



控制与决策

Control and Decision

ISSN 1001-0920, CN 21-1124/TP

《控制与决策》网络首发论文

题目: 非结构化场景下基于改进 JPS 算法的移动机器人路径规划
作者: 周熙栋, 张辉, 陈波
DOI: 10.13195/j.kzyjc.2022.1173
收稿日期: 2022-07-04
网络首发日期: 2022-10-25
引用格式: 周熙栋, 张辉, 陈波. 非结构化场景下基于改进 JPS 算法的移动机器人路径规划[J/OL]. 控制与决策. <https://doi.org/10.13195/j.kzyjc.2022.1173>



网络首发: 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式(包括网络呈现版式)排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

出版确认: 纸质期刊编辑部通过与《中国学术期刊(光盘版)》电子杂志社有限公司签约, 在《中国学术期刊(网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊(网络版)》是国家新闻出版广电总局批准的网络连续型出版物(ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

非结构化场景下基于改进JPS算法的移动机器人路径规划

周熙栋¹, 张辉^{2†}, 陈波²

(1. 长沙理工大学 电气与信息工程学院, 湖南 长沙 410000; 2. 湖南大学 机器人学院, 湖南 长沙 410000)

摘要: 针对移动机器人在大范围非结构化场景下的路径规划问题, 本文在改进跳点搜索(JPS)算法的基础上结合A*搜索, 提出一种基于分层栅格地图的Jump A*(JA*)路径规划算法. 该算法对三维点云地图进行栅格化分层处理, 将环境信息划分为结构层与非结构层, 并建立搜索策略切换规则, 依据图层信息使用不同的搜索策略, 从而有效减少计算量. 为了验证JA*算法的有效性, 在图层比例不同的三维地图中进行仿真, 仿真结果表明, JA*算法相比于传统的A*算法遍历节点更少, 搜索效率更高, 相比于双向A*算法, 具有更高的鲁棒性. 最后将JA*算法应用在公开数据集中, 实验结果表明, JA*算法能有效解决移动机器人在大范围非结构化场景下的路径规划问题.

关键词: 移动机器人; 非结构化场景; 多层栅格地图; A*算法; 跳点搜索

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.1173

引用格式: 周熙栋, 张辉, 陈波. 非结构化场景下基于改进JPS算法的移动机器人路径规划[J]. 控制与决策.

Mobile Robot Path Planning Based on Improved JPS Algorithm in Unstructured Scenarios

ZHOU Xi-dong¹, ZHANG Hui^{2†}, CHEN Bo²

(1. School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410000, China; 2. School of Robotics, Hunan University, Changsha 410000, China)

Abstract: In this paper, aiming at the path planning problem of mobile robots in a large range of unstructured scenes, this paper proposes a Jump A*(JA*) path planning algorithm based on hierarchical grid map, which combines the improved Jump Point Search (JPS) algorithm with A* search. In this algorithm, the 3D point cloud map is rasterized and stratified, and the environmental information is divided into structural layer and non-structural layer. In addition, the search strategy switching rules are established, and different search strategies are used according to the layer information, so as to effectively reduce the computational cost. In order to verify the effectiveness of JA* algorithm, simulation was carried out in 3D maps with different layer proportions. Simulation results show that compared with the traditional A* algorithm, JA* algorithm traverses fewer nodes, has higher search efficiency, and has higher robustness compared with the bidirectional A* algorithm. Finally, JA* algorithm is applied to the public data set, and the experimental results show that JA* algorithm can effectively solve the path planning problem of mobile robot in a large range of unstructured scenarios.

Keywords: Mobile robot; Unstructured scenarios; Multi-layer raster map; A* algorithm; Jump point search

0 引言

近年来移动机器人^[1]在不同的领域中得到了广泛的应用, 例如送餐机器人^[2]、扫地机器人^[3]及巡检机器人^[4]等. 随着移动机器人应用环境的不断扩展, 大范围非结构化场景中的自主导航技术逐渐成为研究热点之一, 而在三维点云地图中进行全局路径规

划目前仍是一项具有挑战的任务.

针对非结构化场景下的全局路径规划, 目前室外场景的全局路径规划算法主要包括智能仿生路径规划算法与传统路径规划算法^[5]. 其中智能仿生算法主要包括遗传算法^[6]、蚁群算法^[7]和粒子群算法^[8]智能仿生算法往往存在搜索效率低、收敛速度

收稿日期: 2022-07-04; 录用日期: 2022-10-10.

基金项目: 国家重大研究计划重点支持项目(92148204); 科技创新2030-“新一代人工智能”重大项目课题(2021ZD0114503); 国家自然科学基金(61971071, 62027810, 62133005); 和湖南省杰出青年科学基金项目(2021JJ10025); 湖南省重点研发计划(2021GK4011, 2022GK2011); 长沙市科技重大专项(kh2003026); 机器人学国家重点实验室联合开放基金(2021-KF-22-17); 中国高校产学研创新基金(2020HYA06006)

†通讯作者. E-mail: zhanghuihy@126.com.

慢以及易陷入局部最优等问题, 因此许多学者提出了各种优化方法. 文献[9]将蚁群算法应用在了水下机器人的动态避障和路径规划问题中, 通过聚类算法识别障碍物分布的复杂性, 并根据复杂程度实现搜索范围的自适应调整, 解决了经典蚁群算法中计算量大、求解速度慢的问题. 文献[10]基于粒子群算法提出三种优化算法, 分别针对快速收敛、随机交叉和精确搜索的需求, 设计了不同的适应性函数和搜索策略, 同时还提出跳出机制和重访机制, 解决了寻路过程中局部最优以及无效搜索的问题. 虽然智能仿生算法可以很好的应用在非结构化场景的全局路径规划中, 但相比于传统路径规划算法仍存在路径规划时间过长, 全局路径可能为非最优解的弊端.

