http://bhxb.buaa.edu.cn jbuaa@buaa.edu.cn
DOI: 10.13700/j.bh.1001-5965.2022.0377

基于 Dijkstra 算法的平滑路径规划方法

巩慧[1]，倪翠[1,*]，王朋[1,2]，程诺[1]

(1. 山东交通学院 信息科学与电气工程学院，济南 250357；2. 山东省科学院 自动化研究所，济南 250013)

摘 要：移动机器人在复杂环境下沿 Dijkstra 算法规划的路径运动时，由于所规划的路径存在转折点多、部分转折角度小等问题，导致移动机器人不得不频繁转向，甚至要暂停才能完成转向，严重影响机器人的工作效率。利用几何拓扑学方法，结合实际场景信息，提出一种基于 Dijkstra 算法的平滑路径规划方法。根据应用场景获取连续化地图，将连续化地图离散化后随机生成离散点阵，计算各点之间的欧氏距离，选取与各离散点距离较近、且连线不跨越障碍的多个点，将其连接并生成离散图。在离散图中利用 Dijkstra 算法搜索最优路径作为引导路径。当移动机器人沿引导路径运动时，结合实际场景信息，采用几何拓扑学计算出移动机器人每一时刻应该采取的最佳动作和运行路线。实验结果表明：所提方法能够有效减少移动机器人运动中的累计转弯角度，增大最小平均转折角度，提高所规划路径的平滑度，从而缩短移动机器人的运动时间，提升机器人的工作效率。

关 键 词：路径规划；Dijkstra 算法；引导路径；路径平滑度；最优路径
中图分类号：TP391.4
文献标志码：A **文章编号**：1001-5965(2024)02-0535-07

随着移动机器人技术的兴起与发展，移动机器人在许多领域都发挥着重要作用，如智能装配、物流运输等。在移动机器人应用中路径规划作为关键技术之一，其主要目标是使移动机器人在应用场景中找到一条从运动起点到目标点的安全无碰撞路径[1]。优秀的路径规划方法可以减少硬件磨损、节约资源、提高移动机器人的运动速度、提升作业效率，为移动机器人在各行业中的应用提供有利保证[2-4]。

路径规划的经典算法包括人工势场法、A*算法、蚁群算法、遗传算法和 Dijkstra 算法等[5-9]。人工势场法规划的路径一般较为平滑，收敛速度快，但不适用于复杂环境[10]。A*算法受到估计代价值的影响，空间需求大，并且很难找到最优路径[11]。蚁群算法具有较强的鲁棒性，但是存在计算量大、易陷入局部最优等问题[12]。遗传算法具有较强的全局搜索能力，鲁棒性高，但是后期搜索速度较为缓慢，训练时间长[13]。Dijkstra 算法作为一种贪心算法，以起始点为中心原点，逐层展开搜索，直到搜索出从中心原点到所有节点的最短路径[14]。其优点是算法简单方便，可以准确获得全局最优路径，但是偶尔会出现冗余的拐点，效率低[15]。上述算法普遍存在移动机器人运动时间长、工作效率低，且未考虑路径平滑问题。李全勇等[16]在 Dijkstra 算法的基础上引入了路径平滑度，通过时间权重优化目标求解最优路径，减少了移动机器人的运行时间，不过仍存在冗余拐点问题。Zhang 等[17]在 A*算法的基础上，提出一种结合人工势场法的新启发式函数，其中包括距离和障碍物信息，用于自动陆地车辆的路径规划，但是也没有考虑到实际转弯半径的大小。Luan 和 Thinh[18]在混合遗传算法的基础上结合三次贝塞尔曲线为移动机器人提供平滑路径，

收稿日期：2022-05-18；**录用日期**：2022-06-23；**网络出版时间**：2022-09-15 11:22
网络出版地址：link.cnki.net/urlid/11.2625.V.20220914.1950.002
基金项目：中国博士后科学基金 (2021M702030)；山东省交通运输厅科技计划项目 (2021B120)
* **通信作者**. E-mail：1149993471@qq.com

引用格式：巩慧，倪翠，王朋，等. 基于 Dijkstra 算法的平滑路径规划方法 [J]. 北京航空航天大学学报，2024，50（2）：535-541.
GONG H，NI C，WANG P，et al. A smooth path planning method based on Dijkstra algorithm [J]. Journal of Beijing University of Aeronautics and Astronautics，2024，50（2）：535-541（in Chinese）.

