DOI: 10.3901/JME.2020.18.205

具有确保安全距离的 A* 路径优化方法*

段书用^{1,2} 王启帆^{1,2} 韩旭^{1,2} 刘桂荣³

- (1. 省部共建电工装备可靠性与智能化国家重点实验室(河北工业大学) 天津 300131;
2. 天津市新能源汽车动力传动与安全技术重点实验室(河北工业大学) 天津 300131;
3. 辛辛那提大学航空工程和机械工程系 辛辛那提 45221 美国)

摘要: 机器人或机械臂在工作空间中, 路径的合理规划是提高搜索效率、减小搜索距离、确保作业安全的关键。现有的路径规划算法搜索过程复杂, 在减小搜索距离的同时牺牲了对安全性的高可靠度考量。因此, 本文提出一种改进的 A* 路径规划算法, 在保证搜索效率的同时, 确保沿着路径执行任务时的安全性。通过 4 邻域搜索 A* 算法与多邻域搜索 A* 算法的搜索时间、搜索距离、转折点数对比, 选择搜索时间较短, 转折点数较少的 4 邻域搜索作为搜索方式; 在标准 A* 算法中加入安全距离矩阵, 并改进其启发函数。提出针对不同环境条件下保证路径安全性要求的具体方法; 提出基于不同环境工况条件下可确保安全距离的轨迹优化方法。研究结果表明本研究提出的改进 A* 算法的路径转折点少, 搜索时间短, 后处理简便, 且能确保沿着路径执行任务时的安全性。

关键词: 机器人; 路径规划; A* 算法; 安全距离矩阵; 启发函数; 路径轨迹优化

中图分类号: TP242.6

Improved A-star Algorithm for Safety Insured Optimal Path with Smoothed Corner Turns

DUAN Shuyong^{1,2} WANG Qifan^{1,2} HAN Xu^{1,2} LIU Guirong³

- (1. State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300131;
2. Tianjin Key Laboratory of Power Transmission and Safety Technology for New Energy Vehicles, Hebei University of Technology, Tianjin 300131;
3. Aeronautical Engineering and Mechanical Engineering, University of Cincinnati, Cincinnati 45221, USA)

Abstract: Optimal path planning for automated operators, such as robots or mechanical arms, in the workspace is of importance to improve the efficiency in terms of minimized operation distance and ensure the operation safety along the optimized working path. The existing path planning algorithms, however, still have some limitations, such as complicated searching process, and most importantly omissions of safety considerations while the searching distance was reduced. An improved A-star optimal path planning algorithm is proposed, which produces safety-insured optimal path that are further fine-tuned with smoothing turning corners. First, a comparison study is carried out for a number of neighborhoods searching algorithms in the family of the A-star approach. It is found that the simplest four neighborhood A-star searching algorithm is often most practical in the searching time, searching distances and the number of turning points. Next, we introduce a matrix in the A-star algorithm, that ensures safety distances for operations along the optimal path. In addition, the heuristic functions used in A-star algorithms are modified to ensure the optimal safety path under different environments. Finally, the optimized path trajectory is further fine-tuned for smoothing corner turnings. The proposed algorithms are tested using a number of examples with complicated obstacles. The results show that our improved A-star algorithm has fewer and smoother turning corners, shorter searching time, and simpler post-processing. Most importantly, it ensures the safety for operations along the optimized path.

Key words: robot; path planning; A-star algorithm; safety distance matrix; heuristic function; path trajectory optimization

* 河北省重点研发计划资助项目(19227208D)。20191025 收到初稿, 20200516 收到修改稿

0 前言

路径规划是指依据某种判定准则,在工作空间中寻找一条从初始位置到目标位置避开障碍物的路径^[1],它不仅关系到机器人在作业空间是否能顺利完成任务,而且也关系到机器人的寿命、运行的安全性、稳定性、可靠性和运行的效率等^[2-4]。

