doi:10.6041/j.issn.1000-1298.2026.17.029

基于转折点优化算法的移动机器人路径规划研究

陈久朋[1,2]　赵泽仲[1]　伞红军[1,2]　张　帆[1,2]　龚梦莹[3]

(1.昆明理工大学机电工程学院,昆明 650500;2.云南省先进装备智能制造技术重点实验室,昆明 650500;
3.玉溪新天力农业装备制造有限公司,玉溪 652699)

摘要:为提升自主移动设备在复杂环境下的适应能力,解决传统 A* 算法路径规划时存在搜索时间长、转折点多和路径平滑度差等问题,提出基于 A* 的转折点优化算法 TP-A*。该算法首先在 A* 算法的基础上,采用对角线距离作为启发函数并完成动态加权,保证算法前期以搜索速度为主,后期以搜索最优路径为主,从而提升自主移动设备的路径规划效率。其次,为了提升自主移动设备的效率和稳定性,采用"两过程、两准则和一个判断机制"完成路径优化。在普通栅格、复杂栅格、狭窄通道、迷宫 4 种场景下比较了机器人在 A*、JPS、TP-A* 3 种算法中的运行情况,仿真结果表明:所提出的 TP-A* 算法在平均规划时间、路径长度、路径节点数以及转折点数等方面均优于其他算法。最后,基于 Turtlebot2 机器人平台完成了室内全局代价地图的创建,比较了 A* 算法和 TP-A* 算法在实际运行过程中的应用效果,实验结果表明:所提出的 TP-A* 算法相比于传统的 A* 算法不仅路径更为平滑,且在寻路时间和路径长度上分别减少了 88.53% 和 21.99%,验证了所提算法的有效性。

关键词:TP-A* 算法;复杂环境;启发函数;路径规划;转折点优化算法

中图分类号:TP242;TH112　　**文献标识码**:A　　**文章编号**:1000-1298(2026)17-0312-12　　**OSID**:

Mobile Robot Path Planning Based on Turning Point Optimization Algorithm

Chen Jiupeng[1,2]　Zhao Zezhong[1]　San Hongjun[1,2]　Zhang Fan[1,2]　Gong Mengying[3]

(1. *Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming* 650500, *China*
2. *Key Laboratory of Advanced Equipment Intelligent Manufacturing Technology of Yunnan Province, Kunming* 650500, *China*
3. *Yuxi Xintianli Agricultural Equipment Manufacturing Co., Ltd., Yuxi* 652699, *China*)

Abstract: In order to improve the adaptability of autonomous mobile devices in complex environments and solve the problems of long search time, multiple turning points and poor path smoothness existing in traditional A* algorithm path planning, a turning point optimization algorithm TP-A* based on A* was proposed. Based on A* algorithm, diagonal distance was used as heuristic function and dynamic weighting was completed, which ensured that the algorithm focused on searching speed in the early stage and searching optimal path in the late stage, thus improving the path planning efficiency of autonomous mobile devices. Secondly, in order to improve the efficiency and stability of autonomous mobile devices, path optimization was completed by using "two processes, two criteria and one judgment mechanism". The performances of A*, JPS and TP-A* algorithms were compared in four scenarios: ordinary grid, complex grid, narrow passage and maze. The simulation results showed that the proposed TP-A* algorithm was superior to other algorithms in average planning time, path length, number of path nodes and turning points. Finally, the indoor global cost map was created based on Turtlebot2 robot platform, and the application effect of A* algorithm and TP-A* algorithm in the actual operation process was compared. Experimental results showed that the proposed TP-A* algorithm not only had smoother path than the traditional A* algorithm, but also reduced the path-finding time and path length by 88.53% and 21.99%, respectively, which verified the effectiveness of the proposed algorithm.

Key words: TP-A* algorithm; complex environment; heuristic function; path planning; turning point optimization algorithm

收稿日期:2025-05-22　修回日期:2025-06-04

基金项目:云南省重大科技专项(202502AC080001)、国家自然科学基金项目(52565057)和云南省"彩云博士后"创新项目

作者简介:陈久朋(1993—),男,副教授,博士,主要从事机器人技术研究,E-mail:18314490225@163.com

通信作者:伞红军(1976—),男,副教授,博士,主要从事机器人技术研究,E-mail:sanhjun@163.com

0　引言

随着科技的不断进步，移动机器人在工业、物流、医疗等多个领域得到广泛应用。导航技术对于机器人的自主运行至关重要，其大致可分为 3 个模块，分别是信息感知、路径规划和运动控制[1]。其中，信息感知采用定位与建图技术，让机器人在陌生环境中实现自主定位和地图构建。路径规划作为移动机器人导航系统的关键组成部分，旨在规划出一条最优且无碰撞的路径。此外，运动控制是保证机器人稳定运行的关键[2]。按照机器人对周围环境信息的感知程度，路径规划可分为全局路径规划和局部路径规划[3-4]。全局路径规划是指机器人在了解地图环境的情况下，根据预先设定的目标点位置，规划出一条到达目标点的安全路径[5-6]。局部路径规划是机器人在按照全局路径运动的过程中，根据传感器实时反馈的信息，进行局部避障。

近年来，随着计算能力的提升和人工智能技术的突破，路径规划正向智能化、自适应化方向演进。尽管现代智能算法快速发展，但传统算法仍在特定场景中展现出不可替代的价值。Dijkstra、A* 和 D* 算法作为经典的非智能规划方法，奠定了机器人路径规划的理论基础，这类算法依赖于对周围环境的了解，因此在应用时需要先对环境进行精确建模。Dijkstra 算法以起始点为中心向外层扩展，直至扩展到终点为止，得到最短路径[7]。针对传统的 Dijkstra 算法路径规划速度慢、效率低等问题，文献[8-9]通过优化代价函数，以时间为优化目标求解最优路径，在缩短规划时间的基础上规划出最短路径。

A*（A-Star）算法是一种启发式算法，该算法在 Dijkstra 算法的基础上引入了启发函数，通过平衡实际代价和启发式估计代价，高效地搜索最优路径[10]。尽管 A* 算法在大多数场景下都具有较高的鲁棒性，但随着场景的不断变大，其计算量大、规划效率低等问题便逐渐凸显出来。针对以上问题，鲁毅等[11]将最佳优先搜索算法作为启发式算法，给启发函数增加权重，改进启发函数，提升搜索路径的能力，然后使用双向搜索策略提高搜索的精度和效率。赵晓等[12]在 A* 算法的基础上，结合跳点搜索算法，用跳点代替最低代价节点，大大减少了计算量，提升了算法搜索效率。此外，针对 A* 算法所规划出来的路径转折点和冗余节点过多的问题[13]，Liang 等[14]通过向量叉积判断三点共线性，去除中间冗余节点，接着结合碰撞检测来生成平滑轨迹，最后通过贝塞尔曲线优化，提升路径平滑度。Sun 等[15]通过跳过冗余节点，筛选出关键的路径节点，然后使用三次 B 样条曲线对路径的平滑度进行优化。迟旭等[16]提出冗余点删除策略，对冗余节点进行剔除，只保留必要的转折点，有效减小了路径长度。姚得鑫等[17]通过动态检测节点间可达性，在保证安全性的前提下，对路径进行优化，显著提升路径的简洁性和平滑性。

D* 算法由 A* 算法发展而来，最初用于火星探测器的动态避障，与 A* 算法不同的是，D* 算法是由终点向起点开始搜索。针对 D* 算法存在搜索效率低、拐点多、路径不平滑等问题，Yu 等[18]采用逆距离加权（IDW）插值方法缩短路径长度，并引入 Dubins 算法对路径进行平滑处理。秦旭等[19]优化了子节点的选取方式，改进代价估计函数，引入平滑度函数，减少了路径拐点数，提升了规划效率和机器人运动过程中的安全性。

基于上述研究可知，Dijkstra 虽能保证全局最优，但其计算效率低，仅适用于小场景。A* 算法通过引入启发函数显著提升效率，但其路径常因冗余节点和转折点过多导致路径长度和复杂度增加，平滑性不足。D* 算法虽支持动态环境，但依赖反向搜索和重复计算，实时性较差。针对以上问题，本文提出 TP-A* 算法，该算法在 A* 算法的基础上对启发函数进行改进，使用对角线距离作为启发函数，并且对启发函数进行动态加权，使其前期以搜索速度为主，后期以搜索最优路径为主，提升路径规划的速度。然后使用转折点优化算法（Turning point optimization algorithm，TP）对路径进行优化，删除多余转折点和冗余路径点，减小路径长度，使最终规划出来的路径更短、更平滑。

1　TP-A* 算法

1.1　环境描述

在进行移动机器人全局路径规划时，需要通过传感器来感知周围环境信息，进行环境地图的构建。栅格法是目前移动机器人路径规划中最常用的方法，它将环境划分为一个由均匀格网组成的网格地图。栅格法简单、直观，特别适用于静态场景下的路径规划，因此，本文采用栅格法对地图环境进行建模[20]。如图 1 所示，在 5×5 的栅格地图中，网格序号按照从上到下、从左到右的顺序排列。其次，根据环境信息，可以将栅格分为两种状态，一种是空闲栅格，即图中白色区域，另一种是障碍物栅格，即图中黑色区域[21]。在栅格地图中可以将栅格索引转换为所在行坐标，表达式为

$$i=\begin{cases} W & (\mathrm{mod}(N,M)=0) \\ \mathrm{mod}(N,M) & (\mathrm{mod}(N,M)\neq 0) \end{cases} \tag{1}$$

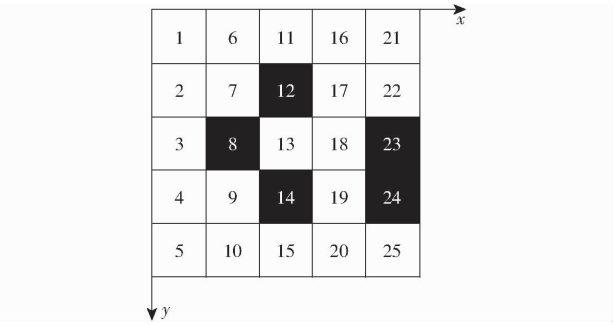


图 1 栅格地图

Fig. 1 Grid map

将栅格索引转换为所在列坐标,表达式为

$$j = \mathrm{ceil}\left(\frac{N}{M}\right) \tag{2}$$

将行列坐标转换为栅格索引,表达式为

$$N = W(j-1) + i \tag{3}$$

式中 mod( )——取余函数

ceil( )——向上取整函数

M——栅格总行数

N——当前栅格索引

i——栅格所在行的位置坐标

j——栅格所在列的位置坐标

W——栅格总列数

例如,对于 5×5 的栅格地图,当前栅格索引为 12 时,其坐标为(2,3);坐标为(4,5)的栅格,其索引为 24。

为实现机器人静态环境下的全局路径规划,对地图做出以下假设:①在机器人运动过程中,障碍物均为静态,且周围环境不发生改变。②斜向不存在障碍物时,机器人可沿斜向移动,如图 2 所示。

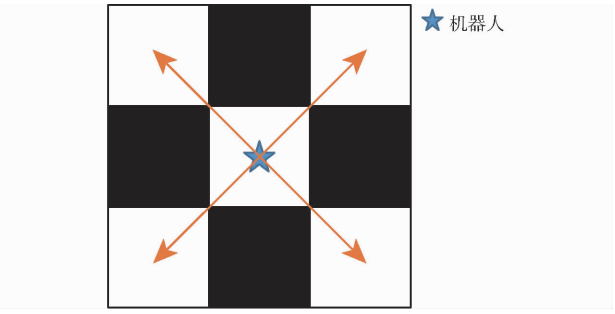


图 2 机器人运动方向

Fig. 2 Motion direction of robot

1.2 传统 A* 算法

A* 算法是一种启发式算法,它通过计算当前节点到目标点的估计代价,再结合实际路径代价来优先搜索最可能的路径。A* 算法的核心是代价函数 $f(n)$,表达式为

$$f(n) = g(n) + h(n) \tag{4}$$

式中 $g(n)$——起点到当前节点的实际代价,即路径累计长度

$f(n)$——当前节点的总代价

$h(n)$——当前节点到目标点的启发式估计代价[22]

$g(n)$使用欧几里得距离来表示,即

$$g(n) = \sqrt{(x_c - x_p)^2 + (y_c - y_p)^2} \tag{5}$$

其中,(x_c, y_c)为当前邻居节点坐标,(x_p, y_p)为当前父节点坐标。$h(n)$用曼哈顿距离表示,即

$$h(n) = |x_c - x_t| + |y_c - y_t| \tag{6}$$

式中 (x_t, y_t)——目标点坐标

A* 算法在寻路的过程中采用 8 邻域搜索方式,即从起始点开始,向周围的 8 个邻居节点扩展。在寻路过程中,需要判断邻居节点的位置是否在地图范围内,是否是障碍物节点。如果该节点为自由节点,则会被放入到开放列表(OpenList)中,并计算其 $f(n)$值,然后选择 OpenList 中 $f(n)$最小的节点作为下一步搜索的扩展节点,同时将该节点从 OpenList 中移除,并移入关闭列表(CloseList)中,循环执行这一过程,直到扩展至目标点为止。A* 算法的具体流程如下:

(1)初始化地图环境。

(2)初始化 OpenList 和 CloseList,将起点放入到 OpenList 中。

(3)判断 OpenList 是否为空,若 OpenList 为空,则寻路失败,若 OpenList 不为空,继续下一步骤。

(4)从 OpenList 中选出 $f(n)$最小的节点,将其设为父节点并移入 CloseList 中,并判断该节点是否是终点,若是终点,则寻路成功,生成最终路径,若不是终点,则执行下一步骤。

(5)依次判断父节点周围的 8 个邻居节点,若该邻居节点在地图范围内,不是障碍物节点,且不在 OpenList 和 CloseList 中,则计算该点 $f(n)$值,并将该邻居节点放入到 OpenList 中;若该邻居节点在 OpenList 中,且 $f(n)$值比 OpenList 中的小,则更新 OpenList 中该节点 $f(n)$值。

(6)重复上述步骤(3)~步骤(5),直到找到终点为止。

1.3 TP-A* 算法

针对传统 A* 算法在进行路径规划时存在搜索速度慢、效率低、最终规划路径长、转折点和冗余路径点较多等问题,本文对传统 A* 算法进行改进,提出 TP-A* 算法。该算法使用对角线距离作为启发函数并进行动态加权,从而提高搜索速度,其次,基于转折点对路径进行优化,进一步缩短路径、提升平滑度。其流程如图 3 所示。

1.3.1 改进启发函数

在 A* 算法中,启发函数是决定搜索速度的关

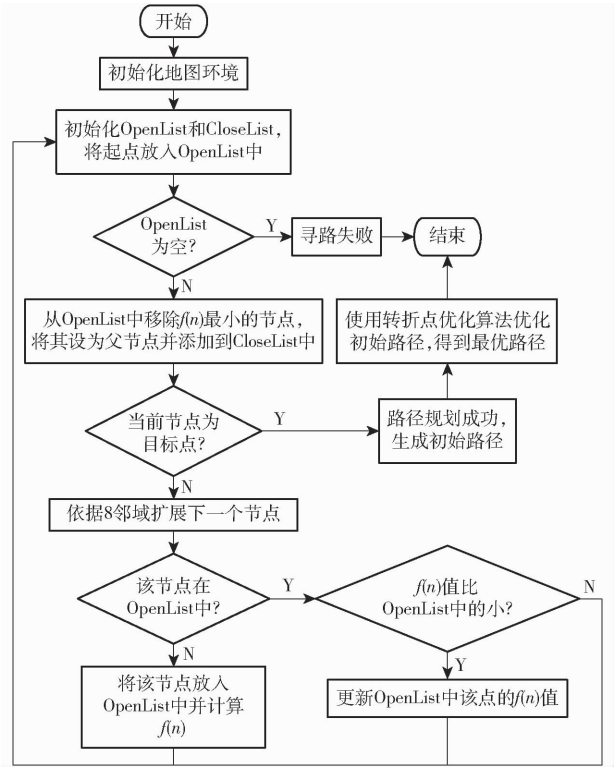


图 3　TP－A* 算法流程图

Fig. 3　Flowchart of TP－A* algorithm

键，好的启发函数能提高算法的性能和效率，所以可以通过改进启发函数来提升算法效率。理想情况下，启发函数的代价值应尽可能接近从当前节点到目标点的实际代价，这样能保证寻找到最佳路径，但因其是一个估计代价，所以不必完全准确。其次，启发函数在整个搜索过程中应该保持一致。在大多数情况下，机器人可以沿任意角度移动，因此可以使用欧几里得距离作为启发函数，即

$$h(n)=\sqrt{(x_c-x_t)^2+(y_c-y_t)^2} \tag{7}$$

本文中，机器人允许沿着斜对角方向移动，所以使用更接近实际代价的对角线距离作为算法的启发函数[23]，即

$$h(n)=\mathrm{d}x+\mathrm{d}y+(\sqrt{2}-2)\min(\mathrm{d}x,\mathrm{d}y) \tag{8}$$

