

DOI: 10.13382/j.jemi.B2407223

改进 A* 算法的安全高效室内全局路径规划*

杨国^{1,2} 吴晓^{1,2} 肖如奇^{1,2} 张若愚^{1,2} 吴文^{1,2}

(1. 南京理工大学电子工程与光电技术学院 南京 210094; 2. 近程射频感知芯片与微系统教育部重点实验室 南京 210094)

摘要:针对 A* 算法生成路径存在斜穿障碍物、转折点多及不平滑的情况,本文提出一种改进的 A* 算法。首先通过排除所有强迫邻居节点来优化搜索邻域,避免生成斜穿障碍物的路径,提升路径的安全性和可靠性;其次设置安全距离,提取优化邻域后生成路径的必经转折点,减少路径冗余,简化路径结构;最后使用贝塞尔曲线对必经转折点进行插值,根据相邻两必经转折点的位置和连线斜率确定每段贝塞尔曲线控制点的数量和位置,进行分段平滑。仿真实验表明,改进的 A* 算法较原 A* 算法生成的路径安全性平均提升了 33.68%,转折点个数平均减少了 37.00%,同时机器人转向角和路径曲率连续,保证了路径的平滑。改进的 A* 算法最终生成的路径与障碍物均保持在安全距离内,转折点少且平滑,可应用到移动机器人室内路径规划中。

关键词: 路径规划; A* 算法; 安全; 转折点; 路径平滑

中图分类号: TP242.6 文献标识码: A 国家标准学科分类代码: 510.80

Improved A* algorithm for secure and efficient indoor global path planning

Yang Guo^{1,2} Wu Xiao^{1,2} Xiao Ruqi^{1,2} Zhang Ruoyu^{1,2} Wu Wen^{1,2}

(1. School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China;

2. Key Laboratory of Near-Range RF Sensing ICs & Microsystems (NJUST), Ministry of Education, Nanjing 210094, China)

Abstract: This paper presents an enhanced A* algorithm aimed at resolving issues of diagonal traversal of obstacles, excessive numbers of turning points, and non-smooth paths produced by the conventional A* algorithm. Initially, by excluding all forced neighboring nodes, the search neighborhood is optimized to prevent the generated path from diagonally crossing obstacles, thereby enhancing path safety and dependability. Subsequently, a safe distance is established, and crucial turning points are extracted from the generated path after optimizing the neighborhood, reducing path redundancies and simplifying the path structure. Lastly, the algorithm employs Bezier curves to interpolate essential turning points, determining the quantity and position of control points for each segment based on the positions of adjacent necessary turning points and the connecting line slopes to achieve segmental smoothing. Simulation experiments demonstrate that compared to the original A* algorithm, the improved A* algorithm exhibits an average 33.68% enhancement in path safety and a corresponding average reduction of 37.00% in the number of turning points. Additionally, the robot's turning angle and path curvature are continuous, ensuring path smoothness. The paths generated by the refined A* algorithm are smooth, with few turning points, and maintain a safe distance from obstacles, which can be applied to indoor path planning for mobile robots.

Keywords: path planning; A* algorithm; safety; turning point; smooth path

0 引言

移动机器人已广泛运用在生活生产中的各个领域^[1-3],路径规划是其中的关键技术之一。与室外环境相

比,室内空间面积窄,可通行区域少,障碍物种类形状复杂、障碍物变化更为频繁^[4-5],大多数机器人设备较为昂贵,若与障碍物发生碰撞容易造成损坏,因此路径的安全性应作为机器人室内路径规划中非常重要的考虑因素。由于地面摩擦、车轮打滑及定位、转角等方面的误差,实

实际行驶的路径与规划的路径存在一定误差^[6], 为避免因误差而导致碰撞, 路径规划算法生成的路径需可通行并应尽量与障碍物保持一定距离, 这也将为后续路径的平滑处理能留下一定的空间。

移动机器人室内路径规划方法可分为传统算法^[6-12]、仿生算法^[13-16]和人工智能算法^[17-18]。其中传统算法和仿生算法内存占用小, 常用于已知环境中的全局路径规划^[19]; 人工智能算法能很好地处理环境中的不确定性, 常用于环境中存在未知信息的局部路径规划^[17]。

快速搜索随机树法 (rapid-exploration random tree, RRT)、蚁群算法 (ant colony optimization, ACO) 和 A* 算法是常见的全局路径规划算法。RRT 算法对地图进行完整探索, 能够找到路径, 但该路径不一定为最优; 算法对环境不敏感, 当环境中存在大量障碍物或狭窄通道时, 算法收敛缓慢, 效率急剧下降^[19]。ACO 算法具有正反馈、并行计算的特点以及良好的鲁棒性性能, 但存在收敛速度慢、容易陷入局部最优等缺点^[11]。A* 算法的规划效率最高, 能够找到最短路径, 算法复杂度低, 二维表现力高, 内存需求量弱^[4, 18, 20], 已广泛使用在移动机器人路径规划中^[1]。但 A* 算法将机器人视作不具有形状和大小质点的质点, 生成的路径会出现斜穿障碍物顶点的情况, 而这在空间狭小、障碍物密集的室内环境中, 路径的安全可靠性将大大下降; 算法生成的路径转折点多, 增加了行驶时间和转向次数; 算法生成的路径并不平滑, 需要停下转向, 增加了机器人的加速、减速的次数, 不但能耗增加, 而且影响机器人运动的稳定性和精准性。

近年来, 多位学者对 A* 算法的不足进行了改进。文献[21]考虑节点与障碍物中心点的距离是否超过所设的安全距离来处理路径斜穿障碍物的情况, 但该方法并不是在 A* 算法生成时就避免产生斜穿障碍物的路径。文献[22]和[23]设置一个栅格单位的安全距离, 在算法的搜索阶段即排除所有障碍物周围一个栅格单位的节点, 而这在具有狭窄通道的室内可能会错失最优路径。文献[24]考察当前路径点相邻 3 个路径点的夹角以及前后两点连线是否经过障碍物来去除冗余点。文献[25]从起点出发, 依次连接剩余的点, 保留路径可行且编号最大的点, 再以该点为起点, 重复上述过程, 直至找到终点。以上两种方法能够有效去除冗余路径点, 但会出现新规划的路径距离障碍物过近的情况, 路径的安全性不能得到保证, 并且若路径点很多, 计算量会很大。文献[21]和[26]虽有考虑安全距离, 但生成的路径还是存在与障碍物较近的情况。文献[27]使用贝塞尔曲线进行路径平滑, 但并未具体说明贝塞尔曲线控制点的选取原则, 而控制点的数量和位置将影响生成路径点的曲率, 进而影响到路径跟踪。

本文提出一种改进的 A* 算法, 生成的路径转角少,

兼具安全性和平滑性。算法基于原 A* 算法做出 3 点改进: 首先, 优化搜索邻域, 以避免生成的路径斜穿障碍物; 其次, 提取路径中的必经转折点, 用于减少路径中的冗余; 最后, 使用贝塞尔曲线进行路径平滑。

1 原 A* 算法

A* 算法常使用栅格地图来进行环境建模, 把每一个栅格视作一个节点, 每个节点具有一些属性, 包括父节点、子节点和总代价。除起点和终点外, 每个节点均有父节点和子节点, 子节点在节点的邻居节点中选取。

A* 算法的寻路过程可以简单表述为: 从起点开始, 每次从 open list 列表中选取具有最小总代价的节点来作为当前节点, 再检查与其相邻的邻居节点, 将不在 open list 中的邻居节点加入 open list, 更新邻居节点与当前节点的父子关系, 计算邻居节点的总代价, 不断重复上述过程, 直至终点加入到 open list 中。最后从终点出发, 沿着父节点一路找到起点, 即为 A* 算法找到的路径。A* 算法的具体寻路过程参见文献[9]。

总代价可由式(1)表示为:

$$F(n) = G(n) + H(n) \tag{1}$$

其中, n 表示节点, $F(n)$ 表示节点 n 的总代价, $G(n)$ 表示从节点 n 经由父节点到起点的移动代价, $H(n)$ 表示从节点 n 到终点的估计移动代价 (忽略障碍物)。