常用的路径规划方法有 Dijkstra 算法、神经网络算法、人工势场法、蚁群算法、快速随机树搜索算法、A*算法等。其中 Dijkstra 算法在 1959 年由荷兰科学家 DIJKSTRA^[5]提出,该算法为单源路径算法,采用遍历搜索方式^[6],搜索模式较简单,主要用于求解一个顶点到其余各个顶点的最短路径问题,可以较好地应用在铁路物资调配^[7]与 AGV 在仓库中的调度问题中^[8],但当环境较为复杂时,Dijkstra 算法计算的节点较多,占用的内存空间较大,严重降低了运算效率。神经网络算法具有自主学习的能力,在大量样本数据的训练下能够快速寻找最优解,在自主移动机器人的路径规划及避障作业中得到了广泛应用^[9],然而,该方法需要大量的训练样本及耗时的训练时间,并且对神经网络模型的要求较高^[10-14]。人工势场法在 1986 年由 KHATIB^[15]提出,此方法通过在目标点与障碍物周围构建出引力场与斥力场,再通过搜索势函数的下降方向来规划出无碰撞的最优路径。人工势场法擅长处理动态避障问题^[16],在实时避障方面上已经得到较多应用^[17]。然而人工势场法在障碍物较多的环境中容易出现零势能点,导致局部最小值出现。尽管改进的人工势场法可以逃离局部最小值,但增加了算法的复杂度^[18-20]。蚁群算法由意大利学者 DORIGO 等^[21]以蚂蚁协同寻找食物为数学模型提出的,该算法具有正反馈机制,鲁棒性强,易于与其他算法结合^[22-23],但是该算法收敛速度较慢,搜索时间较长。快速随机树搜索算法是一种基于采样的搜索算法,该方法主要通过对搜索树上的已有节点进行重复扩展从而找到目标点。该方法的搜索速度较快,搜索能力强^[24],不需要对地图进行预处理,在高维度空间搜索广泛应用,尤其是机械臂在三维空间中的路径规划^[25]。但是该方法搜索精度较低,路径的可重复搜索性差^[26-27],路径平滑程度较差^[28]。标准 A*算法在 1968 年由 Stanford 研究院的 HART 等^[29]共同发表,该算法在 Dijkstra 算法的基础上加入了启发函数,通过评价节点的代价值,获取下一个需要拓展的最佳节点,直至扩展到

目标点。由于标准 A*算法比上述其他路径规划算法简便,因此,呈现了很多基于标准 A*算法的改进研究^[30-38]。如:采用曼哈顿距离-欧式距离混合法对标准 A*算法中的启发函数进行加权的方式提高其搜索效率^[30-31];采用双向 24 邻域算法可沿着正反两个方向同时进行搜索的特点,将其引入到标准 A*算法中提高其搜索效率^[32];采用拓展邻域的方法改善 A*算法的搜索距离^[33-36]。由于在路径规划及避障作业过程中,搜索路径的安全性始终是最重要的、是一切功能实现的前提,因此,有研究提出了通过改变邻域矩阵来提高 A*算法中路径规划的安全性^[37],但是这将导致安全距离矩阵的搜索方向巨增,严重降低搜索效率。还有一些研究通过引入节点与障碍物的距离代价来提高 A*算法中路径规划的安全性^[38],由于在障碍物较多的情况下,该方法需要计算每一个障碍物与当前节点的距离,这将显著降低 A*算法的搜索效率。

综上所述,本文将基于标准 A*算法的搜索效率高、搜索过程简便的优点,结合标准及改进 A*算法的不足,提出在标准 A*算法中加入安全距离矩阵,同时改进标准 A*算法启发函数的方法,以期达到既保留标准 A*算法的优点,又可较好的保证其搜索的安全性。本文中的安全距离矩阵不同于节点与障碍物之间的距离代价。节点与障碍物之间的距离代价指的是在一定范围内所有障碍物到节点的距离代价。而本文中提出的安全距离矩阵可以视为一个矩形的范围,当前节点在矩形的中心位置,此时只需要考虑矩形四个角所对应的单元格是否有障碍物存在,如果存在则当前点的代价增加,从而实现避开障碍物。该方法不需要精确的环境地图,也不需要重构地图,算法简便易操作,较好的解决了 A*算法安全性差的问题。此外,为了提高 A*算法改进方法的稳定性和实用性,本文针对不同的环境工况提出了相应的路径优化方法,并进行了模拟验证。

1 标准 A*算法

1.1 标准 A*算法的步骤

标准 A*算法结合了 Dijkstra 算法搜索效率高与 Breadth-First Search 算法搜索路径短的优势,即标准 A*算法既考虑了从起点到当前点的代价,又考虑了从当前点到目标点的代价,从而保证了搜索效率的同时又得到了较短搜索路径。因此,在静态路网的路径规划中,标准 A*算法成为求解最短路

径的最简单有效的直接搜索方法之一, 其代价的计算如式(1)所示

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

式中, $f(n)$ 表示当前点的总代价, $g(n)$ 表示起始点到当前点的代价, $h(n)$ 表示当前点到目标点的代价。在计算 $g(n)$ 和 $h(n)$ 时, 本文采用了应用最广泛的欧几里德距离作为判断代价的标准如式(2)所示

$$\begin{cases} g(n) = \sqrt{(x_{curr} - x_{start})^2 + (y_{curr} - y_{start})^2} \\ h(n) = \sqrt{(x_{curr} - x_{goal})^2 + (y_{curr} - y_{goal})^2} \end{cases} \quad (2)$$

式中, x_{curr} 和 y_{curr} 分别代表当前点的横坐标与纵坐标, x_{goal} 和 y_{goal} 分别表示目标点的横坐标与纵坐标, x_{start} 和 y_{start} 分别代表初始点的横坐标与纵坐标。

标准 A*算法具体搜索过程的流程如图 1 所示, 其流程的详细说明如下。

(1) 构建环境地图(详细建立过程见第 1.2 节)。

(2) 设置开放列表与封闭列表。其中开放列表存储相对于当前点而言的下一步待定节点, 封闭列表存储符合 A*算法搜索判断条件的待定节点。将起点添加到开放列表中, 封闭列表为空。