其中　　$\mathrm{d}x=|x_c-x_t|\quad \mathrm{d}y=|y_c-y_t|$

式中，$\min(\mathrm{d}x,\mathrm{d}y)$ 表示取 $\mathrm{d}x$、$\mathrm{d}y$ 二者中的最小值。为了提高算法的搜索速度，需要对启发函数进行动态加权，即在 $h(n)$ 前增加一个权重系数 w[23]。w 的大小除了影响搜索速度之外，还决定能否找到一条最优的路径。按照实际需求，在开始搜索时，以搜索速度为主；随着搜索过程的推进，机器人逐渐接近目标点，此时以找到最优路径为主。综上所述，可以将改进后的代价函数表示为

$$f(n)=g(n)+wh(n) \tag{9}$$

其中　　$w=1+d/D$

式中　d——当前邻居节点到目标点的距离

D——起点到目标点的距离

d、D 均使用对角线距离来表示。

1.3.2　TP 算法

传统的 A* 算法规划出的路径冗余节点多，转折点多，导致最终的路径过长且不平滑。在现实环境中，这些因素会导致机器人工作效率低，运行不稳定，在转弯角度过大时，甚至会发生侧翻。为了提高移动机器人实际运行过程中的稳定性，缩短路径，本文提出了一种基于转折点的路径优化方法，即转折点优化算法（Turning point optimization algorithm，TP），算法流程如图 4 所示。

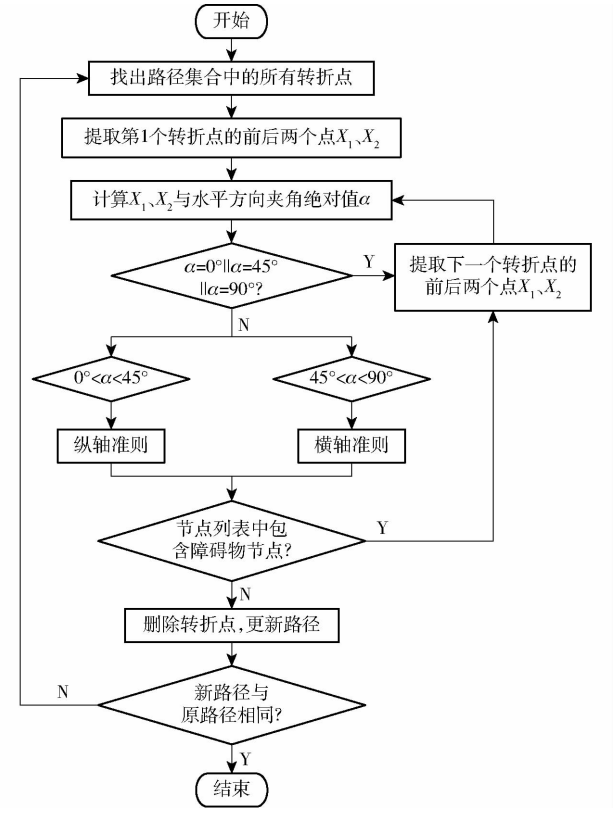


图 4　转折点优化算法流程图

Fig. 4　Flowchart of turning point optimization algorithm

转折点优化算法主要包括：两个过程、两种准则和一个判断机制。其中，两个过程分别是：提取转折点前后两个路径点和路径更新过程；两种准则分别是横轴准则和纵轴准则；一个判断机制是通过角度和坐标确定两个路径点之间的关系。算法的具体思路为：找出初始路径中所有转折点，判断转折点前后两点之间的最短路径（两点连线）是否穿过障碍物，根据判断结果对初始路径进行更新，实现对初始路径的平滑处理，得到一条路径节点更少、平滑度更优、距离更短的路径，提升机器人的效率和稳定性。

在提取转折点前后两个路径点之前，首先需要找出路径中的转折点。本文根据斜率来寻找路径中

的转折点。在初始路径集合 $P=\{P_1,P_2,\cdots,P_n\}$ 中，通过比较路径 $P_1\rightarrow P_2$ 的斜率 K_1 和路径 $P_2\rightarrow P_3$ 的斜率 K_2 来判断路径点 P_2 是不是转折点。如果 $K_1\neq K_2$，证明 3 个路径点不共线，P_2 是转折点；如果 $K_1=K_2$，证明 3 个路径点共线，P_2 不是转折点。以此类推，依次比较相邻两段路径的斜率，从而找出路径中的所有转折点。

提取第 1 个转折点的前后两个路径点 X_1、X_2，假设其坐标为 (x_1,y_1)、(x_2,y_2)，通过反正切函数计算 X_2 相对于 X_1 的角度绝对值 α，即

$$\alpha=\arctan\left(\frac{\mathrm{d}x}{\mathrm{d}y}\right) \tag{10}$$

其中 $\mathrm{d}x=|x_2-x_1|$ $\mathrm{d}y=|y_2-y_1|$

在确定转折点前后两路径点的角度时，有以下 3 种情况不需要进行路径优化：

(1) 如果 $\alpha=0°$，两路径点和转折点之间的关系如图 5a、5b 所示，此时，两点之间的最短路径穿过了障碍物，需要转折点来改变路径，实现避障。所以，转折点无法被优化舍弃。

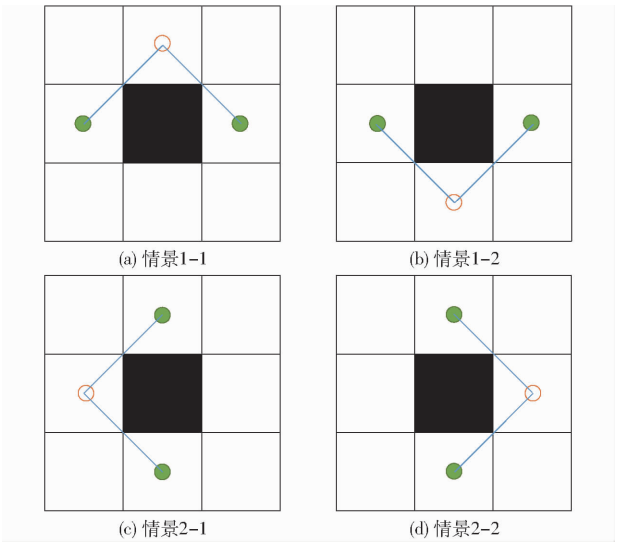


图 5 无需路径优化的情况示意图

Fig. 5 Schematics of cases without path optimization

(2) 如果 $\alpha=90°$，两路径点和转折点之间的关系如图 5c、5d 所示，两点之间的最短路径也穿过了障碍物，转折点无法被优化。

(3) 由于机器人允许斜向运动，当 $\alpha=45°$ 时，两路径点直接构成最短路径，不发生转折，而发生转折的原因是两路径点之间存在障碍物，所以，$\alpha=45°$ 也不需要进行路径优化。

角度判断分为两种情况，一种是 $0°<\alpha<45°$，另一种是 $45°<\alpha<90°$。根据夹角 45°将后一个路径点 X_2 所在区域划分为 8 块，0 为前一个路径点 X_1 的位置，如图 6 所示。两者结合即可判断出后一个路径点 X_2 所在的位置区域，如表 1 所示。

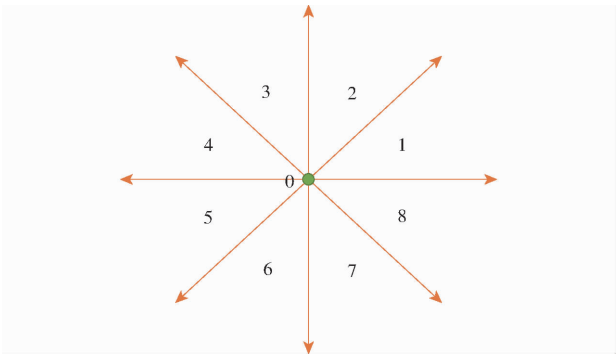


图 6 两路径点位置关系

Fig. 6 Spatial relationship of two path points

表 1 根据角度和坐标确定后一个路径点所在位置

Tab. 1 Calculation of next waypoint's position using angle and coordinates

α 角度范围/(°)	坐标条件	位置区域
(0,45)	$x_2>x_1,y_2>y_1$	区域 8
(0,45)	$x_2>x_1,y_2<y_1$	区域 5
(0,45)	$x_2<x_1,y_2>y_1$	区域 1
(0,45)	$x_2<x_1,y_2<y_1$	区域 4
(45,90)	$x_2>x_1,y_2>y_1$	区域 7
(45,90)	$x_2>x_1,y_2<y_1$	区域 6
(45,90)	$x_2<x_1,y_2>y_1$	区域 2
(45,90)	$x_2<x_1,y_2<y_1$	区域 3

