

控制与决策

Control and Decision

非平坦地形下移动机器人安全路径规划

黄志清, 李鼎鑫, 王庆文

引用本文:

黄志清, 李鼎鑫, 王庆文. 非平坦地形下移动机器人安全路径规划[J]. *控制与决策*, 2022, 37(2): 323–330.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.1221>

您可能感兴趣的其他文章

Articles you may be interested in

移动机器人运动规划中的深度强化学习方法

Deep reinforcement learning for motion planning of mobile robots

控制与决策. 2021, 36(6): 1281–1292 <https://doi.org/10.13195/j.kzyjc.2020.0470>

基于16方向24邻域改进蚁群算法的移动机器人路径规划

Mobile robots path planning based on 16–directions 24–neighborhoods improved ant colony algorithm

控制与决策. 2021, 36(5): 1137–1146 <https://doi.org/10.13195/j.kzyjc.2019.0600>

基于 $\text{pm}3\sigma$ 正态概率区间分族遗传蚁群算法的移动机器人路径规划

Path planning of mobile robot based on $\text{pm}3\sigma$ normal probability interval population division using genetic ant–colony algorithm

控制与决策. 2021, 36(12): 2861–2870 <https://doi.org/10.13195/j.kzyjc.2020.0745>

一种基于免疫机理的确定性移动机器人路径规划算法

A path planning algorithm of deterministic mobile robot based on immune mechanism

控制与决策. 2021, 36(10): 2418–2426 <https://doi.org/10.13195/j.kzyjc.2020.0059>

凸优化与A*算法结合的路径避障算法

Convex optimization and A–star algorithm combined path planning and obstacle avoidance algorithm

控制与决策. 2020, 35(12): 2907–2914 <https://doi.org/10.13195/j.kzyjc.2019.0351>

非平坦地形下移动机器人安全路径规划

黄志清^{1†}, 李鼎鑫¹, 王庆文²

(1. 北京工业大学 信息学部, 北京 100124; 2. 北京眸视科技有限公司, 北京 100085)

摘要: 提出一种基于双分辨率 2.5D 分层栅格地图的 Secure A* (SA*) 路径规划方法, 以解决移动机器人在非平坦地形下的安全路径规划问题. 首先, 设计一种双分辨率 2.5D 分层栅格地图, 利用双分辨率栅格对环境中的障碍物信息与高程信息进行存储, 以节约地图的存储空间; 然后, 结合移动机器人运动能力, 将环境中的高程信息转化为约束因子, 结合移动机器人尺寸, 以移动机器人到目标点的距离作为自适应因子, 引入 A* 算法的代价函数中, 以保证移动机器人在非平坦地形下的路径符合其运动能力; 最后, 通过仿真实验结果表明, 所提方案相比 3D 栅格地图下的传统 A* 算法, 可将地图存储空间减少 53.8%, 路径的高程标准差降低 63.9%, 可以有效地确保机器人在非平坦地下的安全.

关键词: 非平坦地形; 2.5D 地图; A* 算法; 移动机器人

中图分类号: TP242 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2020.1221

引用格式: 黄志清, 李鼎鑫, 王庆文. 非平坦地形下移动机器人安全路径规划 [J]. 控制与决策, 2022, 37(2): 323-330.

Safe path planning of mobile robot in uneven terrain

HUANG Zhi-qing^{1†}, LI Ding-xin¹, WANG Qing-wen²

(1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; 2. Beijing Elitenect Technologies Co., Ltd., Beijing 100085, China)

Abstract: In this paper, a Secure A* (SA*) path planning method based on a dual resolution 2.5D layered raster map is proposed to solve the problem of safe path planning for mobile robots in uneven terrain. Firstly, a dual-resolution 2.5D hierarchical grid map is designed, which uses the dual-resolution grid to store obstacle information and elevation information in the environment for saving the storage space of the map. Then, combined with the movement capabilities of the mobile robot, the elevation information in the environment is converted into a constraint factor. And considering the size of the mobile robot, the distance between the mobile robot and the target point is taken as an adaptive factor, and the cost function of the A* algorithm is introduced to ensure that the path of the mobile robot in uneven terrain conforms to its motion ability. Finally, the results of the simulation experiment show that compared with the traditional A* algorithm under the 3D raster map, this scheme can reduce the map storage space by 53.8% and the path elevation standard deviation by 63.9%, which can effectively ensure that the robot is able to be safe in uneven ground.

Keywords: uneven terrain; 2.5D map; A* algorithm; mobile robots

0 引言

随着移动机器人在各个领域的广泛应用, 非平坦地形(如农田、油田等环境)逐渐成为移动机器人的主要应用场景^[1]. 非平坦地形环境复杂, 这对移动机器人的路径规划方法提出了更多的挑战. 安全可靠的路径规划方法是非平坦地形下移动机器人安全导航的重要保障.

针对 3D 环境下的路径规划方法, 文献[2]列举了近几年中主要的 3D 环境中移动机器人的路径规划

算法, 其中基于数学模型的算法通过微分方程来描述所有约束条件, 可以提高算法的安全性和可靠性, 但这种方法给计算机带来了沉重的计算负担. 文献[3-4]通过模仿生物的行为对算法进行设计, 这种生物启发式方法可以很好地处理复杂和动态的非结构化环境. 但这种算法通过突变(迭代优化)来优化路径, 使得算法处理时间较长, 效率相对较低. 文献[5-6]是基于采样的算法, 该方法具有使用初始猜测的优点, 随机的初始猜测可确保逃避局部最小值, 并且这种算法

收稿日期: 2020-09-03; 录用日期: 2020-12-03.

基金项目: 国家自然科学基金项目(61502018).

责任编辑: 方勇纯.

[†]通讯作者. E-mail: zqhuang@bjut.edu.cn.