如图 1 所示, 当前节点的邻居节点为 1~8 时, 搜索邻域为 8 邻域, 邻居节点仅为 2/4/5/7 时, 搜索邻域为 4 邻域。若设置横向移动的代价为 1, 斜向移动的代价为 1.4, 则根据当前节点到其父节点的横向或斜向移动的代价, 加上其父节点的 G 值, 可计算每个节点的 G 值。

| | | |
|-------|-------|-------|
| 7.4 | 6 | 5.4 |
| 邻居节点1 | 邻居节点2 | 邻居节点3 |
| 1.4 6 | 1 5 | 1.4 4 |
| 7 | F | 5 |
| 邻居节点4 | 当前节点 | 邻居节点5 |
| 1 6 | G H | 1 4 |
| 7.4 | 6 | 5.4 |
| 邻居节点6 | 邻居节点7 | 邻居节点8 |
| 1.4 6 | 1 5 | 1.4 4 |

图 1 算法邻域与代价函数示意图

Fig. 1 Diagram of algorithm neighborhood and cost function

$H(n)$ 又称为启发函数, 有多种经典的计算方法, 包括曼哈顿距离、欧几里得距离、切比雪夫距离和对角线距离。其中, 从当前位置 (x_n, y_n) 移动到终点 (x_g, y_g) 的欧几里得距离 d_e 可由式(2)表示。

$$d_E = \sqrt{(x_g - x_n)^2 + (y_g - y_n)^2} \quad (2)$$

2 改进的 A* 算法

改进的 A* 算法对原 A* 算法做出 3 个方面的改进,分别是优化邻域、提取必经转折点和路径平滑。优化邻域后生成的路径点作为提取必经转折点这一处理的输入,提取必经转折点后生成的必经转折点作为路径平滑的输入,路径平滑后生成的路径点相连即为改进的 A* 算法最终生成的路径。在改进过程中,均保证了路径的安全性。算法的结构如图 2 所示。



图 2 改进的 A* 算法结构

Fig. 2 Improved A* algorithm structure

本文认为生成的路径依次满足以下条件,路径的安全性依次升高:1)不穿过障碍物;2)不斜穿障碍物顶点;3)不与障碍物靠的过近。安全性的具体数值衡量以安全路段占全路径长度的百分比来代替。

认为生成的路径若满足以下条件,则路径具有平滑性:1)机器人朝向与水平方向的夹角(转向角)在行驶过程中不发生突变;2)曲率连续。

2.1 优化邻域

机器人在斜向移动的过程中,由于自身有一定的体积,使用原 A* 算法,若按照八邻域搜索,生成的路径可能存在斜穿障碍物的情况,如图 3 所示。

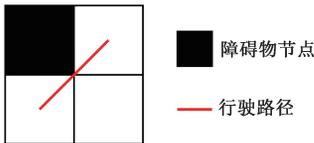


图 3 行驶路线斜穿障碍物

Fig. 3 Driving route diagonally crossing obstacles

为避免这种情况,采用一种搜索邻域优化的策略:遍历当前节点的 8 邻域邻居节点,若当前节点的四邻域方向存在障碍物,则将障碍物水平或垂直方向上的相邻节点称为“强迫邻居节点”^[28],在原搜索邻域中排除强迫邻居节点(即不将其作为拓展节点)和障碍物所在的节点,进而完成对搜索邻域的优化,具体如表 1 所示,表中的邻

居节点的编号来自图 1。

表 1 根据障碍物所在位置排除的邻居节点

Table 1 Neighbor nodes excluded based on the location of obstacles

| 障碍物所在位置 | 排除的邻居节点 |
|---------|------------|
| 邻居节点 2 | 邻居节点 1/2/3 |
| 邻居节点 4 | 邻居节点 1/4/6 |
| 邻居节点 5 | 邻居节点 3/5/8 |
| 邻居节点 7 | 邻居节点 6/7/8 |

排除强迫邻居节点后的各式搜索邻域如图 4 所示。

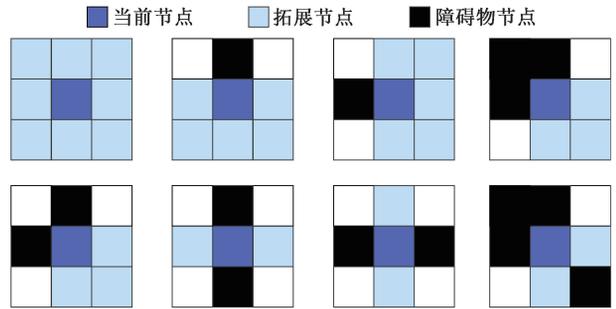


图 4 排除强迫邻居节点后的各式搜索邻域

Fig. 4 Various search neighborhoods after excluding forced neighbor nodes

2.2 提取必经转折点

本文对优化邻域后生成的路径点进行必经转折点的提取处理,以减少路径冗余和机器人的转向次数。为达到提取必经转折点的效果,首先需要提取路径中的转折点,然后通过判断转折点前后两点连线是否经过障碍物以及是否与障碍物靠的过近来去除转折点中的冗余点,留下的即为必经转折点。

1) 提取转折点

由于原路径存在相邻两点在同一直线上的情况,这样的点可视作冗余点去除,因此首先提取路径中的转折点。

具体做法是:输入优化邻域后生成的一系列路径点: $n_0, n_1, n_2, \dots, n_j, \dots, n_J, n_{j+1}$, 其中 $n_0 = n_s$ 为起点, $n_{j+1} = n_g$ 为终点, n_j 为中间路径点, $j = 1, 2, \dots, J$, 考察 n_{j-1}, n_j, n_{j+1} 3 点是否共线,若 3 点共线,则 n_j 为冗余点,可去除;若 3 点不共线,则 n_j 为转折点,保留。保留下来的这些路径点即为路径中的转折点。

对算法生成的路径进行转折点提取,效果如图 5 所示,其中圆点表示生成的路径点,实线表示路径点依次连接生成的路径。

2) 保留必须经过的转折点

在 2.2 节 1) 中提取了路径中的转折点,但这些转折点仍然存在冗余,在保证路径安全的情况下,可将冗余的

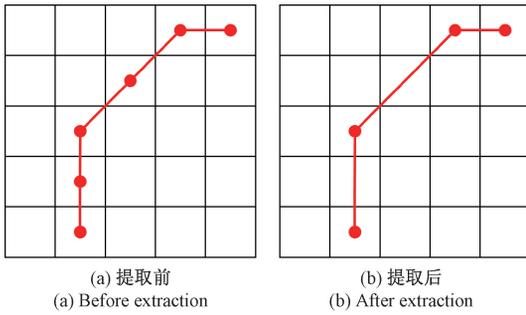


图 5 提取路径中的转折点前后对比

Fig. 5 Comparison before and after extracting turning points in the path

转折点去除,保留下必须经过的转折点。

保留必须经过的转折点的具体做法是:输入提取到的一系列转折点 $s_0, s_1, s_2, \dots, s_l, \dots, s_L, s_{L+1}$, 其中 $s_0 = n_s$ 为起点, $s_{L+1} = n_g$ 为终点, s_l 为中间转折点, $l = 1, 2, \dots, L$ 。对当前转折点 s_l 的前后两点 s_{l-1} 和 s_{l+1} , 考察两点连线 $s_{l-1}s_{l+1}$ 是否满足以下条件:(1) 不经过障碍物;(2) 不与障碍物靠的过近。若均满足,则将 s_l 去除,若不能同时满足,则保留 s_l 。保留下来的这些转折点即为路径中的必须经过的转折点(必经转折点)。

判断两点连线是否经过障碍物以及是否与障碍物靠的过近,首先都需要得到两点连线经过的方格坐标,而要得到方格坐标,首先需要计算两点连线与坐标轴的交点坐标,再根据交点坐标计算出连线经过的方格坐标。具体步骤如下:

(1) 获取考察范围:计算两点 $s_{l-1}(x_{l-1}, y_{l-1})$ 、 $s_{l+1}(x_{l+1}, y_{l+1})$ 连线与坐标轴的交点,首先需要获取考察 x 的范围 x_{ar} 和考察 y 的范围 y_{ar} , 计算方法可分别由式(3)和式(4)表示,其中 $ceil(\cdot)$ 表示向上取整, $floor(\cdot)$ 表示向下取整, x_{\min}, x_{\max} 为两点 x 轴坐标的最小值和最大值, y_{\min}, y_{\max} 为两点 y 轴坐标的最小值和最大值。 P 的值根据栅格地图中网格线的设置来定,若网格线为整数,则 P 为 1,若网格线为带 .5 的小数,则 P 为 0.5。

$$x_{ar} = [ceil(x_{\min} - P) + P, floor(x_{\max} - P) + P] \quad (3)$$

$$y_{ar} = [ceil(y_{\min} - P) + P, floor(y_{\max} - P) + P] \quad (4)$$

(2) 求交点坐标:令 $x_i \in x_{ar}, y_i \in y_{ar}$, 依次递增 x_i, y_i (每次递增 1), 可由式(5)和(6)求得交点的坐标,其中两点所在的直线连线方程满足: $y = kx + b$ 。

$$\begin{cases} y = kx + b \\ x = x_i \end{cases} \rightarrow y = kx_i + b \quad (5)$$

$$\begin{cases} y = kx + b \\ y = y_i \end{cases} \rightarrow x = (y_i - b)/k \quad (6)$$

(3) 计算经过的方格坐标:根据式(5)和(6)求得交点的坐标 $(x_{c,i}, y_{c,i})$ 、 $(x_{c,i+1}, y_{c,i+1})$, 可进一步计算经过

的方格的坐标 $(x_{w,i}, y_{w,i})$, 如式(7)和(8)所示。

$$x_{w,i} = \max(ceil(x_{c,i} - P), ceil(x_{c,i+1} - P)) \quad (7)$$

$$y_{w,i} = \max(ceil(y_{c,i} - P), ceil(y_{c,i+1} - P)) \quad (8)$$

效果如图 6 所示,先计算两点连线与网格的交点,再根据交点计算连线经过的方格坐标。

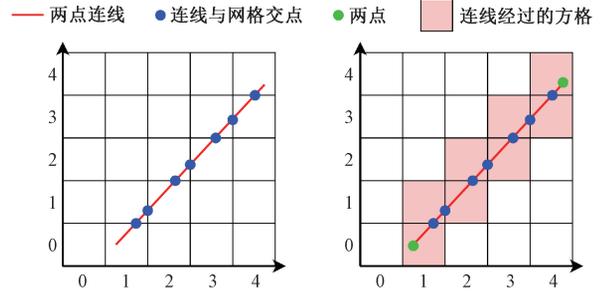


图 6 两点连线经过的方格坐标过程示意

Fig. 6 Schematic diagram of the grid coordinate process passed by the line connecting two points

(4) 排序去重:将连线经过的方格坐标按照 x 轴的大小排序,并删除重复的方格坐标,得到 $s_{l-1}s_{l+1}$ 经过的一系列方格坐标 $w_{l,0}, w_{l,1}, w_{l,2}, \dots, w_{l,e}, \dots, w_{l,E}$, $e = 0, 1, 2, \dots, E$ 。

求得连线经过的方格坐标后,可通过查看方格上是否有障碍物来判断连线是否经过障碍物。而考察连线是否与障碍物靠的过近,需要设置安全距离,再计算连线与障碍物的距离,与安全距离相比较,来确定路径是否可行。

判别两点连线经过的方格周围是否与障碍物靠的过近的过程示意如图 7 所示,具体的做法是:遍历连线经过的方格坐标 $w_{l,e}$, 先观察 $w_{l,e}$ 的 8 邻域内是否有障碍物;若存在障碍物,则计算障碍物 4 个顶点到连线的垂直距离 d , 若不存在障碍物,则不存在与障碍物靠的过近的情况;最后将 d 与安全距离 r_s 相比较,若同时满足:(1) d 小于 r_s ; (2) 垂足正好落在线段上,则说明连线与障碍物靠的过近,若不能同时满足,则不存在与障碍物靠的过近的情况。

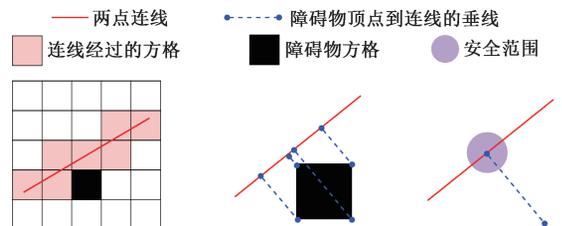


图 7 判别两点连线是否与障碍物靠的过近的过程示意

Fig. 7 Diagram of the process of distinguishing whether the connection between two points is too close to the obstacle

在本文中,设置栅格的大小为 1, 机器人半径 $r = 0.3$, r_s 应大于 $r/2$, 小于栅格大小的 $1/2$, 综合考虑,本文

设置为 0.4。

3) 具体步骤

综合 2.2 节 1) 和 2) 中提到的方法,提取必经转折点具体步骤如图 8 所示。对于输入的路径点(此处为优化邻域后生成的路径点),首先提取其中的转折点,然后遍历这些转折点,考察每一个转折点相邻两点的连线是否经过障碍物,连线周围是否与障碍物靠的过近,若均满足这两个条件,则认为当前转折点为必经转折点。

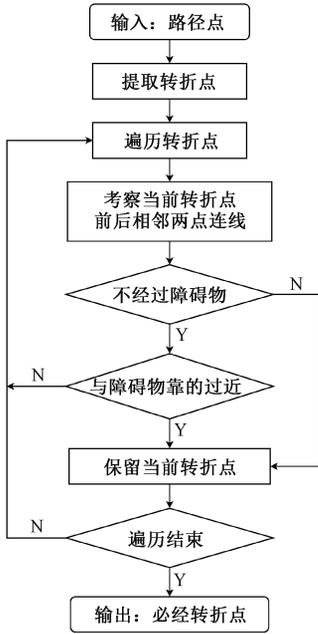


图 8 提取路径中的必经转折点流程

Fig. 8 Process diagram for extracting necessary turning points in the path

2.3 路径平滑

若定义转向角为机器人朝向与水平方向 x 轴的夹角,则任意两路径点 $p_v(x_v, y_v)$ 、 $p_{v+1}(x_{v+1}, y_{v+1})$ 连线生成的转向角 θ_v 如图 9 所示。

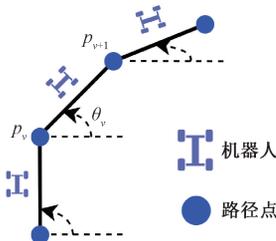


图 9 任意两路径点连线生成的转向角

Fig. 9 Turning angle generated by connecting any two path points

提取必经转折点后生成了一系列路径点(加上起点终点),连线后得到的路径为折线,可由式(9)计算出对

应的转向角,计算可知转向角在必经转折点处并不连续,会发生突变。因此有必要对生成路径进行平滑处理,本文在路径平滑阶段使用贝塞尔曲线来对提取的必经转折点分段路径平滑。

$$\theta_v = \arctan\left(\frac{y_{v+1} - y_v}{x_{v+1} - x_v}\right) \quad (9)$$

1) 贝塞尔曲线

贝塞尔曲线可用于路径平滑,与其他常用的路径平滑方法^[21]相比,能够生成曲率连续的曲线,满足移动机器人的行驶路径在方向和曲率上均连续^[29]的要求。

给定一共 $M + 1$ 个控制点 $P_0, P_1, P_2, \dots, P_m, \dots, P_M$, 则 M 阶贝塞尔曲线可由式(10)和(11)表示。

$$B(t) = \sum_{m=0}^M \binom{M}{m} P_m (1-t)^{M-m} t^m \quad (10)$$

$$\binom{M}{m} = \frac{M!}{m! (M-m)!} \quad (11)$$

其中, P_m 表示第 m 个点的 x 轴或 y 轴坐标, $t \in [0, 1]$ 控制生成贝塞尔曲线的点数,若取 t 的步长为 0.1,则一共生成 11 个点,若取 t 的步长为 0.01,则一共生成 101 个点。

