



计算机科学

COMPUTER SCIENCE

基于行为树调度的多无人机未知室内空间探索方法

史殿习, 苏雅倩文, 李宁, 孙亦璇, 张拥军

引用本文

史殿习, 苏雅倩文, 李宁, 孙亦璇, 张拥军 [基于行为树调度的多无人机未知室内空间探索方法](#) [J]. 计算机科学, 2022, 49(11A): 210900083-11.

SHI Dian-xi, SU Ya-qian-wen, LI Ning, SUN Yi-xuan, ZHANG Yong-jun. [Multi-UAV Cooperative Exploring for Large Unknown Indoor Environment Based on Behavior Tree](#) [J]. Computer Science, 2022, 49(11A): 210900083-11.

相似文献推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[传感器唤醒机制下的智能干扰源定位方法](#)

Intelligent Jammers Localization Scheme Under Sensor Sleep-Wakeup Mechanism

计算机科学, 2022, 49(11A): 211000165-6. <https://doi.org/10.11896/jsjcx.211000165>

[基于多智能体强化学习的端到端合作的自适应奖励方法](#)

Adaptive Reward Method for End-to-End Cooperation Based on Multi-agent Reinforcement Learning

计算机科学, 2022, 49(8): 247-256. <https://doi.org/10.11896/jsjcx.210700100>

[高分辨率斜视聚束SAR回波仿真加速算法研究](#)

Study on Acceleration Algorithm for Raw Data Simulation of High Resolution Squint Spotlight SAR

计算机科学, 2022, 49(8): 178-183. <https://doi.org/10.11896/jsjcx.210600066>

[基于改进势场法的机器人路径规划](#)

Robot Path Planning Based on Improved Potential Field Method

计算机科学, 2022, 49(7): 196-203. <https://doi.org/10.11896/jsjcx.210500020>

[海上风电场通用运维路径规划模型优化及仿真](#)

Optimization and Simulation of General Operation and Maintenance Path Planning Model for Offshore Wind Farms

计算机科学, 2022, 49(6A): 795-801. <https://doi.org/10.11896/jsjcx.210400300>

基于行为树调度的多无人机未知室内空间探索方法

史殿习^{1,2,3} 苏雅倩文¹ 李 宁² 孙亦璇² 张拥军¹

1 军事科学院国防科技创新研究院 北京 100166

2 国防科技大学计算机学院 长沙 410073

3 天津(滨海)人工智能创新中心 天津 300457

(dxshi@nudt.edu.cn)

摘 要 文中提出了一种在无 GPS 信号的未知室内空间中利用行为树框架调度多无人机和路径规划算法进行协作探索的方法。该方法的核心是提出了一种未知区域动态目标追踪算法 Tracking-D* Lite 对未知地形中的运动目标进行跟踪,同时结合基于 Bug 算法的 Wall-Around 算法在未知室内空间中导航无人机,最后利用行为树对多架无人机以及这两种算法进行调度和切换。该方法基于 ROS,使用 Gazebo 进行模拟和可视化。设计并实现该方法与其他未知室内空间探索方法的对比实验,结果表明它可以有效地完成探索任务并最终绘制出整个未知室内空间的边界轮廓图。一旦扩展到现实世界中,该方法可以应用于地震后的危险建筑、危险气体工厂、地下矿井或其他搜救场景。

关键词: 多无人机;行为树;无 GPS 信号;协同探索;路径规划

中图法分类号 TP391

Multi-UAV Cooperative Exploring for Large Unknown Indoor Environment Based on Behavior Tree

SHI Dian-xi^{1,2,3}, SU Ya-qian-wen¹, LI Ning², SUN Yi-xuan² and ZHANG Yong-jun¹

1 National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing 100166, China

2 School of Computer Science, National University of Defense Technology, Changsha 410073, China

3 Tianjin Artificial Intelligence Innovation Center, Tianjin 300457, China

Abstract This paper proposes a method of using behavior tree framework to schedule multiple UAVs and path planning algorithms for collaborative exploration in a large unknown indoor space without GPS signals. The core of this method is to use the Tracking-D* Lite algorithm to track moving targets in unknown terrain, combined with the Wall-Around algorithm based on the Bug algorithm to navigate the UAV in the unknown indoor environment. Finally, the behavior tree is used to schedule and switch multiple UAVs and these two algorithms. This method is based on ROS and uses Gazebo for simulation and visualization. It designs and implements comparative experiments with other unknown indoor environment exploration methods. Experimental results show that it can effectively complete the exploration task and finally draw the boundary contour map of the entire unknown indoor environment. Once extended to the real world, this method can be applied to dangerous buildings after earthquakes, hazardous gas factories, underground mines, or other search and rescue scenarios.

