

基于动态精简式混合地图的移动机器人自主探索

李秀智, 邱欢[†], 贾松敏, 龚月

(北京工业大学 信息学部, 北京 100124)

摘要: 针对未知环境下移动机器人自主探索和地图创建问题, 在机器人操作系统的框架下, 提出一种基于动态精简式混合地图的移动机器人自主探索方法. 首先, 提出一种基于几何规则的候选目标点生成方法, 用于快速提取当前的前沿目标点; 然后, 从信息收益和路径成本的角度, 引入一种改进的效用函数来评价候选目标点; 最后, 利用缓存增量式的原理优化拓扑节点, 进而构建精简式混合地图. 实验结果表明, 通过拓扑图构建策略的改进, 所提出方法具有良好的导航性能.

关键词: 地图创建; 机器人操作系统; 混合地图; 信息收益; 自主探索

中图分类号: TP24

文献标志码: A

Mobile robot autonomous exploration based on dynamically simplified hybrid map

LI Xiu-zhi, QIU Huan[†], JIA Song-min, GONG Yue

(Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

Abstract: For mobile robot autonomous exploration and mapping in the unknown environment, the paper presents a mobile robot autonomous exploration method based on the dynamically simplified hybrid map under the robot operating system(ROS) framework. Firstly, to fast extract current frontier target points, a method of generating target points based on geometry rules is proposed. Then, from the perspective of the information gain and the cost of path, an improved value function is introduced to measure the candidate target points. Finally, an incremental caching theory is adopted to optimize topological nodes, and then construct the simplified hybrid map. Experimental results demonstrate that the proposed method has good navigation performance by improving the strategies of building topological map.

Keywords: mapping; ROS; hybrid maps; information gain; autonomous exploration

0 引言

未知环境下移动机器人自主探索是机器人领域的一个历久弥新的研究热点, 涉及机器人搜救^[1]、行星探测^[2]和 水下探测^[3]等诸多领域. 在自主探索的过程中, 机器人通过感知环境来获取一系列的局部目标点, 进而指导机器人遍历环境^[4]. 几乎所有的机器人探索策略均可简化为一个生成最佳目标点和运动到该目标点的过程^[5]. 一种较为常见的探索策略是在前沿^[6]处生成候选目标点, 然后通过效用函数评价当前候选目标点, 进而实现最优选择^[7].

前沿检测的合理性及鲁棒性是上述探索策略中的核心. 现有的研究中, 通常借用数字图像处理技术在栅格地图上识别前沿区域^[5,8]. 然而, 随着环境规模

的增大, 在栅格地图上进行前沿检测和路径规划的计算量会急剧膨胀, 继而导致机器人探索效率显著降低.

为了提高机器人的探索效率, Ge等^[9]利用分层拓扑图表示全局环境信息, 实现了未知环境下同时路径规划和拓扑建图(SP²ATM)的环境探索算法. 该方法避开了基于栅格地图的前沿检测问题, 但在提取的可能目标点中存在非前沿区域的目标点时会产生不必要的探索开销. 在作者所在研究团队的前期工作^[10]中, 利用改进的SP²ATM算法探索环境, 完成了多机器人地图拼接的任务. 然而, 该方法创建的拓扑地图中冗余节点较多, 在计算回溯路径时会占用较多的计算资源, 并且未引入合适的效用函数来评价候选

收稿日期: 2016-01-28; 修回日期: 2016-06-14.

基金项目: 北京工业大学智能机器人“大科研”推进计划项目(002000514316009); 北京工业大学日新人才计划支持项目(2015-RX-L03).

作者简介: 李秀智(1979—), 男, 副教授, 博士, 从事移动机器人、机器视觉等研究; 邱欢(1991—), 男, 硕士生, 从事移动机器人的研究.

[†]通讯作者. E-mail: qiuhuanbjut@163.com

目标点的优劣. 基于栅格-拓扑混合地图的环境探索方法^[11-12], 由于可利用的信息更加充分, 一度成为人们关注的热点. 该方法利用混合地图中的拓扑地图信息进行路径规划和导航, 以此来提高自主探索效率. 其中 Gim 等^[11] 先构建栅格地图表达环境信息, 再根据栅格地图来获取拓扑地图信息. 但该方法存在不能同步构建栅格地图和拓扑地图的问题. Jia 等^[12] 在粒子滤波的基础上, 结合激光雷达数据增量式地构建了栅格-拓扑混合地图模型. 但该方法存在冗余拓扑节点过多的问题.

