无人平台、仿生可变形与可重构机器人等。

E-mail:liujinguo@sia.cn