(3) 选取开放列表中具有最小代价的节点 n , 将其从开放列表中转移到封闭列表中。如开放列表中只有起点时(即此时起点就是代价最小的节点), 可将起点移到封闭列表中。

图 1 标准 A*算法的流程

(4) 判断封闭列表中的节点 n 是否为目标节点, 如果是, 则根据它的前向指针生成路径, 并结束循环。如果不是, 则根据搜索邻域的方向生成节点 n 的后继扩展节点 m_f (f 表示搜索邻域的方向个数, 即后继扩展节点个数。例如 4 邻域搜索, 则 $f=4$, 即存在 4 个后继扩展节点 m_1 、 m_2 、 m_3 、 m_4 ; 8 邻域搜索时, 则 $f=8$, 即存在 8 个后继扩展节点 m_1 、 m_2 、 m_3 、 m_4 、 m_5 、 m_6 、 m_7 、 m_8 。), 搜索邻域的详细选取过程见第 1.3 节。

(5) 判断 m_f 是否在环境地图中的障碍物上, 如果在则删去此点, 不在则保留, 并且分别计算保留点的总代价, 并更新 m_f 。

(6) 基于步骤(5), 增加判断语句, 判断保留的节点是否在开放列表中。如果没有, 则把节点纳入开放列表中; 如果有, 则先比较相同点总代价的大小, 选取有较小总代价的节点(节点坐标相同, 但对应的总代价与前向指针不同), 将其纳入到开放列表中, 并把其余没在开放列表中的节点纳入开放列表中。

(7) 返回步骤(3), 继续循环, 直至找到目标节点 n , 进而得到路径图, 结束搜索。

1.2 环境地图的构建

已知环境状况下, 进行路径规划的首要步骤是建立环境地图。在诸多建立环境地图的方法中^[39-41], 栅格法是较常用的环境地图建立方法, 它首先将环境地图表示为具有固定行列数的矩阵。在矩阵中用“0”表示无障碍物, 在地图上表现为白色, 用“1”表示有障碍物, 在地图上用黑色表示^[42]。栅格的大小即矩阵行列数可以自主选取, 栅格划分地越精细, 对环境的描述越精确, 但占用的储存空间越大, 搜索效率越低。

在栅格地图上构建障碍物时, 通常会按照障碍物的原始形状如圆形、多边形等来构建, 此种构建障碍物的方法会使障碍物的边缘部分失真, 影响搜索效率和路径的安全性。现有研究大多使用矩形包围障碍物原有形状的方法来避免边缘局部失真的情况。因此, 本文中的障碍物统一使用矩形来表示, 图 2 分别将原有形状为圆形、菱形、六边形的障碍物统一用矩形包络其实际外形(图 2 中虚线为原有形状, 实线为矩形包络后的形状)。此时只需保证矩形障碍物的安全性, 即可保证原障碍物的安全性。

图 2 矩形障碍物代替原始障碍物的示意图

1.3 搜索邻域的选择

搜索邻域表示机器人可以运动的方向，常用的搜索邻域为 8 邻域如图 3a 和 24 邻域如图 3b 所示。图 3 所示的邻域搜索矩阵中，“2”代表机器人目前所在的位置，“1”代表机器人下一步允许到达的位置。机器人最终到达的位置取决于该处的代价是否最小，如果该处代价最小则机器人下一步要到达的位置即该位置。图 3 中的箭头表示机器人可以移动的方向，使用 8 邻域搜索矩阵(图 3a)的机器人每一步都可以有 45° 的转角，使用 24 邻域搜索矩阵(图 3b)的机器人每一步都可以有 15° 的转角。当搜索邻域变多时，搜索的效率会变低，但是搜索的方向增加，最终得到的路径的总长度会减小。图 4 为基于不同邻域搜索矩阵的标准 A*算法得到的路径图，其中图 4a 为基于 8 邻域搜索矩阵得到的路径图，图 4b 为基于 24 邻域搜索矩阵得到的路径图，这两种方法的搜索方向均从初始点向着目标点进行。在环境、起始点与目标点相同的情况下，基于 8 邻域搜索矩阵与 24 邻域搜索矩阵所得到的路径图可以看出，邻域搜索数越多搜索路径的长度越短，但是路径转折点越多(转折点个数见表 1)，造成路径的后处理十分复杂，而且转折点过多会使机器人运动时的启停过多，电机输入电流的峰值增加，严重影响机器人使用寿命和可靠性^[43]。