但是当移动机器人距离障碍物过近时，需要移动贝塞尔曲线中的相应控制点，这会导致规划好的整条路径都发生改变，无法保证得到最优路径。

本文在Dijkstra算法的基础上充分考虑路径平滑度对移动机器人运动的影响，提出一种基于Dijkstra算法的平滑路径规划方法，通过几何拓扑学方法计算移动机器人在每一时刻应采取的最佳动作和运行路线，提升移动机器人的工作效率。

1 相关工作

1.1 移动机器人与模拟地图的建立

移动机器人的底盘根据转向方式可以分为差速底盘、全向轮底盘、麦克纳姆轮底盘和阿克曼底盘等。本文所用的移动机器人和模拟环境下所用移动机器人底盘均为阿克曼底盘，其特点是移动机器人的车轮系统由4个车轮构成，前后各2个，后轮负责驱动，前轮负责转向。因此，阿克曼底盘的移动机器人在转向时存在最小旋转半径。

移动机器人进行运动规划时需要先采集环境信息并建立地图模型，然后在建立的地图模型上进行路径规划。建立的地图模型主要包括栅格地图和矢量地图2类。栅格地图是按照一定间距排列的一系列阵列数据，其内容和色彩要与实际地形图一致。矢量地图是将与地形图内容一致的空间进行矢量化而得到的地图。本文采用矢量地图，根据真实场景选用尺寸为9×5像素（随机生成尺寸）的迷宫网格，采用Prim算法生成模拟地图，如图1所示。其中，黑色线段代表障碍，白色区域为机器人的活动区域。地图生成表述式为

$$\boldsymbol{M}=\mathrm{Prim}(W,H) \tag{1}$$

式中：$\boldsymbol{M}$为迷宫矩阵；Prim为矩阵生成函数；W为模拟地图的宽度；H为模拟地图的高度。

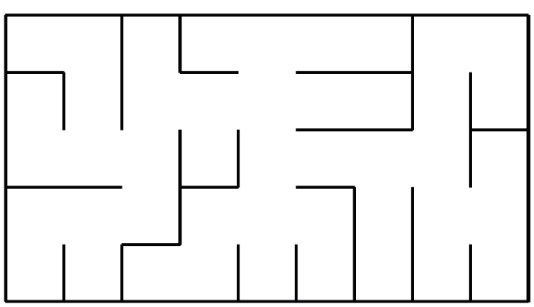


图1 Prim算法生成的模拟地图

Fig. 1 Simulation map generated by Prim's algorithm

1.2 基于Dijkstra算法的路径规划

Dijkstra算法是求解最短路径问题的经典算法。其核心思想是以起始点为原点，逐层向外扩展，直至到达终点。将Dijkstra算法应用到移动机器人路径规划中，为降低计算复杂度，将连续环境下的路径规划任务转化为离散环境下的路径规划任务，以简化路径规划过程。

首先，将模拟地图离散化，随机生成n个离散点，n的计算式如(2)所示，并保证各离散点之间的欧氏距离不小于32像素（根据地图尺寸计算），其目的是使点相对均匀地充满整个空间，避免出现一大堆点聚集在一起，无法连接到终点的现象。最终生成的离散点分布如图2所示。

$$n=\frac{4WH}{D^2} \tag{2}$$

式中：D为模拟地图中迷宫网格的个数。

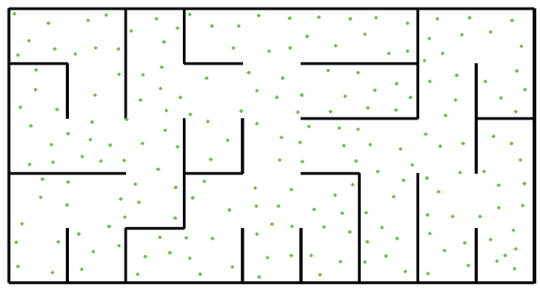


图2 模拟地图中随机生成的离散点示意图

Fig. 2 A sketch of randomly generated discrete points on a simulated map

然后，将n个离散点逐个与其距离最近、但不跨越障碍的m个点（本文取3个）连接，形成离散图。并将移动机器人的运动起点和终点添加到离散图中，如图3所示。其中，黑色“■”号代表移动机器人的起点，红色圆点代表终点。

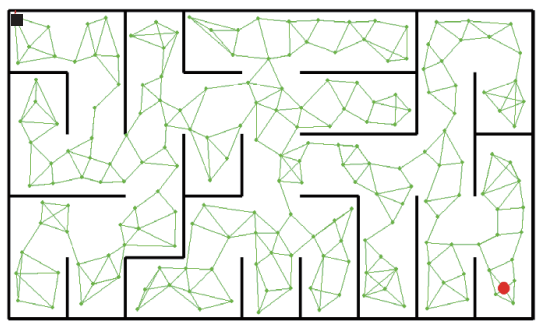


图3 由离散点生成的离散图

Fig. 3 A discrete graph generated from discrete points

最后，采用Dijkstra算法在离散图中搜索，以获得移动机器人从运动起点到终点的路径，如图4所示。

由图4可知，传统Dijkstra算法在离散图中进行路径规划时，生成的路径中转折点较多，累计转弯角度过大。实际应用时，移动机器人在沿着路径前进过程中需要频繁停下来转向，导致运行时间延长，降低了移动机器人的工作效率。并且路径中的

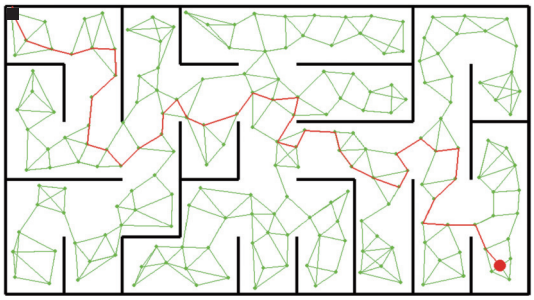


图 4　Dijkstra 算法在离散图中规划的路径

Fig. 4　A path planned by Dijkstra's algorithm in a discrete graph

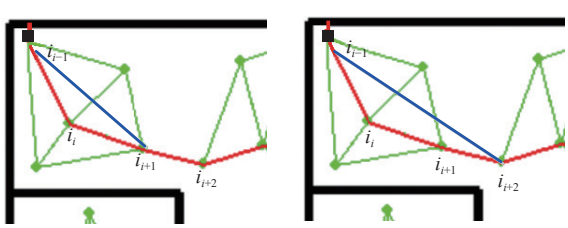
某些转折角度过小, 实际应用时, 可能无法满足移动机器人的转弯半径要求, 难以达到预期的转弯效果, 造成移动机器人无法按所规划的路径行走。

2　平滑路径规划方法

采用几何拓扑学方法, 通过计算移动机器人在每一时刻应采取的最佳动作, 对 Dijkstra 算法规划的路径进行优化。

获取引导路径。对 Dijkstra 算法规划的路径进行遍历。图 5 为引导路径获取方法, 设路径中的节点集合 $I=\{i_1,i_2,\cdots,i_n\}$, 当前节点为 i_i。先判断下一节点 i_{i+1} 是否为空, 若为空, 则表示该地图的引导路径构建完成; 若不为空, 则判断当前节点 i_i 是否需要删除, 如果 i_i 的前一个节点 i_{i-1} 与后一个节点 i_{i+1} 相连, 连线 (i_{i-1},i_{i+1}) 不会穿过障碍, 则删除当前节点 i_i, 将其前后节点 i_{i-1} 与 i_{i+1} 相连接, i_{i-1} 与 i_{i+1} 成为相邻节点, (i_{i-1},i_{i+1}) 成为现阶段的引导路径, 如图 5(a) 所示。然后, 将 i_{i+1} 作为当前节点, 连接 i_{i-1} 与 i_{i+2}, 若连线 (i_{i-1},i_{i+2}) 不会穿过障碍物, 则将节点 i_{i+1} 删除, 现阶段引导路径为 (i_{i-1},i_{i+2}), 如图 5(b) 所示。以此类推, 完成整个地图的引导路径构建, 如图 6 所示, 图中红色线表示 Dijkstra 算法规划的路径, 蓝色线表示获取的引导路径。

令移动机器人沿引导路径运动, 结合移动机器人周围实际场景信息, 通过几何拓扑学方法计算移动机器人在每一时刻应采取的最佳动作; 如图 7 所示, 假设移动机器人的运动路径为 $A\rightarrow B\rightarrow C$, 其中, A、B、C 这 3 点的坐标分别为 (x_1,y_1)、(x_2,y_2)、(x_3,y_3),



(a) 第1阶段引导路径　　(b) 第2阶段引导路径

图 5　引导路径获取方法

Fig. 5　Method for obtaining boot path

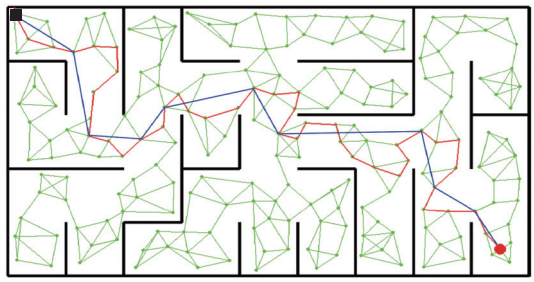


图 6　获取的引导路径示意图

Fig. 6　Acquired boot path diagram

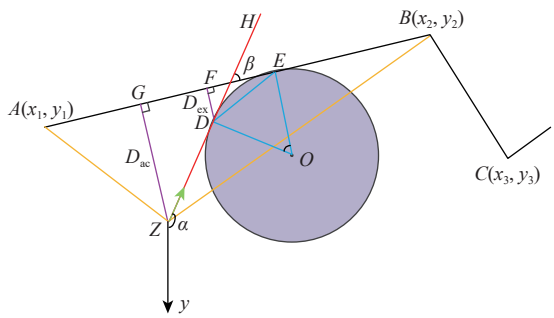


图 7　移动机器人行走轨迹示意图

Fig. 7　Mobile robot walking track diagram

移动机器人从当前位置 Z 点沿绿色箭头方向前进, 其前进方向所在直线 ZH 与目标路径 AB 之间的夹角 β 为

$$\beta=\arctan\frac{x_2-x_1}{y_2-y_1}-\alpha \tag{3}$$

式中: α 为机器人前进方向与 y 轴正方向的夹角。

存在一个圆心 O, 半径 R 的圆, 分别与机器人前进方向所在的直线 ZH 和目标路径 AB 相切。切点分别为 D 和 E。半径 R 即为移动机器人的最小转弯半径。过点 D 和点 Z 分别做直线 AB 的垂线, 交 AB 于点 F 和 G, 移动机器人转向时, 与目标路径之间的预期距离为线段 DF 的长度, 记为 D_{ex}, 计算式如 (4) 所示。与目标路径之间的实际距离为线段 ZG 的长度, 记为 D_{ac}, 计算式如 (5) 所示。

$$D_{ex}=2R\cos\left(90-\frac{\beta}{2}\right)\sin\frac{\beta}{2} \tag{4}$$

$$D_{ac}=\frac{x_1y_2+x_2y_0+x_0y_1-x_1y_0-x_2y_1-x_0y_2}{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}} \tag{5}$$

式中: x_0、y_0 为机器人所在点的横、纵坐标。

根据计算得出的 D_{ex} 和 D_{ac} 的值, 确定移动机器人的当前状态和应选择的最佳运动方案。

1) 当 $D_{ac}>D_{ex}$ 时, 表示移动机器人未到达预期的转向点, 此时应保持继续前进。

2) 当 $D_{ac}=D_{ex}$ 时, 表示移动机器人到达预期的转向点, 此时, 若 $\beta>0$, 则右转, 反之, 则左转。

3) 当 $D_{ac}<D_{ex}$ 时, 表示移动机器人已经超过预

期转向点，此时需要进行路径矫正。

矫正路径如图8所示。假设移动机器人的行走路径为$A\to B\to C$，A、B、C这3点的坐标分别为(x_1,y_1)、(x_2,y_2)、(x_3,y_3)，点Z表示移动机器人当前位置，绿色箭头表示移动机器人前进方向，点N为下一条路径的预期转向起点。为了保证机器人能够在下一条路径的预期转向起点之前完成矫正，就需要计算机器人目前矫正路径的终点X的坐标。移动机器人在矫正路径的时候会经过3个圆弧，分别对应图中的圆O_1、圆O_2(半径为$2R$)、圆O_3中的红色部分，最后通过圆O_4中的圆弧进入下一条路径。