判断转折点前后两路径点之间的最短路径是否穿过障碍物时，需要找出这条最短路径穿过的所有栅格，此处使用纵轴准则和横轴准则来进行寻找。由于 α 在 0°~45°之间路径与纵轴的交点更多，所以将之命名为纵轴准则，其原理是通过斜率确定路径与纵轴交点的位置，从而得到交点左右两侧栅格的位置坐标，同理，横轴准则则是因为 α 在 45°~90°之间路径与横轴的交点更多，其原理是通过斜率确定路径与横轴交点的位置，从而得到交点上下两侧栅格的位置坐标。综上所述，当后一个路径点位于 8、5、1、4 这 4 个区域时，如图 7 所示，图中红色栅格为转折点前后两路径点之间最短路径穿过的所有栅格，此时使用纵轴准则计算所有栅格坐标，纵轴准则判别计算方法如表 2 所示；当后一个路径点位于 7、6、2、3 这 4 个区域时，如图 8 所示，图中红色栅格为转折点前后两路径点之间最短路径穿过的所有栅格，此时使用横轴准则计算所有栅格坐标，横轴准则判别计算方法如表 3 所示。

通过上述过程，得到转折点前后两路径点之间最短路径穿过的所有栅格。然后，判断这些栅格中是否有障碍物栅格，如果存在一个及以上的障碍物栅格，表明这两路径点之间无法通行。

以图 9a 所示使用改进启发函数后的 A* 算法在 7×7 的栅格地图中规划出来的初始路径为例，使用

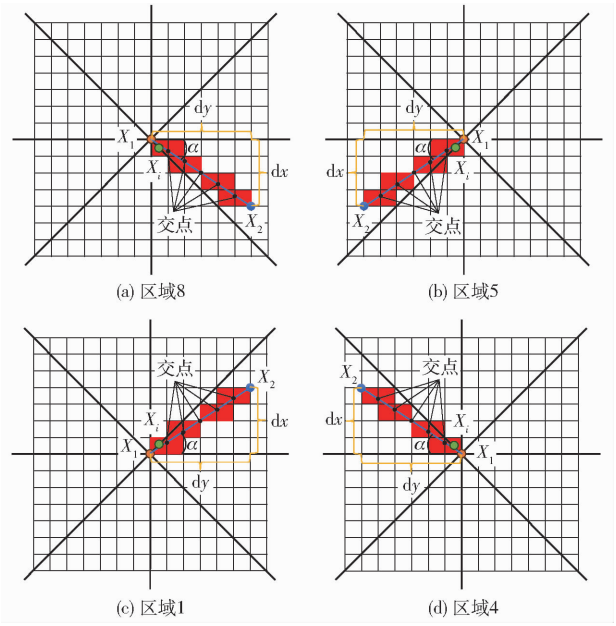


图 7　纵轴准则下路径穿过的栅格示意图

Fig. 7　Grid diagrams of path crossing under vertical axis criterion

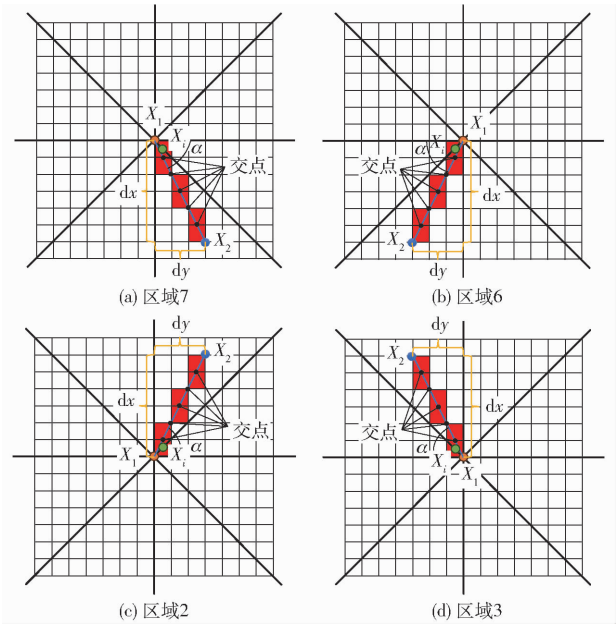


图 8　横轴准则下路径穿过的栅格示意图

Fig. 8　Grid diagrams of path crossing under horizontal axis criterion

表 2　纵轴准则表

Tab. 2　Vertical axis criteria

区域	判别条件	左右两侧栅格坐标
8	$\frac{x_1+q+1-x_i}{y_1+p+1-y_i}>k$	$(x_1+q,y_1+p),(x_1+q,y_1+p+1)$
	$\frac{x_1+q+1-x_i}{y_1+p+1-y_i}=k$	$(x_1+q,y_1+p),(x_1+q+1,y_1+p+1)$
5	$\frac{x_1+q+1-x_i}{y_i-(y_1-p)}>k$	$(x_1+q,y_1-p-1),(x_1+q,y_1-p)$
	$\frac{x_1+q+1-x_i}{y_i-(y_1-p)}=k$	$(x_1+q+1,y_1-p-1),(x_1+q,y_1-p)$
1	$\frac{x_i-(x_1-q)}{y_1+p+1-y_i}>k$	$(x_1-q,y_1+p),(x_1-q,y_1+p+1)$
	$\frac{x_i-(x_1-q)}{y_1+p+1-y_i}=k$	$(x_1-q,y_1+p),(x_1-q-1,y_1+p+1)$
4	$\frac{x_i-(x_1-q)}{y_i-(y_1-p)}>k$	$(x_1-q,y_1-p-1),(x_1-q,y_1-p)$
	$\frac{x_i-(x_1-q)}{y_i-(y_1-p)}=k$	$(x_1-q-1,y_1-p-1),(x_1-q,y_1-p)$

注：(x_i,y_i) 为前 1 个路径点的栅格中心坐标，$x_i=x_1+0.5$，$y_i=y_1+0.5$；$k=\mathrm{d}x/\mathrm{d}y$；$p$、$q$ 是迭代次数，范围分别是：$0\leqslant p<\mathrm{d}y$，$0\leqslant q\leqslant \mathrm{d}x$，$p$ 每进行一次迭代，都要对 q 进行迭代计算，直到找到满足判别条件的值为止。

表 3　横轴准则表

Tab. 3　Horizontal axis criteria

区域	判别条件	上下两侧栅格坐标
7	$\frac{x_1+p+1-x_i}{y_1+q+1-y_i}<k$	$(x_1+p,y_1+q),(x_1+p+1,y_1+q)$
	$\frac{x_1+p+1-x_i}{y_1+q+1-y_i}=k$	$(x_1+p,y_1+q),(x_1+p+1,y_1+q+1)$
6	$\frac{x_1+p+1-x_i}{y_i-(y_1-q)}<k$	$(x_1+p,y_1-q),(x_1+p+1,y_1-q)$
	$\frac{x_1+p+1-x_i}{y_i-(y_1-q)}=k$	$(x_1+p,y_1-q),(x_1+p+1,y_1-q-1)$
2	$\frac{x_i-(x_1-p)}{y_1+q+1-y_i}<k$	$(x_1-p-1,y_1+q),(x_1-p,y_1+q)$
	$\frac{x_i-(x_1-p)}{y_1+q+1-y_i}=k$	$(x_1-p-1,y_1+q+1),(x_1-p,y_1+q)$
3	$\frac{x_i-(x_1-p)}{y_i-(y_1-q)}<k$	$(x_1-p-1,y_1-q),(x_1-p,y_1-q)$
	$\frac{x_i-(x_1-p)}{y_i-(y_1-q)}=k$	$(x_1-p-1,y_1-q-1),(x_1-p,y_1-q)$

注：$(x_i,\ y_i)$ 为前 1 个路径点的栅格中心坐标，$x_i=x_1+0.5$，$y_i=y_1+0.5$；$k=\mathrm{d}x/\mathrm{d}y$；$p$、$q$ 是迭代次数，范围分别是：$0\leqslant p<\mathrm{d}x$，$0\leqslant q\leqslant \mathrm{d}y$，$p$ 每进行一次迭代，都要对 q 进行迭代计算，直到找到满足判别条件的值为止。

转折点优化算法对该路径进行优化。在该路径中，起始点的栅格坐标为 $S(7,1)$，终点栅格坐标为 $T(1,7)$。

(1)通过计算初始路径中相邻两个路径点之间的斜率，找到路径中的所有转折点：$P_1(6,2)$、$P_2(5,2)$、$P_5(2,5)$、$P_6(2,6)$。

(2)提取第 1 个转折点 P_1 的前后两个路径点 $S(7,1)$、$P_2(5,2)$，计算 P_2 相对于 S 的角度绝对值 $\alpha=63°$。因为 $45°<\alpha<90°$，且 $x_2<x_1$，$y_2>y_1$，所以，根据表 1 可得，P_2 位于区域 2，应使用横轴准则。

(3)根据表 3，S 的栅格中心点标为 $(7.5,1.5)$，$k=2$，p、q 的范围分别是：$0\leqslant p<2$，$0\leqslant q\leqslant 1$，计算得到路径 $S\rightarrow P_2$ 所经过的全部栅格为：$(6,1)$、$(7,1)$、$(5,2)$、$(6,2)$。

(4)由于 $S\rightarrow P_2$ 这条路径经过的所有栅格均为空闲栅格，所以，将转折点 P_1 从路径中删除，更新后的路径集合为 $\{S,P_2,P_3,P_4,P_5,P_6,T\}$。

(5)对后续转折点逐一进行上述优化，得到第 1

次优化后的新路径集合为{S,P_2,P_3,P_4,P_6,T}。

(6)新路径与初始路径不同,重复上述步骤(1)~(5)继续对新路径进行优化,直到生成的新路径与上一次优化后的路径相同为止,最优路径集合为{S,P_2,P_3,P_6,T}。

具体路径优化过程如图 9 所示。图 9b 为第 1 次使用转折点优化算法后得到的路径,图 9c 为第 2 次优化后的路径,此时路径已经无法继续优化,该路径即为 TP-A* 算法规划出的最终路径。

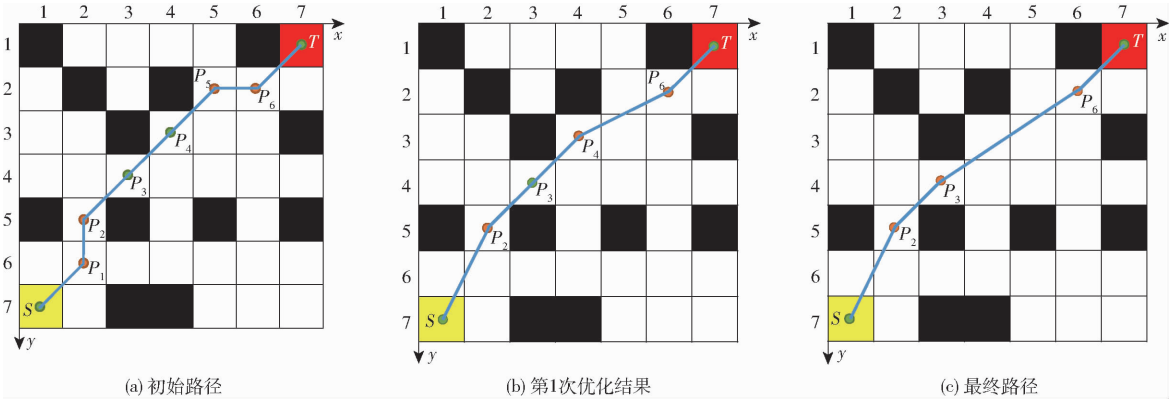


图 9 路径优化示意图

Fig. 9 Schematics of path optimization

2 仿真

为了验证 TP-A* 算法在不同地图环境中路径规划的有效性和可靠性,本文设计了 4 种地图,包括普通 20×20 的栅格地图、范围更大及障碍物更为复杂的 50×50 的栅格地图、狭窄通道地图和迷宫地图。在每一种类型的地图中分别对 A* 算法、JPS 算法和 TP-A* 算法进行 50 次重复仿真实验,并对实验结果取平均值。3 种算法的基础参数如表 4 所示。

表 4 3 种算法基础参数

Tab. 4 Basic parameters of three algorithms

算法	启发函数	权重系数 w	代价函数 $f(n)$	节点扩展方式	路径优化
A*	曼哈顿距离	1	$g(n)+h(n)$	8 邻域逐步扩展(逐步遍历相邻栅格)	无
JPS	曼哈顿距离	1	$g(n)+h(n)$	跳跃扩展(跳过非关键节点)	无
TP-A*	对角线距离	$1+d/D$	$g(n)+wh(n)$	8 邻域扩展+动态启发函数引导方向	转折点优化

为了全面评估算法的性能,本文选取平均规划时间、路径长度、路径节点数及转折点数 4 个指标来对实验结果进行评估分析。路径规划时间主要受启发函数和搜索过程中的扩展节点数影响,扩展节点数越多,搜索时间越长,但随着地图范围扩大,复杂度提高,规划所需时间也会变长。不同算法的搜索策略会直接影响最终的路径长度,其次,剔除路径中非必要的冗余路径点也可能缩短路径长度。路径节点数受算法节点扩展方式的影响,节点扩展方式不同,最终路径的节点数也不同,路径节点数的多少除了影响路径复杂度之外,还可能影响路径长短。转折点数反映路径平滑度,冗余转折点会增加路径长度,优化转折点可同时提升平滑度与缩短长度。

为确保实验可复现性,增强算法对比的公平性,突出 TP-A* 算法的合理性和优越性,本文对 4 种仿真实验场景的障碍物比例、障碍物类型和障碍物分布特性进行说明,如表 5 所示。

表 5 仿真场景参数

Tab. 5 Simulation scene parameters

地图类型	障碍物比例/%	障碍物类型	分布特性
20×20 普通栅格	33	大小不同的矩形障碍物	随机均匀分布,障碍物较为分散
50×50 复杂栅格	40	无固定形状和规则的障碍物	随机均匀分布,障碍物较为密集
狭窄通道	43	有明确边界的不规则障碍物	障碍物之间分布紧密,形成多个“瓶颈”区域
迷宫	50	连续的厚度不一的墙体	障碍物分布紧密,入口到出口之间至少存在一条可行路径

图 10~13 分别展示了 A* 算法、JPS 算法和 TP-A* 算法在 4 种不同的栅格地图中进行路径规划的结果,图中黄色栅格表示路径起点,紫色栅格表示目标点,蓝色线条即为规划出的路径。其中,图 10 为