因此，为了减少转折点的个数，本文采用 4 邻域搜索矩阵如图 5 所示，在搜索方向上它不会从初

图3 不同邻域搜索矩阵

图4 基于不同邻域搜索矩阵的标准 A*算法得到的路径图

始点向着目标点搜索，而是向着避开初始点与目标点之间的众多障碍物的方向搜索，并且基于 4 邻域的标准 A*算法与基于 8 邻域、24 邻域搜索的标准 A*算法都能准确到达目标点，具有相同的搜索精度。在与图 4 相同环境、起始点与目标点条件下的基于 4 邻域搜索矩阵的标准 A*算法得到的路径图如图 6 所示。本文中障碍物与边界空隙较大，因此机器人可以绕行，即便空隙较小，机器人无法从障碍物外侧绕行时，基于 4 邻域搜索方法得到的搜索路径也比其他邻域搜索算法得到的路径平滑。使用 4 邻域搜索矩阵、8 邻域搜索矩阵和 24 邻域搜索矩阵的搜索时间、搜索距离、搜索路径转折点个数、准确性统计如表 1 所示，4 邻域搜索方法与 8 邻域和 24 邻域搜索方法相比，其搜索距离较长，但是 4

图5 4邻域搜索矩阵

图6 基于4邻域搜索矩阵的标准 A*算法得到的路径图

邻域搜索方法有两个优点, 第一个优点是搜索时间短, 第二个优点是搜索路径较平滑, 因此综合 4 邻域搜索的优缺点考虑, 采用 4 邻域搜索。

表 1 不同搜索邻域所对应的搜索时间-距离-转折点数与搜索准确性

搜索邻域数	搜索时间/s	搜索距离/m	转折点个数	准确性
4	1.5	875	1	准确
8	1.5	696	8	准确
24	2.7	663	10	准确

2 改进 A*算法

机器人与障碍物间存在适当的安全距离, 对机器人自身及其周围环境都是一种很好的保护, 且一定程度上降低了对机器人路径准确性及可靠性的要求。传统 A*算法进行路径搜索时, 没有考虑机器人与障碍物间的安全距离, 其规划的路径紧贴障碍物, 这就造成对机器人运动控制方面和可靠性方面的要求很高, 同时极大地增加了机器人与障碍物间发生碰撞的可能性。本文提出了安全距离矩阵和与之配合的启发函数来改进传统 A*算法, 进而提高路径规划的安全性, 一定程度上降低了对机器人运动控制方面和可靠性方面的要求。本文提出的安全距离矩阵如式(3)所示, 为一个 $n \times n$ 的矩阵, 矩阵的大小可以根据实际需要控制的安全距离的大小进行自主选取。矩阵 K_n 中“1”代表的是机器人下一步可能到达的位置, 与图 3 邻域搜索矩阵中的“1”意义相同。矩阵中的 4 个角表示待检测的 4 个位置, 为了方便解释, 本文将待检测的 4 个位置随意附上数字“3”, 该数字“3”仅表示待检测的 4 个位置, 不具有其他任何意义。在路径搜索的过程中, 检测每一个“1”点即下一步可能选取点的代价时, 都会继续检测“1”位置处的安全距离矩阵 K_n 中“3”的位置是否存在障碍物(需要检测 4 个位置)。当存在障碍物时, 则在传统 A*算法的启发函数基础上增加一项 $K(n)$ 如式(4)所示, 此时 $K(n)$ 的值就会增加 1。不存在障碍物时, $K(n)$ 的值就保持原值即退化为传统 A*算法的启发函数($K(n)$ 代表的是一个数值, K_n 表示的是一个矩阵)。

$$K_n = \begin{pmatrix} 3 & 0 & \cdots & \cdots & \cdots & 0 & 3 \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & 1 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ 3 & 0 & \cdots & \cdots & \cdots & 0 & 3 \end{pmatrix} \quad (3)$$

$$f(n) = g(n) + h(b) + aK(n) \quad (4)$$

式中, a 为代价 $K(n)$ 的权值系数, 其值的大小一般设为 10, 可根据实际情况进行选取。

为了更清晰地说明安全距离矩阵式(3)的作用, 以单个障碍物场景为例, 如图 7 所示, 黑色矩形块代表障碍物, A 点代表初始点, C 点代表目标点, 假设此场景的边界在无限远处。图中的 $A \rightarrow B \rightarrow F \rightarrow C$ 为不使用安全距离矩阵搜索代价的路径, 其对应的矩阵代价图如图 8 所示。图中的 $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F \rightarrow C$ 路径为使用安全距离矩阵搜索代价的路径, 其对应的矩阵代价图如图 9 所示。图 8、9 地图中的格子代表机器人可以到达的路径点, 每个格子的长度为单位像素, 地图中没有数字的格子表示与搜索过程无关的路径点, 存在数字的格子为与搜索过程有关的路径点。存在数字的格子中有四个数字分别代表式(4)中的 $g(n)$ 、 $h(n)$ 、 $K(n)$ 和 $f(n)$, 其中 a 取为 10。 $g(n)$ 、 $h(n)$ 、 $K(n)$ 和 $f(n)$ 在格子中的相应位置如图 10 所示分别位于左上、右上、左下、右下。图 8、9 中黑色格子表示障碍物, 颜色最深的灰色格子表示初始点与目标点, 颜色较浅的格子表示搜索过程中选择的路径点。地图的原点设为左下角, 初始点的坐标即图 7 中 A 的坐标为(2, 12), 目标点的坐标即图 7 中 C 点的坐标为(5, 0)。