传统路径规划算法主要包括人工势场法^[11]、A*算法^[12]、快速扩展随机树法^[13]和概率路标算法^[14]等. 文献[15]使用图形用户界面来对比研究这几种全局路径规划算法的路径长度、稳定性和收敛速度, 发现A*算法在复杂环境中的综合表现最好, 证明了A*算法对于最优路径的求解是最有效的. 但A*算法在大范围场景下进行最优路径求解时, 由于评价函数计算量增大, 导致规划时间过长, 搜索效率降低. 针对A*算法计算量过大的问题, 文献[16]通过改进A*算法中评价函数的具体计算方式减少计算时间, 再通过改进评价函数的权重比例, 减少生成路径中的冗余点和拐点. 针对A*算法的搜索效率低的问题, 文献[17]提出了双向自适应A*算法, 采用方向搜索策略来提高扩展的效率, 同时使用自适应步长和自适应权值策略提高算法的搜索速度. 文献[18]针对A*算法子节点搜索策略的弊端, 提出跳点搜索算法(JPS), 通过筛选跳点有效提高了路径搜索效率, 但搜索效率较高的JPS算法对于非结构化场景并不适用.

为解决全局路径规划算法在大范围的非结构化场景下, 存在计算量大, 搜索效率低等问题, 本文针对非结构化场景, 提出一种分层栅格地图表示方法, 对三维点云地图进行栅格化分层处理, 将环境信息划分为结构层与非结构层, 有效降低地图复杂度. 另外, 在分层栅格地图的基础上提出一种基于改进JPS算法的快速路径规划算法, 分区域的使用跳跃搜索和A*搜索, 从而减少三维点云地图中不必要节点的计算, 有效提高路径搜索效率. 最后将改进后的算法应用在公开数据集中, 并与传统A*算法对比, 验证了所提出方法在大范围非结构化场景下的有效性和优越性.

1 相关算法

1.1 A*算法

A*算法是目前在静态地图中求解最优路径最有效的直接搜索方法, 它结合了迪杰斯特拉算法和贪心算法, 即融合了广度优先和深度优先两种思想, 是一种启发式^[19]的搜索算法. A*算法使用两个状态表来存储寻路过程中所遍历的节点, 其中OpenList由待考察节点组成, CloseList由已考察节点组成. 每次搜索循环中, 从OpenList中选取最优节点添加到CloseList中, 并将该最优节点的子节点(邻接节点)存入OpenList中, 重复执行这一寻路过程, 直至终点被添加到OpenList中, 则判断寻路完成. A*算法通过代价函数评判节点的优劣, 其代价函数为

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

其中 $f(n)$ 是从起点经过节点 n 到达终点的代价估计. $g(n)$ 为实际代价, 即从起点到节点 n 实际走过的距离, 并在每次搜索循环中将其更新为更短距离, 从而保证从起点到节点 n 为最优路径. $h(n)$ 为估计代价, 即节点 n 到终点之间的估计直线距离. 当 $g(n) = 0$ 时, A*算法则变为贪心算法; $h(n) = 0$ 时, 则变为迪杰斯特拉算法, 如图1所示, 黑色栅格表示障碍物, 灰色栅格表示遍历的节点. 贪心算法虽然遍历节点少、计算速度快, 但路径不一定最优; 迪杰斯特拉算法遍历节点多、效率低, 但能搜索到最优路径; A*算法则在保证路径最优的情况下也有较高的搜索效率.

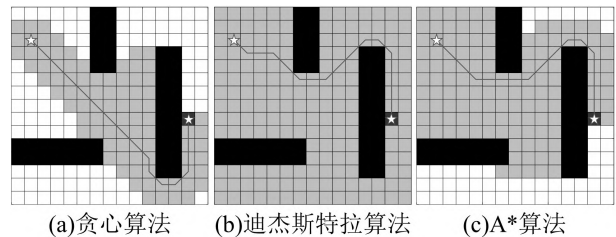


图1 不同算法的搜索效率对比

1.2 JPS算法

JPS算法在A*算法的基础上提出了新的子节点搜索策略, 将跳点作为A*算法中的子节点. 跳点就是在搜索过程中可以直接跳跃的节点, 两个跳点所组成路径上的中间节点不会被遍历, 从而有效减少了算法在寻路过程中的计算量和内存消耗. JPS算法首先需要定义节点 n 的自然邻节点与强制邻节点, 如图2和图3所示, 浅灰色栅格表示未被障碍占据的自由空间, 白色栅格表示节点 n 的自然邻节点, 深灰色栅格表示节点 n 的强制邻节点.

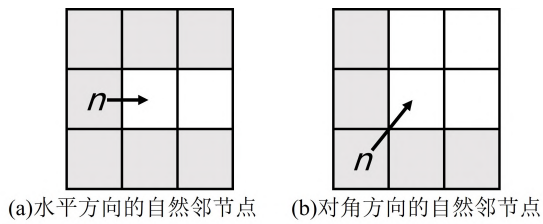


图2 自然邻节点搜索策略

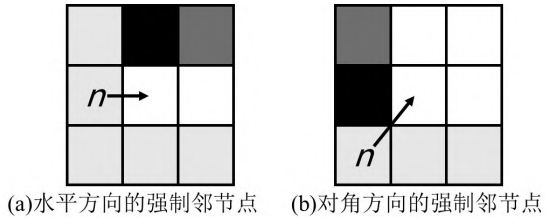


图3 强制邻节点搜索策略

在文献[20]中定义了跳点筛选规则, 对跳点进行搜索, 并删除跳点之外的不必要节点. 在JPS算法的跳点搜索循环中包括直线搜索与斜向搜索, 直线搜索是指对父节点 n 水平方向的自然邻节点依次进行扩展, 直到扩展到地图边界或者跳点; 斜向搜索是指对父节点 n 对角方向的自然邻节点进行扩展, 并判断该自然邻节点是否为地图边界或者跳点. 在每个父节点的跳点搜索循环中, 首先对未被扩展过的邻接节点方向进行直线搜索或斜向搜索, 如均未搜索到跳点, 则将斜向搜索到的邻接节点作为新的父节点, 循环以上操作, 直到搜索到跳点. 然后再执行A*算法的操作, 循环遍历子节点直到搜索到终点, 如图4所示, 灰色虚线表示已扩展过的区域, 灰色栅格表示跳点.

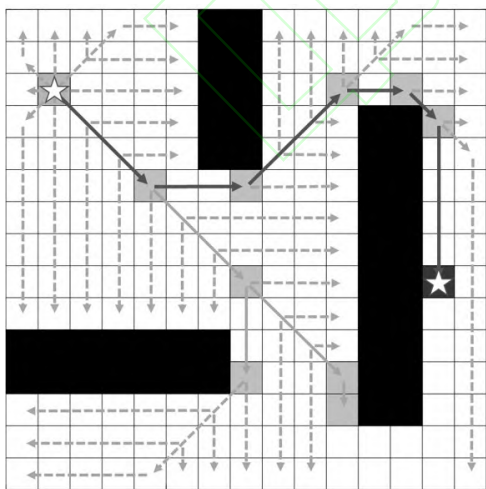


图4 JPS算法寻路过程

尽管JPS算法的跳点搜索策略避免了遍历不必要节点, 但是对于非结构场景, 即无法仅通过两个节点的二维坐标计算其之间距离, JPS算法仍然无法有效的提高路径搜索效率. 同时JPS算法的跳点筛选规

则依赖于障碍物, 对于无法对障碍物进行有效定义的三维点云地图, 传统的JPS算法并不适用.

2 改进规划算法

本文针对大范围的非结构化场景提出了一种快速路径规划算法, 该算法分为两个部分, 一是三维点云地图的栅格化分层处理, 二是基于改进JPS算法的路径搜索算法. 对于栅格化分层处理, 是将栅格化后的三维点云地图划分为结构层与非结构层, 即划分为平面区域与非平面区域. 路径搜索算法结合了A*算法中广度优先的思想以及JPS算法中节点跳跃的思想, 因此称其为Jump A*(JA*)算法. JA*算法在非结构层遍历所有邻接节点, 即A*搜索; 在结构层对跳点进行搜索, 即跳跃搜索. 因此保证了在路径规划中仅对必要节点进行遍历, 从而提高了路径搜索的效率.

2.1 环境模型构建

在移动机器人的三维路径规划中, 由于非结构化的三维场景无法使用传统的二维栅格地图来进行描述, 因此只能使用三维点云地图. 目前LIO-SAM^[21]、LeGO-LOAM^[22]等方法被广泛应用于3D激光SLAM技术, 其构建的稠密点云地图是一种能够详细表示周围环境信息的模型, 如图5所示. 为了能够在三维点云地图中进行快速路径规划, 我们首先需要对原始的三维点云地图进行降采样和插值拟合的预处理操作, 将三维点云地图转化为能被路径规划算法所使用的栅格化地图.

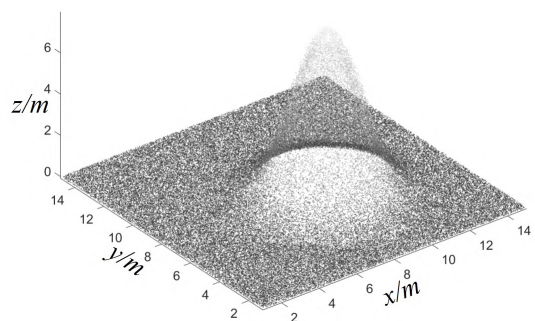


图5 原始点云地图

稠密点云地图中往往存在大量点云对同一区域的三维环境进行重复描述, 这些冗余的点云会造成路径规划算法中的不必要访问, 消耗过多的内存, 导致寻路效率降低, 因此我们需要对点云地图进行降采样预处理. 首先将三维点云数据按照 x 轴或 y 轴进行排序, 再每隔一定区间重采样一次数据, 可将同一区域中的冗余点云数据删除, 其所保留的三维点云数据仍能不失真的表示周围的环境信息, 如图6所示.

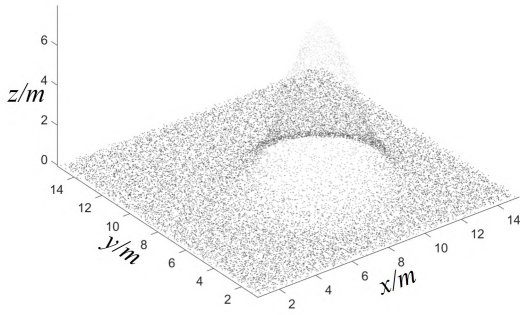


图6 点云降采样

由于非等距的离散点云在基于栅格法的三维路径规划算法中无法构建子节点搜索规则, 因此需要对三维点云地图进行插值处理, 将其转化为栅格化地图. 首先需要对 x 轴和 y 轴的坐标进行插值, 生成等间距的网格采样点, 并构建均匀分布的二维网格坐标, 再在该均匀网格上对离散的 z 轴坐标进行数据插值. 为了保证插值后的数据能尽可能的还原真实环境, 我们采用样条函数在规则网格上进行分段三维三次样条插值, 如图7所示.