不太依赖环境的表示方式. 除此之外, 针对非平坦地形的路径规划常用的是基于3D栅格地图和高程地图的路径规划方法. 文献[7]提出了一种变维度状态空间的建图方法, 该方法在机器人周围的局部区域利用3D栅格地图进行建图, 其他区域使用2D栅格地图进行建图, 这种2D与3D相结合的建图方法, 即可以提高移动机器人局部路径的精度, 又可以保证全局路径的高效性. 文献[8]通过相机对室内的非平坦环境进行3D重建, 以解决带有楼梯与斜坡的3D环境中的移动机器人导航问题. 文献[9]针对非平坦地形, 利用激光雷达和车轮里程计的信息获得地面的高程等数据, 对非平坦地形进行3D栅格建图, 与2D栅格建图相比, 该方法可以更准确地感知周围的环境信息, 但3D栅格地图由于状态空间维数的增加必然导致路径规划时间和计算机内存需求的增加, 尤其是当环境尺寸增加时, 这个问题尤其突出. 为了节省地图存储所占用的空间, 一些学者提出用基于2D栅格地图的高程图法^[10-12], 利用2D栅格地图以及每一个栅格的高程信息组成2.5D栅格地图对环境进行建模. 虽然利用栅格存储单位面积内的高程信息会导致部分环境信息的缺失, 但可以显著降低数据存储空间并提高运算速度.

文献[13]提出了一种基于改进A*的启发式搜索方法, 利用高程地图在非平坦环境中寻找有效路径, 但其设计的启发式函数无法利用高程信息对路径的风险作出判断. 文献[14]通过计算移动机器人在高程地图上的能耗模型, 根据能耗模型提出了一种基于Dijkstra算法的路径规划方法, 该方法通过缩短路径长度以减少移动机器人的能源消耗, 容易产生局部最优解. 文献[15]实现了一种对环境高程信息进行分级的3D栅格地图上的路径规划算法, 对环境中的高程信息进行等级划分, 移动机器人每次移动只能在相同或者相邻等级的栅格间运动, 保证了移动机器人路径的相对平缓, 但是当环境高程差相差过大, 或分级不均匀时, 会直接导致移动机器人无法完成路径规划. 文献[16]提出了一种基于拓扑高程模型的室外三维环境建模和路径规划方法, 虽然该方案可以有效地构建环境并将相邻栅格间的高程差作为路径规划的约束条件之一, 但是由于栅格的高分辨率, 使得相邻栅格之间的距离在环境中只能表示几厘米, 不能根据移动机器人的运动能力进行调节, 因此不适用于其他尺寸的移动机器人.

综上所述, 非平坦地形下的移动机器人路径规划已具有一定的研究成果, 但目前的这些方法没有根据

移动机器人的运动能力和环境信息对路径进行优化, 以保证移动机器人在非平坦地形下的安全. 对此, 本文针对非平坦地形环境, 提出一种双分辨率2.5D分层栅格地图的建图方法, 利用双分辨率栅格对环境中的高程信息与障碍物信息进行分层存储, 可以节约移动机器人的存储空间, 提高运算效率. 另外, 在双分辨率2.5D栅格地图的基础上提出一种基于A*算法^[17-19]的路径规划方法, 该方法根据移动机器人的运动能力^[20]对其路径进行优化, 可以规划出一条保证移动机器人自身安全的路径.

1 双分辨率2.5D栅格地图建模

结合2D栅格地图与3D栅格地图的优缺点, 本文提出一种2.5D的建图方案, 称为Multilevel Map, 其是由多层栅格地图组成的一种复合型地图. 如图1所示, map A与map B为两层栅格地图, map A存储环境中的障碍物信息, map B存储环境中的高程信息. 每一个栅格对应一个节点, 所有的节点都在同一个坐标系下, 即投影到平面下的坐标一致. 例如: 通过函数`get.map[A]`, `map(1, 0)`可以获取到map A下(1, 0)节点的障碍物信息; 通过函数`get.map[B]`, `map(1, 0)`可以获取到map B下(1, 0)节点的高程信息, 通过这种办法可以得到地图中(1, 0)的环境信息. 移动机器人在路径规划的过程中可以对地图中的每一层信息进行选择性读取.

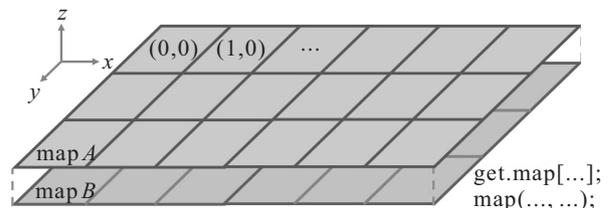


图1 2.5D分层地图

1.1 障碍物信息层

障碍物信息层利用占据栅格地图存储环境中的障碍物信息, 如图2所示. 白色栅格表示free状态,

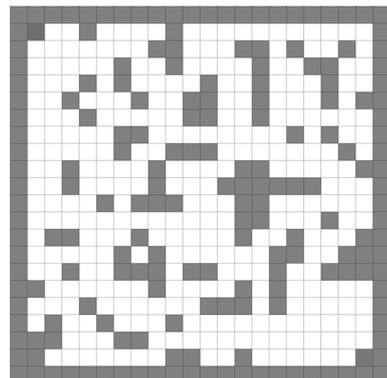


图2 障碍物信息层地图

机器人可以在白色栅格中自由移动;灰色栅格表示 occupied 状态,即有障碍物存在,机器人无法到达该区域。

1.2 高度信息层

高度信息层中的每个节点存储地图的高程信息,每一个节点中的高程值代表着一个范围(根据分辨率设定)内的高程平均值。图3所示是一张尺寸为 20×20 栅格的高度信息层地图,每个节点中存储单位面积的高程信息。比如,(3,4)节点中的高程值为30,表示在单位面积中偏离水平面的平均距离为30 cm。