为了完善上述方法中的不足, 本文提出一种基于动态精简式混合地图的移动机器人自主探索方法. 首先, 提出一种基于几何规则的候选目标点快速生成方法, 可同时提高目标点检测的可靠性及效率, 根据当前激光数据提取当前前沿区域的候选目标点, 为避免重复探索, 利用目标点全局动态维护以保证候选目标点全局最优; 然后, 从信息收益和路径代价的角度, 利用一种改进的效用函数评价候选目标点; 最后, 通过缓存增量式原理优化拓扑地图的构建, 一定程度上剔除了冗余的拓扑节点, 得到更加精简的拓扑地图. 此外, 本文在 ROS^[13] 框架下实现了自主探索算法的模块化设计, 采用多线程的运行机制, 进一步提高了自主探索算法的实时性和混合地图创建的同时性.

1 算法框架

本文的混合地图为栅格-拓扑地图, 算法框架如图 1 所示. 在底层, 基于 GMapping 算法^[14] 完成同时定位与地图创建 (SLAM), 根据激光测距仪数据和里程计数据优化机器人位姿并创建栅格地图. 在上层, 本文结合作者所在研究团队的前期工作^[10], 在候选目标点生成、拓扑地图更新及效用函数等方面做出改进, 以构建精简式的拓扑地图.

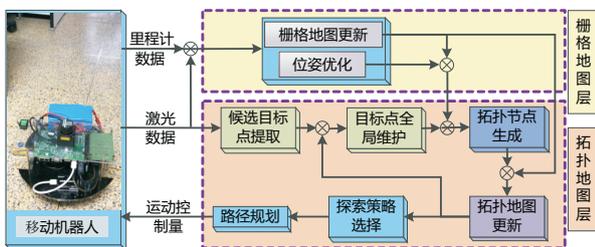


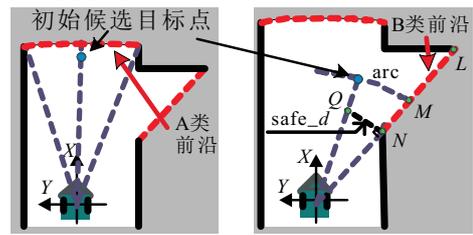
图 1 算法框架

2 候选目标点提取

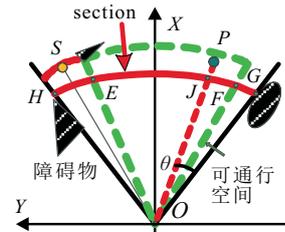
2.1 基于几何规则的初始候选目标点生成

本文将当前视野下的前沿分成 A、B 两类, A 类前沿在传感器最大量程处, 如图 2(a) 所示, 其弧长大于移动机器人的宽度; B 类前沿由障碍物遮挡而产生,

该处深度差大于机器人宽度, 如图 2(b) 所示. 候选目标点在 A 和 B 两类前沿处产生.



(a) A类初始目标点 (b) B类初始目标点



(c) 初始目标点调整示意

图 2 A、B 两类初始候选目标点的选取及调整示意

为使生成的候选目标点处在当前可通行空间^[9]中, 本文生成 A 类初始候选目标点的几何规则如下:

- 1) 将前沿弧的中点与激光测距仪的中心相连;
- 2) 取连线上略小于传感器最大量程的点作为初始的候选目标点, 将略小于的数量 δ 设定为 0.1 m, 如图 2(a) 所示.

令 $safe_d$ 略大于二分之一的机器人宽度, 本文生成 B 类初始候选目标点的几何规则如下:

- 1) 将 B 类前沿的两端分别标记为 N 、 L ;
- 2) 在 NL 上取一点 M , 使 $MN = safe_d$;
- 3) 以激光测距仪的中心位置为圆心, 作经过点 M 的圆弧 arc ;
- 4) 在过 N 点的垂线上取一点 Q , 使 $NQ = safe_d$, 且点 Q 在自由区域内;
- 5) 作一条由激光测距仪的中心指向点 Q 的射线, 射线与圆弧 arc 的交点即为初始的候选目标点, 如图 2(b) 所示.

若用点 P 表示候选的目标点, 则其数据结构为

$$P = \{d, idx, type\}. \quad (1)$$

其中: d 表示目标点的深度信息, idx 表示点 P 对应激光束的索引号, $type$ 表示目标点的类型.

2.2 对初始候选目标点进行调整

研究表明, 上述规则并不能保证生产的候选目标点一定出现在当前可通行空间^[9]中, 因此有必要进行二次优化和调整. 如图 2(c) 所示, O 为激光测距仪的中心, 扇形区域表示一个当前的可通行空间, 点 S 为一个初始的候选目标点, 满足 A 类初始候选目标点的

生成条件,但该点不在可通行空间中,且机器人不能沿直线到达该目标点,故需对该点做出调整。

引入 section^[9]来辅助调整初始的候选目标点,如图2(c)中圆弧所示。section定义为

$$\text{section} = \{r, \text{start_idx}, \text{end_idx}\}. \quad (2)$$

其中: r 表示该 section 对应的半径, start_idx 和 end_idx 分别表示该 section 左右两端的点对应激光束的索引号。如图2(c)所示, P 为当前初始的候选目标点,实线标记的圆弧表示与 OP 相交的一个 section,交点记为 J , section 的首末端点分别标记为 H 、 G ,与可通行区域边界的交点分别为 E 、 F 。

A类候选目标点成立的条件为:

- 1) $\widehat{HJ} > \widehat{HE}$;
- 2) $\widehat{JG} > \widehat{FG}$;
- 3) $|OP| > d_{\text{tho}}$.

其中: d_{tho} 是一个给定阈值,限定A类候选目标点与激光测距仪中心的最小距离。为简化计算,条件1)和条件2)可简化为:

- 1) $\widehat{HJ} > \text{safe_dis}$;
- 2) $\widehat{JG} > \text{safe_dis}$.

令 $\Delta\theta$ 表示激光测距仪扫描的角度增量,用弧度制表示,则 section 对应的圆心角为

$$\beta = (\text{end_idx} - \text{start_idx}) \cdot \Delta\theta, \quad (3)$$

因此有

$$\begin{cases} \theta = (\text{idx} - \text{start_idx}) \cdot \Delta\theta, \\ \widehat{HJ} = (\beta - \theta) \cdot r, \\ \widehat{JG} = \theta \cdot r. \end{cases}$$

其中: θ 表示 OP 与对应 section 右边线的夹角,通过调整 θ 和 $|OP|$ 使目标点 P 满足自身成立的条件,其初始的 $|OP| = d$ 。与A类候选目标点相比,基于几何位置的考虑,仅对B类候选目标点的方向做出调整,故B类候选目标点仅需满足条件1)和条件2)即可。

针对所有与 OP 相交的 section,若调整后的 θ 和 $|OP|$ 均能够使 P 满足自身成立的条件,则根据调整后的 θ 和 $|OP|$ 计算调整后的候选目标点;否则,舍弃该初始的候选目标点。

2.3 目标点全局动态维护

当前的候选目标点仅由最新的激光数据生成,未包含全局信息,因此,目标点可能出现在已探索的区域。为剔除可能出现的该类目标点,需结合所有已生成的拓扑节点对目标点做全局动态维护^[10],以保证所有的候选目标点全局最优,从而提高探索效率。

3 全局拓扑地图层缓存式更新

3.1 创建拓扑节点

为方便导航,每个拓扑节点中应包含所对应机器人的位姿和候选的目标点等信息。因此,拓扑节点 T 的数据结构定义为

$$T = \{\text{pose}_m, P_c, \text{parent}, \text{children}\}. \quad (4)$$

其中: pose_m 为优化后的机器人位姿, parent 和 children 分别为该节点的父节点和子节点, P_c 为当前节点的候选目标点集合。针对 P_c 中的任一元素 P 可扩展为

$$P = \{\text{position}, d, \text{idx}, \text{type}, \text{effectiveFun}\}. \quad (5)$$