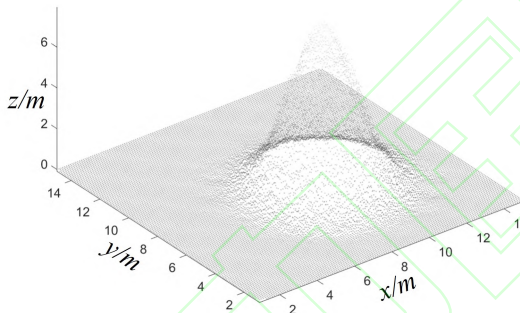


图7 等距插值栅格化

将栅格化后的三维地图分层处理首先需要定义标准平面高度, 即在水平方向的连续栅格中出现最多的高度数值, 将此高度下的区域定义为结构层, 一般情况下该区域为水平面. 在结构层中, 由于没有高度落差, 因此可以根据该平面内任意两点的二维坐标计算其之间的距离. 对于结构层以外的区域则定义为非结构层, 在非结构层中, 任意两点之间的距离无法直接计算, 而是必须在固定路径下累加相邻坐标之间的距离. 因此结构层仅保留了三维点云地图中的二维坐标数据, 非结构层在此基础上增加一个高度数据, 相比于二维栅格地图中的自由空间和障碍物, 分层栅格地图中的环境信息变为了平面信息与高度信息. 分层栅格地图如图8所示, 白色栅格为结构层, 灰色栅格为非结构层.

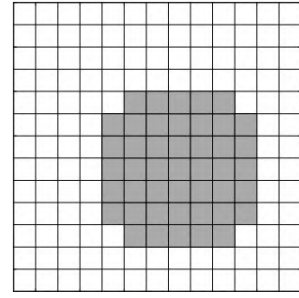


图8 分层栅格地图

为了路径规划算法能应用在不同的三维场景, 本文的研究与实验均在以下合理的前提条件中开展:

(1)通过预处理后的三维点云地图以及仿真生成的模拟地图分辨率均为1米; 将机器人视为质点, 地形不对机器人移动造成影响.

(2)移动机器人可向自身的八个方向移动, 即水平四个方向与对角四个方向, 本文中所描述的直线距离为两个三维节点之间的欧几里得距离, 映射距离为两节点在水平面上映射点之间的距离, 如图9所示. 图9(a)中实线箭头为直线距离, 虚线为节点在水平面上的映射关系, 图9(b)中虚线箭头为映射距离.

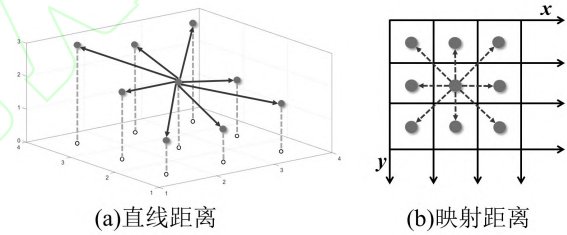


图9 距离代价规则

(3)路径规划过程中, 均为静态场景且不存在动态障碍.

2.2 改进算法

在分层栅格地图中, 传统JPS算法的跳点筛选规则并不适用, 因为它是通过寻找环境信息中的障碍物边界进行无碰撞路径规划, 而在分层栅格地图中, 环境信息为平面信息与高度信息, 因此JA*算法在分层栅格地图的基础上需要建立新的跳点筛选规则. 具体筛选条件如下:

$$L(\langle n, x_i, g \rangle) \leq L(\langle n, g \rangle) \quad (2)$$

其中函数 $L()$ 表示节点的映射距离, n 为跳跃搜索的父节点, g 为当前搜索方向上离 n 最近的非结构层节点, x_i 为 g 在结构层上的邻接节点, $L(\langle n, x_i, g \rangle)$ 表示路径 $n \rightarrow x_i \rightarrow g$, 该路径经过 g 的结构层邻接节点, $L(\langle n, g \rangle)$ 表示 n 与 g 之间的映射距离. 满足筛选条件的节点 x_i 即为跳点, 如图10所示, 深灰色栅格表示非

结构层, 浅灰色栅格表示跳点.

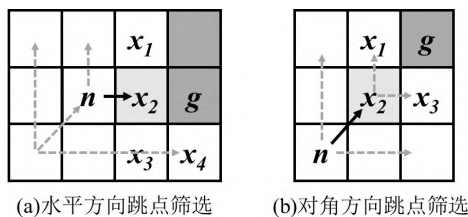


图 10 JA*跳点搜索策略

由于分层栅格地图的特性, 可以使JA*算法在结构层进行跳跃搜索, 为实现路径在结构层与非结构层之间的连接, 我们还需建立跳跃搜索和A*搜索的开始与结束规则, 即搜索算法的切换规则. 在跳跃搜索循环中, 当斜向搜索阶段搜索到跳点时, 结束跳跃搜索, 并将搜索到的跳点添加到OpenList中, 选取代价最小的跳点作为A*搜索的父节点. 在A*搜索循环中, 当A*搜索的子节点为未被扩展的结构层时, 结束A*搜索, 选取子节点 x_i 作为跳跃搜索的父节点, 并添加到OpenList中. 作为跳跃搜索父节点的 x_i 有以下筛选条件:

$$\min(L(\langle x_i, t \rangle) + \|x_i - n\|) \quad (3)$$

其中 t 为终点, $L(\langle x_i, t \rangle)$ 为 x_i 与 t 之间的映射距离, $\|x_i - n\|$ 为 x_i 与 n 之间的直线距离. 对每一个 x_i 进行求解, 选择计算值最小的 x_i 作为跳跃搜索的父节点.