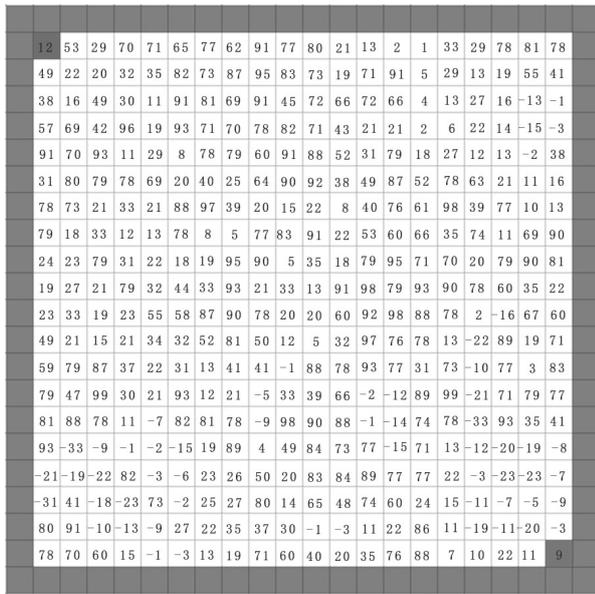


图3 高度信息层地图

1.3 双分辨率分层

对于大尺寸环境,利用高分辨建图可能使移动机器人在路径规划时超出其运算能力,无法快速地获得有效路径^[21]。因此,构建高分辨率的大地图是不可取的。同时,分辨率过大会导致移动机器人路径规划的精度降低,增加移动机器人碰撞障碍物的风险,因此适当的分辨率对大尺寸环境的路径规划有着重要的影响。使用双分辨率地图是一种很好的解决方法。障碍物信息层保证移动机器人能准确地识别障碍物的边界,以实现移动机器人的高效避障,分辨率可以设置的相对较小。为避免高度信息层分辨率过低导致大量高程信息丢失,影响移动机器人路径规划,本文中高度信息层的分辨率根据移动机器人的尺寸进行设定,通常选取移动机器人自身长度的二分之一作为高度信息层的分辨率大小,这样即节省了存储地图所需的空间,又提高了建图的效率。如图4所示,上层地图为障碍物信息地图,下层地图为高程信息地图,其分辨率比例为1:3。为方便理解,以下讨论中将使用分辨率为1:3的Multilevel Map地图进行说明。

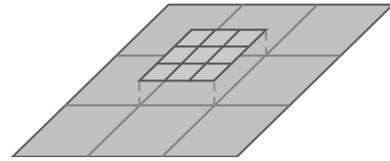


图4 双分辨率分层地图

2 Secure A*(SA*)算法

在双分辨率 2.5D 栅格地图的基础上,本文提出一种基于 A* 算法的安全路径规划算法,称其为 Secure A*(SA*)算法。传统 A* 算法的代价函数为

$$f(n) = g(n) + h(n).$$

其中: $g(n)$ 表示从起始节点到当前节点的真实消耗; $h(n)$ 表示从当前节点到目标节点的估算消耗,也称为启发值。

SA* 算法与传统算法 A* 不同的是:SA* 算法将传统 A* 算法代价函数中的 $g(n)$ 与 $h(n)$ 进行了重新定义,分别称为 $G_{cost}(n)$ 和 $H_{cost}(n)$ 。新的定义使移动机器人在路径规划时充分考虑环境中高程信息对路径的影响,从而实现一条符合机器人运动能力以及保证机器人自身安全的平缓路径。SA* 算法的代价函数为

$$F_{cost}(n) = xG_{cost}(n) + (1 - x)H_{cost}(n), \quad (1)$$

其中 x 是自适应调节因子,将在 2.4 节中介绍。

2.1 符号说明

为了表达方便,本文引入一些符号作为标记,这些符号的表示及其含义如表 1 所示。

表 1 符号说明

变量	变量描述
$P_\lambda(x_\lambda, y_\lambda)$	障碍物信息层节点坐标
$P_{h\lambda}(x_{h\lambda}, y_{h\lambda})$	高度信息层映射节点坐标
$P_{d\lambda}(x_{d\lambda}, y_{d\lambda})$	高度信息层投影节点坐标
θ_λ	坡度值
θ_{min}	机器人最小可爬坡度
θ_{max}	机器人最大可爬坡度
H_λ	节点高程信息
N_k	障碍物信息层邻近节点集合
M_k	高度信息层邻近节点集合
S_d	坡度距离
U_c	单位代价值
L_r	机器人长度

注 1 $\lambda = \{s, f, n, t\}$ 。其中: s 为起始节点, f 为父节点, n 为当前节点, t 为目标节点。

2.2 $G_{cost}(n)$ 的定义

2.2.1 双分辨率地图的映射关系

$G_{cost}(n)$ 是利用当前节点周围的高程信息进行辨别下一步的移动方向,因为障碍物信息层地图和高

度信息层地图使用的是双分辨率,所以在计算当前节点 P_n 的 $G_{\text{cost}}(n)$ 时需要先找到在高度信息层地图上相对应的映射节点 P_{hn} , 获取到其中的高程信息 H_{hn} 后再进行计算. 其映射关系通过算法1实现.

算法1 GetHeightMapPoint.

input: $P_f(x_f, y_f), P_n(x_n, y_n)$;

output: N_k .