图 7 保证安全性的原理图

使用 4 邻域 A*算法进行搜索时的代价图如图 8 所示。初始点处的 $g(n)$ 为 0, $h(n)$ 经式(2)计算得 $h(n)=12.4$, 未使用安全距离矩阵, $K(n)=0$, 则 $f(n)$ 的值由式(4)计算得 12.4。由于本文采用的是 4 邻域搜索算法, 因此, 下一步可能的位置存在 4 个, 分布在初始点的四周, 此时 4 个点的 $f(n)$ 都可以计算, 选择未搜索过并且 $f(n)$ 最小的点作为下一位置, 依次类推最终得到图 8 中的路径即对应于图 7 中 $A \rightarrow B \rightarrow F \rightarrow C$ 路径。由图 7 或图 8 均可看出不加安全距离矩阵的路径 BC 段紧贴障碍物, 路径的安全性较差, 如果环境地图不精确或机器人控

制有误差将导致机器人很容易与障碍物相撞。

图8 不使用安全距离矩阵搜索代价图

当 A*算法中加入了式(5)所示的安全距离矩阵和与之匹配的启发函数时, 搜索的代价图如图9所示。加入安全距离矩阵后, B点下方的 $f(n)$ 远大于向右运动的 $f(n)$ (将B点下方的点放置在式(5)中的“1”位置处, 则式(5)中左下方“3”位置处存在障碍物, $K(n)$ 为 1, 因此 B 点下方的 $f(n)$ 较大)。因此, 机器人会向右运动, 继续搜索, 选择未被搜索

图9 使用安全距离矩阵搜索代价图

过并且 $f(n)$ 较小的点作为下一位置, 最终得到图9所示的路径图即对应于图7中 $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F \rightarrow C$ 路径。由图7或图9均可看出加安全距离矩阵的路径远离障碍物, 路径的安全性较好, 此时, 对环境地图的精确或机器人控制精度等均要求不高, 可有效避免机器人与障碍物的碰撞。

由上述分析知安全距离矩阵的维度越大, 机器人运动的路径距离障碍物越远, 安全性越好。若环境中有多障碍物或者环境地图较小, 则安全距离矩阵的维度不能设置过大, 否则会出现机器人与障碍物相撞或超出环境地图的问题, 因此要根据机器人的大小选择合适维度的安全距离矩阵。式(6)为计算安全距离矩阵维度的公式, 其中 d_n 表示安全距离矩阵的维度, r_n 表示选取的安全距离, 单位为 m, α 表示地图的分辨率, 单位为像素/m。假设此时环境地图的分辨率为 α , 如机器人为圆形, 半径为 r m, 可选择半径作为安全距离, 代入式(6)中, 可求出安全距离矩阵的维度; 若机器人为矩形, 较长边的长度为 L m, 可选择 $L/2$ 作为安全距离, 代入式(6)中求出矩阵的维度, 安全距离的选择并不固定, 上述为不同形状机器人推荐的安全距离, 实际情况下的安全距离可以自由选取。例如图7中 BD 的长度, 则由安全距离矩阵的维度决定, 维度越大, BD 的长度越大。

$$K_n = \begin{pmatrix} 3 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 3 \end{pmatrix} \quad (5)$$

$$d_n = 2 \times (r_n \times \alpha) + 1 \quad (6)$$

$$\begin{pmatrix} g(n) & h(n) \\ K(n) & f(n) \end{pmatrix}$$

图10 代价图中各位置的含义

为了进一步说明安全距离矩阵的维度对路径的影响, 选择4邻域搜索的传统A*算法在一个特定场景条件下的路径为例, 即不使用安全距离矩阵和与之相配合的启发函数, 得到的路径如图11所示, 其路径紧贴障碍物。当分别使用 3×3 和 5×5 的安全距离矩阵和与之相配合的启发函数时得到的路径分别如图12、13所示。由图11、图12和图13对比分析知, 使用安全距离矩阵所规划出的路径比不使用安全距离矩阵规划出的路径更安全, 并且安全距离矩阵的维度越大规划出的路线距离障碍物越远, 路径越安全。

图 11 无安全距离矩阵的标准 A* 算法得到的路径图

图 12 施加 3×3 安全距离矩阵的改进 A* 算法得到的路径图

图 13 施加 5×5 安全距离矩阵的改进 A* 算法得到的路径图

虽然对障碍物进行膨胀处理^[44]与引入安全距离矩阵都可以保证路径规划的安全性,但是对障碍物进行膨胀处理是对环境地图的所有障碍物进行操作,对地图的精确性要求较高。如果地图不精确,那么膨胀后的地图会扩大不精确程度。而且当障碍物间距较小时,对其进行膨胀处理后,极有可能导致机器人本体无法按照正常路径行驶,可能会增加搜索路径长度、搜索时间和降低搜索路径平滑性。