其中: position 是该目标点的全局位置, effectiveFun 是评价该目标的效用函数值。

在进行效用评价时考虑目标点的信息收益和路径成本,效用函数 $F(p)$ 为

$$F(p) = A(p) \cdot e^{-(k_1 \cdot L + k_2 \cdot \gamma)}. \quad (6)$$

其中: p 为候选目标点, $A(p)$ 为该目标点处的信息收益^[15], L 为机器人当前位置 T_{cur} 与 p 之间的欧氏距离, γ 为 T_{cur} 和 p 的连线与机器人正方向夹角的绝对值, k_1 和 k_2 是通过实验确定的参数。 $F(p)$ 越大,表示该目标点被探索的价值越大。

3.2 全局拓扑地图缓存式更新

基于缓存增量式的原理,本文将拓扑节点分成当前节点、缓存节点、终极节点和标记节点。当前节点是根据最新传感器数据生成的拓扑节点;缓存节点是在当前节点前一时刻生成的拓扑节点;终极节点是指已经融入全局拓扑地图的节点;标记节点是指最终没有被添加到全局拓扑地图中的节点,用来标记机器人探索过的区域。缓存节点、终极节点和标记节点均属于已生成的拓扑节点。除初始的拓扑节点外,此后的每个节点均需由当前节点过渡到缓存节点,最后才有可能成为终极节点。当环境探索完毕时,最后一个缓存节点默认为终极节点。

在更新全局拓扑地图时,需判断是否将缓存节点添加到全局拓扑地图中。缓存节点添加到拓扑地图的条件如下:

- 1) 当前节点和与之较近的终极节点间不是直线可达的;
- 2) 缓存节点至少还存在一个未探索的候选目标点。

上述两个条件中,若任有一条成立,则将缓存节点融入全局拓扑地图中。对于条件1),虚构一个矩形框,如图3(a)中虚线框所示,矩形框的宽度为 w (w 为

大于机器人宽度的给定阈值),在栅格地图上利用 ray-traversal^[16] 算法沿着矩形框的两长边和中线投射3条射线,如图3(b)中射线所示.若3条射线在向前蔓延的过程中均没有遇到障碍物,则认为当前节点和与之最近的终极节点间是直线可达的;否则,不是直线可达的.若条件1)成立,如图3(c)和图3(f)所示,则将缓存节点添加到全局拓扑地图,其属性变为终极节点,同时将当前节点的属性改为缓存节点.若条件2)成立,则全局拓扑地图更新操作如图3(b)和图3(e)所示;若条件1)和条件2)均不成立,则将缓存节点压入标记节点集合,并将当前节点改为缓存节点,如图3(a)和图3(d)所示.

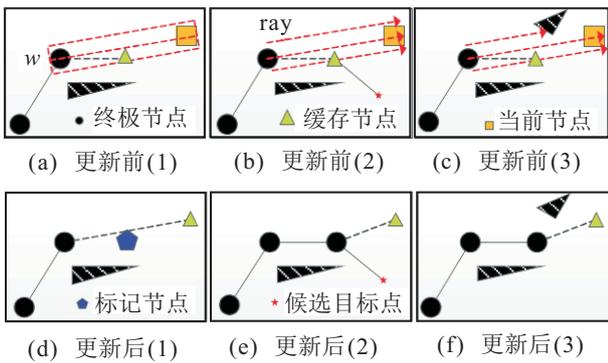


图3 拓扑地图的一次更新示意

4 基于ROS的模块化实现

ROS是一款开源软件^[13],旨在构建一个能够整合不同研究成果、实现算法发布、代码重用的机器人软件平台.“节点”、“消息”、“话题”和“服务”是ROS的4个基本概念.其中,“节点”是与机器人功能相关的任务进程,相当于功能模块.本文假设读者对ROS有一定的了解.

如图4所示,圆角矩形框表示“节点”,即功能模块. SLAM-GMapping^[17]“节点”是GMapping算法^[14]模块化后的产物. Topology“节点”由作者完成,对应本文拓扑地图层的创建部分.