- 1) function GetHeightMapPoint ($P_f(x_f, y_f), P_n(x_n, y_n)$)
- 2) if $P_f(x_f, y_f) = P_{df}(x_{df}, y_{df}) \& P_n(x_n, y_n) = P_{dn}(x_{dn}, y_{dn})$ then
- 3) $N_k \leftarrow \text{GetSurroundPoint}(P_f(x_f, y_f))$
- 4) $M_k \leftarrow \text{GetSurroundPoint}(P_{df}(x_{df}, y_{df}))$
- 5) return MappingRelation (N_k, M_k)
- 6) end if
- 7) if $P_f(x_f, y_f) = P_{df}(x_{df}, y_{df}) \& P_n(x_n, y_n) \neq P_{dn}(x_{dn}, y_{dn})$ then
- 8) $N_k \leftarrow \text{GetSurroundPoint}(P_f(x_f, y_f))$
- 9) $M_k \leftarrow \text{GetSurroundPoint}(P_{dn}(x_{dn}, y_{dn}))$
- 10) return MappingRelation (N_k, M_k)
- 11) end if
- 12) end function
- 13) function GetSurroundPoint ($P(x, y)$)
- 14) for $i = x - 1 \rightarrow x + 1$ do
- 15) for $j = y - 1 \rightarrow y + 1$ do
- 16) if IsCanreach (i, j) then
- 17) return N_k
- 18) end if
- 19) end for
- 20) end for
- 21) end function
- 22) function MappingRelation (N_k, M_k)
- 23) for $i = 1 \rightarrow i = k$ do
- 24) $N_k[i] = M_k[i]$
- 25) return N_k
- 26) end for
- 27) end function

将父节点 P_f 与当前节点 P_n 引入算法1中, 首先 GetHeightMapPoint 函数会判断父节点 P_f 与当前节点 P_n 的高度信息层投影节点 $P_{df}(x_{df}, y_{df})$ 与 $P_{dn}(x_{dn}, y_{dn})$ 是否为同一个节点. 如图5所示, 为了方便观察, 障碍物信息层只截取了一次迭代所要遍历的节点. 左图表示父节点 P_f 与当前节点 P_n 的高度信息层投影节点 $P_{df}(x_{df}, y_{df})$ 与 $P_{dn}(x_{dn}, y_{dn})$ 不

是同一个节点的情况, 右图表示两者的投影节点是同一个节点的情况. 对于前者情况, GetSurroundPoint 函数以节点 P_f 和节点 $P_{dn}(x_{dn}, y_{dn})$ 为中心, 以相同顺序去寻找周围节点, 并分别存入集合 N_k 和集合 M_k 中, 其中 k 值表示集合中节点的个数. 对于后者情况同理, GetSurroundPoint 函数以节点 P_f 和节点 $P_{df}(x_{df}, y_{df})$ 为中心寻找周围节点, 并分别存入集合 N_k 和集合 M_k 中. 最后利用 MappingRelation 函数将集合 N_k 和集合 M_k 中的节点相对应.

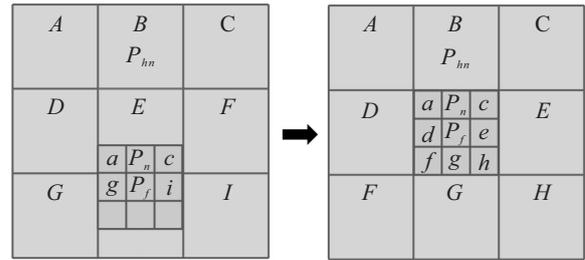


图5 双分辨率地图的映射关系

根据算法1可以得到: 图5中左图障碍物信息层节点 $\{a, c, g, i\}$ 对应的高度信息层的映射节点为 $\{A, C, G, I\}$; 图5中右图障碍物信息层中节点 $\{a, c, d, e, f, g, h\}$ 对应的高度信息层映射节点为 $\{A, C, E, F, G, H\}$, 此时 $N_k = \{A, C, E, F, G, H\}$, $k = 8$.

2.2.2 $G_{\text{cost}}(n)$ 的计算

图6是图5中右图的立体图, 计算当前节点 P_n 的坡度信息如下式所示:

$$\theta_n = \arctan\left(\frac{|H_{hn} - H_f|}{\sqrt{(x_{hn} - x_f)^2 + (y_{hn} - y_f)^2}}\right). \quad (2)$$

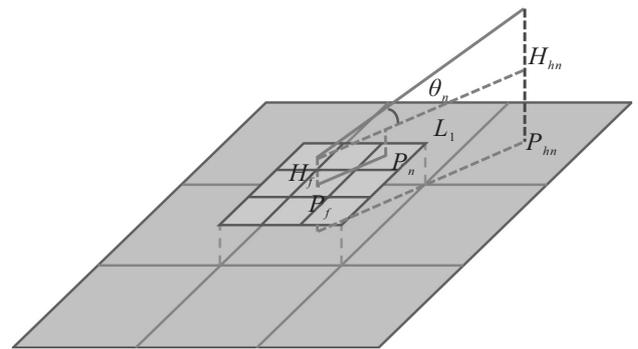


图6 当前节点的坡度

根据文献[14]提出的轮式机器人运动能力的性能指标, 本文以最大可爬坡度与最小可爬坡度作为移动机器人运动能力的性能指标. 设移动机器人最大可爬坡度为 θ_{max} , 即大于 θ_{max} 的坡(坑)机器人无法通过; 最小可爬坡度为 θ_{min} , 即小于 θ_{min} 的坡(坑)并不影响移动机器人的正常移动, 可以忽略不计. 考虑到移动机器人的运动能力, 对 θ_n 进行处理, 有

$$\theta_n = \begin{cases} \theta_{n_1}, & |\theta_n| \leq \theta_{\min}; \\ \theta_{n_2}, & |\theta_{\min}| < \theta_n \leq \theta_{\max}; \\ \theta_{n_3}, & |\theta_n| > \theta_{\max}. \end{cases} \quad (3)$$

其中: θ_{n_1} 、 θ_{n_2} 、 θ_{n_3} 在数据上等于 θ_n ; 在优先级上, $n_1 > n_2, n_3$ 不予考虑。

定义1 n_a 是周围正方向节点集合, n_b 是斜方向节点集合, 且优先级上, $n_a > n_b$, 则有 $N_k = n_a \cup n_b$. 将算法1中输出的 N_k 集合, 采用算法2对其进行优先级排序。

算法2 SortSurroundPoint.

input: N_k ;

output: N_k .