本文提出的安全距离矩阵的方法并不是通过改

变地图来实现安全性,而是从算法本身的搜索方式上进行了修改,这样降低了在搜索过程中对环境地图精度的要求,为机器人运行的安全性提供了一种简便的新途径。

3 不同环境下的路径优化处理

采用 A* 算法规划出的路径是具有多个转折点的非光滑曲线,机器人按此路线行走会经历多次启停及加减速,增加电机输入电流的峰值,严重影响机器人的使用寿命,因此路径的平滑处理十分必要,以下简称平滑处理。路径平滑处理的流程如图 14 所示。首先根据实际情况,判断路径是否可以简化。当节点个数少于 5 个点时,路径不能简化,当节点个数不小于 5 时,根据图 14 判断路径是否可以简化。图 15a 为不可简化的路径,之后对路径直接进行圆弧过渡处理如图 15b 所示,其中过渡圆弧的半径长度选取一般为路径转折点较短临边长度的 1/4(也可以根据实际需求自主选取,过渡圆弧半径越短,路径越安全)。当路径可以简化时如图 16 所示,处理过程包括两步,第一步是基于初始搜索路径如图 16a 对路径进行隔项简化。首先建立一个空的 Delete 表,输入节点个数 k 。从第 i 个

图 14 路径平滑优化流程图

图 15 不可简化搜索路径图

点开始($i=3$ 到 $k-2$), 判断 i 点与 $i-2$ 点的连线是否与障碍物有交点, 没有则将 $i-1$ 点放入 Delete 表中, 有则继续检测 i 点与 $i+2$ 的连线与障碍物是否相交, 不相交则将 $i+1$ 点放入 Delete 表中, 相交则不对点 $i+1$ 进行处理, 此时 i 点进行了一次循环。之后用 $i+1$ 点代替 i 点继续进行循环, 直到检测完第 $k-2$ 个点, 接着将 Delete 表中的节点都去除 (Delete 表中无节点则不去除节点), 保留剩下的节点并依次连接, 此时便完成了隔项简化如图 16b 所示, 图 16b 隔项简化后的路径与 8 邻域路径搜索的路径不相似。8 邻域搜索虽然路径短, 但是转折点更多, 简化路径时更加复杂。从图 4a 与图 6 可以明显的看出 4 邻域搜索的简便性, 4 邻域搜索再加隔项简化可以使路径更加平滑, 转折点更少。第二步是对隔项简化后的路径进行圆弧过渡处理如图 17 所示, 其过渡圆弧半径的选取原则与图 15b 中过渡圆弧半径的选取原则相同。

本文选择了圆弧过渡法与隔项简化法作为简化路径的方法, 任意的环境工况只会出现两种结果, 第一种情况是只能进行圆弧过渡, 另外一种情况是可以进行隔项简化, 简化后再进行圆弧过渡处理, 由于任意路径的优化措施都只有这两种情况, 因此

本文只介绍了图 15~17 所示的两种典型的例子。

图 16 可简化搜索路径图

图 17 隔项简化与圆弧过渡优化后的路径图

4 结论

本文在传统 A*算法的基础上, 提出了改进 A*算法。该算法加入了安全距离矩阵, 并对与之相配合的启发函数进行了改进。主要研究结论如下所述。

(1) 4 邻域搜索 A*算法规划出的路径相较于于其他的多邻域搜索 A*算法规划出的路径转折点个数少, 路径更加平滑, 是简单而实用的算法。

(2) 改进的 A*算法, 可以针对不同的环境情况选择不同维度的安全距离矩阵来确保沿着路径执

行任务时的安全性, 与使用膨胀地图来确保安全性相比, 降低了搜索过程中对环境地图精度的要求。

(3) 本文提出了两种路径后处理方案, 一种是对满足后处理简化条件的路径, 采取隔项简化法与圆弧过渡法; 另一种是对不满足后处理简化条件的路径, 直接采用圆弧过渡法。这两种路径后处理方法使路径变得更加光滑, 从而减少了机器人电机启停次数及加速度幅值, 从而减小电机输入电流峰值, 增加了机器人的使用寿命。

参 考 文 献

- [1] 陈慧岩, 熊光明, 龚建伟, 等. 无人驾驶汽车概论[M]. 北京: 北京理工大学出版社, 2014.
CHEN Huiyan, XIONG Guangming, GONG Jianwei, et al. Introduction to self-driving car[M]. Beijing: Beijing Institute of Technology Press, 2014.
- [2] DUCHON F, BABINEC A, KAJAN M, et al. Path planning with modified a star algorithm for a mobile robot[J]. Procedia Engineering, 2014, 96(7): 59-69.
- [3] HOU Y B, WANG W, LU X Y. Mobile robot path planning and research in the improved artificial immune algorithm[J]. Advanced Materials Research, 2012, 466(467): 864-869.
- [4] BERG J P V D, OVERMARS M H. Roadmap-based motion planning in dynamic environments[J]. IEEE Transactions on Robotics, 2005, 21(5): 885-897.
- [5] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269-271.
- [6] DONALD E K. A generalization of Dijkstra's algorithm[J]. Information Processing Letters, 1977, 6(2): 1-5.
- [7] FRANK S, DOROTHEA W, KARSTEN W. Dijkstra's algorithm on-line: An empirical case study from public railroad transport[J]. Lecture Notes in Computer Science, 2000, 5(3): 12-17.
- [8] NEPOMNIASCHAYA A S, DVOSKINA M A. A simple implementation of Dijkstra's shortest path algorithm on associative parallel processors[J]. Fundamenta Informaticae, 2000, 43(1): 227-243.
- [9] POMERLEAU D A. Neural-network-based vision processing for autonomous robot guidance[J]. Proceedings of SPIE the International Society for Optical Engineering, 1991, 1469(17): 121-128.
- [10] PARK J H, HUH U Y. Local path planning for mobile robot using artificial neural network: Potential field algorithm[J]. Transactions of the Korean Institute of Electrical Engineers, 2015, 64(10): 1479-1485.
- [11] KASSIM A A, KUMAR B V K V. Path planners based on the wave expansion neural network[J]. Robotics & Autonomous Systems, 1999, 26(1): 1-22.
- [12] ZHU Y J, WANG S G, CHANG J. A kind of path planning algorithm for mobile robot based on neural network[J]. High Technology Letters, 2002, 12(9): 42-45.
- [13] WANG G, LIU X, ZHAO Y, et al. Neural network-based adaptive motion control for a mobile robot with unknown longitudinal slipping[J]. Chinese Journal of Mechanical Engineering, 2019, 32(1): 1-9.
- [14] CAI Y, WANG H, CHEN X, et al. Vehicle detection based on visual saliency and deep sparse convolution hierarchical model[J]. Chinese Journal of Mechanical Engineering, 2016, 29(4): 765-772.
- [15] KHATIB O. Real-time obstacle avoidance system for manipulators and mobile robots[J]. International Journal of Robotics Research, 1986, 5(1): 90-98.
- [16] SHI W R, HUANG X H, ZHOU W. Path planning of mobile robot based on improved artificial potential field[J]. Journal of Computer Applications, 2010, 30(8): 2021-2023.
- [17] WANG W, WANG H. Real-time obstacle avoidance trajectory planning for missile borne air vehicle based on constrained artificial potential field method[J]. Journal of Aerospace Power, 2014, 29(7): 1738-1743.
- [18] PARK M G, LEE M C. A new technique to escape local minimum in artificial potential field based path planning[J]. Journal of Mechanical Science, 2003, 17(12): 1876-1885.
- [19] LIU J, YANG J, LIU H, et al. Robot global path planning based on ant colony optimization with artificial potential field[J]. Transactions of the Chinese Society of Agricultural Machinery, 2015, 46(9): 18-27.
- [20] LI C, JIANG X, WANG W, et al. A simplified car-following model based on the artificial potential field[J]. Procedia Engineering, 2016, 137(7): 13-20.
- [21] DORIGO M, MEMBER S. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Transaction on Evolutionary Computation, 1996, 1(1): 53-66.
- [22] WU Y F, ZHANG X X, WU J Q. Using cellular ant colony algorithm for path-planning of robots[J]. Applied Mechanics & Materials, 2012, 182(183): 1776-1780.