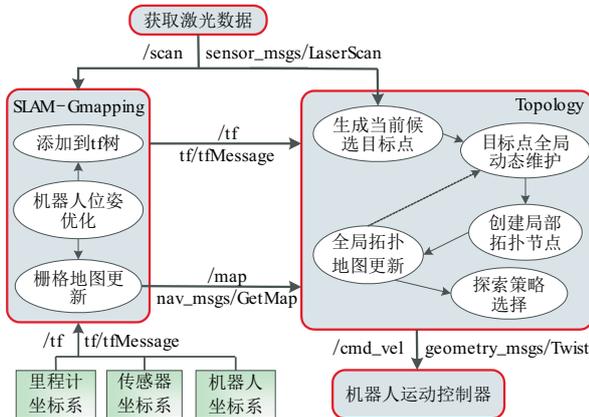


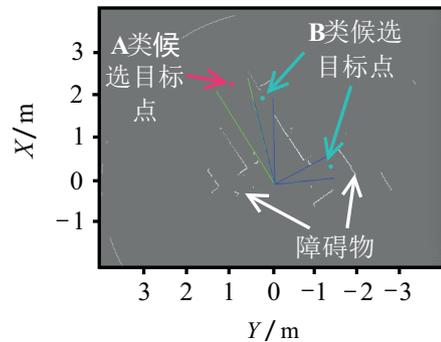
图4 自主探索模块化结构

5 实验结果

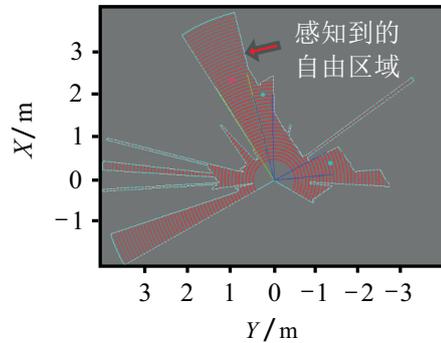
实验硬件为作者所在团队自主研发的移动机器人平台,如图1中所示.控制单元采用研华单板电脑,CPU为Intel赛扬J1900,主频为2 GHz, RAM为2 GB. 配备有URG-04LX-UG01激光测距仪,其扫描范围为240°,水平安装在移动机器人顶部.通过CAN总线收发测控指令.

5.1 提取候选目标点实验

如图5所示,图5(a)为由一组激光数据生成的局部几何地图,图5(b)为其对应的自由区域示意图.由图5可知,在A、B两类前沿处均能生成对应的候选目标点,从而验证了基于几何规则的候选目标点生成算法的可行性.



(a) 局部几何地图

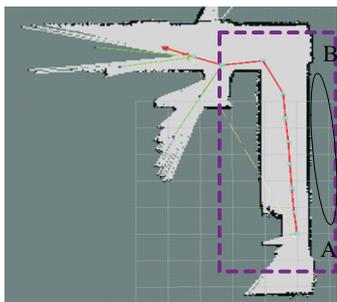


(b) 自由区域

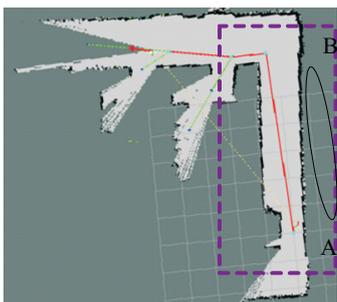
图5 候选目标点生成实验

5.2 拓扑地图更新实验

为验证全局拓扑地图缓存增量式更新策略的有效性,本文设计了一个对比实验,如图6所示.图6(a)为采用文献[10]中拓扑地图增量式更新策略生成的部分拓扑地图,图6(b)为利用本文提出的拓扑地图缓存增量式更新策略生成的部分拓扑地图.图6中较大的点表示拓扑节点,较小的点表示未探索的候选目标点,线条连接未探索的候选目标点与其所属的节点,同时,能够直线到达的相邻两节点也用线条连接,箭头末端为机器人所在的位置,箭头所指方向为机器人在该位置处的正方向.