通过以上建立的跳点筛选规则以及搜索算法的切换规则, JA*算法可在分层栅格地图中进行快速路径规划. JA*算法在结构层使用跳跃搜索算法, 在非结构层使用A*搜索算法, 避免了遍历不必要节点, 提高了搜索效率. JA*算法寻路过程如图11所示.

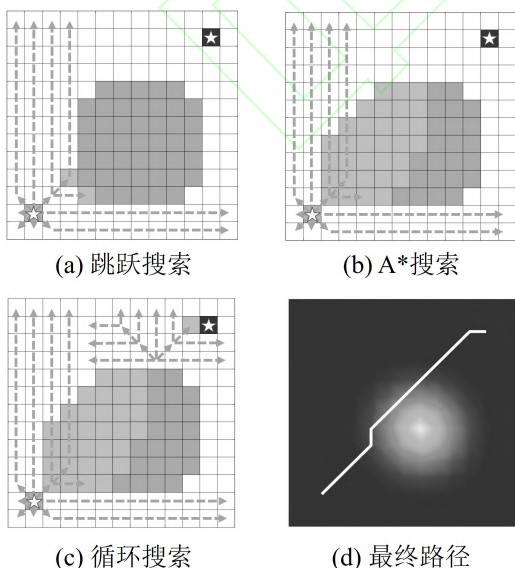


图 11 JA*搜索寻路过程

在JA*算法中, 首先将起点添加到OpenList中, 并判断起点位于什么图层, 示例中由于起点在结构层,

因此进行跳跃搜索, 向八个邻接节点方向进行扩展, 将搜索到的跳点添加到OpenList中, 并计算其代价如图11(a)所示。由于其中一跳点满足结束跳跃搜索的条件, 因此选取OpenList中代价最小的节点作为A*搜索的父节点, 并执行A*搜索, 如图11(b)所示。在A*搜索的节点扩展中, 搜索到满足结束A*搜索条件的节点, 将该节点的子节点作为跳跃搜索的父节点, 并执行跳跃搜索, 直到搜索到终点, 如图11(c)所示。JA*算法流程图如图12所示。

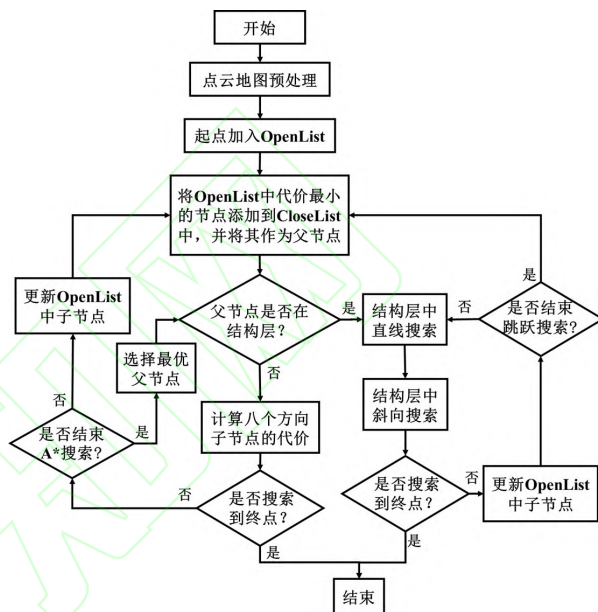


图 12 JA*搜索流程图

3 仿真及实验验证

3.1 仿真验证

为验证JA*算法的有效性,本文分别对传统A*算法、双向A*算法[17]和JA*算法进行仿真实验,仿真实验使用文献[23]中的环境建模算法,构建山峰高度、分布位置不同的三维地图.实验采用的三维地图规模均为 $100\text{m} \times 100\text{m}$ 、山峰数量均为10个,计算机配置为: Windows10操作系统, i5-1035处理器, 主频1.0GHz, 16G运行内存.

仿真实验场景选用四种山峰分布均不同的场景,其中起点坐标为(7, 7), 终点坐标为(90, 99), 如图13所示。实验结果如图14和图15所示, 我们选择两种典型山峰分布场景进行展示, 其中俯视图中的浅色区域为山峰, 灰色圆点表示被遍历过的节点, 白色折线表示最终生成的路径, 空心圆标记表示切换为A*搜索的节点, 矩形标记表示切换为跳跃搜索的节点。其他场景中的仿真结果如表1所示, 场景复杂程度通过非结构层与结构层的面积之比表示, 可以看出, JA*算法遍历节点数量最多减少67.5%, 搜索时间最多减少60.6%。

表 1 算法的搜索时间与扩展节点数量对比

场景 序号	复杂 程度	搜索时间			遍历节点数量			算法执行次数之比 跳跃搜索:A*搜索
		A*算法	双向A*算法	JA*算法	A*算法	双向A*算法	JA*算法	
1	1.61	3.23	1.88	3.18	4514	3019	3772	31:30
2	0.43	4.62	15.44	1.82	6134	16930	1989	2:1
3	0.45	3.98	3.79	2.46	4664	5509	2408	20:19
4	1.05	4.47	5.02	4.17	5700	7031	3980	30:29