- [23] WANG L, HUI X. Improved ant colony-genetic algorithm for information transmission path optimization in remanufacturing service system[J]. Chinese Journal of Mechanical Engineering, 2018, 31(6): 106-117.
- [24] 刘佳顺, 刘检华, 张之敬, 等. 基于任意时间 RRT 算法的三维自动布线技术[J]. 机械工程学报, 2016, 52(13): 156-165.
- LIU Jiashun, LIU Jianhua, ZHANG Zhijing, et al. Anytime RRT based cable automatic routing under three-dimensional environment[J]. Journal of Mechanical Engineering, 2016, 52(13): 156-165.
- [25] CHU H K, MILLS J K, CLEGHORN W L. Automated dual-arm micromanipulation with path planning for micro-object handling[J]. Robotics & Autonomous Systems, 2015, 74(5): 166-174.
- [26] DU M, MEI T, CHEN J, et al. RRT-based motion planning algorithm for intelligent vehicle in complex environments[J]. Robot, 2015, 37(4): 443-450.
- [27] MOON C B, CHUNG W. Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree[J]. Industrial Electronics, 2015, 62(2): 1080-1090.
- [28] HSIEN I L. 2D-span resampling of Bi-RRT in dynamic path planning[J]. International Journal of Automation and Smart Technology, 2015, 4(4): 39-48.
- [29] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science & Cybernetics, 1968, 4(2): 100-107.
- [30] 王维, 裴东, 冯璋. 改进 A*算法的移动机器人最短路径规划[J]. 计算机应用, 2018, 38(5): 1523-1526.
- WANG Wei, FEI Dong, FENG Zhang. The shortest path planning for mobile robots using improved A* algorithm[J]. Journal of Computer Applications, 2018, 38(5): 1523-1526.
- [31] 王小红, 叶涛. 基于改进 A*算法机器人路径规划研究[J]. 计算机测量与控制, 2018, 26(7): 282-286.
- WANG Xiaohong, YE Tao. Research on robot path planning based on improved A* algorithm[J]. Computer Measurement & Control, 2018, 26(7): 282-286.
- [32] 王淼弛. 基于 A*算法的移动机器人路径规划[D]. 沈阳: 沈阳工业大学, 2017.
- WANG Miaochi. Path planing of mobile robots based on A* algorithm[D]. Shenyang: Shenyang University of Technology, 2017.
- [33] 任玉洁, 付丽霞, 张勇, 等. 拓展搜索邻域的平滑 A*算法机器人路径规划[J]. 电子科技, 2018, 31(5): 33-43.
- REN Yujie, FU Lixia, ZHANG Yong, et al. Smoothing A* algorithm extended search neighborhood for robot path planing[J]. Electronic Sci.&Tech., 2018, 31(5): 33-43.
- [34] 辛煜, 梁华为, 杜明博, 等. 一种可搜索无限个邻域的改进 A*算法[J]. 机器人, 2014, 36(5): 627-633.
- XIN Yu, LIANG Huawei, DU Mingbo, et al. An improved A* algorithm for searching infinite neighbourhoods[J]. Robot, 2014, 36(5): 627-633.
- [35] 张敬寒, 陶兆胜, 彭澎, 等. 基于扩大搜索邻域 A*算法的平滑路径规划[J]. 长春理工大学学报, 2018, 41(6): 124-127.
- ZHANG Jinghan, TAO Zhaosheng, PENG Peng, et al. Smooth path planning based on extended search neighborhood A* algorithm[J]. Journal of Changchun University of Science and Technology, 2018, 41(6): 124-127.
- [36] 崔宝侠, 王淼弛, 段勇. 基于可搜索 24 邻域的 A*算法路径规划[J]. 沈阳工业大学学报, 2018, 40(2): 180-184.
- CUI Baoxia, WANG Miaochi, DUAN Yong. Path planning for A* algorithm based on searching 24 neighborhoods[J]. Journal of Shenyang University of Technology, 2018, 40(2): 180-184.
- [37] 陈若男, 文聪聪, 彭玲, 等. 改进 A*算法及其在室内机器人路径规划中的应用[J]. 计算机应用, 2019, 39(4): 78-83.
- CHEN Ruonan, WEN Congcong, PENG Ling, et al. Improve A* algorithm and apply to indoor path planning for mobile robots[J]. Journal of Computer Applications, 2019, 39(4): 78-83.
- [38] 张红梅, 李明龙, 杨乐. 基于改进 A*算法的移动机器人安全路径规划[J]. 计算机仿真, 2018, 35(4): 324-329.
- ZHANG Hongmei, LI Minglong, YANG Le. Safe path planning of mobile robot based on improved A* algorithm[J]. Computer Simulation, 2018, 35(4): 324-329.
- [39] CHENG X, QI Y. Indoor indicator path planning algorithm based on grid method[J]. Journal of Chinese Inertial Technology, 2018, 26(2): 236-240.
- [40] WEI P, SUN J, MENG X. A ray path calculation method based on grid traveltime[J]. Oil Geophysical Prospecting, 2018, 53(4): 722-729.
- [41] PI J, ZHANG W, ZHANG D, et al. Research on robot

- manipulator path planning based on Delaunay map[J]. Academic Journal of Manufacturing Engineering, 2018, 16(1): 135-143.
- [42] 刘梅. 基于栅格化视觉的机器人路径优化研究[J]. 计算机与数字工程, 2018, 46(8): 1548-1552.
- LIU Mei. Research on robot path optimization based on rasterized vision[J]. Computer & Digital Engineering, 2018, 46(8): 1548-1552.
- [43] 高峰, 郭为忠. 中国机器人的发展战略思考[J]. 机械工程学报, 2016, 52(7): 1-5.
- GAO Feng, GUO Weizhong. Thinking of the development strategy of robots in China[J]. Journal of Mechanical Engineering, 2016, 52(7): 1-5.
- [44] 刘晨曦. 基于改进 A*算法的仿人机器人路径规划研究[D]. 北京: 北京建筑大学, 2018.
- LIU Chenxi. Research on path planning based on improved A-star algorithm of humanoid robot[D]. Beijing: Beijing University of Civil Engineering and Architecture, 2018.

作者简介: 段书用, 女, 1984 年出生, 副教授, 硕士研究生导师。主要研究方向为计算反求技术、机器人智能感知、智能算法。

韩旭(通信作者), 男, 1968 年出生, 教授, 博士研究生导师。主要研究方向为计算反求技术、机器人可靠性、机器人智能感知、智能算法。

E-mail: xhan@hebut.edu.cn