若将式(10)视作式(12)和(13)所示的参数方程,则贝塞尔曲线生成的点集中每一点的曲率 κ 可由式(14)^[30]表示,其中 $B'(t)$ 为 x 或 y 方向对参数 t 的一阶导数, $B''(t)$ 为 x 或 y 方向对参数 t 的二阶导数。

$$x = B_x(t) \quad (12)$$

$$y = B_y(t) \quad (13)$$

$$\kappa = \frac{B'_x(t)B''_y(t) - B'_y(t)B''_x(t)}{((B'_x(t))^2 + (B'_y(t))^2)^{3/2}} \quad (14)$$

路径曲率反映了路径的平滑度,曲率变化越小,累积误差越小^[8]。曲率若为正数,曲线的方向为凹,反之则为凸;曲率的绝对值数值越大,曲线的弯曲程度越大,可将直线看成特殊的曲线,它的曲率为 0。

2) 控制点的选取

由于使用全部路径点生成贝塞尔曲线会出现曲线穿过障碍物的情况,因此本文仅对路径中的必经转折点使用贝塞尔曲线来进行分段路径平滑,这一处理涉及到具体控制点个数和位置的选取。

贝塞尔曲线的阶数 $M \geq 3$ 时,能保证曲线的曲率连续。由于直线的曲率为 0,若进行分段平滑,需保证曲线在 P_0 和 P_M 两端点处的曲率为 0。为使曲率连续,本文根据相邻两必经转折点的位置情况和连线斜率,来确定控制点的数目和位置。

由于在提取必经转折点的过程中已经去除了冗余点,提取后的相邻两个必经转折点可能出现 3 种情况,如图 10 所示。其中圆点为当前必经转折点,菱形为下一个必经转折点,两必经转折点位置的 3 种情况具体表现为:

(1) 在 8 邻域内,对应图中的浅色方格;(2) 不在 8 邻域内,但在 15 邻域内,对应图中的深色方格;(3) 在 15 邻域外,对应图中的白色方格。

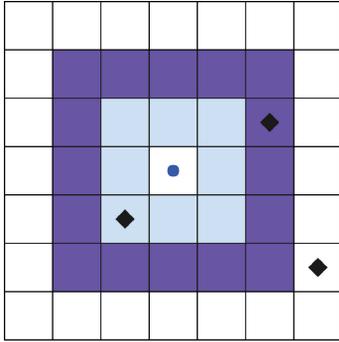


图 10 必经转折点的可能分布范围

Fig. 10 Possible distribution range of necessary turning points

由于两必经转折点之间的位置情况复杂,拟采用不同的控制点选取方法,可能选取到的控制点如图 11 所示。其中 Q_0, Q_4, Q_{11}, Q_{15} 为提取到的必经转折点,如图 11 中圆点所示; Q_1, Q_7, Q_8, Q_{14} 为自适应斜率控制点计算方法求得的控制点,如图 11 中正方形所示; Q_2 为 Q_1 和 Q_4 的中点, Q_6 为 Q_4 和 Q_7 的中点, Q_9 为 Q_8 和 Q_{11} 的中点, Q_{13} 为 Q_{11} 和 Q_{14} 的中点,如图 11 中三角形所示; Q_3 为 Q_2 和 Q_4 的中点, Q_5 为 Q_4 和 Q_6 的中点, Q_{10} 为 Q_9 和 Q_{11} 的中点, Q_{12} 为 Q_{11} 和 Q_{13} 的中点,如图 11 中五边形所示。

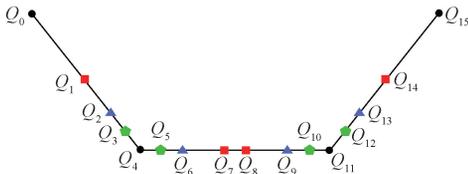


图 11 可能选取的控制点位置分布

Fig. 11 Possible distribution of control point positions to be selected

自适应斜率控制点计算方法基于两点斜率来进行控制点的位置计算,生成的贝塞尔曲线与原折线接近,但不与障碍物靠的太近。具体的做法是:输入 $A(x_a, y_a)$ 、 $B(x_b, y_b)$ 两点坐标,两点满足方程 $y = kx + b$ 。首先根据两点横坐标是否相等,判断 AB 线段是否为竖直线,若为竖直线,则直接根据式 (15) 和 (16) 得到输出 (x_{next}, y_{next}) ;若不为竖直线,则计算两点的斜率 k 和偏置 b ,根据斜率的大小,由式 (19) 分情况求得横坐标的增减量 $addx$,然后再根据式 (17) 和 (18) 得到输出 (x_{next}, y_{next}) 。式 (19) 中, $abs(k)$ 表示取斜率的绝对值,将其与 0.5 相比较来求 $addx$,该值的选取由大量的试验分析得出,能保证生成的路径不交叉。算法最后输出的 (x_{next}, y_{next}) 即为所求自适应斜率控制点的坐标。

$$x_{next} = x_a \tag{15}$$

$$y_{next} = \begin{cases} y_a - 1, & y_a > y_b \\ y_a + 1, & y_a < y_b \\ y_a, & y_a = y_b \end{cases} \tag{16}$$

$$x_{next} = \begin{cases} x_a - addx, & x_a > x_b \\ x_a + addx, & x_a \leq x_b \end{cases} \tag{17}$$

$$y_{next} = kx_{next} + b \tag{18}$$

$$addx = \begin{cases} 1, & abs(k) < 0.5 \\ 1/k, & abs(k) \geq 0.5 \end{cases} \tag{19}$$

求解 Q_1 时将 Q_0 视作 A 点, Q_4 视作 B 点;求解 Q_7 时将 Q_4 视作 A 点, Q_{11} 视作 B 点;求解 Q_8 时将 Q_{11} 视作 A 点, Q_4 视作 B 点;求解 Q_{14} 时将 Q_{11} 视作 A 点, Q_{15} 视作 B 点。

得到选取的控制点位置坐标后,采用如图 12 所示的方法,选择控制点。2.2 节求得的必经转折点为: $p_0, p_1, p_2, \dots, p_v, \dots, p_V, p_{V+1}$, 其中 $p_0 = n_s$ 为起点, $p_{V+1} = n_g$ 为终点, p_v 为必经转折点, $v = 1, 2, \dots, V$ 。遍历必经转折点 p_v , 对于每一个 p_v , 将其视作图 11 中的 Q_4, p_{v-1} 视作 Q_0 , p_{v+1} 视作 Q_{15} , 若 $v = V$, 则不存在 Q_{15} , 否则将 p_{v+2} 视作 Q_{15} 。最后保存生成的点集,按照顺序依次相接,再加入起点终点,即为路径平滑阶段生成的路径点。

3 算法仿真与分析

使用 Python 语言来进行算法的仿真与分析,版本为 3.10.10,电脑处理器型号为 Intel (R) Core (TM) i5-8250U CPU@ 1.60 GHz 1.80 GHz,内存为 20.0 GB。环境建模使用栅格地图,栅格的大小设置为 1,启发函数使用欧几里得距离,机器人半径大小为 0.3,安全半径为 0.4,贝塞尔曲线的阶数根据图 12 中的情况为四阶或五阶,点数为 11。栅格地图中圆点所在的方格为起点,五边形所在的方格为终点,深色方格为障碍物,障碍物的位置和大小是已知的,且不随时间的变化而改变。

在本章的所有仿真中,由于算法运行时间与电脑 CPU 频率、内存占用比例等有联系,不同时刻运行同一算法所耗费的时间会产生波动,因此取 10 次运行程序的平均值视作算法的运行时间。路程这一指标的单位为栅格的大小。