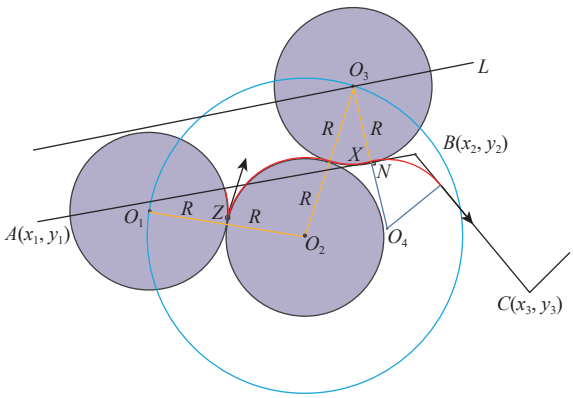


图8 移动机器人矫正路径示意图

Fig. 8 Path correction diagram of mobile robot

计算圆O_2的圆心坐标(x_{O_2},y_{O_2})，计算式为

$$x_{O_2}=x_0-R\sin\left(\alpha-\frac{\pi}{2}\right) \tag{6}$$

$$y_{O_2}=y_0-R\sin\left(\alpha-\frac{\pi}{2}\right) \tag{7}$$

过圆O_3的圆心做直线AB的平行线L，其函数形式如式(8)所示。以O_2为圆心，O_2O_3为半径作圆，如图8中所示圆O_2，函数表达如式(9)所示。

$$y=\frac{y_2-y_1}{x_2-x_1}(x-x_1-R\sin\alpha)+y_1+R\cos\alpha \tag{8}$$

$$(x-x_{O_2})^2+(y-y_{O_2})^2=(2R)^2 \tag{9}$$

通过式(8)和式(9)求O_3的圆心坐标。当$D_{ac}<D_{ex}$时，图8中的圆O_2与直线L存在2个交点，其横坐标函数表达式分别为

$$x_a=\frac{-u+\sqrt{u^2-4(1+k^2)v}}{2(1+k^2)} \tag{10}$$

$$x_b=\frac{-u-\sqrt{u^2-4(1+k^2)v}}{2(1+k^2)} \tag{11}$$

式中：x_a、x_b分别为2个交点的横坐标；k为直线L的斜率，u为一次项系数，v为常数项系数，其计算式为

$$k=\frac{y_2-y_1}{x_2-x_1} \tag{12}$$

$$u=-2k^2(x_1+R\sin\alpha)+2k(y_1+R\cos\alpha)-2y_0k-2x_0 \tag{13}$$

$$v=k^2x^2-2k(x_1+R\sin\alpha)(y_1+R\cos\alpha)+y^2+2y_0k(x_1+R\sin\alpha)-2y_0(y_1+R\cos\alpha)+y_0^2+x_0^2-4R^2 \tag{14}$$

根据式(10)和式(11)求出的x_a,x_b，可以计算得到O_3的圆心坐标(x_{O_3},y_{O_3})，计算式为

$$x_{O_3}=\begin{cases}x_a & \dfrac{x_a-x_b}{x_2-x_1}>0\\ x_b & \dfrac{x_a-x_b}{x_2-x_1}\leqslant 0\end{cases} \tag{15}$$

$$y_{O_3}=k(x_{O_3}-x_1-R\sin\alpha)+y_1+R\cos\alpha \tag{16}$$

根据圆O_3的圆心坐标计算移动机器人进入目标轨迹AB的位置点$X(x_X,y_X)$，计算式为

$$x_X=x_{O_3}+R\sin\alpha \tag{17}$$

$$y_X=y_{O_3}-R\cos\alpha \tag{18}$$

计算点X和点N到目标点A的距离，分别记为实际矫正距离d_1和预期矫正距离d_2，计算式为

$$d_1=\sqrt{(x_X-x_1)^2+(y_X-y_1)^2} \tag{19}$$

$$d_2=\sqrt{(x_N-x_1)^2+(y_N-y_1)^2} \tag{20}$$

式中：x_N和y_N分别表示点N的横坐标和纵坐标。

根据d_1和d_2即可判断机器人应该采取的动作，规则如下：

1) 当$\beta>0$，$d_1>d_2$时，移动机器人左转倒车；

2) 当$\beta<0$，$d_1>d_2$时，移动机器人右转倒车；

3) 当$\beta>0$，$d_1<d_2$时，移动机器人右转前进；

4) 当$\beta<0$，$d_1<d_2$时，移动机器人左转前进。

在离散图中运用Dijkstra算法进行路径规划时，虽然离散点的个数越多路径越趋于平滑，但当离散点的个数增多时，计算量也会显著增大，不利于实际应用。本文综合考虑路径平滑度和计算效率，选取n个离散点，将Dijkstra算法求解得到的路径作为引导路径，结合几何拓扑学方法，计算出移动机器人在每一时刻应采取的最佳动作，这样既能保证规划出的路径更加平滑，又能兼顾移动机器人的工作效率。

3 实验结果与分析

为了验证本文方法的有效性，实验选择Windows 10操作系统下的PyCharm平台，利用TensorFlow 2.1框架实现。

本实验根据真实环境随机生成20幅模拟地图，以及20组移动机器人的起始点坐标和目标点坐标，然后分别利用传统Dijkstra算法和本文方法进行路径规划，图9为部分路径规划的结果。其中红色线为传统Dijkstra算法规划的路径，蓝色线为