1) function SortSurroundPoint N_k

2) JudgmentPriority (N_k)

3) for $i = 1 \rightarrow i = k$ do

4) if $N_k[i] \in n_a \& N_k[i].\theta_i \in \theta_{n_1}$ then

5) $n_c \leftarrow N_k[i]$

6) Delete($N_k[i]$)

7) if $N_k[i].\theta_i \in \theta_{n_2}$ then

8) $n_b \leftarrow (N_k[i].\theta_i - \theta_{\min})$

9) end if

10) end if

11) end for

12) Sort(n_c)

13) Sort(n_b)

14) $N_k \leftarrow \text{Splice}(n_c, n_b)$

15) end function

16) function JudgmentPriority N_k

17) for $i = 1 \rightarrow i = k$ do

18) if $N_k[i].\theta_i \leq \theta_{\min}$ then

19) $N_k[i].\theta_i \in \theta_{n_1}$

20) end if

21) if $N_k[i].\theta_i \geq \theta_{\max}$ then

22) $N_k[i].\theta_i \in \theta_{n_3}$

23) Delete($N_k[i]$)

24) else

25) $N_k[i].\theta_i \in \theta_{n_2}$

26) end if

27) end for

28) return N_k

29) end function

算法2的实施流程是: 首先判断集合 N_k 中节点的优先级, 按照式(3)将节点的坡度值分为 n_1 、 n_2 、 n_3 三级, 并将集合 N_k 中优先级为 n_3 的节点删除。然后

将集合 N_k 中剩余的节点根据位置关系划分为 n_a 和 n_b 两个子集. 对于集合 n_a 中的节点, 如果节点的优先级为 n_1 , 则将其保存到一个集合 n_c 中, 并将该节点在集合 n_c 中删除. 如果集合中节点的优先级是 n_2 , 则将该节点的 θ_{n_2} 进行 $\theta_{n_2} - \theta_{\min}$ 处理后加到集合 n_b 中, 对集合 n_c 和 n_b 中的节点按照值升序排序. 最后将集合 n_c 和 n_b 中的节点首尾相接返回到集合 N_k 中, 完成对集合 N_k 中节点的排序工作。

图7是用算法2对父节点周围节点进行优先级排序的一个简单例子. 为了直观, 截取高度信息层中的 3×3 个栅格大小, 并且对应的障碍物信息层节点都是 free 状态(可到达)。

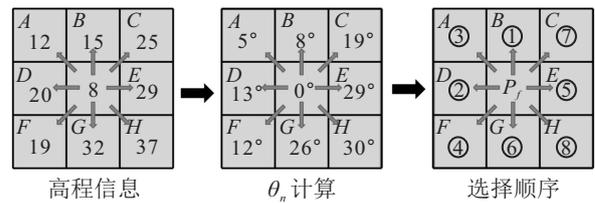


图7 高程信息选择顺序图

在这个例子中, 设定机器人的最小可爬坡度 $\theta_{\min} = 10^\circ$, 最大可爬坡度 $\theta_{\max} = 30^\circ$. 每一个节点的高程信息通过式(2)计算得到对应的 θ_n , 此时 $N_k = \{A, B, C, D, E, F, G, H\}, k = 8$. 算法2中 SortSurroundPoint 函数首先对集合 N_k 中的节点进行优先级判断可以得出: 节点B的优先级为 n_1 , 其他节点的优先级为 n_2 , 无优先级为 n_3 的节点. 然后对集合 N_k 中的节点进行分类得到 $n_a = \{B, D, E, G\}, n_b = \{A, C, F, H\}$, 对节点的 θ 值进行优先级排序, 可以看到节点B的 $\theta_B = 8^\circ$, 在 n_a 最小并且优先级为 n_1 , 所以优先级最高. 将节点B放入 n_c 集合中并在 n_a 删除, 因为 N_c 集合只有B一个节点所以无需排序, 其他 n_a 中的节点 θ 值都大于 θ_{\min} 小于等于 θ_{\max} , 所以优先级属于 n_2 , SortSurroundPoint 函数会将 N_a 集合中现有的节点的 θ 进行 $\theta_{n_2} - \theta_{\min}$ 处理, 即 $\theta_{n_2} - 10^\circ$, 然后加入到 n_b 集合中, 此时参数 $\theta_D = 3^\circ, \theta_E = 19^\circ, \theta_G = 20^\circ, n_b = \{A, C, F, H, D, E, G\}$. 根据 n_b 中各节点的 θ 值再次排序得到 $n_b = \{D, A, F, E, G, C, H\}$. 最后将 n_c 和 n_b 集合按顺序加入 N_k 集合得到 $N_k = \{B, D, A, F, E, G, C, H\}$, 排序完成. 完成排序后, 需要对节点进行 cost 值的赋值。

定义2 集合 $B = 1, 2, \dots, m$, 集合 S : cost 标准库; $f : \{N_k\}_{k=1}^m \rightarrow S$, 其中 $S \triangleq \bigcup_{B_0 \subset B} B_0, m = 2U_c - 1$.

利用下式对 N_k 中节点的 cost 值进行赋值:

$$f(N_k) = \begin{cases} \left\{ \left\lfloor \frac{m}{2} \right\rfloor + 1 - \frac{k-1}{2}, \dots, \left\lfloor \frac{m}{2} \right\rfloor + \frac{k-1}{2} \right\}, \\ k \in 2\mathbf{Z} + 1; \\ \left\{ \left\lfloor \frac{m}{2} \right\rfloor - \frac{k}{2} + 1, \dots, \left\lfloor \frac{m}{2} \right\rfloor, \left\lfloor \frac{m}{2} \right\rfloor + 2, \dots, \left\lfloor \frac{m}{2} \right\rfloor + \frac{k}{2} + 1 \right\}, k \in 2\mathbf{Z}. \end{cases} \quad (4)$$

其中 $[a]$ 为高斯函数。

$$G_{\text{near}}(n) = f(N_k(n)),$$

则由图6可得, $k = 8$. 设 $U_c = 5$, 则 $m = 9$, $B = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $G_{\text{near}}(n) = \{1, 2, 3, 4, 6, 7, 8, 9\}$. SA* 算法中的 $G_{\text{cost}}(n)$ 计算方式如下:

$$G_{\text{cost}}(n) = G_{\text{cost}}(n-1) + G_{\text{near}}(n).$$

2.3 $H_{\text{cost}}(n)$ 的定义

传统 A* 算法常用的距离计算方式有曼哈顿距离 (Manhattan distance)、欧氏距离 (Euclidean distance) 等. 为了使路径整体较为平缓, 本文提出一种新的距离计算方式——坡度距离 (slope distance), 简称 S_d . 如图8所示, L 表示欧氏距离, 因为高度信息层地图的存在, 可以求出图中的 S_d 距离, 这个距离在实际地图中比欧氏距离更加接近真实值.

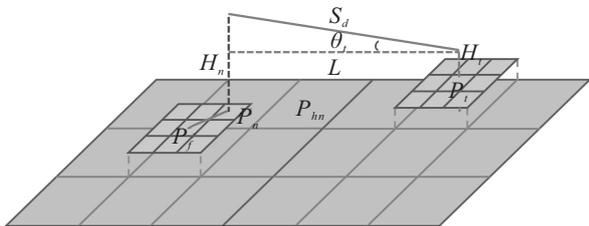


图8 坡度距离

S_d 的计算方式为

$$S_d = (|H_t - H_n|) / \sin \theta_t, \quad (5)$$

其中 $\theta_t = \arctan\left(\frac{|H_t - H_n|}{\sqrt{(x_t - x_n)^2 + (y_t - y_n)^2}}\right)$, 则 SA* 算法中的 $H_{\text{cost}}(n)$ 计算方式为

$$H_{\text{cost}}(n) = S_d \times U_c.$$

2.4 自适应因子

为了提高算法的效率与精确度, 本文在式(1)中加入一个比例调节因子 x , 用于移动机器人在规划路径时的自适应调节. 算法可以根据比例调节因子 x 改变代价函数中 $G_{\text{cost}}(n)$ 与 $H_{\text{cost}}(n)$ 的比重, 减少路径中的冗余点和无用的区间搜索, 以提高算法的效率和准确率. 比例调节因子 x 的值根据当前节点到目标节点的坡度距离 S_d 进行变化. 当 S_d 很大时, 说明移动机器人距离目标节点较远, 此时可以稍微增加 $H_{\text{cost}}(n)$ 的权重比例, 减少路径搜索的范围, 以实现算法加速的目的. 随着 S_d 的减小, 可以慢慢减小

$H_{\text{cost}}(n)$ 的权重比例, 增加 $G_{\text{cost}}(n)$ 的权重比例, 提高算法的准确性. S_d 与 x 的对应关系根据地图的尺寸和机器人自身的尺寸进行设定, 如下所示:

$$x = \begin{cases} \alpha, & |S_d| \leq uL_r; \\ \frac{(\beta - \alpha)(S_d - L_r)}{(v - u)L_r} + \beta, & |uL_r| < S_d < |vL_r|; \\ \beta, & |S_d| \geq |vL_r|. \end{cases} \quad (6)$$

3 实验结果及分析

为了验证所提出的基于双分辨率 2.5D 分层地图的安全路径规划方法的有效性, 实验中随机生成 50 幅地图进行仿真实验. 实验中需要设定的参数如表 2 所示.

表2 参数设定

参数名称	设定值
机器人最小可爬坡度 $\theta_{\min}/(^{\circ})$	10
机器人最大可爬坡度 $\theta_{\max}/(^{\circ})$	30
机器人长度 L_r/cm	30
单位代价值 U_c	5
机器人长度最小倍数 u	5
机器人长度最大倍数 v	10
比例系数 α	0.4
比例系数 β	0.6
地图分辨率比值	1:3

首先, 针对不同尺寸的环境分别进行 2D、2.5D 和 3D 栅格建图以比较不同建图方式下的存储空间大小, 如图9所示. 其中: 2D 和 3D 栅格地图的分辨率都为 $0.1 \text{ m} \times 0.1 \text{ m}$, 2.5D 栅格地图的障碍物信息层分辨率为 $0.1 \text{ m} \times 0.1 \text{ m}$, 高度信息层分辨率为 $0.3 \text{ m} \times 0.3 \text{ m}$.

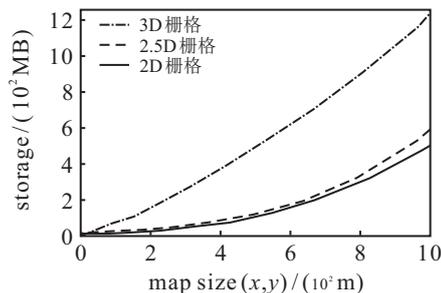


图9 地图面积与存储空间的关系

当地图面积为 $1000 \text{ m} \times 1000 \text{ m}$ 时, 3D 栅格的存储空间为 1254.7 MB, 2.5D 栅格地图存储空间为 579.2 MB, 2D 栅格地图存储空间为 490.4 MB. 相比之下, 2.5D 栅格地图的存储空间比 3D 栅格地图减少了 53.8%, 比 2D 栅格地图增长了 18.1%. 随着建图面积的不断增大, 3D 栅格地图所需要的存储空间明显高于其他两种建图方式, 2.5D 栅格地图和 2D 栅格地图所需存储空间增幅较为平缓, 且 2.5D 栅格地图相比

2D栅格地图所需的存储空间增幅不大.