Keywords Multi-UAV, Behavior tree, GPS-denied environment, Collaborative exploration, Path planning

1 引言

无人机(Unmanned Aerial Vehicle, UAV)作为一种新兴的人工智能技术,不仅成为近几年最受关注的领域之一^[1],而且凭借高效便捷的辅助手段替代了原有工具服务于各行各业的日常工作中。凭借成本低、性价比高、机动性能好、使用方便等优势,无人机在农业植保、物流配送、事故危害现场探测及救援、警用监控等领域得到了越来越多的应用,降低了人工操作的风险,提高了任务执行的安全性和可控性。

对于一个无 GPS 信号的未知复杂环境而言,无人机的自主飞行问题主要有两方面的难点。一方面,环境中存在不利于飞行控制的外部干扰;另一方面,无人机缺少对环境的先验信息,因此需要在运动过程中建立环境地图以满足位姿估计

和运动规划需求。其中,定位与地图创建是自主导航中最基本的问题,解决该问题依赖于具有足够精度的环境地图,而创建地图又以准确定位为基础。针对未知环境中单架无人机的导航问题,可利用同时定位与地图创建(Simultaneous Localization And Mapping, SLAM)^[2]方法进行探索。但是为了获得全局一致的运动状态和地图估计,SLAM 方法往往需要较高的计算性能和存储性能,无法满足低配置无人机飞行控制中状态反馈的需要。所以对于单架无人机而言,其有限的计算能力和存储性能很难使用 SLAM 方法快速地对未知环境进行高精度的探索。因此,为了提高无人机在多变复杂的任务场景中完成任务的能力并减少无人机的计算成本和搜索未知室内空间的时间,可以在无 GPS 信号的未知室内空间中使用一种多无人机协同探索策略,这种策略能够利用多无人机

以相对较小的成本快速探索整个未知的室内空间,并为后续任务提供及时有效的信息,例如事故或灾难现场的探索和救援、建筑和公共设施检查等。

随着当前人工智能技术的不断发展,日新月异的新技术不断颠覆不同的传统行业和领域。针对多无人机智能规划任务目标,自适应调度路径规划算法,具有模块化、层次化、灵活和表达性强的行为树技术^[3-4]得到了广大研究者的重视。行为树在构建复杂的任务切换和执行逻辑时具有很大的便利性^[5],且能够针对类似的场景进行复用。通过将多无人机探索未知室内空间任务和行为树相结合,利用行为树对多无人机和路径规划算法进行协调与调度。

因此,利用多无人机探索无 GPS 信号的未知室内空间主要面临 3 个挑战。1) 使用多架无人机对未知室内空间进行探索,使用行为树构建多无人机之间的协作探索模型,设计一种具有鲁棒性和灵活性的探索策略。在无 GPS 信号的未知室内空间中,搜索策略的设计不仅需要自主协调多架无人机之间的相互协作,更需要对每架无人机搭配能够自主定位、飞行和避障的导航算法。2) 在无人机探索过程中,导航策略应避免无人机之间的重复探索,并且每一架无人机应尽可能高效地搜索未知区域,覆盖更大的范围。3) 保证单架无人机出现故障不会造成搜索过程的失败,从而提高整个策略的稳定性。

综上所述,针对无 GPS 条件下未知室内空间及单架无人机能力受限问题,本文提出了一种基于行为树调度的多无人机协作探索方法。1) 提出了一种基于 Bug 算法^[6]思想的 Wall-Around 算法,采用能力受限的微型无人机自主绕墙飞行,进行未知室内空间自主探索,从而绘制出飞行轨迹和未知室内空间的轮廓,为上层应用提供支撑;2) 将 D* Lite^[7]和 I-ARA*^[8]算法有机结合,提出了一种基于动态目标追踪的多无人机协作探索算法 Tracking-D* Lite,无人机之间通过动态目标追踪和接力的方式,实现未知室内空间的探索,扩大搜索范围,提升搜索效率;3) 基于行为树设计一种调度策略,实现多无人机对未知室内空间的协同探索;4) 设计了一系列 Tracking-D* Lite 算法对比实验以及与其他多无人机协同探索未知室内空间方法的对比实验^[19];5) 在 ROS 平台上设计实现了一个基于行为树调度的多无人机协同探索未知室内空间的原型系统,并在 Gazebo 仿真环境中搭建的未知室内空间场景中进行验证,仿真实验表明本文提出的方法能够有效提升室内未知空间的探索效率。

2 相关工作

2.1 多无人机室内协同探索方法

目前,针对无 GPS 信号的未知室内空间探索问题,研究者们提出了各种各样的解决方案。He 等^[9]针对无 GPS 环境下的单无人机的定位和导航问题进行研究,配合无人机搭载的激光雷达对未知环境进行感知,并基于 BRM^[10]算法设计了一个卡尔曼滤波模型对无人机位置进行估计,但该方法只针对无人机在未知空间中的导航和定位,并未考虑多无人机的协同探索问题。Pravitra 等^[11]提出了一种利用无人机探索未知室内空间的策略,该探索策略使用激光雷达提供的 2D 距离信息,通过将基于传感器的随机树边界规划器与沿墙速度场生成器混合生成速度命令,但该探索策略中无人机需要搭载高精度的激光雷达,通过 SLAM 方法进行建图和定位,

提高了无人机配置要求,而且未考虑多无人机的协同探索问题。Stirling 等^[12]使用群体无人机对一个封闭的无 GPS 信号的未知室内空间进行探索。他们将无人机分为信标无人机和探索无人机:信标无人机处于静止状态,其被固定在天花板上,以节约能源并形成无人机传感器网络;探索无人机在信标无人机的指引下不断朝未知空间前进,当到达探索边界时,探索又会转化为信标无人机为其他探索无人机指引方向。该工作中使用多架信标无人机组成了一个巨大的指引网络为探索无人机指引正确的前进方向,同时该指引网络的存在成功解决了无人机的导航定位问题。然而,该方法为了构造指引网络,需要用到大量的信标无人机,因为在整个多无人机系统中大部分无人机都用于定位和指引,只有很少一部分无人机用于真正的探索任务。Grzonka 等^[13]将适用于地面机器人的导航方法扩展到了无人机上,采用带有处理器和各种传感器的专用无人机来实时处理所获取的空间信息,可以在未知环境中实时构建地图和导航。Bi 等^[14]在无人机上配备了激光雷达和深度摄像头,使用激光雷达感知周围的障碍物,利用深度摄像头探测障碍物。基于 SLAM 的方法在未知室内空间探索领域最为经典和精准,该方法需要无人机配备较高性能的处理单元和存储设备以对复杂场景进行精确建模和对无人机进行自主控制,但是该方法并不能适用于计算能力弱、存储性能低的普通无人机,无法完成 SLAM 方法所必须的大量计算。

随着近年来深度强化学习的不断发展,人们开始将其应用于未知室内空间探索任务来解决无人机在室内环境中的定位导航和避障动作策略学习问题。Sampedro 等^[15]使用深度学习和强化学习方法来识别无人机拍摄的视频图像,设计了一种在室内环境中进行搜索和救援的应用系统。Pham 等^[16]提出了一种使用 PID+Q 学习算法来训练无人机学习在未知环境中导航到目标点的方法。Walker 等^[17]提出了一种基于深度强化学习的无人机探索室内环境的框架,该框架分为马尔可夫决策过程(Markov Decision Process, MDP)和部分可观察的马尔可夫决策过程(Partially Observable MDP, POMDP)。Walker 等^[18]又提出了一个目标发现框架,该框架将传统的基于 POMDP 规划与深度强化学习方法相结合,与文献[17]不同的是,其使用了多个无人机来探索相同的室内环境。基于深度强化学习的方法利用单架无人机对复杂未知空间进行导航、探索和路径规划能够取得一定的效果,但是该方法的大部分研究工作都是基于单架无人机,缺乏对多无人机的有效利用。

McGuire 等^[19]提出了一种群体梯度错误算法(Swarm Gradient Bug Algorithm, SGBA),该方法使用群体微型无人机探索无 GPS 信号的未知室内空间。微型无人机相比于普通无人机具有更小的体型和更轻的重量,该文献中所使用的微型无人机重量只有 27 g,机身上搭载了一款 STM32 处理器和一颗内存大小为 192 kB 的存储器。该微型无人机机身上配备了 6 个激光雷达用于测量前后左右上下 6 个方向上到障碍物的距离,从而起到避障的效果。同时,该微型机器人的底部还装有一块基于光流的视觉里程计,用于计算微型无人机在飞行过程中移动的距离和方向,可以达到辅助定位的效果。该文献中使用这款微型无人机的硬件支持再搭配适当的调度算法和路径规划算法提出了 SGBA 方法,该方法派出多架

微型无人机朝不同的优先方向前进,利用简单的 Bug 算法进行导航和返回,最后绘制出所有无人机围绕障碍物所飞行的路线,从而得到整个未知室内空间的边界轮廓图。该文献所使用的 SGBA 方法避免了在微型无人机上使用过于复杂的方法(如 SLAM),能够有效地在无 GPS 信号的未知室内空间中进行搜索和导航。但是,该方法对每架微型无人机所分配的任务可能会产生重合,比如多架微型无人机可能对同一片区域进行重复搜索,导致资源上的浪费。同时,该方法也并未将微型无人机的探索效率发挥到最大,由于微型无人机经得起损耗,所以微型无人机可以将全部能源用于探索新的边界,进而探索更大的空间。

为了能够在无 GPS 信号的未知室内环境中使用无人机进行有效的探索和导航,本文提出了 Wall-Around 算法和 Tracking-D* Lite 算法结合的基于传统路径规划算法的多架微型无人机协同探索方法。该方法能够应用在一些计算能力弱、内存小的无人机(特别是微型无人机)上,非常适用于对室内环境进行探索,并且成本代价更小。

2.2 行为树

行为树是一种以层次化的方式表示计划以及决策的工具,比有限状态机拥有更好的可扩展性和更高的模块化程度,是取代有限状态机的理想模型。行为树发源于游戏产业并广泛用于大型游戏中非玩家角色(Non-player Character, NPC)的行为策略控制。行为树是一种只包含执行结点和控制结点的树状结构,执行结点处于整个树状结构的叶结点,用于执行动作和判断条件;控制结点属于树中的内部结点,用于执行逻辑和任务切换。执行结点可分为条件结点、动作结点;常用的控制结点主要包括顺序结点、选择结点、并行结点和装饰器结点。行为树的执行过程从根结点以固定频率触发,使用深度搜索的方式执行每一个结点并将结果以逐层反馈的方式传递至上层结点。

行为树具有模块化、层次化、代码可重用、反应快速、可读性强以及表达性强的特点^[20-21]。模块化是指系统的各个组成部分可以分解为多个构件并重新组合的程度,模块化系统可以一次设计、实施、测试和重用同一模块。因此,通过在设计、实施和测试中启用分而治之的方法,系统越复杂,模块化的好处就越能得到体现。层次化代表行为树是分层的,它包含多个决策层,因为行为树的每一层中的每个结点都会自动定义层次结构中的一个决策。代码可重用表示任何子树都可以在行为树的多个位置重用。反应快速是指快速有效地对变化做出反应的能力,因为不断从根结点触发执行会形成闭环执行,提高了行为树的应激性。行为树由于其树状结构和模块化而易于阅读。行为树控制体系结构必须足够表达多种行为,它至少具有与有限状态机相同的表现力。

目前,由于具有种种优点,行为树已经开始在机器人人工智能领域得到更多的重视和应用^[20-23]。Marzinotto 等^[24]提出了用于机器人行为控制的统一行为树库,并利用 NAO 机器人对该库进行验证。Jones 等^[25]利用行为树为群体机器人设计了一个调度控制器,能够控制数以千计的机器人完成觅食任务。Kuckling 等^[26]采用行为树作为群体机器人控制软件体系结构,使用行为树替换有限自动机,从而达到了更好的控制效果。

3 场景描述

在本文中,我们针对无 GPS 信号的未知室内空间,基于行为树调度,使用多架无人机进行自主协同探索,并绘制出整个未知室内空间的边界轮廓图。图 1 展示了利用 4 队 8 架无人机对一个未知室内空间进行探索的场景,该未知室内空间在不同高度的二维平面轮廓是一致的。首先,所有无人机位于整个地图的中心位置,如图 1(a)所示,4 架无人机分别从不同的方向出发,无人机飞行于某一固定高度(如 1.5 m),遇到边界时使用 Wall-Around 算法进行边界环绕,其中不同的无人机飞行轨迹使用不同的颜色表示。当正在进行边界探索的无人机电量不足或者出现意外情况时,从出发点派出下一架无人机使用 Tracking-D* Lite 算法进行接力追踪,如图 1(b)虚线所示。最后,通过每队无人机的接力和探索过程,得到图 1(c)所示的边界轮廓图。

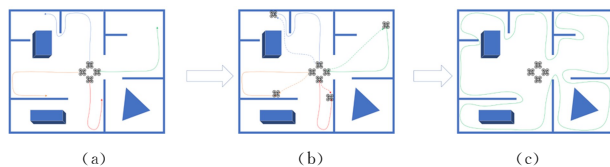


图 1 场景描述示意图

Fig. 1 Scenario description diagram

本文中的多无人机协同探索系统由 1 架无人机和 1 个基站组成,基站和无人机都部署于未知室内空间的中心位置。基站上安装有总控制软件,与多架无人机通过无线信号进行通信,完成无人机调度、实时无人机轨迹显示以及地图轨迹拼接的功能。总控制软件发出指令启动无人机,命令无人机进行边界环绕和追踪接力。

4 多无人机协同探索方法

针对图 1 中描述的未知室内空间探索问题,我们提出了一种基于行为树调度的多无人机未知室内空间协同探索方法,该方法包括 1 个调度框架和 3 个机制:行为树调度框架、Wall-Around 机制、接力追击机制和边界轮廓构建机制。

行为树调度框架主要完成对无人机的调度和算法的切换,每架无人机都运行相同的行为树,根据无人机感知周围环境的信息做出算法切换和与基站的通信。3 个机制的具体算法调度框架如图 2 所示,主要结合 Wall Around 算法和 Tracking-D* Lite 算法,包括输入、Wall Around 算法、Tracking-D* Lite 算法以及输出。

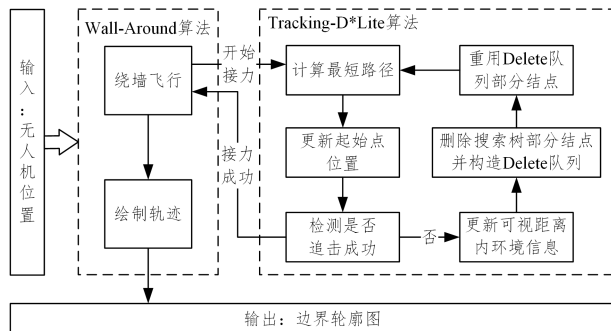


图 2 算法框架图

Fig. 2 Algorithm framework

输入是在初始状态下无人机所在的位置,将其输入之后

直接调用 Wall Around 算法进行绕墙运行并且实时绘制无人机的运行轨迹; Wall Around 算法主要包括绘制轨迹和绕墙飞行,这两项是同步工作的,在无人机停止工作(电量不足或意外情况)时,会触发下一架无人机的接力并调用 Tracking-D* Lite 算法; Tracking-D* Lite 算法主要包括 6 个部分,这 6 个部分概要地表示了 Tracking-D* Lite 算法的流程,直到追上前一架无人机时才会从 Tracking-D* