

基于并行一交替式双向 JPS 算法的机器人路径规划

苗红霞^{1,2}, 郭章旺¹, 齐本胜¹, 邹 扬¹, 李成林¹

(1. 河海大学 物联网工程学院, 江苏 常州 213000;

2. 江苏省输配电装备技术重点实验室, 江苏 常州 213000)

摘要: 针对跳点搜索 (JPS, jump point search) 算法在障碍物位置随机的栅格地图中路径规划时间较长的问题, 提出了并行一交替式双向跳点搜索 (PA-BJPS, parallel alternate bidirectional jump point search) 算法; 首先, 在起始点与目标点间确定一个中心热点区域; 其次, 采用改进了预计代价函数的并行式双向跳点搜索算法, 分别规划从起始点抵达中心热点区域以及目标点抵达中心热点区域的路径; 然后, 采用交替式双向跳点搜索算法, 规划中心热点区域内部的路径; 最后, 提出迭代式路径修正方法来改良危险路径, 并采用 3 次 B-样条曲线替代拐角来平滑路径; 仿真结果表明, 并行一交替式双向跳点搜索算法有效地缩短了路径规划时间, 同时提高了路径的安全性和平滑性。

关键词: 路径规划; 跳点搜索算法; 预计代价函数; 路径修正; 3 次 B-样条曲线

Robot Path Planning Based on Parallel Alternate Bidirectional JPS Algorithm

MIAO Hongxia^{1,2}, GUO Zhangwang¹, QI Bensheng¹, ZOU Yang¹, LI Chenglin¹

(1. College of Internet of Things Engineering, Hohai University, Changzhou 213000, China;

2. Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology, Changzhou 213000, China)

Abstract: Aiming at the problem of long path planning time of traditional jump point search (JPS) algorithm in grid map with random obstacle location, a parallel alternate bidirectional jump point search algorithm (PA-BJPS) is proposed. Firstly, a central hot spot area is determined between the starting point and the target point; Secondly, the parallel bidirectional jump point search algorithm with the improved expective cost function is used to plan the paths from the starting point and the target point to the central hot spot area respectively; Then, the alternate bidirectional jump point search algorithm is used to plan the path inside the central hot spot area; Finally, an iterative path correction method is proposed to improve the dangerous path, and the cubic B-spline curve is used to replace the corners to smooth the path. The simulation results show that the PA-BJPS algorithm effectively shortens the path planning time and simultaneously improves the security and smoothness of the planned path.

Keywords: path planning; jump point search algorithm; estimated cost function; path correction; cubic B-spline curve

0 引言

路径规划是移动机器人研究领域的关键内容^[1], 它是指在障碍物环境下, 机器人自主地规划出一条从起始点运行至目标点的无碰撞路径。为解决机器人的路径规划问题, 各国科研人员主要提出了以下四类路径规划算法^[2]: 1) 基于智能搜索算法, 如: 蚁群算法、粒子群算法、快速搜索随机数算法等; 2) 基于人工智能算法, 如: 深度学习、Q-Learning 算法等; 3) 基于几何模型算法, 如: Dijkstra 算法、Voronoi 图等; 4) 基于局部避障算法, 如: 人工势场法、动态窗口法等。

在上述的四类路径规划算法中, 基于智能搜索算法的路径规划其最大的特点是具有随机性, 它首先随机生成初始解或采样点, 然后通过迭代的方式来逼近最优解, 因此

它的解是渐进最优且非唯一的, 当地图中的最优路径比较狭窄时, 该方法可能会规划出较长的冗余路径。基于人工智能算法的路径规划能够让机器人自主对环境进行学习以实现自主路径规划, 其通常只使用在具有大量数据样本的场景中。基于避障算法的路径规划主要适用于移动机器人在局部地图中的避障场景, 主要用于提高移动机器人的安全性。基于几何模型的路径规划算法能够实时调节根据最优策略得到的可行解, 其在工业、农业、医疗等领域的移动机器人路径规划中均得到广泛应用, 极大地提高了生产效率, 适用于大规模静态场景下的移动机器人路径规划问题。

基于几何模型的路径规划先需根据已知的环境信息构建地图几何模型, 再在几何模型中规划合适的路径。常见的地图模型有 3 种: 1) 语义地图; 2) 栅格地图; 3) 拓扑

收稿日期: 2022-01-10; 修回日期: 2022-02-23。

基金项目: 江苏省输配电装备技术重点实验室开放课题资助(2021JSSPD05)。

作者简介: 苗红霞(1968-), 女, 河北邯郸人, 博士, 副教授, 主要从事机器学习方向研究。

齐本胜(1969-), 男, 湖北枣阳人, 博士, 副教授, 主要从事智能信息系统设计方向研究。

引用格式: 苗红霞, 郭章旺, 齐本胜, 等. 基于并行一交替式双向 JPS 算法的机器人路径规划[J]. 计算机测量与控制, 2022, 30(7): 233-239.

地图。构建语义地图^[3]需要与同步定位与建图技术相结合,其主要应用于识别环境中独立个体,以应对复杂场景及完成更加智能的任务。栅格地图能够反应物体真实的物理尺寸,在实时环境中有助于机器人的定位与导航^[4]。拓扑地图是对地理空间的抽象和形式化描述^[5],其采用节点和连线来表述环境信息,通常关键位置以节点的形式表示,连线表示节点之间是否允许通行。相较于只表示不同节点之间连通关系的拓扑地图而言,栅格地图能够唯一确定地表示机器人、障碍物、可行路径的确切位置信息。与此同时,栅格地图还具有易于构建和保存的特点,因此各国的工程师和科研人员研究提出了诸多建立在栅格地图基础上的路径规划算法。

A*算法是一种可应用在栅格地图中的启发式路径规划算法,其在搜索过程中具有一定的方向性,具有运行效率较高的特点^[6]。该方法广泛应用于单机器人在全局静态环境中的路径规划。文献[7]通过在估价函数中加入车身轮廓代价和障碍物距离代价对A*算法进行改进;文献[8]参考了人工势场法的引力思想,提升了A*算法的效率。即便如此,在障碍物栅格数量较少的空旷区域,A*算法仍需对栅格进行逐个搜索,这导致了冗余的搜索栅格增多,降低了算法的效率^[9]。

针对A*算法在空旷区域搜索效率十分有限的问题,跳点搜索算法(JPS, jump point search)在A*算法的基础上进行了优化。文献[10]的验结果表明,由于跳点搜索算法在扩展过程中裁剪了无用的节点,其路径规划速度高于A*算法。文献[11]提出了一种正六边形栅格跳点搜索算法,并以此解决智能体在障碍物地图中的路径规划问题;文献[12]提出通过“块”操作的方法减少跳点搜索过程中不必要的节点,进一步加快跳点搜索算法的速度。然而,在障碍物位置分布随机的情况下,障碍物个数的增加导致了跳点数量的攀升,跳点搜索算法的路径规划的时间仍大幅度增加。此外,由于跳点搜索算法没有考虑机器人的体积大小,其规划出的路径存在紧贴障碍物、轨迹不平滑^[13]问题,给机器人的运行带来安全风险。

在障碍物位置随机的栅格地图中,针对跳点搜索算法路径规划时间大幅度增加的问题,本文提出了并行-交替式双向跳点搜索(PA-BJPS, parallel alternate bidirectional jump point search)算法,其首先在障碍物位置随机的栅格地图里确定一个中心热点区域,在该中心热点区域的外部采用并行运算的方式规划路径,在中心热点区域内部采用交替运算的方式规划路径,有效地缩短了路径规划的时间。针对跳点搜索算法规划的路径存在安全性和平滑性不足的问题,本文提出迭代式路径修正方法来改良危险路径,并采用3次B样条曲线替代拐角来平滑路径。仿真实验结果表明,本文算法有效地缩短了路径规划的时间,同时兼顾了路径的安全性和平滑性,具有良好的工程应用前景。

1 跳点搜索算法及其不足

跳点搜索算法^[14]在保留A*算法框架的同时,优化了

A*算法对后续节点扩展的操作。通过对扩展节点的筛选和处理,跳点搜索算法不仅缩短了路径规划的时间,而且减小了节点信息的内存占用量。然而,跳点搜索算法只将移动机器人视为一个质点,因此由跳点搜索算法规划的路径可能会由于移动机器人紧贴障碍物而产生碰撞风险。此外,若采用跳点搜索算法进行双向的路径规划,还可能会造成路径规划时间增长和路径冗余的问题。

1.1 跳点搜索算法

在进行路径规划时,跳点搜索算法对扩展过程中的无用节点进行裁剪,只保留了用于扩展的跳点。当跳点搜索算法扩展至目标点后,即得到一条从起始点到目标点的无障碍路径。

跳点搜索算法进行路径规划时,首要筛选栅格中的强迫邻居(Forced Neighbour)。当栅格节点 x 的八邻域中有障碍物,且节点 x 的父节点 p 经过 x 到达节点 n 的代价距离,比父节点 p 经过任何其他节点到达节点 n 的代价距离更小,则称 n 为 x 的强迫邻居,强迫邻居的筛选规则如式(1):

$$\text{len}(\langle p(x), x, n \rangle) < \text{len}(\langle p(x), \dots, n \rangle \setminus x) \quad (1)$$

强迫邻居主要包含两种类型,横向强迫邻居如图1(a)所示,在图1(a)中画圈的节点为 x 的斜向强迫邻居;斜向强迫邻居如图1(b)所示,在图1(b)中画圈的节点为 x 的横向强迫邻居。

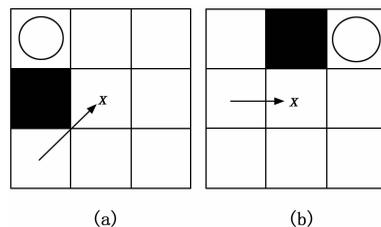


图1 强迫邻居

在筛选出强迫邻居的基础上,还需对跳点进行筛选。如果节点 x 满足下列3个条件之一,则该节点为跳点。

- 1) 节点 x 是起始点或目标点;
- 2) 节点 x 至少有一个强迫邻居;
- 3) 在斜向搜索情况下,节点 x 的水平或垂直方向上有满足1), 2)的节点。