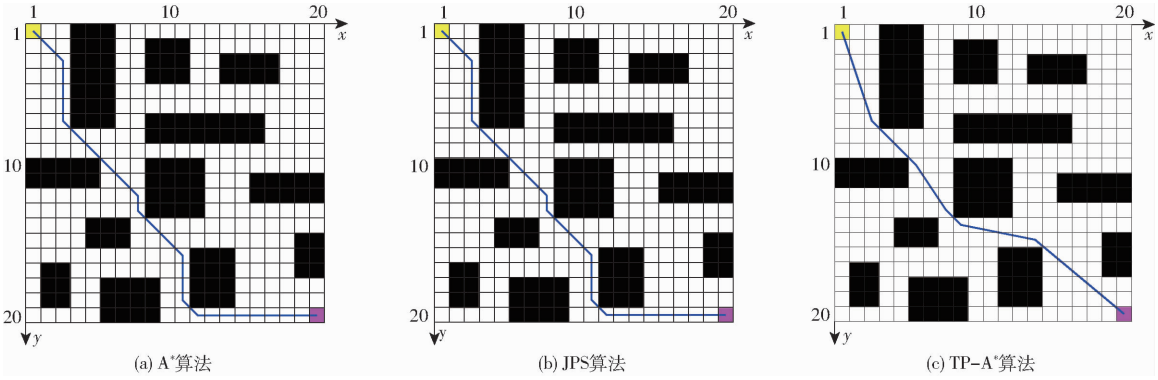


图 10　20 × 20 普通栅格地图规划结果

Fig. 10　20 × 20 regular grid map planning results

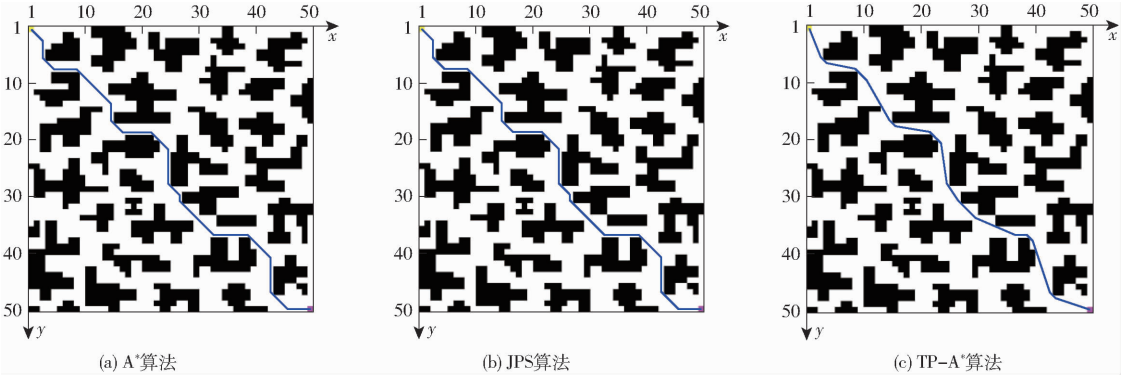


图 11　50 × 50 复杂栅格地图规划结果

Fig. 11　50 × 50 complex grid map planning results

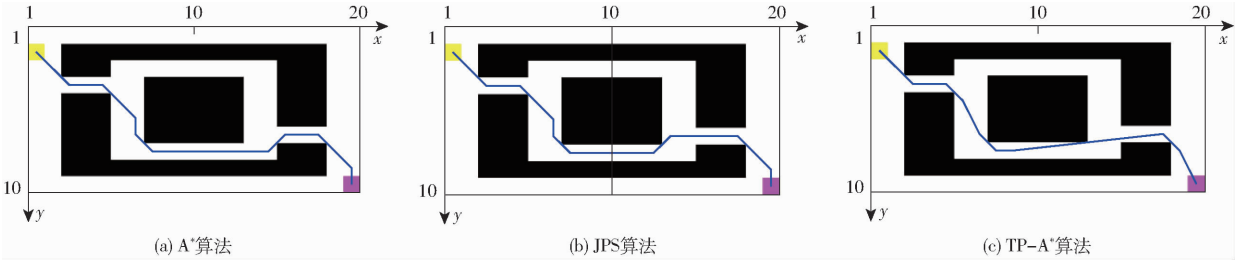


图 12　狭窄通道地图规划结果

Fig. 12　Narrow channel map planning results

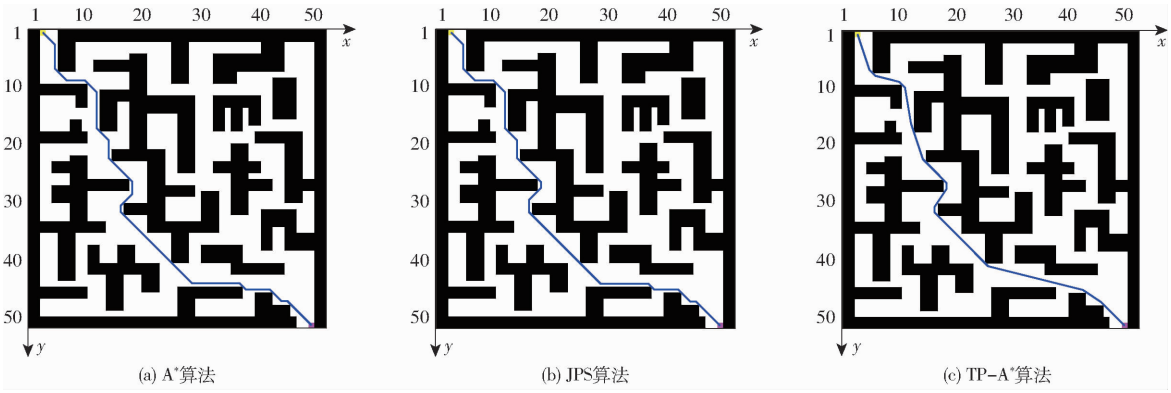


图 13　迷宫地图规划结果

Fig. 13　Maze map planning results

20 × 20 的普通栅格地图，图中所有障碍物均为方形规则障碍物；图 11 为范围较大的 50 × 50 复杂栅格地图，地图中的障碍物较多且不规则；图 12 为障碍物之间相对紧密的狭窄环境，此环境中，路径规划易受算法和机器人自身条件的限制；图 13 为迷宫地图，用来测试算法能否脱离局部最优，顺利找到可通

行路径。表 6 给出了 3 种算法的仿真结果,所有结果均为 50 次仿真实验数据的平均值。

表 6 不同地图中算法仿真结果对比

Tab. 6 Comparison of algorithm simulation results in different grid maps

地图类型	算法	平均规划时间/s	路径长度	路径节点数	转折点数
20×20 普通栅格	A*	0.198 6	31.556	28	7
	JPS	0.207 5	31.556	15	7
	TP-A*	0.187 6	28.496	9	5
50×50 复杂栅格	A*	0.347 5	80.426	69	17
	JPS	0.838 2	80.426	34	17
	TP-A*	0.323 3	76.289	23	16
狭窄通道	A*	0.195 3	24.314	22	9
	JPS	0.181 7	24.314	14	9
	TP-A*	0.170 7	23.598	12	7
迷宫	A*	0.339 5	78.669	66	18
	JPS	0.991 3	78.669	31	18
	TP-A*	0.317 2	75.429	29	13

从图 10~13 不同地图的路径规划结果可以看出,A* 算法和 JPS 算法的路径大部分都是相同的,原因是两者的核心(启发函数和权重系数)是一致的,均使用曼哈顿距离作为启发函数,并且两种算法均以搜索最优路径为目的,导致最终路径相同。由于 TP-A* 算法的启发函数和权重系数与 A*、JPS 都不相同,并且对路径进行了优化,所以得到的最终路径在长度和平滑度上均优于前二者。

通过表 6 的仿真结果,对 4 种地图环境下 TP-A* 算法的实验结果逐一做综合比较分析。在 20×20 的普通栅格地图中,与 A* 算法相比,TP-A* 算法在规划时间上减少 5.54%,路径长度减少 9.69%,路径节点数减少 67.86%,转折点数减少 28.57%;在 50×50 的复杂栅格地图中,TP-A* 算法的规划时间减少 6.96%,路径长度减少 5.14%,路径节点数减少 66.66%,转折点数减少 5.88%;在狭窄通道地图中,TP-A* 算法的规划时间减少 12.59%,路径长度减少 2.94%,路径节点数减少 45.45%,转折点数减少 22.22%;在迷宫地图中,TP-A* 算法的规划时间减少 6.57%,路径长度减少 4.12%,路径节点数减少 56.06%,转折点数减少 27.77%。

根据实验结果对比,分别对平均规划时间、路径长度、路径节点数和转折点数 4 个指标做了如下分析:从路径规划时间上看,A* 和 TP-A* 在不同地图中均能快速规划出最优路径,JPS 因其在复杂环境中需频繁执行强制邻域检测,单次跳点扩展的计算复杂度增加,使得路径规划所需的时间更长。从路径长度上看,TP-A* 通过转折点优化,使得其在 4 种地图环境中所规划出的最终路径都比 A*、JPS 短,JPS、A* 路径长度完全相同,因 JPS 仅优化了搜索策略,通过跳过非必要节点来加速搜索,但未改变路径生成逻辑,所以,二者路径长度一致。从路径节点数上看,A* 因其采用 8 邻域依次扩展的策略,导致路径冗余节点数较多,JPS 在搜索过程中跳过冗余节点,只搜索关键的跳点,路径节点数大幅减少,TP-A* 则通过剔除冗余路径点来减少路径节点数。从转折点数上看,TP-A* 显著减少了路径中的转折点,使得路径更平滑,JPS 转折点数与 A* 一致,均未优化转折点。通过对 A*、JPS 和 TP-A* 在 4 种地图环境(普通栅格、复杂栅格、狭窄通道、迷宫)中的仿真实验对比,TP-A* 算法在规划效率、路径长度和平滑度上相比于 A*、JPS 均展现出显著优势。