3.1 仿真 1: 各个改进的逐步仿真与分析

使用文献[26]中的栅格地图进行逐步仿真和分析,该地图大小为 20(宽)×20(高),障碍物密集程度适中,适用于查看每一步仿真的效果。详细结果数据如表 2 所示;对生成的路径绘制曲线如图 13 所示,其中起点坐标为 (1,1),终点坐标为 (18,19),粗虚线、粗实线、细实线、细虚线分别为“ A^* 算法”、“优化邻域处理”、“优化邻域+提取必经转折点处理”、“优化邻域+提取必经转折点+路

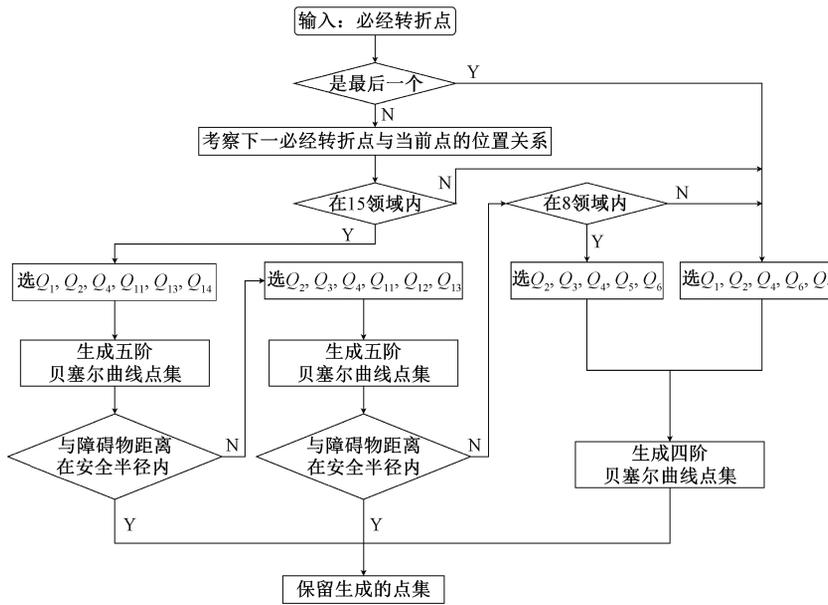


图 12 控制点的选取流程

Fig. 12 Flow chart for selecting control points

表 2 逐步仿真结果

Table 2 Gradual simulation results

| 序号 | 算法 | 运行时间/ s | 路径长度 (/栅格大小) | 安全路段 占比/% | 安全性 提升量/% | 转折点 个数 | 转折点个数 变化量/% | 平滑性 |
|------------------|-------------------------|------------|-----------------|--------------|--------------|-----------|----------------|-----|
| 1 ⁽¹⁾ | A* 算法 | 0.029 5 | 27.38 | 79.34 | | 10 | | 不平滑 |
| 2 ⁽²⁾ | 优化邻域处理 | 0.033 2 | 29.73 | 100.00 | 26.03 | 12 | 20.00 | 平滑 |
| 3 ⁽³⁾ | 优化邻域+提取必经转折点 | 0.043 5 | 28.27 | 100.00 | 0.00 | 5 | -58.33 | 平滑 |
| 4 ⁽⁴⁾ | 优化邻域+提取必经转折点+路径平滑(本文算法) | 0.045 2 | 27.64 | 100.00 | 0.00 | 5 | 0.00 | 平滑 |

注:比较关系为:(2)的比较对象为(1),(3)的比较对象为(2),(4)的比较对象为(3)。

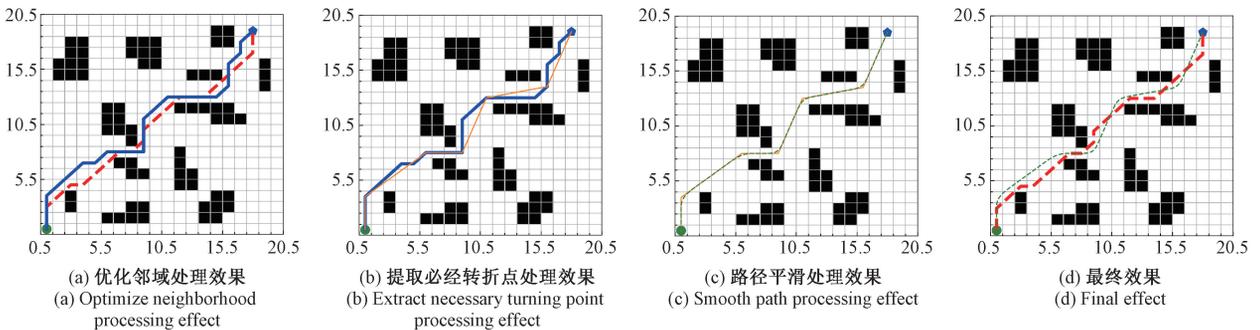


图 13 逐项处理生成路径对比

Fig. 13 Comparison of generation paths through item by item processing

径平滑处理”生成的路径。

图 13(a)将原 A* 算法与优化邻域处理后生成的路径对比,优化邻域后生成的路径不再出现斜穿障碍物顶点的情况,安全路段占比 100.00%,较原 A* 算法提升了 20.66%,但代价是生成的路径比之前略长,转折点个数由 10 个变成了 12 个,这是由于优化邻域这一处理在选

择当前节点的邻域时排除了强迫邻居节点,将原本斜穿障碍物的斜线变成了需要拐弯的两段直线,因此新生成的路径比原来更长,转折点个数更多。

图 13(b)对优化邻域后生成的路径点进行了必经转折点的提取,提取必经转折点后的路径长度由原来的 29.73 减少到了 28.27,转折点的个数由 12 个减少到了 5

个,减少了 58.33%。这是由于提取必经转折点这一处理在保证生成路径安全的前提下,减少了原路径中的冗余点,使转折点的个数减少且路径长度缩短。

图 13(c)对提取的转折点进行了路径平滑,路径长度变短了,这是由于路径平滑这一处理在保证平滑后曲线与障碍物保持一定距离的前提下,对必经转折点进行了分段平滑,把原来转向角不连续的折线变成转向角连续的曲线,在某些情况下会使路径长度缩短。

图 13(d)将原 A*算法与改进的 A*算法生成的路径在地图上进行了对比,改进的 A*算法生成的路径较原 A*算法略长,转折点的个数由 10 个变为了 5 个,减少了 50%,而路径与障碍物均保持一定的距离,安全性大大提升,并在转角处保持了平滑。

图 14 将原 A*算法与改进的 A*算法生成路径的转向角根据时间变化绘制了一段预估曲线。为简化计算,设置机器人行驶一个栅格使用的时间为 0.3 s,全程线速度恒定,忽略跟踪两种算法生成路径时产生的误差;跟踪 A*算法生成的路径时,忽略在转角处转向所花费的时间;跟踪改进 A*算法生成的路径时,曲线部分进行角速度恒定的圆周运动,不同曲线段的角速度随路径点的曲率和时间而变化,如图 15 所示。

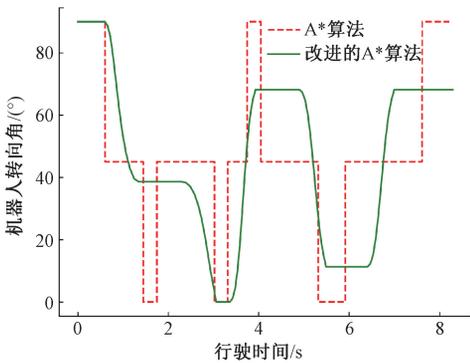


图 14 A*算法与改进的 A*算法生成路径转向角随时间变化的预估曲线

Fig. 14 A* algorithm and improved A* algorithm generate an estimated curve of path turning angle over time

观察图 14 中曲线可知,原 A*算法的转向角在转角处存在突变,而改进的 A*算法全程转向角不存在突变,这是由于进行了路径平滑的处理;观察图 15 中曲线可知,改进的 A*算法生成的路径曲率连续,具有平滑性。