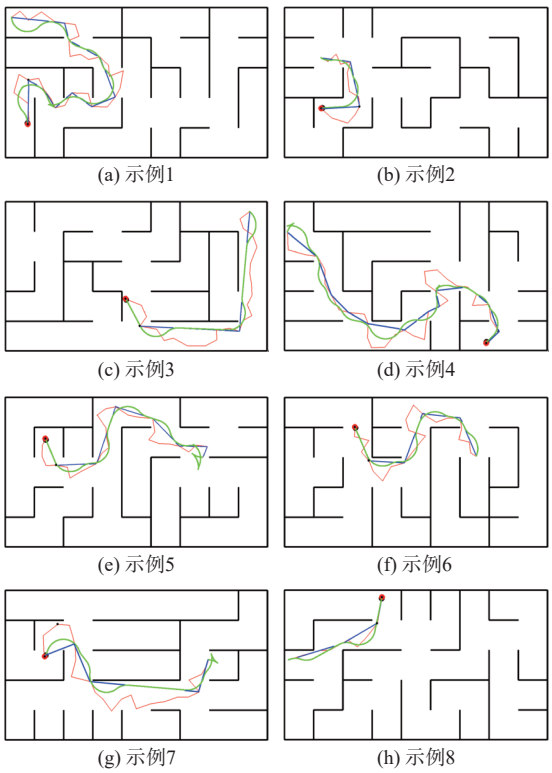


图 9　路径规划部分结果示意图

Fig. 9　Schematic diagram of partial result of path planning

优化后的引导路径, 绿色线为本文方法规划的路径。由图可知, 本文方法所规划的路径更加平滑, 能够有效降低移动机器人的转向频率, 避免移动机器人在运动过程中因转折角度过小导致暂停转向的问题, 保证了移动机器人运动的连续性和平稳性。

移动机器人运动过程中所转过的累计角度是评价一个路径规划方法所规划的路径是否平滑的重要指标。在上述随机生成的 20 幅模拟地图中, 分别对本文方法与传统 Dijkstra 算法、A*算法及文献 [17]、文献 [18] 改进的算法在累计转弯角度方面进行了比较, 结果如图 10 所示。由图可知, 本文方法累计转弯角度最小, 其次为文献 [18] 改进的算法, 再次为文献 [17] 改进的算法, Dijkstra 算法和

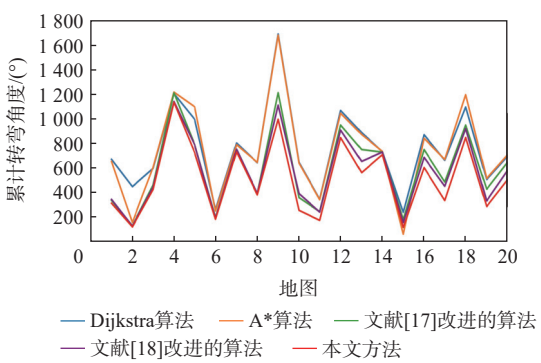


图 10　累计转弯角度比较

Fig. 10　Comparison of cumulative turning angles

A*算法的累计转弯角度最大。本文方法与传统 Dijkstra 算法相比, 平均累计转弯角度减少了 31.82%, 比 A*算法减少了 31.15%, 比文献 [17] 改进的算法减少了 14.15%, 比文献 [18] 改进的算法减少了 9.563%。结果证明本文方法所规划的路径最为平滑。

如果规划的路径中存在转折角度过小的情况, 可能无法满足移动机器人转弯半径要求, 造成机器人无法按照所规划的路径运动。将本文方法分别与传统 Dijkstra 算法、A*算法及文献 [17], 文献 [18] 改进的算法在最小转折角度方面进行了比较, 结果如图 11 所示。由图可知, 本文方法规划的路径中最小转折角度最大, 其次为文献 [17] 和文献 [18] 改进的算法, A*算法和传统 Dijkstra 算法的最小转折最小。本文方法与传统 Dijkstra 算法相比, 最小平均转折角度增大了 54.18%, 比 A*算法增大了 51.13%, 比文献 [17] 改进的算法增大了 17.63%, 比文献 [18] 改进的算法增大了 8.65%。结果证明本文方法规划的路径能有效避免因转折角度过小导致移动机器人无法按照规划路径行走的问题。

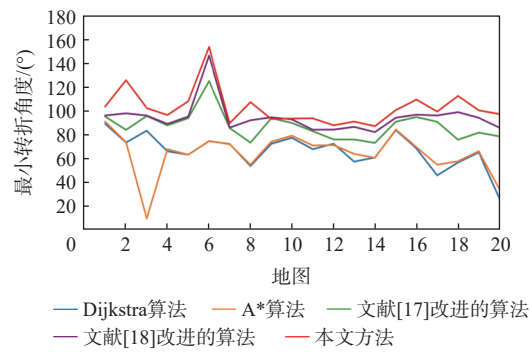


图 11　最小转折角度比较

Fig. 11　Comparison of minimum turning angles

在路径规划中, 移动机器人运动路程的总和称为总路程, 总路程与起点到终点的距离之比可以用于评价一个方法在路径规划方面的最优性, 以及该方法进行路径规划所带来的收益。将本文方法与传统 Dijkstra 算法、A*算法及文献 [17]、文献 [18] 改进的算法进行比较, 计算移动机器人运动的总路程与起点到终点的距离之比, 结果如图 12 所示。由图可知, 在 20 幅模拟地图中, 传统 Dijkstra 算法的平均距离比值为 1.192, A*算法的平均距离比值为 1.171 5, 文献 [17] 改进的算法平均距离比值为 1.178, 文献 [18] 改进的算法平均距离比值为 1.172 5, 本文方法平均距离比值为 1.174 5。本文方法所规划的总路程比传统 Dijkstra 算法、A*算法和文献 [18] 改进的算法长, 这是为了提高移动机器人运动的平滑性和连续性而增大转弯半径的结果。

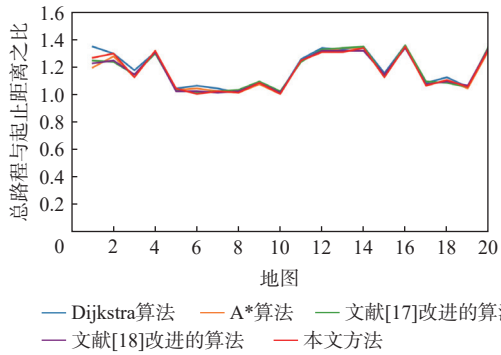


图 12 总路程与起止距离之比

Fig. 12 Ratio of total distance to start-stop distance

此外，移动机器人沿规划路径的运动时间也是评价一个路径规划方法好坏的重要指标。在移动机器人运动时间方面分别对本文方法、传统 Dijkstra 算法、A*算法及文献 [17]、文献 [18] 改进的算法进行了比较，结果如图 13 所示。由图可知，在 20 幅模拟地图中，本文方法移动机器人的运动时间最少，其次为文献 [17] 和文献 [18] 改进算法，A*算法和传统 Dijkstra 算法移动机器人的运动时间最多。与传统 Dijkstra 算法相比，本文方法移动机器人运动时间比传统 Dijkstra 算法缩短了 27.43%，比 A*算法缩短了 19.41%，比文献 [17] 改进的算法缩短了 7.75%，比文献 [18] 改进的算法缩短了 7.45%。由此可知，本文方法在移动机器人工作效率方面具有优势。

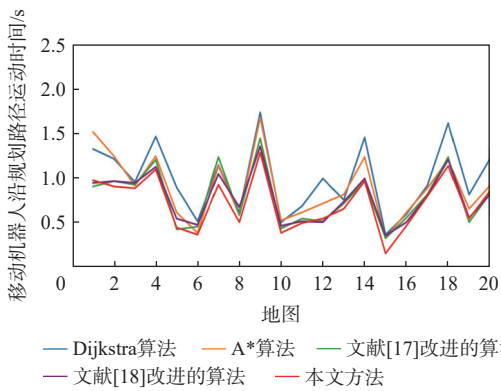


图 13 移动机器人沿规划路径运动时间比较

Fig. 13 Comparison of motion time of mobile robot along planned path

4 结 论

1）采用几何拓扑学方法，计算出移动机器人在每一时刻应该采取的最佳动作和运行路线，有效避免了路径规划过程中转折点过多导致移动机器人频繁暂停再转向，以及转折角度过小导致移动机器人无法按所规划的路径行走等问题。

2）实验结果表明，本文方法与其他算法相比，所规划的路径累计转弯角度更小，最小转折角度更大，移动机器人的运动路径和时间更短，能够有效改善移动机器人运动的平滑性和连续性，提升移动机器人的工作效率，更加符合实际生产生活的应用。

参考文献（References）

[1] LU H W. Artificial intelligence and robotics[M]. Berlin: Springer, 2021: 267-275.

[2] KO J H. An adaptive path-planning for intelligent AGV system[J]. Journal of the Institute of Electronics and Information Engineers, 2017, 54(4): 115-121.

[3] CHEN X, ZHAO M Y, YIN L Y. Dynamic path planning of the UAV avoiding static and moving obstacles[J]. Journal of Intelligent & Robotic Systems, 2020, 99(3-4): 909-931.

[4] 殷建军, 董文龙, 梁利华, 等. 复杂环境下农业机器人路径规划优化方法[J]. 农业机械学报, 2019, 50(5): 17-22.
YIN J J, DONG W L, LIANG L H, et al. Optimization method of agricultural robot path planning in complex environment[J]. Transactions of the Chinese Society for Agricultural Machinery, 2019, 50(5): 17-22(in Chinese).

[5] SANG H Q, YOU Y S, SUN X J, et al. The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations[J]. Ocean Engineering, 2021, 223: 108709.

[6] ZHONG X Y, TIAN J, HU H S, et al. Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment[J]. Journal of Intelligent & Robotic Systems, 2020, 99(1): 65-77.