在进行节点扩展时,如果搜索到符合要求的跳点,则将该跳点加入OpenList列表。在OpenList列表中,每次扩展只选取评价函数最小的节点作为下一个待扩展的节点^[15]。跳点搜索的评价函数公式定义如式(2):

$$f(n) = g(n) + h(n) \quad (2)$$

式(2)中, $f(n)$ 为节点 n 的评价函数,它由实际代价函数 $g(n)$ 和预计代价函数 $h(n)$ 组成。实际代价函数 $g(n)$ 反映了从起始点到当前节点的实际距离;预计代价函数 $h(n)$ 估算了当前节点到目标点的剩余距离,通常预计代价函数 $h(n)$ 采用欧几里得距离^[16]、曼哈顿距离^[17]或切比雪夫距离^[18]进行计算。

跳点搜索算法的路径规划结果如图2所示。图2中,栅

格 s 为起始点, 栅格 g 为目标点, 黑色栅格为障碍物, 灰色栅格为路径规划过程中扩展的跳点, 箭头标识了从起始点到目标点依次经过的路径。从图 2 中可知, 跳点搜索算法对路径规划过程中的冗余节点进行了裁剪, 实现了对节点的跳跃搜索, 从而加快了路径规划的速度。

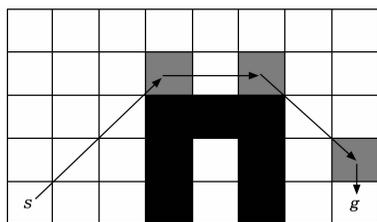


图 2 跳点搜索路径规划

1.2 跳点搜索算法的不足

与 A^* 算法相比, 跳点搜索算法虽然加速了路径规划的速度, 但是它仍存在一些不足。在路径规划时, 跳点搜索算法只将移动机器人视为一个质点, 并未考虑移动机器人的体积大小, 因此由跳点搜索算法规划出来的路径存在机器人紧贴障碍物的问题^[19]。紧贴障碍物的危险路径示意图如图 3 所示, 在图 3 中用银白色圆圈标记了危险路径的位置, 如果机器人按照图 3 中的路径运行, 将在银白色圆圈位置与障碍物发生碰撞。此外, 由跳点搜索算法规划的路径由栅格节点依次相连而成, 因此路径存在不连续, 拐角很多且不够平滑的缺点。在路径的拐角处, 移动机器人必须停下, 原地旋转至设定角度, 再向前运行。频繁的急停急起操作, 不仅不符合移动机器人的运动特性, 还加大了移动机器人的后期维护成本。

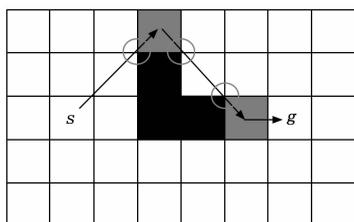


图 3 紧贴障碍物的危险路径

在移动机器人路径规划问题中, 若采用双向搜索策略, 从起始点和目标点同时规划路径, 将有效地提高路径规划的效率^[20]。然而, 当使用跳点搜索算法进行双向的路径规划时, 可能会发生以下 3 种可能导致冗余路径增多或路径规划时间增长的情况。

情况 1: 如图 4 所示, 当障碍物形状为对称图形时, 从起始点规划的正向路径从障碍物的上方经过, 从目标点规划的反向路径从障碍物的下方经过, 正向和反向的路径有一定概率分别从两侧经过障碍物两侧, 规划出了两条不相交的路径。若发生这种情况, 路径规划的时间不减反增, 同时还增加了移动机器人路径规划的计算量。

情况 2: 跳点搜索算法采用了跳跃搜索的方式, 它裁剪了路径规划的过程中不能成为跳点的栅格。跳跃搜索的特

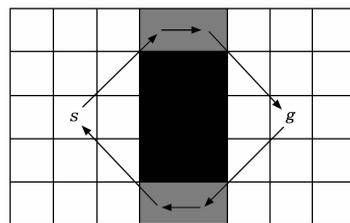


图 4 对称障碍物下的双向路径规划

性虽然能够达到较少内存及提高路径规划的速度目的, 但是在双向路径规划的情况下, 可能会在一侧跳点扩展的过程中, 越过另一侧路径正在扩展的跳点, 从而导致路径冗余的产生, 同时也大大增加了路径规划的时间。

情况 3: 在障碍物位置随机的栅格地图中, 从起始点和目标点出发的跳点搜索算法的方向和距离均存在很大的不确定性。由于障碍物的位置是不规则的, 且跳点搜索算法具有跳跃扩展的特性, 所以从起始点和目标点规划的两条路径, 其相遇节点可能距离两点连线的中心较远, 这也将导致路径规划的时间大大增加。

为避免发生上述导致路径规划时间增长的情况, 同时进一步提升机器人路径规划的速度, 并兼顾运行路径的安全性和平滑性, 本文提出了基于并行一交替式双向跳点搜索算法的机器人路径规划方法。