(a) 拓扑地图增量式更新



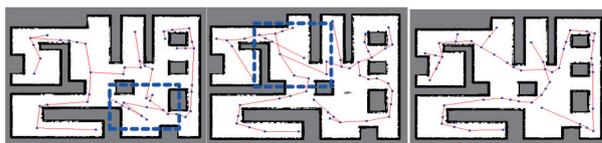
(b) 拓扑地图缓存增量式更新

图6 全局拓扑地图更新对比实验

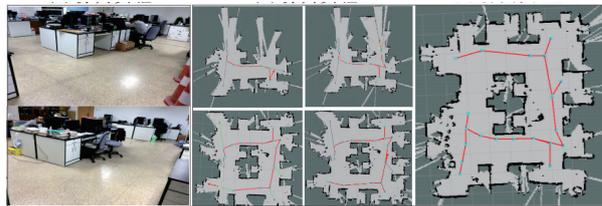
初始拓扑节点为A,拐角处的拓扑节点为B. 移动机器人从A节点运动到B节点的过程中,生成了多个拓扑节点(如图6(a)中椭圆内的节点). 采用文献[10]中的拓扑地图更新策略,椭圆内的拓扑节点均被添加进全局拓扑地图中. 然而,这些拓扑节点在环境表达及导航方面的作用并不大,在利用全局拓扑地图规划路径时,这些拓扑节点反而会增加算法的复杂度. 如图6(b)所示,采用本文提出的拓扑地图缓存增量式更新策略,椭圆内的拓扑节点均被剔除,得到了更加精简的全局拓扑地图,且没有影响拓扑地图在环境表达和导航方面的作用,一定程度上降低了路径规划算法的复杂度. 实验结果表明,利用缓存增量式原理构建精简的拓扑地图是可行的.

5.3 自主探索实验

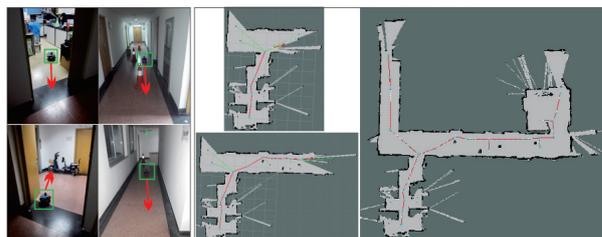
为验证改进的效用函数的有效性,在仿真环境下做了3组实验,如图7(a)、图7(b)和图7(c)所示. 图7(a)和图7(b)中,分别选择当前时刻距离机器人最远和最近的目标点所在区域作为机器人将要探索的区域,虽然两图中所示的环境探索方法可以实现对环境的全局遍历,但是探索性能较差,如图中虚线包围的矩形区域,机器人在该区域内出现了重复性的探索行为;图7(c)中,机器人利用改进的效用函数进行环境探索,综合考虑当前时刻目标点处的信息收益和到达该目标点的路径成本,能使移动机器人完整地遍历未知环境,且路径重复率较低. 通过上述对比实验,有力表明了改进的效用函数的优势.



(a) 仿真实验1 (b) 仿真实验2 (c) 仿真实验3



(d) 实验环境1 (e) 实验1探索过程



(f) 实验环境2 (g) 实验2探索过程

图7 自主探索实验图

为验证自主探索算法的可行性,在真实环境下进行了完整的导航测试. 在探索的过程中,各个功能模块同时运行,移动机器人的线速度 $v = 0.15 \text{ m/s}$,角速度 $\omega = 0.3 \text{ rad/s}$,机器人在前进一个采样步长 S_{tep} 或者到达目标点时,生成一个拓扑节点. 由于实验环境的复杂度是未知的,为尽可能多地提取环境中的候选目标点,在本实验中将 S_{tep} 设定为 0.9 m .

实验1中选择单个房间作为实验场所(如图7(d)所示);实验2中选择房间和走廊作为实验场所,并在走廊中放置了一定数量的障碍物(如图7(f)所示). 在两次实验中,机器人每采集一次激光数据就会生成一个拓扑节点,从生成的全局拓扑地图来看,该地图较好地表达了整个实验环境,再次验证了利用缓存增量式原理构建拓扑地图的可行性、有效性. 同时,从图7(e)和图7(g)中的探索过程可以看出,随着探索的进行,栅格地图和拓扑地图是增量式地同步创建的. 可见,算法能够保证栅格地图与拓扑地图创建的同时性. 在候选目标点选择方面,在利用改进的效用函数评价候选目标点的情况下,移动机器人能够较好地完成环境遍历.

实验结果表明,改进的效用函数可行且有效,所提出的移动机器人自主探索策略使机器人能够完整、安全、准确、实时地遍历整个环境,具有良好的导航性能.