综合上述仿真结果,与原 A*算法相比,优化邻域后生成路径在安全性能方面大大提升,虽然优化邻域这一处理导致路径长度和转折点个数增加了,但经过提取必经转折点和路径平滑这两项处理,改进的 A*算法生成的路径长度缩短至与原 A*算法十分接近,且转折点的个数大大减少,路径曲率连续,转向角不存在突变,同时保证

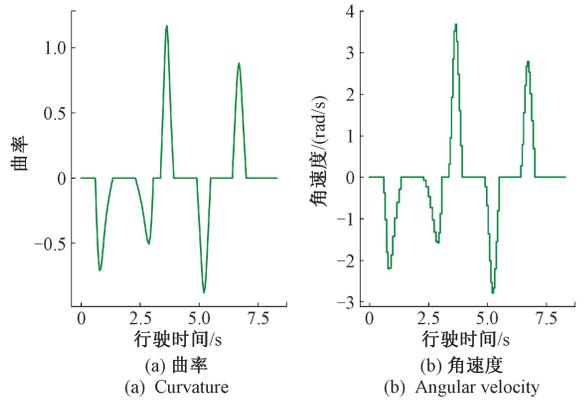


图 15 改进的 A*算法路径点曲率与角速度随时间的变化曲线

Fig. 15 The time varying curve of path point curvature and angular velocity in improved A* algorithm

了路径的安全性。

为进一步验证改进算法的有效性,选取文献[31]中同等大小,但障碍物更为密集的栅格地图进行逐步仿真,效果如图 16 所示。观察可知,改进 A*算法生成的路径能够自主避开障碍物多的区域,如图 16(a)所示;算法能删除路径中的冗余点,如图 16(b)所示;有效处理两必经转折点过近的情况并进行平滑,如图 16(c)所示;最终能够适应障碍物更为密集的场景,如图 16(d)所示。

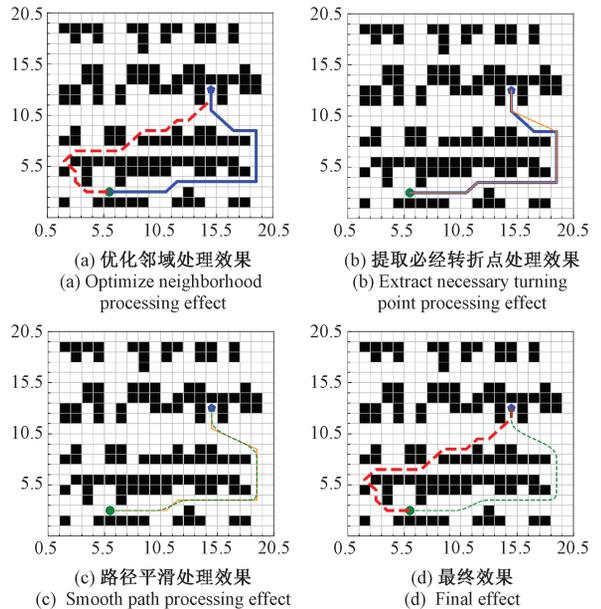


图 16 逐项处理生成路径对比(障碍物更密集)

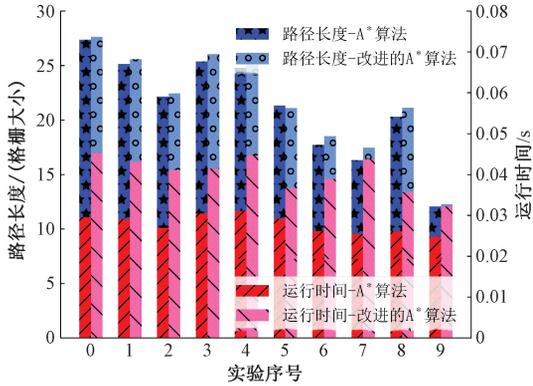
Fig. 16 Comparison of generation paths through item by item processing (more dense obstacles)

A*算法与改进的 A*算法的运行时间分别为 0.034 7 和 0.047 2 s,路程分别为 22.31 和 23.96,转折点个数分

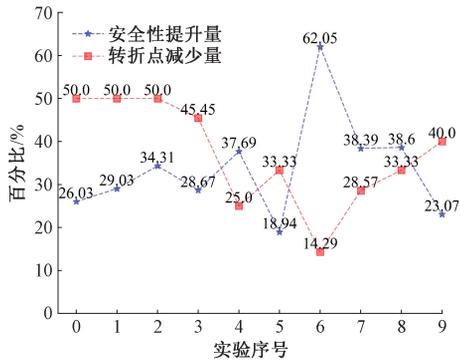
别为 11 和 5 个,安全路段占比分别为 64.51% 和 100.00%。与 A* 算法相比,改进的 A* 算法运行时间及生成的路径长度略长,但转折点个数减少了 54.55%,安全性提升了 55.02%,生成的路径平滑。

3.2 仿真 2:选取不同起点终点进行仿真与结果分析

仍选用文献[26]中的栅格地图,但选取 10 处不同的起点终点,进行 A* 算法和改进的 A* 算法的仿真,两种算法生成路径的对比数据曲线如图 17 所示。



(a) 路径长度和运行时间
(a) Path length and runtime



(b) 安全性提升量与转折点减少量
(b) Security improvement and turning point reduction

图 17 两种算法在选取 10 处起点终点实验中数据曲线
Fig. 17 Two algorithms for selecting data curves at 10 starting and ending points in the experiment

改进的 A* 算法与原 A* 算法相比,运行时间均在 0.05 s 内,路径长度在绝大多数情况下增加了,平均增加 2.86%,但安全性平均提升了 33.68%,转折点个数也平均减少了 37.00%,同时具有平滑性。

3.3 仿真 3:模拟室内环境的仿真与结果分析

使用教学楼内的平面地图,粗略绘制了一张室内环境的栅格地图,大小为 50(宽)×40(高),起点为(49,1),终点为(1,39),在栅格地图上进行了 A* 算法和改进的 A* 算法的仿真(分别用粗虚线和细虚线表示),效果如图 18 所示。

两种算法的运行时间分别为 0.268 1 和 0.298 3 s,



图 18 两种算法在室内地图环境中生成的路径对比
Fig. 18 Comparison of paths generated by two algorithms in indoor map environments

路程分别为 74.87 和 72.21,转折点个数分别为 10 个和 3 个,安全路段占比分别为 95.47% 和 100.00%。与 A* 算法相比,改进的 A* 算法运行时间略长,但路程更短,转折点个数减少了 70.00%,同时兼具安全性和平滑性。

本节仿真较 3.2 节中的算法运行时间增加了一个数量级,这是由于本节使用的栅格地图规模更大,算法需要更多的时间来搜索周围节点,并且需要更多的时间来处理数据。

改进的 A* 算法生成的路径的长度并不总是大于 A* 算法,考虑其中的改进:优化邻域这一处理使路径长度增加,但提取必经转折点这一处理将缩短路径,而路径平滑这一处理可能使路径长度增加也可能减小。综合上述处理,改进的 A* 算法最终生成的路径长度取决于 3 项处理对路径长度变化的叠加影响。综合本章前 3 节所有仿真数据,改进的 A* 算法最终生成的路径长度在大多数情况下比 A* 算法略长,但增加程度不超过 8%,在某些情况下可以十分接近甚至比 A* 算法要小。

3.4 仿真 4:与其他学者提出的算法比较分析

使用文献[23]中的栅格地图,大小为 50(宽)×50(高),起点为(5,46),终点为(45,6),进行了 A* 算法、weighted A* 算法^[32]、JPS 算法^[33]、文献[23]中的改进算法和本文算法共 5 种算法的仿真,绘制曲线效果如图 19 所示,详细结果数据如表 3 所示。