2 并行一交替式双向跳点搜索算法

与传统的跳点搜索算法不同, 并行一交替式跳点搜索算法首先在起始点与目标点间确定了一个中心热点区域, 从而增加双向并行搜索的方向性; 其次, 采用并行式双向跳点搜索算法, 分别规划从起始点抵达中心热点区域以及目标点抵达中心热点区域的路径, 在该过程中无需进行信息交换和状态同步, 因此缩短了移动机器人路径规划的时间; 最后, 采用交替式双向跳点搜索算法, 相向地规划中心热点区域内部的路径, 将中心热点区域外部的路径与内部的路径拼接, 即得到一条从起始点到目标点的完整路径。

2.1 中心热点区域

在双向的跳点搜索算法进行路径规划时, 从起始点和目标点出发的两条路径, 其搜索方向存在不确定性, 因此可能会发生在 1.2 节中所述的导致冗余搜索节点增多、路径规划时间增加的 3 种情况。

为解决上述问题, 本算法首先在栅格地图中确定一个如图 5 所示的中心热点区域, 再采用并行式双向跳点搜索算法, 分别规划从起始点和目标点抵达中心热点区域的路径。确定中心热点区域之后, 并行式双向跳点搜索算法以中心热点区域作为全局路径规划初始目标。中心热点区域不仅能够增强双向路径规划的方向性, 还能够将两侧路径的相遇位置限制在中心热点区域内部, 避免了冗余路径的产生。本算法所确定的中心热点区域的形状是一个圆形, 其轨迹的计算公式如式 (3):

$$(x - x_{center})^2 + (y - y_{center})^2 = R^2 \quad (3)$$

式 (3) 中, x_{center} 和 y_{center} 是圆心的横坐标和纵坐标, 它是起始点与目标点中点, R 是圆的半径。经过多次实验测试得出, 当中心热点区域的半径长度为起始点与目标点欧式距离的 $1/6$ 左右时, 本算法能较好地兼顾路径规划的时效性和方向性。

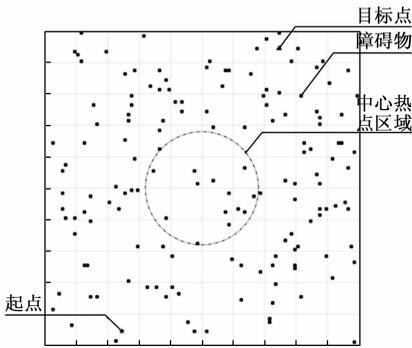


图 5 中心热点区域

在中心热点区域外部, 采用并行的方式规划路径, 当扩展的跳点进入中心热点区域内部时, 停止并行规划, 并记录下由并行式双向跳点搜索算法规划的进入中心热点区域的第一个跳点。在中心热点区域内部, 从记录下的跳点开始, 采用双向交替的方式相向地规划路径, 从而确保两条相向的路径能够连接在一起。

在障碍物位置随机的栅格地图中, 非障碍物节点之间往往具有良好的连通性, 因此当起始点和目标点并行规划的路径抵达中心热点区域之后, 在中心热点区的内部亦能相向地规划出一条相遇的路径, 最后将中心热点区域内部和外部的路径相连, 就得到一条从起始点到目标点的全局路径。

2.2 并行式双向跳点搜索算法

在中心热点区域外部, 并行式双向跳点搜索算法分别了从起始点和目标点规划抵达中心热点区域的路径。在进行路径规划的过程中, 其采用了并行计算的思想, 并且在该过程中无需进行信息交换和状态同步, 因此加快了路径规划的速度。以从起始点侧出发的并行式双向跳点搜索算法为例, 进行路径规划的步骤为:

- 步骤 1: 将起始点加入 $OpenList_S$;
- 步骤 2: 寻找 $OpenList_S$ 中 $h(n)$ 最小的点, 设置为 $current_S$;
- 步骤 3: 判断路径 1 的当前节点 $current_S$ 是否位于中心热点区域内部, 若是则结束路径 1 搜索, 并将 $current_S$ 标记为 A_0 , 若否则进行步骤 4;
- 步骤 4: $OpenList_S$ 删除当前节点 $current_S$, 在 $CloseList_S$ 加入当前节点 $current_S$;
- 步骤 5: 根据跳点扩展规则, 拓展下一个跳点 $point$;
- 步骤 6: 判断跳点 $point$ 是否在 $OpenList_S$ 中, 如果否则将跳点 $point$ 加入 $OpenList_S$, 如果是则更新 $g(n)$ 的值以及父节点的位置, 并返回步骤 2。

由于并行式双向跳点搜索算法在路径规划过程中不进

行信息交换和状态同步, 因此, 为了进一步加强双向搜索的方向性, 并行式跳点搜索算法对预计代价函数进行了改进。按照改进后的预计代价函数进行路径规划, 能够同时兼顾中心热点区域的位置和目标点的位置。