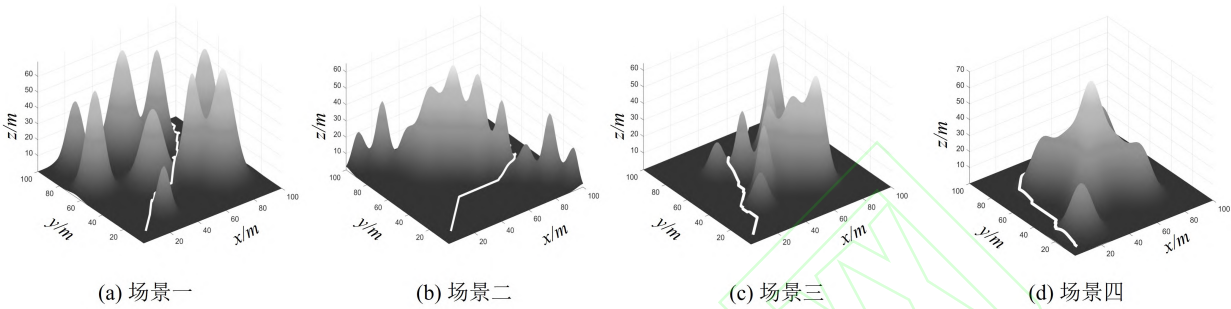


图 13 仿真实验场景

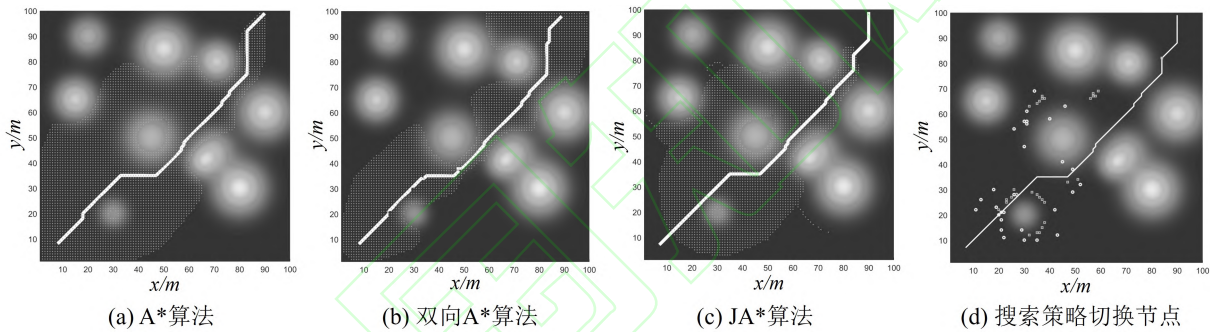


图 14 场景一下算法结果对比

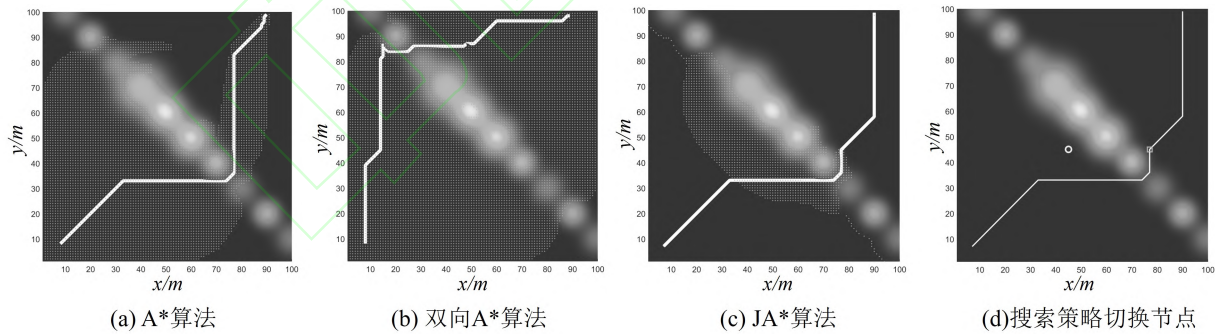


图 15 场景二下算法结果对比

仿真结果对比中可以看出，双向A*算法在场景一和场景三中性能优于A*算法，且在场景一中性能优于JA*算法，但在场景二和场景四中性能不及A*算法，且在场景二中规划效率大幅降低。这是由于JA*算法在非结构层分布较为稀疏的场景中，会大量的使用A*搜索，而双向A*算法中的反向规划会降低A*算法的搜索时间。但是双向A*算法在大范围的结构层中并不能有效提高搜索效率，其反向规划

会造成遍历更多的非必要节点，如图15(b)所示。因此双向A*算法在某些场景中虽然具有更高的搜索效率，但其并不能在复杂场景中保持高效的规划性能。在结构层占比较小的场景中，JA*算法的性能有所提升，当场景的结构层占比增大时，JA*算法所遍历的节点大大减少，性能有明显的提升，遍历节点数量减少一个数量级，且在非结构层分布更集中的场景中，搜索方式的切换次数更少，搜索效率也越高。同

时JA*算法规划的路径转折点更少, 这是由于两种规划算法均是路径长度最优算法, 且转折点的数量对路径的长度优劣并不造成影响, 而JA*算法中的跳跃搜索对非必要节点进行了跳跃, 因此JA*算法规划的路径相比于A*算法规划的路径在两个必要节点之间的转折点更少. 仿真实验结果表明, JA*算法不仅能减少遍历节点和转折点数量, 还能有效提高搜索速度, 且场景结构层占比越大, 性能提升越明显, 面向多种复杂场景具有更高的鲁棒性.

3.2 实验验证

为验证本文所提出JA*算法的有效性, 将JA*算法分别应用在公开数据集上, 由于双向A*算法并不具有适应复杂场景的规划性能, 因此仅对A*算法与JA*算法进行对比. 公开数据集采用武汉大学测绘遥感信息工程国家重点实验室发布的地面站扫描点云配准基准数据集(WHU-TLS), 我们选择其中的山地点云数据集, 该数据集包含了 19.61×10^7 个三维点, 场景大小为 $336\text{m} \times 312\text{m}$.对原始数据集进行降采样与插值操作, 将稠密点云地图转化为三维栅格化地图, 如图16所示. 预处理之后的地图分辨率为1m, 三维点数量为105481个.