本文算法与 A* 算法相比,运行时间略有增加,生成的路径长度减少 3.84%,安全性提升 10.17%,转折点个数减少 62.50%,路径变平滑,总转弯角减少 78.44%,预估行驶时间减少 3.84%。

weighted A* 算法和 JPS 算法生成的路径大致相同,在路径长度、平滑性和预估行驶时间这 3 项指标上与 A* 算法一致,安全性能提升,转折点数减少,总转弯角变小,

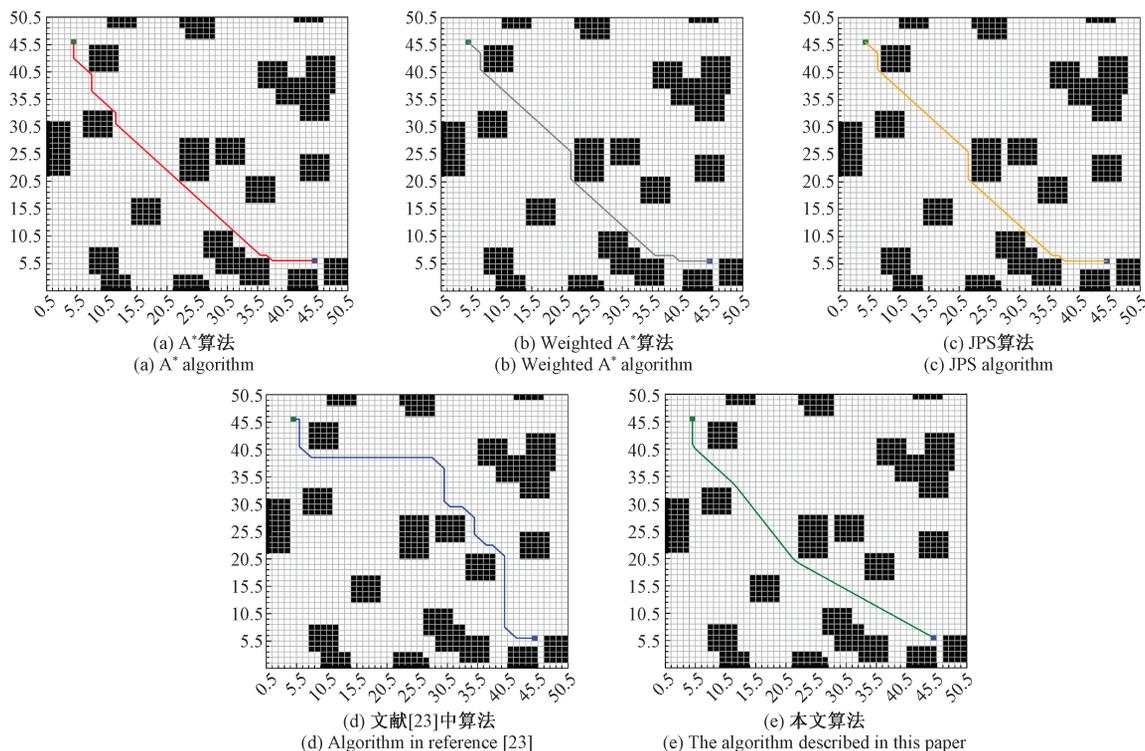


图 19 5 种算法生成的路径对比

Fig. 19 Comparison of paths generated by five algorithms

表 3 5 种算法仿真结果

Table 3 Simulation results of five algorithms

| 序号 | 指标 | A* 算法 | weighted A* 算法 | JPS 算法 | 文献[23]中算法 | 本文算法 |
|----|-------------|----------|----------------|----------|-----------|----------|
| 1 | 运行时间/s | 0.669 8 | 0.654 5 | 0.008 6 | 9.747 0 | 0.713 9 |
| 2 | 路径长度(/栅格大小) | 61.25 | 61.25 | 61.25 | 72.38 | 58.90 |
| 3 | 安全路段占比/% | 90.77 | 94.46 | 92.61 | 100.00 | 100.00 |
| 4 | 转折点个数 | 8 | 7 | 7 | 15 | 3 |
| 5 | 平滑性 | 不平滑 | 不平滑 | 不平滑 | 平滑 | 平滑 |
| 6 | 总转弯角/(°) | 360 | 315 | 315 | 675 | 77.6 |
| 7 | 预估行驶时间/s | 18.376 5 | 18.376 5 | 18.376 5 | 21.715 4 | 17.670 7 |

但生成的路径仍不能保证全路段安全和平滑,且转折点个数减少的幅度小。本文算法虽在运行时间上长于上述两种算法,但在其他指标上均更优,生成的路径绝对安全且平滑,转折点个数也大大减少。

本文算法与文献[23]中的改进算法相比,各项指标均更优。文献[23]中设置安全距离,将障碍物周围的栅格也全部认为是障碍物,这种做法缩小了搜索范围,在图 19 中错失了最佳路径,生成的路径更长、转角更多;而本文算法仅在搜索领域中排除完全不可走的强迫邻居节点,在尽可能不减小搜索范围的同时也保证了路径的安全。文献[23]中采用路径平滑策略,将路径中的直角转弯变成 45°斜向转弯来去除冗余点,这种做法有一定的效果,但没有去除路径中全部的冗余点;而本文对路径点循

环遍历,在保证安全的情况下,一一排除冗余点,并在必经转折点处进行路径平滑,缩短了路径长度、减少了转角次数,并且在全路段中保持了安全和平滑。

总体而言,文献[23]中的改进算法虽保证了路径的绝对安全,但错失了最佳路径,并且对冗余点的去除不够充分,而本文提出的改进策略能够有效解决上述问题,在各方面的性能均更优。

4 结 论

本文针对 A* 算法不安全、转折点多、不平滑的不足,进行了 3 项改进,包括优化邻域、提取必经转折点和路径平滑,在改进的过程中均保证了路径的安全,并进行了 4

次仿真实验,首先详细分析了每一步改进的结果数据和可能的原因,验证算法的可行性;其次选取不同的起点终点进行多次仿真比较,验证算法的通用性;然后采用室内环境中的栅格地图进行了仿真,验证算法在室内环境中的适用性;最后将算法与其他文献提出的方法在同一地图上仿真并分析比较,验证算法的有效性。仿真结果表明,改进后算法生成的路径与原 A* 算法相比:安全性大大提升,平均提升了 33.68%;转折点个数更少,平均减少了 37.00%;路径更加平滑,曲率与转向角连续;时间略有增加,路径长度在大多数情况下也略有增加。总体而言,改进后的算法具有安全、转折点少、平滑的特点,可用于移动机器人室内全局路径规划。