以从起始点侧出发的并行式双向跳点搜索算法为例, 其预计代价函数计算公式如式 (4):

$$h_s(n) = p \cdot h_{s1}(n) + (1 - p) \cdot h_{s2}(n) \quad (4)$$

式 (4) 中, p 是权值, 是节点 n 到中心热点区域圆心的欧几里得距离, 节点 n 到目标点的欧几里得距离。

从目标点侧出发的路径, 与从起始点出发的路径规划是并行运算的, 其路径规划步骤与起点侧的步骤对称一致, 它的预计代价函数的计算公式如式 (5):

$$h_g(n) = q \cdot h_{g1}(n) + (1 - q) \cdot h_{g2}(n) \quad (5)$$

式 (5) 中, q 是权值, 是节点到中心热点区域圆心的欧几里得距离, 是节点到起始点的欧几里得距离。

2.3 交替式双向跳点搜索算法

在规划中心热点区域内部的路径之前, 先读取由并行式双向跳点搜索算法规划的跳点标 A_0 和 B_0 , 再采用交替式双向跳点搜索算法在 A_0 点和 B_0 点之间相向地路径规划。交替式双向跳点搜索算法的步骤为:

- 步骤 1: 将 A_0 和 B_0 分别放进 $OpenList_A$ 和 $OpenList_B$ 中;
- 步骤 2: 寻找 $OpenList_A$ 中最小的点, 将其设置为 $current_A$;
- 步骤 3: $OpenList_A$ 删除 $current_A$ 节点, 在 $CloseList_A$ 中加入当前节点;
- 步骤 4: 根据跳点扩展规则, 寻找不在 $CloseList_A$ 中的跳点 A_1 ;
- 步骤 5: 判断 A_1 是否在 $OpenList_A$ 中, 若否则加入 $OpenList_A$, 若是则更新的值以及父节点的位置;
- 步骤 6: 广播 A_1 的位置;
- 步骤 7: 寻找 $OpenList_B$ 中最小的点, 将其设置为 $current_B$;
- 步骤 8: $OpenList_B$ 删除 $current_B$ 节点, 在 $CloseList_B$ 中加入当前节点;
- 步骤 9: 根据跳点扩展规则, 寻找不在 $CloseList_B$ 中的跳点 B_1 ;
- 步骤 10: 判断 B_1 是否在 $OpenList_B$ 中, 若否则加入 $OpenList_B$, 若是则更新的值以及父节点的位置;
- 步骤 11: 广播 B_1 的位置;
- 步骤 12: 判断是否搜索到一个公共重合节点, 若是找到完整路径, 并结束搜索, 若否则返回步骤 2。

在交替式双向跳点搜索算法中, 每进行一次跳点扩展, 都需要对新跳点的位置坐标进行广播。当另一侧进行跳点扩展时, 需以本侧广播的新跳点作为目标位置。

从 A_0 点一侧规划的路径, 在进行跳点扩展时, 其预计代价函数的计算公式如式 (6):

$$h(n) = \sqrt{(x_{B_cur} - x_{A_cur})^2 + (y_{B_cur} - y_{A_cur})^2} \quad (6)$$

从 B0 点一侧规划的路径, 在进行跳点扩展时, 其预计代价函数的计算公式如式 (7):

$$h(n) = \sqrt{(x_{A_cur} - x_{B_cur})^2 + (y_{A_cur} - y_{B_cur})^2} \quad (7)$$

式 (6) 和式 (7) 中, x_{A_cur} 、 y_{A_cur} 为 A0 点一侧当前跳点位置坐标, x_{B_cur} 、 y_{B_cur} 为 B0 点一侧当前跳点位置坐标。

得到中心热点区域内部的完整路径后, 将中心热点区域外部的路径与内部的路径拼接, 即可得到一条从起始点到目标点的完整路径。

3 局部路径优化

跳点搜索算法在进行路径规划时, 没有考虑移动机器人的体积, 因此规划的路径存在移动机器人与障碍物发生碰撞的风险。此外, 跳点搜索算法规划的路径由多个跳点栅格依次相连而成, 因此规划出来的路径具有不连续、拐角多、不平滑的缺陷。这不仅不符合机器人的运动学特征, 同时也给机器人带来安全隐患。针对上述问题, 提出了迭代式路径修正方法, 并采用 3 次 B-样条曲线对路径进行平滑处理。

3.1 迭代式路径修正

在基于跳点搜索算法的路径规划过程中, 根据跳点扩展的规则, 路径仅先保留了跳点的信息, 再将跳点依次相连, 继而得到从起始点到目标点路径。因此, 由跳点搜索算法规划的路径具有稀疏性, 要获得路径的完整坐标信息, 需要对路径跳点进行双线性插值^[21]处理。

相邻跳点之间的栅格数是不固定的, 因此将插值点间的插值步长设定为单个栅格的长度。双线性插值完成后, 得到一条从起始点到目标点的完整路径, 该条路径中每个栅格的坐标信息均已悉知, 双线性插值的公式如式 (8):

$$\begin{cases} y = \frac{x_1 - x}{x_1 - x_0}y_0 + \frac{x - x_0}{x_1 - x_0}y_1 \\ x = \frac{y_1 - y}{y_1 - y_0}x_0 + \frac{y - y_0}{y_1 - y_0}x_1 \end{cases} \quad (8)$$

式 (8) 中, 和是插值点两端跳点的坐标, 是插值点的坐标。

在获取全局路径的信息之后, 采用迭代式路径修正方法对存在安全隐患的局部路径进行修正, 迭代式路径修正方法的步骤为:

步骤 1: 循环遍历路径 route;