3 实验验证

3.1 路径规划实验

为验证本文算法的可行性,将 TP-A* 算法应用到 ROS(机器人操作系统)中,实验对象为 Turtlebot2 移动机器人,使用搭载 Ubuntu 18.04,melodic 版本的计算机作为上位机,采用 A2M7 激光雷达来获取周围的环境信息。通过两根数据线分别将移动机器人与雷达和上位机相连接,实现三者之间的通信。实验平台如图 14 所示。

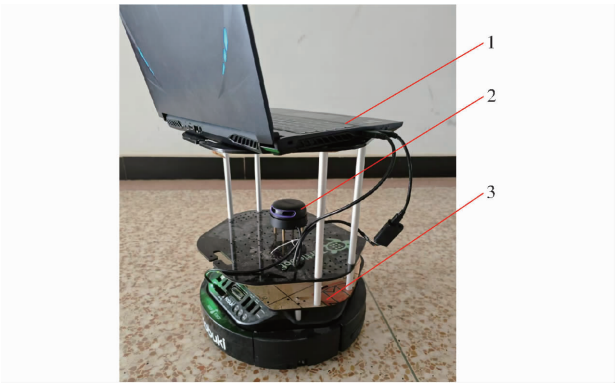


图 14 Turtlebot2 移动机器人平台

Fig. 14 Turtlebot2 mobile robot platform

1. 上位机 2. A2M7 激光雷达 3. Turtlebot2 移动机器人

实验主要分为建图、定位和路径规划 3 个步骤。本文使用 gmapping 算法来构建环境地图。首先启动移动机器人底盘和激光雷达,激光雷达通过发射激光束并接收反射信号,测量周围物体与机器人的距离,生成点云数据,使用键盘控制节点控制机器人移动,gmapping 可以根据移动机器人里程计数据和激光雷达数据来绘制包含障碍物信息的二维栅格地图。接下来,利用地图服务器中的 map_saver 节点保存地图,再利用 map_server 节点读取地图数据,将

之传入 move_base 模块中，作为本实验的全局代价地图。实验场景如图 15 所示，对应的建图结果如图 16 所示。



图 15　实验场景

Fig. 15　Experimental scenario

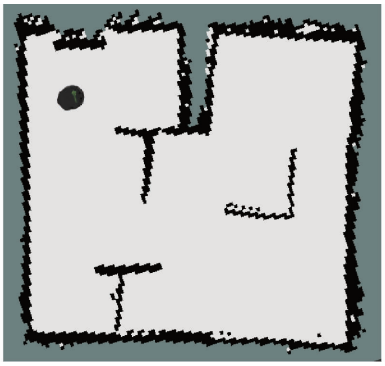


图 16　建图结果

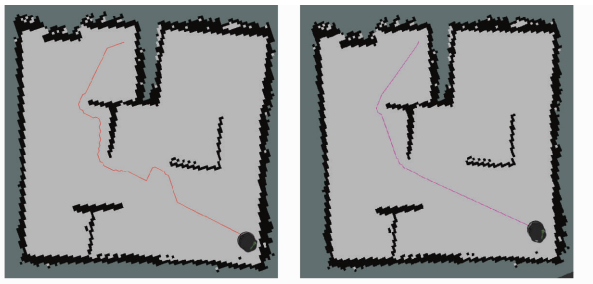
Fig. 16　Mapping result

在机器人路径规划开始之前，需要确定机器人在地图中的位置，本文使用基于粒子滤波的自适应蒙特卡罗定位算法（amcl）对地图中的机器人进行定位。通过激光雷达所感知到的信息确定机器人在地图中的大致位置，然后再使用 Rviz 中的 2D Pose Estimate 对机器人的位姿进行调整。

为实现本文算法的路径规划效果，需要将 TP－A* 算法以 plugin 插件的形式植入到 move_base 模块中，替换原有的全局规划器。接着启动导航 launch 文件，进入 Rviz 界面，将 Rviz 中的全局路径规划算法改为植入的 TP－A* 算法，从而实现本文算法在当前实际场景中的全局路径规划过程[24]。

为了评估改进算法的效果，将 TP－A* 算法与传统 A* 算法做比较。如图 17 所示，右下角为机器人初始位置，图 17a 中的红色路径为传统 A* 算法的路径规划结果，图 17b 中的紫色路径为 TP－A* 算法的路径规划结果。从图中可以看出，TP－A* 算法规划出的路径更短，转折点更少，且路径更为平滑。

表 7 为传统 A* 算法和 TP－A* 算法寻路时间和路径长度实验结果对比，相较于传统的 A* 算法，TP－A* 算法寻路时间减少 88.53%，路径长度减少 21.99%。总体而言，TP－A* 算法不仅能规划出一条更优的路径，而且路径规划的时间更少，路径也更为平滑，能够显著提升机器人导航过程中的效率和稳定性。



(a) A*　　(b) TP-A*

图 17　A* 算法和 TP－A* 算法的路径规划结果

Fig. 17　Path planning results of A* and TP－A*

表 7　A* 与 TP－A* 实验结果对比

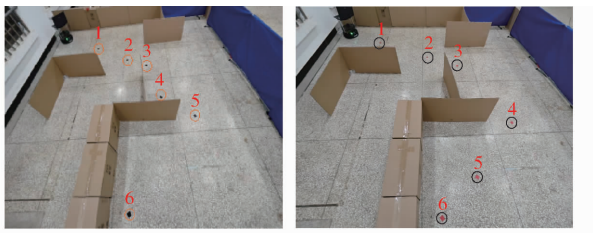
Tab. 7　A* and TP－A* comparison of experimental results

算法	寻路时间/ms	路径长度/m
A*	21.412 3	6.987 6
TP－A*	2.456 6	5.451 2

3.2　轨迹跟踪实验

为验证 TP－A* 算法所规划路径的实际可行性以及在自主移动设备中的性能效果，本文使用 Turtlebot2 移动机器人对传统 A* 算法和 TP－A* 算法所规划的路径进行跟踪，通过对比两条路径的跟踪实验结果进一步说明 TP－A* 算法的路径规划效果。

首先，在图 17 两种算法的路径规划结果中选出 6 个路径点，并根据这 6 个点的位置坐标将其映射到实际场景中，如图 18 所示。使用机器人分别对 A* 和 TP－A* 的 6 个路径点进行 10 次重复跟踪实验并记录跟踪距离误差，实验结果取 10 次跟踪误差的平均值，如表 8 所示。



(a) A*　　(b) TP-A*

图 18　两种算法路径跟踪点

Fig. 18　Two algorithms path tracking points

表 8　A* 和 TP－A* 路径跟踪误差

Tab. 8　A* and TP－A* path tracking error　m

跟踪点	1	2	3	4	5	6
A*	0.066	0.078	0.093	0.129	0.158	0.048
TP－A*	0.023	0.028	0.032	0.115	0.025	0.034

从两种算法的路径跟踪误差结果来看，TP－A* 算法的路径跟踪误差显著低于 A* 算法。A* 算法的误差峰值为 0.158 m，TP－A* 算法的误差峰值为 0.115 m，两者的误差峰值均出现在同一区域，从路

径规划结果中可以看出,该区域路径转折角度过大且存在障碍物,机器人在避障的同时需保证行走路径更为平滑,从而导致跟踪误差较大。结合两种算法的路径规划实验结果可以得出:TP - A* 算法所规划的路径转折点较少,路径整体上更为平滑,路径跟踪误差更小,机器人运行更加稳定。

4 结束语

针对传统 A* 算法在全局路径规划过程中规划时间长、路径非最优等不足,不能满足机器人在某些复杂场景中平稳导航,本文提出了 TP - A* 算法。该算法使用更为接近实际距离的对角线距离作为启发函数,并进行动态加权,让算法在距离目标点较远时以搜索速度为主,在接近目标点时以得到最优路径为主,提高搜索效率。为使最终路径更短、更平滑,使用 TP 算法对初始路径进行优化,减少不必要的转折点和冗余路径点,提升机器人导航过程中的效率和稳定性。通过与传统 A* 算法和 JPS 算法在不同栅格地图环境下的仿真对比分析,TP - A* 算法能够显著提高搜索效率,转折点和冗余路径点更少,最终路径更为平滑。最后将算法部署到 Turtlebot2 移动机器人平台上进行实验验证,结果表明 TP - A* 算法能够满足实际要求,并且导航效果更好,轨迹跟踪误差更小。

参 考 文 献

[1] Mac T T, Copot C, Tran T D, et al. Heuristic approaches in robot path planning: a survey[J]. Robotics and Autonomous Systems, 2016, 86: 13 - 28.