参考文献

- [1] LIU H X, ZHANG Y H. ASL-DWA: An improved A* algorithm for indoor cleaning robots[J]. IEEE Access, 2022, 10: 99498-99515.
- [2] OHNEBERG C, STOBICH N, WARMBEIN A, et al. Assistive robotic systems in nursing care: A scoping review[J]. BMC Nurs, 2023, 22(1): 72.
- [3] HONG B, LIN Z X, CHEN X, et al. Development and application of key technologies for guide dog robot: A systematic literature review [J]. Robotics and Autonomous Systems, 2022: 154.
- [4] ZHANG H C, ZHUANG Q J, LI G. Robot path planning method based on indoor spacetime grid model [J]. Remote Sensing, 2022, 14(10): 2357.
- [5] JIANG L, ZHAO P C, DONG W, et al. An eight-direction scanning detection algorithm for the mapping robot pathfinding in unknown indoor environment [J]. Sensors, 2018, 18(12): 4254.
- [6] YASUDA S, KUMAGAI T, YOSHIDA H. Safe and efficient dynamic window approach for differential mobile robots with stochastic dynamics using deterministic sampling [J]. IEEE Robotics and Automation Letters, 2023, 8(5): 2614-2621.
- [7] WANG C Q, WANG J K, LI C M, et al. Safe and robust mobile robot navigation in uneven indoor environments [J]. Sensors, 2019, 19(13): 2993.
- [8] LI Q Q, XU Y Q, BU S Q, et al. Smart vehicle path planning based on modified PRM algorithm [J]. Sensors, 2022, 22(17): 6581.
- [9] AMMAR A, BENNACEUR H, CHÂARI I, et al. Relaxed dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments [J]. Soft Computing, 2015, 20(10): 4149-4171.
- [10] ZHANG W, XU G J, SONG Y, et al. An obstacle avoidance strategy for complex obstacles based on artificial potential field method [J]. Journal of Field Robotics, 2023, 40(5): 1231-1244.
- [11] 赵倩楠, 黄宜庆. 融合 A* 蚁群和动态窗口法的机器人路径规划 [J]. 电子测量与仪器学报, 2023, 37(2): 28-38.
- [12] ZHAO Q N, HUANG Y Q. Robot path planning combining A* ant colony and dynamic window method [J]. Journal of Electronic Measurement and Instrumentation, 2023, 37(2): 28-38.
- [12] 郭世凯, 孙鑫. 基于改进粒子群算法的移动机器人路径规划 [J]. 电子测量技术, 2019, 42(3): 54-58.
- [13] GUO SH K, SUN X. Mobile robot path planning based on improved particle swarm optimization algorithm [J]. Electronic Measurement Technology, 2019, 42(3): 54-58.
- [13] 孙浩涵. 动态环境下履带式无人平台路径规划研究 [D]. 哈尔滨: 哈尔滨工业大学, 2020.
- [14] SUN H H. Research on path planning of tracked unmanned platform in dynamic environment [D]. Harbin: Harbin Institute of Technology, 2020.
- [14] 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.
- [15] WAHAB M N A, NEFTI-MEZIANI S, ATYABI A. A comparative review on mobile robot path planning: Classical or metaheuristic methods? [J]. Annual Reviews in Control, 2020, 50: 233-252.
- [16] 李艳生, 万勇, 张毅, 等. 基于人工蜂群-自适应遗传算法的仓储机器人路径规划 [J]. 仪器仪表学报, 2022, 43(4): 282-290.
- [17] LI Y SH, WAN Y, ZHANG Y, et al. Path planning for storage robots based on artificial bee colony adaptive genetic algorithm [J]. Chinese Journal of Scientific Instrument, 2022, 43(4): 282-290.
- [17] PATLE B K, BABU L G, PANDEY A, et al. A review: On path planning strategies for navigation of mobile robot [J]. Defence Technology, 2019, 15(4): 582-606.
- [18] ULLAH Z, XU Z W, ZHANG L, et al. RL and ANN based modular path planning controller for resource-constrained robots in the indoor complex dynamic environment [J]. IEEE Access, 2018, 6: 74557-74568.
- [19] LIU L X, WANG X, YANG X, et al. Path planning techniques for mobile robots: Review and prospect [J]. Expert Systems with Applications, 2023, 120254.
- [20] 王园. 基于 3DVFH+ 和 DWA 算法的无人机轨迹规划研究 [D]. 成都: 电子科技大学, 2022.
- [21] WANG Y. Research on UAV trajectory planning based on 3DVFH+ and DWA algorithms [D]. Chengdu: University of Electronic Science and Technology of China, 2022.
- [22] YIN X, CAI P, ZHAO K W, et al. Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusion algorithms [J]. Sensors, 2023, 23(8): 4102.
- [22] ZHANG H X, TAO Y D, ZHU W L. Global path

- planning of unmanned surface vehicle based on improved A-star algorithm[J]. *Sensors*, 2023, 23(14): 6647.
- [23] WANG H W, LOU S J, JING J, et al. The EBS-A* algorithm; An improved A* algorithm for path planning[J]. *Plos One*, 2022, 17(2): e0263841.
- [24] LI Y B, WANG Z X, ZHANG S Y. Path planning of robots based on an improved A* algorithm[C]. 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2022,5: 826-831.
- [25] FU B, CHEN L, ZHOU Y T, et al. An improved A* algorithm for the industrial robot path planning with high success rate and short length [J]. *Robotics and Autonomous Systems*, 2018, 106: 26-37.
- [26] LI H B, ZHENG H, LIU G, et al. Research on mobile robot path planning based on IHBS-A* algorithm[C]. 2022 5th International Conference on Robotics, Control and Automation Engineering (RCAE), 2022: 195-200.
- [27] JIANG C C, WANG C Y, WANG M L. Research on path planning for mobile robots based on improved A* algorithm[C]. 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC), 2023,7: 723-727.
- [28] MIN J G, RUY W S, PARK C S. Faster pipe auto-routing using improved jump point search [J]. *International Journal of Naval Architecture and Ocean Engineering*, 2020, 12: 596-604.
- [29] LI F F, DU Y, JIA K J. Path planning and smoothing of mobile robot based on improved artificial fish swarm algorithm[J]. *Scientific Reports*, 2022, 12(1): 659.
- [30] SHENTU S Z, GONG Z, LIU X J, et al. Hybrid navigation system based autonomous positioning and path planning for mobile robots [J]. *Chinese Journal of Mechanical Engineering*, 2022, 35(1):1-13.
- [31] TANG G, TANG C, CLARAMUNT C, et al. Geometric A* algorithm; An improved A* algorithm for AGV path planning in a port environment[J]. *IEEE Access*, 2021, 9: 59196-59210.
- [32] LAI X, LI J, CHAMBERS J. Enhanced center constraint weighted A* algorithm for path planning of petrochemical inspection robot [J]. *Journal of Intelligent & Robotic Systems*, 2021, 102(4): 78.
- [33] HARABOR D D, GRASTIEN A. Online graph pruning for pathfinding on grid maps [C]. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011,2: 1114-1119.

作者简介



杨国, 1999 年于南京理工大学获得学士学位, 2002 年于南京理工大学获得硕士学位, 2007 年于南京理工大学获得博士学位, 现为南京理工大学副研究员, 主要研究方向为微波毫米波近程探测与通信技术。
E-mail: yangguo@njust.edu.cn

Yang Guo received his B. Sc. degree from Nanjing University of Science and Technology in 1999, M. Sc. degree from Nanjing University of Science and Technology in 2002 and Ph. D. degree from Nanjing University of Science and Technology in 2007, respectively. Now he is a research associate in Nanjing University of Science and Technology. His main research interests include microwave millimeter wave near-range detection and communication technology.



吴晓, 2022 年于江西财经大学获得学士学位, 现为南京理工大学硕士研究生, 主要研究方向为移动机器人路径规划。

E-mail: 452708910@qq.com

Wu Xiao received her B. Sc. degree from Jiangxi University of Finance and Economics in 2022. Now she is a M. Sc candidate at Nanjing University of Science and Technology. Her main research interest includes mobile robot path planning.



肖如奇 (通信作者), 2015 年于南京信息工程大学滨江学院获得学士学位, 现为南京理工大学在读博士研究生, 主要研究方向为天线谐振理论和无源滤波器。

E-mail: Xiaoruqi_ee@njust.edu.cn

Xiao Ruqi (Corresponding author) received his B. Sc. degree from Binjiang College of Nanjing University of Information Science and Technology in 2015. Now he is a Ph. D. candidate at Nanjing University of Science and Technology. His main research interests include the theory of resonant mode and passive filter.



张若愚, 2014 年于哈尔滨工业大学获得学士学位, 2019 年于哈尔滨工业大学获得博士学位, 现为南京理工大学副研究员, 主要研究方向为通信感知一体化和大规模 MIMO。

E-mail: ryzhang@njust.edu.cn

Zhang Ruoyu received his B. Sc. degree from the Harbin Institute of Technology in 2014 and Ph. D. degree from the Harbin Institute of Technology in 2019, respectively. Now he is a research associate in Nanjing University of Science and Technology. His main research interests include integrated sensing and communication and massive MIMO.



吴文, 现为南京理工大学二级教授, 主要研究方向为微波毫米波近程探测与卫星通信技术。

E-mail: wuwen@mail.njust.edu.cn

Wu Wen is a Level-2 professor in Nanjing University of Science and Technology. His main research interests include microwave millimeter wave near-range detection and satellite communication technology.