步骤 2: 判断路径 route 中第 k 个元素的上下左右邻域中是否存在障碍物 obstacle, 如果是执行步骤 3, 如果否退出当前路径修正;

步骤 3: 判断路径 route 中第 $k+1$ 个元素的上下左右邻域中是否存在步骤 2 中得到的障碍物 obstacle, 如果是执行步骤 4, 如果否则退出当前路径修正;

步骤 4: 根据式 (9), 计算路径转角周围需要检测栅格的位置;

$$\begin{aligned} P_1 &= \{(x_k, y_{k+1})\} \\ P_2 &= \{(x_{k+1}, y_k)\} \end{aligned} \quad (9)$$

式 (9) 中, x_k 和 y_k 是路径中第 k 个元素的横坐标和纵坐标, x_{k+1} 和 y_{k+1} 是路径中第 $k+1$ 个元素的横坐标和纵坐标。

步骤 5: 将路径修正到 P_1 或 P_2 对应的位置, 并返回步骤 1。

修正前路径如图 6 (a) 所示, 修正后的路径如图 6 (b) 所示。修正后的路径与障碍物的最短距离维持在半个栅格长度以上, 从而有效地避免了移动机器人在运行过程中与障碍物发生碰撞的问题。

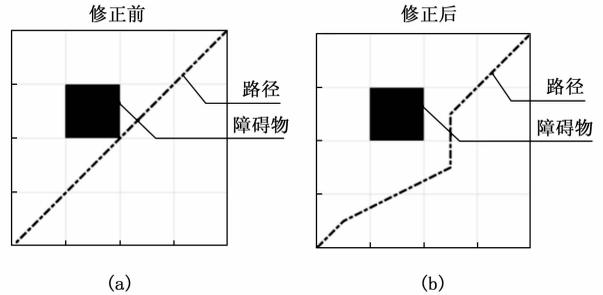


图 6 迭代式路径修正示意图

3.2 路径平滑

移动机器人若按照经过迭代式修正后的路径运行, 虽可以有效地避免与障碍物发生碰撞, 但是路径中拐角数量的增多限制了移动机器人的运行效率。在栅格地图中, 如果按照跳点搜索算法规划的路径运行, 机器人不得不在拐角处停下, 原地旋转至设定角度, 再继续向前运行, 这不符合机器人的运动特性。

非均匀 B-样条具有仿射不变性等优势^[22], 因此广泛应用于计算机辅助设计、制造等工程应用中。递推式 B-样条曲线的基函数的定义如下:

设 $U = \{u_0, u_1, \dots, u_n\}$ 为一不减的实数序列, 以 U 作为样条节点, 第 i 个 p 次 B-样条基函数 $N_{i,p}(u)$ 的定义如式 (10), 并约定, 当式 (10) 出现 0/0 形式的商时, 其比值为 0。

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & u \leq u \leq u_{i+1} \\ 0, & \text{其它} \end{cases} \\ N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{cases} \quad (10)$$

本文采用 3 次 B-样条曲线对路径进行平滑处理。平滑前后的路径对比如图 7 所示, 从图 7 中可以看到, 平滑后的路径对拐角进行了优化, 移动机器人按照平滑后的路径运行, 在拐角处可以柔滑地通过, 无需停下并原地旋转, 更加符合移动机器人的运动特性。

4 实验结果分析

为验证并行一交替式双向跳点搜索算法的有效性, 分别在栅格地图中放置不同比例的随机障碍物进行仿真实验。仿真使用的设备配置为: Windows10 操作系统, 四核 Core i5-6300 处理器, 主频为 2.3 GHz, 内存大小为 8 GB, 仿真运行平台为 MATLAB R2016a。

4.1 全局路径规划结果

在大小为 100×100 , 随机障碍物比例为 1.5% 栅格地图

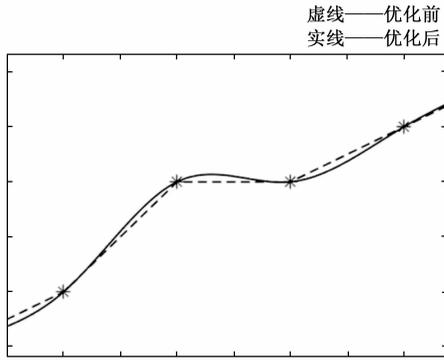


图 7 平滑前后的路径对比示意图

中,使用并行-交替式双向跳点搜索算法跳点搜索算法进行路径规划结果如图 8 所示。在图 8 中,中心热点区域外部的是由并行式跳点搜索算法规划的,A0 和 B0 分别是起始点侧和目标点侧的路径抵达中心热点区域的第一个跳点。A0 与 B0 之间的路径,是采用交替式双向跳点搜索算法规划的。对贴近障碍物的局部路径进行放大观察,优化后的路径(实线)与优化前的路径(虚线)相比,优化后的路径更好地兼顾了安全性和平滑性,按照优化后的路径运行,移动机器人不仅避免了与障碍物发生碰撞风险,同时也减少了在拐角处急停急起的次数,确保了移动机器人安全平稳运行。在中心热点区域外部,并行式双向跳点搜索算法无需进行信息交换与状态同步,在没有冗余路径的前提下,加快了路径规划的速度。在中心热点区域内部,交替式双向跳点搜索算法相向地规划到了重合的节点,得到了区域内部最短路径。将中心热点区域内的外的路径相连,得到从起始点到目标点的全局路径。

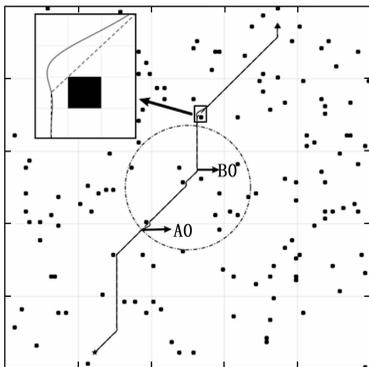


图 8 路径规划仿真结果

4.2 路径规划的时效性

不同随机障碍物比例下的路径规划时间见表 1。从表 1 可以看出,随着障碍物比例逐渐升高,A* 算法的路径规划时间基本保持不变,这是由于 A* 算法需要逐个计算路径中每个节点的信息,而障碍物比例升高对路径中节点数量的影响较小。对于跳点搜索算法而言,障碍物比例的升高导致了跳点数量的攀升,因此跳点搜索算法路径规划时间也随之增长。与跳点搜索算法相比,并行-交替式双向跳

点搜索算法在中心热点区域外部,采用并行的方式进行路径规划,有效地缩短了路径规划的时间。

表 1 不同障碍物比例下的路径规划时间 s

算法	障碍物比例			
	0.5%	1.5%	2.0%	3.0%
A* 算法	1.98	2.01	2.02	1.99
跳点搜索算法	0.45	0.75	0.92	1.01
PA-BJPS	0.30	0.43	0.48	0.53

4.3 路径规划长度

不同随机障碍物比例下的路径规划长度见表 2。从表 2 可以看出,在不同障碍物比例下,基于并行-交替式双向跳点搜索算法路径规划长度都略大于 A* 算法和跳点搜索算法。通过对全局路径位置信息进行逐一分析可知,路径长度增加是由于对局部路径进行修正和平滑造成的。相较于移动机器人可能存在的安全隐患而言,微量的路径长度增加几乎可以忽略不计。

表 2 不同障碍物比例下的路径规划距离 m

算法	障碍物比例			
	0.5%	1.5%	2.0%	3.0%
A* 算法	58.2	60.4	59.1	62.5
跳点搜索算法	58.2	60.4	59.1	62.5
PA-BJPS	59.6	62.1	61.2	63.6

4.4 路径的平滑性

路径的曲率 K 描述了移动机器人运行过程中法向加速度关于弧长参数的变化情况,曲率越大,路径越不平滑。根据式 (11),计算从起始点到目标点完整路径的曲率变化情况。

$$K = \frac{|y''|^2}{(1+y'^2)^{3/2}} \quad (11)$$

使用 3 次 B-样条曲线进行平滑处理前后,路径的曲率变化情况如图 9 所示。如图 9 (a) 所示,平滑处理前的路径曲率数值很大,其变化过程也十分尖锐。经过 3 次 B-样条曲线平滑处理后的路径曲率如图 9 (b) 所示,路径的最大曲率相较于处理前大幅度降低,变化的过程也比平滑前更加平缓,按照平滑处理后的路径运行,移动机器人无需在拐角处停下,避免了急停急起的问题。

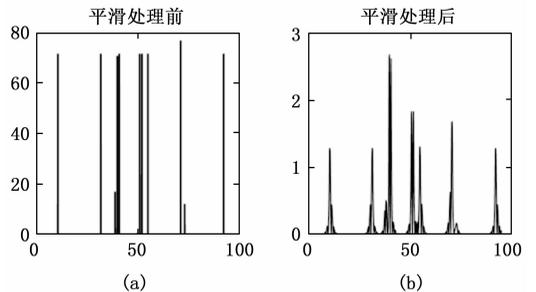


图 9 平滑处理前后曲率变化对比

5 结束语

针对在障碍物随机的栅格地图中,跳点搜索算法路径

规划时间较长的问题,本文提出了基于并行一交替式双向跳点搜索算法的机器人路径规划算法,并行式双向跳点搜索算法采用了并行运算的思想,无需进行信息交互和状态同步,加快了路径规划的速度。交替式双向跳点搜索算法每进行一次跳点扩展,需实时更新跳点的位置信息,以确保找到中心热点区域内部的最短路径。为避免机器人紧贴障碍物而产生的安全问题,本文提出迭代式路径修正方法,并使用3次B样条曲线对路径进行平滑处理。仿真实验结果表明,本文所提算法能够有效地缩短路径规划时间,同时所规划路径兼顾了安全性和平滑性,具有良好的学术研究和工程应用价值。

参考文献:

- [1] 宋晓茹,任怡悦,高嵩,等. 移动机器人路径规划综述[J]. 计算机测量与控制, 2019, 27(4): 1-6.
- [2] 林韩熙,向丹,欧阳剑,等. 移动机器人路径规划算法的研究综述[J]. 计算机工程与应用, 2021, 57(18): 38-48.
- [3] 齐少华,徐和根,万友文,等. 动态环境下的语义地图构建[J]. 计算机科学, 2020, 47(9): 198-203.
- [4] 李雅梅,张瑞,吕猛. 一种多机器人栅格地图拼接方法的研究[J]. 测控技术, 2020, 39(4): 58-61, 100.
- [5] 张标. 基于拓扑地图模型的空间运算研究[D]. 西安:长安大学, 2017.
- [6] 王洪斌,尹鹏衡,郑维,等. 基于改进的A*算法与动态窗口法的移动机器人路径规划[J]. 机器人, 2020, 42(3): 346-353.
- [7] 杨瑶,付克昌,蒋涛,等. 改进A*算法的智能车路径规划研究[J]. 计算机测量与控制, 2020, 28(10): 170-176.
- [8] 汪四新,谭功全,蒋沁,等. 基于改进A*算法的移动机器人路径规划[J]. 计算机仿真, 2021, 38(9): 386-389.
- [9] 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.
- [10] ZAFAR M A, ZHENG Z, WENKAI Y. Mobile Robots Path Planning based on A* Algorithm Improved with Jump Point Search [J]. International Bhurban Conference on Applied Sciences and Technologies (IBCAST), 2021. 536-544.
- [11] 王文明,杜佳璐. 基于正六边形栅格JPS算法的智能体路径规划[J]. 系统工程与电子技术, 2021, 43(12): 3635-3642.
- [12] 宋晓茹,任怡悦. 面向移动机器人快速全局路径规划的改进跳点搜索算法[J]. 科学技术与工程, 2020, 20(29): 11992-11999.
- [13] 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.
- [14] HARABOR D, &GRASTIEN A. Online Graph Pruning for Pathfinding On Grid Maps [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2011, 1114-1119.
- [15] TRAISH, JASON, TULIP, James, Moore, Wayne. Optimization Using Boundary Lookup Jump Point Search [J]. IEEE transactions on computational intelligence and AI in games, 2016, 8(3): 268-277.
- [16] 张洪海,李翰,刘啤,等. 城市区域物流无人机路径规划[J]. 交通运输系统工程与信息, 2020, 20(6): 22-29.
- [17] 王秀红,刘雪豪,王永成. 基于改进A*算法的仓储物流移动机器人任务调度和路径优化研究[J]. 工业工程, 2019, 22(6): 34-39.
- [18] 张庆,刘旭,彭力,等. 融合JPS和改进A*算法的移动机器人路径规划[J]. 计算机科学与探索, 2021, 15(11): 2233-2240.
- [19] 黄智榜,胡立坤,张宇,等. 基于改进跳点搜索策略的安全路径研究[J]. 计算机工程与应用, 2021, 57(1): 56-61.
- [20] 张子然,黄卫华,陈阳,等. 基于双向搜索的改进蚁群路径规划算法[J]. 计算机工程与应用, 2021, 57(21): 270-277.
- [21] 吉星宇,郝惠敏,杨凯,等. 基于双线性插值法的比例方向阀死区补偿方法[J]. 液压与气动, 2021, 45(6): 56-62.
- [22] 王振华,章卫国,李广文,等. 基于非均匀B样条的G2路径平滑方法[J]. 系统工程与电子技术, 2011, 33(7): 1539-1543.
- [12] 韩冰,徐敏,李广宁,等. 双三角翼及其翼身组合的滚转运动特性比较研究[J]. 航空学报, 2014, 35(2): 417-426.
- [13] 向锦武,曾开春,聂璐. 考虑弹性影响的乘波体飞行动力学特性[J]. 北京航空航天大学学报, 2012, 38(10): 1306-1310.
- [14] PEEBLES C. Eleven seconds into the unknown: a history of the hyper-x program [M]. 1801 Alexander Bell Drive, Reston; American Institute of Aeronautics and Astronautics, Inc, 2011: 199-219.
- [15] BAHM C, BAUMANN E, MARTIN J, et al. The X-43A hyper-x mach 7 flight 2 guidance, navigation, and control overview and flight test results [R]. Capua; AIAA, 2005.
- [16] BLOCKER W D, REUBUSH D E. X-43A stage separation system-a flight data evaluation [R]. Capua; AIAA, 2005.
- [17] TARTABINI P V, BOSE D M, THOMBLUM M N, et al. Mach 10 stage separation analysis for the X43-A [R]. Reno;

- [18] HURLEY M J, CARRIE G W. Stage separation of parallel-staged shuttle vehicles-a capability assessment [J]. Journal of Spacecraft and Rockets, 1972, 9(10): 764-771.
- [19] RONCOLI R B, LUDWINSKI J M. Mission design overview for the mars exploration rover mission [R]. Monterey; AIAA, 2002.
- [20] PETTEY T M. Ground handling and launching procedures for an alcohol-oxygen rocket missile [J]. Journal of Jet Propulsion, 1956, 26(6): 474-479.
- [21] BAKER W B, KEEN S, MORGRET C. Validation of weapon separation predictions using F/A-22 flight test results [R]. Woodland Hills; AIAA, 2004.
- [22] HU Z, ZHANG J, LUO X L. A novel design of efficient multi-channel uart controller based on FPGA [J]. Chinese Journal of Aeronautics, 2007, 20(1): 66-74.