[2] Huang J, Chen C, Shen J, et al. A self-adaptive neighborhood search A-star algorithm for mobile robots global path planning [J]. Computers and Electrical Engineering, 2025, 123: 110018.

[3] Liu L X, Wang X, Yang X, et al. Path planning techniques for mobile robots: review and prospect[J]. Expert Systems with Applications, 2023, 227: 120254.

[4] 史恩秀, 陈敏敏, 李俊, 等. 基于蚁群算法的移动机器人全局路径规划方法研究[J]. 农业机械学报, 2014, 45(6): 53 - 57.
Shi Enxiu, Chen Minmin, Li Jun, et al. Research on method of global path-planning for mobile robot based on ant-colony algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2014, 45(6): 53 - 57. (in Chinese)

[5] 张万枝, 赵威, 李玉华, 等. 基于改进 A* 算法 + LM - BZS 算法的农业机器人路径规划[J]. 农业机械学报, 2024, 55(8): 81 - 92.
Zhang Wanzhi, Zhao Wei, Li Yuhua, et al. Path planning of agricultural robot based on improved A* and LM - BZS algorithms [J]. Transactions of the Chinese Society for Agricultural Machinery, 2024, 55(8): 81 - 92. (in Chinese)

[6] 刘天湖, 赖嘉上, 孙伟龙, 等. 基于跳点优化蚁群算法的菠萝田间导航路径规划[J]. 农业机械学报, 2025, 56(4): 387 - 396.
Liu Tianhu, Lai Jiashang, Sun Weilong, et al. Path planning of pineapple planting field based on jump point optimized ant colony algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2025, 56(4): 387 - 396. (in Chinese)

[7] 李忠林, 罗邵屏, 贾玉婷. 移动机器人路径规划算法综述[J]. 现代信息科技, 2024, 8(19): 184 - 188.
Li Zhonglin, Luo Shaoping, Jia Yuting. Review of path planning algorithms for mobile robots[J]. Modern Information Technology, 2024, 8(19): 184 - 188. (in Chinese)

[8] 梁彧. 基于改进 Dijkstra 算法的 AGV 智能车路径规划[J]. 科技与创新, 2020(24): 159 - 161.
Liang Yu. Path planning for AGV intelligent vehicles based on improved Dijkstra algorithm[J]. Science and Technology Innovation, 2020(24): 159 - 161. (in Chinese)

[9] 李全勇, 李波, 张瑞, 等. 基于改进 Dijkstra 算法的 AGV 路径规划研究[J]. 机械工程与自动化, 2021(1): 23 - 25, 28.
Li Quanyong, Li Bo, Zhang Rui, et al. Research on AGV path planning based on improved Dijkstra algorithm[J]. Mechanical Engineering and Automation, 2021(1): 23 - 25, 28. (in Chinese)

[10] 白晓兰, 袁铮, 周文全, 等. 移动机器人路径规划算法研究综述[J]. 机械工程师, 2024(8): 24 - 28, 33.
Bai Xiaolan, Yuan Zheng, Zhou Wenquan, et al. Survey of path planning algorithms for mobile robots[J]. Mechanical Engineer, 2024(8): 24 - 28, 33. (in Chinese)

[11] 鲁毅, 高永平, 龙江腾. A* 算法在移动机器人路径规划中的研究[J]. 湖北师范大学学报(自然科学版), 2022, 42(2): 59 - 65.
Lu Yi, Gao Yongping, Long Jiangteng. A study on A* algorithm in mobile robot path planning[J]. Journal of Hubei Normal University(Natural Science), 2022, 42(2): 59 - 65. (in Chinese)

[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]. 机器人, 2018, 40(6): 903 - 910.
Zhao Xiao, Wang Zheng, Huang Chengkan, et al. Mobile robot path planning based on an improved A* algorithm[J]. Robot, 2018, 40(6): 903 - 910. (in Chinese)

[13] 张文, 刘勇, 张超凡, 等. 基于方向 A* 算法的温室机器人实时路径规划[J]. 农业机械学报, 2017, 48(7): 22 - 28.
Zhang Wen, Liu Yong, Zhang Chaofan, et al. Real-time path planning of greenhouse robot based on directional A* algorithm

[J]. Transactions of the Chinese Society for Agricultural Machinery, 2017, 48(7): 22 – 28. (in Chinese)

[14] Liang C, Wang Y L, Li S B, et al. Intelligent vehicle path planning based on optimized A* algorithm[J]. Sensors, 2024, 24(10): 3149.

[15] Sun Y, Yuan Q N, Gao Q Y, et al. A multiple environment available path planning based on an improved A* algorithm[J]. International Journal of Computational Intelligence Systems, 2024, 17(1): 172 – 190.

[16] 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]. 仪器仪表学报, 2021, 42(3): 132 – 140.
Chi Xu, Li Hua, Fei Jiyou. Research on robot random obstacle avoidance method based on fusion of improved A* algorithm and dynamic window method[J]. Chinese Journal of Scientific Instrument, 2021, 42(3): 132 – 140. (in Chinese)

[17] 姚得鑫, 伞红军, 王雅如, 等. 移动机器人路径规划中 A*算法的改进研究[J]. 系统仿真学报, 2024, 36(11): 2684 – 2698.
Yao Dexin, San Hongjun, Wang Yaru, et al. Improvement of A* algorithm in path planning of mobile robot[J]. Journal of System Simulation, 2024, 36(11): 2684 – 2698. (in Chinese)

[18] Yu J B, Yang M, Zhao Z Y, et al. Path planning of unmanned surface vessel in an unknown environment based on improved D* Lite algorithm[J]. Ocean Engineering, 2022, 266: 112873.

[19] 秦旭, 黄晓华, 马东明, 等. 基于改进 D*算法的巡检机器人路径规划[J]. 组合机床与自动化加工技术, 2022(6): 10 – 13.
Qin Xu, Huang Xiaohua, Ma Dongming, et al. Path planning of inspection robot based on improved D* algorithm[J]. Modular Machine Tool and Automatic Manufacturing Technique, 2022(6): 10 – 13. (in Chinese)

[20] 黄健萌, 吴宇雄, 林谢昭. 移动机器人平滑 JPS 路径规划与轨迹优化方法[J]. 农业机械学报, 2021, 52(2): 21 – 29, 121.
Huang Jianmeng, Wu Yuxiong, Lin Xiezhao. Smooth JPS path planning and trajectory optimization method of mobile robot[J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(2): 21 – 29, 121. (in Chinese)

[21] 张彪, 李永强. 基于动态寻优蚁群算法的移动机器人路径规划[J]. 仪器仪表学报, 2025, 46(3): 74 – 85.
Zhang Biao, Li Yongqiang. Path planning of mobile robot based on the dynamic optimization ant colony algorithm[J]. Chinese Journal of Scientific Instrument, 2025, 46(3): 74 – 85. (in Chinese)

[22] 房殿军, 王少杰, 蒋红琰, 等. 融合改进 A*算法和动态窗口法的 AGV 路径规划[J]. 物流技术, 2023, 42(8): 47 – 52.
Fang Dianjun, Wang Shaojie, Jiang Hongyan, et al. AGV path planning integrating improved A* algorithm and dynamic window method[J]. Logistics Technology, 2023, 42(8): 47 – 52. (in Chinese)

[23] 魏博闻, 严华. 一种面向非结构化环境的改进跳点搜索路径规划算法[J]. 科学技术与工程, 2021, 21(6): 2363 – 2370.
Wei Bowen, Yan Hua. An improved jump point search path-planning algorithm for unstructured environment[J]. Science Technology and Engineering, 2021, 21(6): 2363 – 2370. (in Chinese)

[24] 伞红军, 杨晓园, 陈久朋, 等. 基于拟水流算法在移动机器人路径规划中的应用[J]. 仪器仪表学报, 2024, 45(7): 263 – 278.
San Hongjun, Yang Xiaoyuan, Chen Jiupeng, et al. Research on path planning of mobile robot based on the stream algorithm[J]. Chinese Journal of Scientific Instrument, 2024, 45(7): 263 – 278. (in Chinese)