[7] TIAN X E, LIU L, LIU S A, et al. Path planning of mobile robot based on improved ant colony algorithm for logistics[J]. Mathematical Biosciences and Engineering, 2021, 18(4): 3034-3045.

[8] 魏彤, 龙琛. 基于改进遗传算法的移动机器人路径规划[J]. 北京航空航天大学学报, 2020, 46(4): 703-711.
WEI T, LONG C. Path planning for mobile robot based on an improved genetic algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2020, 46(4): 703-711(in Chinese).

[9] 车建涛, 高方玉, 解玉文, 等. 基于 Dijkstra 算法的水下机器人路径规划[J]. 机械设计与研究, 2020, 36(1): 44-48.
CHE J T, GAO F Y, XIE Y W, et al. Path planning of underwater robots based on the Dijkstra algorithm[J]. Mechanical Design and Research, 2020, 36(1): 44-48(in Chinese).

[10] 邓叶, 姜香菊. 基于改进人工势场法的四旋翼无人机航迹规划算法[J]. 传感器与微系统, 2021, 40(7): 130-133.
DENG Y, JIANG X J. Four-rotor UAV trark planning algorithm based on improved artificial potential field method[J]. Transducer and Microsystem Technologies, 2021, 40(7): 130-133(in Chinese).

[11] TANG G, TANG C Q, CLARAMUNT C, et al. Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment[J]. IEEE Access, 2021, 9: 59196-59210.

[12] HE C W, MAO J A. AGV optimal path planning based on improved ant colony algorithm[J]. EDP Sciences, 2018, 232: 03052.

[13] LI Y H, HUANG Z H, XIE Y. Path planning of mobile robot based on improved genetic algorithm[C]//2020 3rd International Confer-

ence on Electron Device and Mechanical Engineering. Piscataway: IEEE Press, 2020: 691-695.

[14] SUN Y H, FANG M, SU Y X. AGV path planning based on improved Dijkstra algorithm[J]. Journal of Physics, 2021, 1746(1): 012052.

[15] ZHU D D, SUN J Q. A new algorithm based on Dijkstra for vehicle path planning considering intersection attribute[J]. IEEE Access, 2021, 9: 19761-19775.

[16] 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.

LI Q Y, LI B, ZHANG R, et al. Research on AGV path planning based on improved Dijkstra algorithm[J]. Mechanical Engineering and Automation, 2021(1): 23-25(in Chinese).

[17] ZHANG J, WU J, SHEN X, et al. Autonomous land vehicle path planning algorithm based on improved heuristic function of A-Star[J]. International Journal of Advanced Robotic Systems, 2021, 18(5): 172988142110427.

[18] LUAN P G, THINH N T. Hybrid genetic algorithm based smooth global-path planning for a mobile robot[J]. Mechanics Based Design of Structures and Machines, 2023, 51(3): 1758-1774.

A smooth path planning method based on Dijkstra algorithm

GONG Hui[1], NI Cui[1,*], WANG Peng[1,2], CHENG Nuo[1]

(1. College of Information Science and Electrical Engineering, Shandong Jiao Tong University, Jinan 250357, China;
2. Institute of Automation, Shandong Academy of Sciences, Jinan 250013, China)

Abstract: When the mobile robot moves along the path planned by the Dijkstra algorithm in a complex environment, due to the planned path having many turning points and some turning angles being small, the mobile robot has to turn frequently or even pause to complete the turning, which seriously affects the working efficiency of the robot. In this study, the mobile robot's actual scene data is combined with the geometric topology method to propose smooth path planning method based on Dijkstra algorithm. The continuous map is obtained according to the application scenario, and the discrete lattice is randomly generated after the discretization of the continuous map, and the Euclidean distance between the points is calculated. Multiple points which are close to the discrete points and whose connection does not cross the barrier are selected to connect them and generate the discrete graph. The Dijkstra algorithm is used to search the optimal path as the guidance path in the discrete graph. The geometric topology is utilized to determine the optimum action and the running path that the mobile robot should follow at each time as it proceeds along the guidance path in conjunction with the actual scene information. Experimental results show that the proposed method can effectively reduce the cumulative turning angles, increase the minimum average turning angle, and improve the smoothness of the planned path, thus shortening the movement time of the mobile robot and improving the working efficiency of the robot.

Keywords: path planning; Dijkstra algorithm; guide path; path smoothness; optimal path

Received: 2022-05-18; **Accepted**: 2022-06-23; **Published Online**: 2022-09-15 11:22

URL: link.cnki.net/urlid/11.2625.V.20220914.1950.002

Foundation items: China Postdoctoral Science Foundation (2021M702030); Science and Technology Plan Project of Transportation Department of Shandong (2021B120)

* **Corresponding author**. E-mail: 1149993471@qq.com