6 结 论

本文系统地研究了未知环境下移动机器人自主探索的问题,借助于ROS软件平台,保证了算法的实时性和同时性.完成了Topology“节点”的编写,实现了自主探索算法的模块化设计.充分利用ROS点对点的通信机制,同时运行SLAM-GMapping“节点”和Topology“节点”,同步创建了栅格-拓扑混合地图.本文提出的基于几何规则的候选目标点生成方法需结合全局拓扑信息使用,可不依赖于栅格地图.在拓扑地图更新方面,结合栅格地图信息,在保证实时性的前提下充分利用ray-traversal算法,减少了拓扑地图中的冗余节点.在移动机器人自主探索的过程中,能够完整、连贯地探索直至遍历完实验环境.实验结果验证了本文方法的可行性与有效性.

参考文献(References)

- [1] Calisi D, Farinelli A, Iocchi L, et al. Multi-objective exploration and search for autonomous rescue robots[J]. *J of Field Robotics*, 2007, 24(8/9): 763-777.
- [2] Weisbin C R, Rodriguez G. NASA robotics research for planetary surface exploration[J]. *Robotics and Automation Magazine*, 2000, 7(4): 25-34.
- [3] Bennett A A, Leonard J J. A behavior-based approach to adaptive feature detection and following with autonomous underwater vehicles[J]. *IEEE J of Oceanic Engineering*, 2000, 25(2): 213-226.
- [4] Burgard W, Moors M, Fox D, et al. Collaborative multi-Robot exploration[C]. *Proc of IEEE Int Symposium on Robotics and Automation*. San Francisco: IEEE Press, 2000: 476-481.
- [5] Senarathne P G C N, Wang D. Incremental algorithms for safe and reachable frontier detection for robot exploration[J]. *Robotics and Autonomous Systems*, 2015, 72(C): 189-206.
- [6] Yamauchi B. A frontier-based approach for autonomous exploration[C]. *Proc of IEEE Int Symposium on Computational Intelligence in Robotics and Automation*. Monterey: IEE Press, 1997: 146-151.
- [7] Visser Arnoud, Ji Xing Rui, Van Ittersum Merlijn, et al. Beyond frontier exploration[M]. Atlanta: Springer Verlag, 2008: 113-123.
- [8] Keidar M, Kaminka G A. Efficient frontier detection for robot exploration[J]. *Int J of Robotics Research*, 2014, 33(2): 215-236.
- [9] Ge S S, Zhang Q, Abraham A T, et al. Simultaneous path planning and topological mapping (SP²ATM) for environment exploration and goal oriented navigation[J]. *Robotics and Autonomous Systems*, 2011, 59(3/4): 228-242.
- [10] 贾松敏,李雨晨,王可,等. RTM框架下基于分层拓扑结构的多机器人系统地图拼接[J]. *机器人*, 2013, 35(3): 292-298.
(Jia S M, Li Y C, Wang K, et al. Map merging for multi-robot systems based on hierarchical topology structure under RTM framework[J]. *Robot*, 2013, 35(3): 292-298.)
- [11] Gim S, Kim H, Yang T K. Increasing localization accuracies by hybrid maps and scan matching[C]. 2008 Int Conf on Control, Automation and Systems. Seoul: IEEE Press, 2008: 2076-2081.
- [12] Jia S M, Shen H M, Li X Z, et al. Autonomous robot exploration based on hybrid environment model[C]. 2012 Int Conf on Information and Automation. Shenyang: IEEE Press, 2012: 19-24.
- [13] Quigley M, Conley K, Gerkey B, et al. ROS: An open-source robot operating system[C]. ICRA, Workshop on Open Source Software. Munich: IEEE Press, 2009: 1-8.
- [14] Grisetti G, Stachniss C, Burgard W. Improved techniques for grid mapping with rao-blackwellized particle filters[J]. *IEEE Trans on Robotics*, 2007, 23(1): 34-46.
- [15] Gonzalez-Banos H H, Latombe J C. Navigation strategies for exploring indoor environments[J]. *Int J of Robotics Research*, 2002, 21(10): 829-848.
- [16] Amanatides J, Woo A. A fast voxel traversal algorithm for ray tracing[J]. *European Association for Computer Graphics*, 1987, 87(3): 3-10.
- [17] Garage W. SLAM-GMapping[EB/OL]. (2008). <http://wiki.ros.org/gmapping>.

(责任编辑:李君玲)