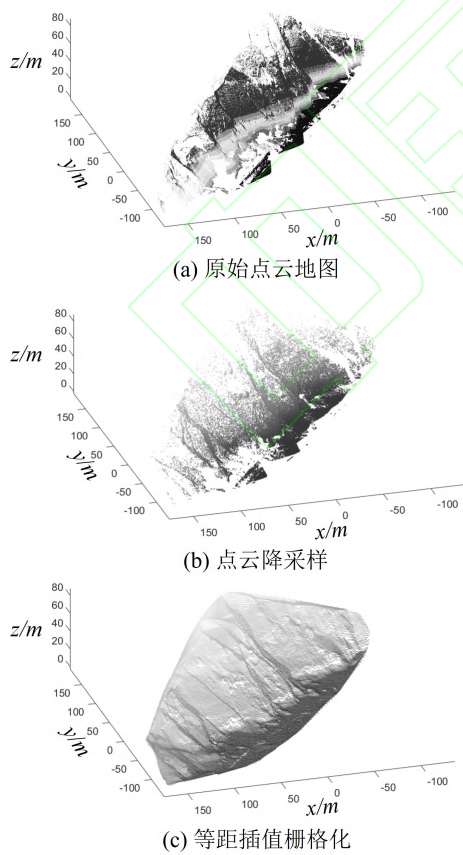


图 16 三维地图预处理

本文实验中, A*算法与JA*算法设置为同一起点和两个不同终点, 生成路径如图17和图18所示.

表2和表3分别列出两种算法对应的路径下的寻路时间与路径长度. 可以看出当起点与终点之间包含了大量坑洼和崎岖不平的路段时, 即并不包含大范围的非结构层时, 频繁的切换搜索策略会导致寻路效率降低, 因此这种路段仅使用A*搜索, 较少的使用跳跃搜索, 相比于A*算法性能有所提升, 搜索时间减少1.5%, 当起点与终点之间存在大范围非结构层时, JA*算法明显优于A*算法, 遍历节点数量减少30.7%, 搜索时间减少24.8%.

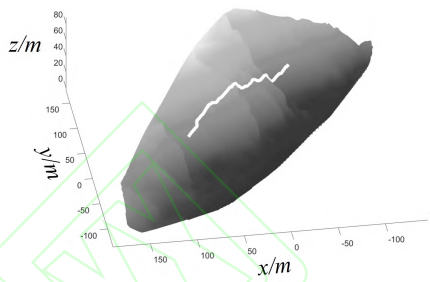


图 17 路径1

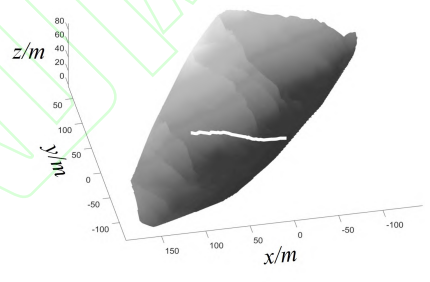


图 18 路径2

表2 路径1下两种算法对比

	A*算法	JA*算法
路径长度/m	210.38	210.38
搜索时间/s	10.03	9.87
遍历节点数量	9976	9960

表3 路径2下两种算法对比

	A*算法	JA*算法
路径长度/m	122.58	122.58
搜索时间/s	2.54	1.91
遍历节点数量	3301	2287

实验结果证明, 基于改进JPS算法的JA*算法能有效的实现三维地图中的路径规划任务, 且相比于A*算法, JA*算法优化了搜索策略, 算法占用内存更小, 寻路性能更好.

4 结 论

针对传统全局路径规划算法在大范围的非结构化场景下, 存在计算量大, 搜索效率低等问题, 本文结合A*算法与JPS算法的思想, 提出一种基于分层栅格地图的Jump A*(JA*)路径规划算法. 首先利用分

层栅格地图对三维环境进行描述, 将三维点地图按结构层与非结构层进行区域划分, 通过对JPS算法进行改进, 实现在分层栅格地图中进行跳跃搜索, 并建立搜索策略切换规则, 依据图层信息使用不同的搜索策略, 从而减少计算量, 提高搜索效率. 仿真实验结果证明, JA*算法在保证传统A*算法的性能前提下, 在结构层占比较大的场景中, 性能明显提升. 最后将JA*算法应用在公开数据集中, 实验结果表明, JA*算法寻路性能优于传统A*算法, 遍历节点数量明显减少, 搜索时间可减少24.8%, 能有效解决移动机器人在大范围非结构化场景下的路径规划问题.