为方便观察,实验中随机生成50幅地图尺寸大小为20×20个栅格的2.5D栅格地图,其障碍物信息层栅格分辨率为0.1m×0.1m,高度信息层栅格分辨率为0.3m×0.3m.为了验证所提出路径规划方法的有效性,实验中在同一环境下对比SA*算法与2DA*算法和3DA*算法在路径高程变化上的差异,仿真计算的结果如表3所示.从表3可以看出,本文所提出的SA*算法在规划时间和扩展点上与2DA*几乎一样,但整体路径高程标准相比传统A*算法降低了63.9%,而所得路径的真实消耗并没有增加太多.因为传统的A*算法在规划的过程中并不会考虑高程对路径的影响,所以尽管3DA*增加了扩展点的数量,但是最终路径和2DA*并无明显差异,反而效率降低.

表3 仿真计算结果

算法名称	时间/ms	扩张点	路径长度	高程均值	高程标准差
2DA*	4.94	88	28	16.5247	30.3051
3DA*	47.86	829	31	16.4641	30.2527
SA*	5.14	92	35	15.3056	10.9174

传统A*算法与SA*算法的路径高程变化情况如图10所示.可以看出:SA*的路径整体高程波动并不大,波动区间为[-20,40],整体高程较为平稳;传统A*算法路径高程起伏较大,波动区间为[-40,80].

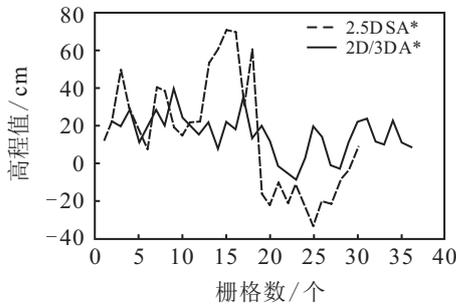


图10 传统A*算法与SA*算法的路径高程变化

实验中,取其中一组实验数据进行分析.为了方便观察,将高度信息层中的高程信息与障碍物信息层合并,图11是传统A*算法的路线,图12是SA*算法的路线.对比发现,SA*算法的路径与传统A*算法的路径并不相同,在坐标(11,7)处,SA*算法与传统A*选择了不同路径.传统A*所选取的(12,8)与(13,8)节点的高度差为31cm,SA*算法所选择的(12,8)与(12,9)节点的高度差为14cm,明显低于传统A*算法选择的路径.此外,传统A*算法所选择的路径高程变化极大,例如(17,9)与(16,9)节点之间的高度变换为50cm,坡度达到59.04°,已经远远超出了移动机器人的最大可爬坡度 θ_{max} ,即已经超出了移动机器人

的运动能力,无法保证移动机器人的安全行进.反观SA*算法的路径更加合理,路径高度更加平缓,可以保证移动机器人在该环境下的安全.

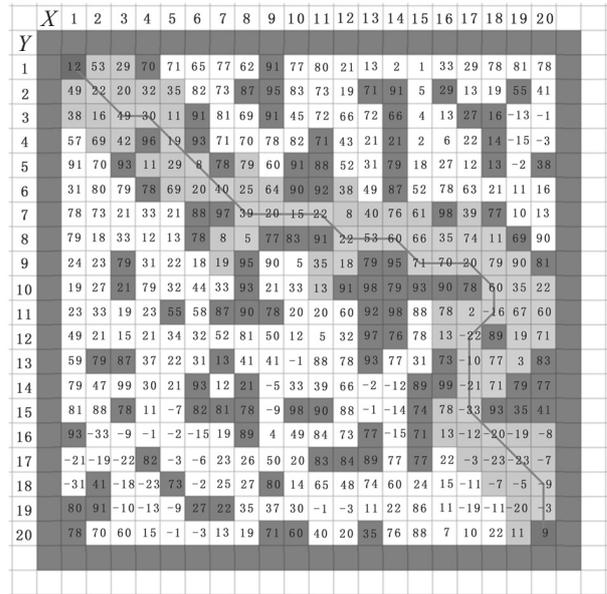


图11 2DA*算法路径

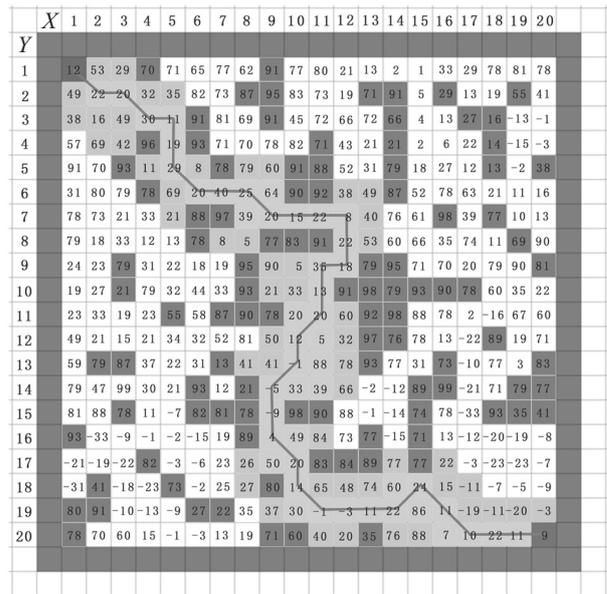


图12 SA*算法路径

4 结论

针对非平坦地形,本文提出了一种基于2.5D栅格地图的安全路径规划方法.首先,利用双分辨率栅格对环境中的高程信息与障碍物信息进行分层存储,使用高分辨率栅格存储障碍物信息,利用低分辨率的栅格存储高程信息,相比3D栅格地图可以节约53.8%的存储空间.在此基础上,提出了一种保证移动机器人安全的路径规划方法,该方法根据移动机器人的运动能力对路径进行优化,使得路径的高程标准差相比传统A*算法降低了63.9%,路面坡度变化更加平缓,保证了移动机器人在非平坦环境下的安

全. 同时,算法中加入了自适应因子,可以减少路径中的冗余点和无用的区间搜索,以提高算法的效率和准确率. 最后,通过仿真实验证实了所提出的规划方法的有效性,该方法能够快速准确地实现一条保证机器人自身安全的路径.

参考文献(References)

- [1] Ganganath N, Cheng C T, Fernando T, et al. Shortest path planning for energy-constrained mobile platforms navigating on uneven terrains[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(9): 4264-4272.
- [2] Yang L, Qi J T, Song D L, et al. Survey of robot 3D path planning algorithms[J]. *Journal of Control Science and Engineering*, 2016, 2016: 1-22.
- [3] Pu X C, Xiong C W, Ji L H, et al. 3D path planning for a robot based on improved ant colony algorithm[J]. *Evolutionary Intelligence*, DOI: <https://doi.org/10.1007/s12065-020-00397-6>.
- [4] Wang L F, Kan J M, Guo J, et al. 3D path planning for the ground robot with improved ant colony optimization[J]. *Sensors*, 2019, 19(4): 815.
- [5] Li M, Sun Q P, Song Q, et al. Path planning of mobile robot based on RRT in rugged terrain[C]. *Proceedings of the 2nd International Conference on Computer Science and Application Engineering*. Hohhot, 2018: 1-5.
- [6] Li X L, Jiang H X, Shi W Q, et al. Path planning of multipoint region attraction RRT* algorithm in complex environment[C]. *2019 Chinese Control Conference (CCC)*. Guangzhou, 2019: 4409-4414.
- [7] 张浩杰, 龚建伟, 姜岩, 等. 基于变维度状态空间的增量启发式路径规划方法研究[J]. *自动化学报*, 2013, 39(10): 1602-1610.
(Zhang H J, Gong J W, Jiang Y, et al. Research on incremental heuristic path planner with variable dimensional state space[J]. *Acta Automatica Sinica*, 2013, 39(10): 1602-1610.)
- [8] Wang C Q, Wang J K, Li C M, et al. Safe and robust mobile robot navigation in uneven indoor environments[J]. *Sensors*, 2019, 19(13): 2993.
- [9] Yu X Y, Wang J C, Chen W D, et al. A map accessibility analysis algorithm for mobile robot navigation in outdoor environment[C]. *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. Irkutsk, 2019: 475-480.
- [10] Choi S, Park J, Lim E, et al. Global path planning on uneven elevation maps[C]. *The 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. Daejeon, 2012: 49-54.
- [11] Ohki T, Nagatani K, Yoshida K. Path planning for mobile robot on rough terrain based on sparse transition cost propagation in extended elevation maps[C]. *2013 IEEE International Conference on Mechatronics and Automation*. Takamatsu, 2013: 494-499.
- [12] Ganganath N, Cheng C T, Tse C K. A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains[J]. *IEEE Transactions on Industrial Informatics*, 2015, 11(3): 601-611.
- [13] Norouzi M, Miro J V, Dissanayake G. Planning high-visibility stable paths for reconfigurable robots on uneven terrain[C]. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura-Algar, 2012: 2844-2849.
- [14] Saad M, Salameh A I, Abdallah S. Energy-efficient shortest path planning on uneven terrains: A composite routing metric approach[C]. *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. Ajman, 2019: 1-6.
- [15] Zhang H, Liu M L, Liu R, et al. Path planning of robot in three-dimensional grid environment based on genetic algorithms[C]. *The 7th World Congress on Intelligent Control and Automation*. Chongqing, 2008: 1010-1014.
- [16] 闫飞, 庄严, 白明, 等. 基于拓扑高程模型的室外三维环境建模与路径规划[J]. *自动化学报*, 2010, 36(11): 1493-1501.
(Yan F, Zhuang Y, Bai M, et al. 3D outdoor environment modeling and path planning based on topology-elevation model[J]. *Acta Automatica Sinica*, 2010, 36(11): 1493-1501.)
- [17] Ono M, Fuchs T J, Steffy A, et al. Risk-aware planetary rover operation: Autonomous terrain classification and path planning[C]. *2015 IEEE Aerospace Conference*. Big Sky, 2015: 1-10.
- [18] 余文凯, 章政, 付雪画, 等. 基于地图预处理及改进A*算法的路径规划[J]. *高技术通讯*, 2020, 30(4): 383-390.
(Yu W K, Zhang Z, Fu X H, et al. Path planning based on map preprocessing and improved A* algorithm[J]. *Chinese High Technology Letters*, 2020, 30(4): 383-390.)
- [19] 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.
(Wang H B, Yin P H, Zheng W, et al. Mobile robot path planning based on improved A* algorithm and dynamic window method[J]. *Robot*, 2020, 42(3): 346-353.)
- [20] Nowac W, González F, MacMahon S, et al. Performance indicators for wheeled robots traversing obstacles[J]. *IEEE Robotics and Automation Letters*, 2020, 5(2): 2881-2888.
- [21] Shamsudin A U, Ohno K, Hamada R, et al. Two-stage hybrid A* path-planning in large petrochemical complexes[C]. *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. Munich, 2017: 1619-1626.

作者简介

黄志清(1970—), 男, 副教授, 博士, 从事机器人SLAM、无人驾驶智能决策控制等研究, E-mail: zqhuang@bjut.edu.cn;

李鼎鑫(1995—), 男, 硕士生, 从事机器人SLAM、无人车导航的研究, E-mail: ldx@emails.bjut.edu.cn;

王庆文(1968—), 男, 高级工程师, 博士, 从事移动机器人SLAM、机器视觉图像识别等研究, E-mail: qingwen.wang@iotcnn.com.