参考文献(References)

- [1] Khan S and Daudpoto J. Development of Low-Cost Autonomous Robot[C]. 2019 Innovations in Intelligent Systems and Applications Conference(ASYU). Izmir, 2019: 1-6.
- [2] Bai J, Cai J, Zhou T, et al. Research on Key Technologies of Meal Delivery Robot Based on the fusion of Lidar and Machine Vision[C]. 2021 7th International Conference on Big Data and Information Analytics(BigDIA). Chong'qing, 2021: 236-242.
- [3] Zhou X, Wang W, Wang T, et al. Bayesian Reinforcement Learning for Multi-Robot Decentralized Patrolling in Uncertain Environments[J]. IEEE Transactions on Vehicular Technology, 2019, 68(99):11691-11703.
- [4] Zhang H, Hong W, Chen M. A Path Planning Strategy for Intelligent Sweeping Robots[C]. 2019 IEEE International Conference on Mechatronics and Automation(ICMA). Tian'jin, 2019: 11-15.
- [5] 王梓强, 胡晓光, 李晓筱, 等. 移动机器人全局路径规划算法综述[J]. 计算机科学, 2021, 48(10): 19-29.
(Wang Z Q, Hu X G, Li X X, et al. Overview of Global Path Planning Algorithms for Mobile Robots[J]. Computer Science, 2021, 48(10): 19-29.)
- [6] 李少波, 宋启松, 李志昂, 等. 遗传算法在机器人路径规划中的研究综述[J]. 科学技术与工程, 2020, 20(02): 423-431.
(Li S B, Song Q S, Li Z A, et al. Research review of genetic algorithm in robot path planning[J]. Science Technology and Engineering, 2020, 20(02): 423-431.)
- [7] 杨萍, 赵珍, 郑海霞. 基于改进蚁群算法的移动机器人全局路径规划方法研究[J]. 机械制造与自动化, 2017, 46(06): 155-157+192.
(Yang P, Zhao Z, Zheng H X. Research on Mobile Robot Global Path Planning Based on Improved Ant Colony Algorithm[J]. Machine Building and Automation, 2017, 46(06): 155-157+192.)
- [8] 胡章芳, 冯淳一, 罗元. 改进粒子群优化算法的移动机器人路径规划[J]. 计算机应用研究, 2021, 38(10): 3089-3092.
(Hu Z F, Feng C Y, Luo Y. Improved particle swarm optimization algorithm for mobile robot path planning[J]. Application Research of Computers, 2021, 38(10): 3089-3092.)
- [9] Liu X, Li Y, Zhang J, et al. Self-Adaptive Dynamic Obstacle Avoidance and Path Planning for USV Under Complex Maritime Environment[J]. IEEE Access, 2019, 7(99): 114945-114954.
- [10] Wang Y, Bai P, Liang X, et al. Reconnaissance Mission Conducted by UAV Swarms Based on Distributed PSO Path Planning Algorithms[J]. IEEE Access, 2019, 7(99): 105086-105099.
- [11] 于振中, 闫继宏, 赵杰, 等. 改进人工势场法的移动机器人路径规划[J]. 哈尔滨工业大学学报, 2011, 43(1): 50-55.
(YU Z Z, YAN J H, ZHAO J, et al. Mobile robot path planning on improved artificial potential field method[J]. Journal of Harbin Institute of Technology, 2011, 43(1): 50-55)
- [12] 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 and Cybernetics, 1968, 4(2): 100-107.
- [13] Wang J, Chi W, Li C, et al. Neural RRT*: learning-based optimal path planning[J]. IEEE Transactions on Automation Science and Engineering, 2020, 17(4): 1748-1758.
- [14] Akbaripour H, Masehian E. Semi-lazy probabilistic roadmap: A parameter-tuned, resilient and robust path planning method for manipulator robots[J]. The International Journal of Advanced Manufacturing Technology, 2016, 89 (5-8): 1401-1430.
- [15] 周滔, 赵津, 胡秋霞, 等. 复杂环境下移动机器人全局路径规划与跟踪[J]. 计算机工程, 2018, 44(12): 208-214.
(Zhou T, Zhao j, Hu Q X, et al. Global Path Planning and Tracking for Mobile Robot in Cluttered Environment[J]. Computer Engineering, 2018, 44(12): 208-214.)
- [16] 王中玉, 曾国辉, 黄勃, 等. 改进A*算法的机器人全局最优路径规划[J]. 计算机应用, 2019, 39(09): 2517-2522.
(Wang Z Y, Zeng G H, Huang B, et al. Global optimal path planning for robots with improved A* algorithm[J]. Computer application, 2019, 39(9): 2517-2522.)
- [17] Wu X, Xu L, Zhen R , et al. Bi-Directional Adaptive A* Algorithm Toward Optimal Path Planning for Large-Scale UAV Under Multi-Constraints[J]. IEEE Access, 2020, 8: 85431-85440.
- [18] Harabor D, Grastien A. The JPS pathfinding system[C]. 5th Annual Symposium on Combinatorial Search. Menlo Park, 2012: 207-208.
- [19] Korf R E. Real-time heuristic search[J]. Artificial Intelligence, 1990, 42(2/3): 189-211.
- [20] 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(06): 903-910.
(Zhao X, Wang Z, Huang C K, et al. Mobile Robot Path Planning Based on an Improved A* Algorithm[J]. Robot,

- 2018, 40(06): 903-910.)
- [21] Shan T, Englot B, Meyers D, et al. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping[C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS). Las Vegas, 2020: 5135-5142.
- [22] Shan T, Englot B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS). Madrid, 2019: 4758-4765.
- [23] 黄书召, 田军委, 乔路, 等. 基于改进遗传算法的无人机路径规划[J]. 计算机应用, 2021, 41(02): 390-397.
(Huang S Z, Tian J W, Qiao L, et al. Unmanned

aerial vehicle path planning based on improved genetic algorithm[J]. Journal of Computer Applications, 2021, 41(02): 390-397.)

作者简介

周熙栋 (1998—), 男, 硕士生, 从事移动机器人路径规划和定位等研究, E-mail: 741170053@qq.com;

张辉 (1983—), 男, 教授, 博士, 从事图像识别和深度学习等研究, E-mail: zhanghuihy@126.com;

陈波 (1997—), 男, 博士生, 从事多机器人区域覆盖搜索的研究, E-mail: cbc233@gs.zzu.edu.cn.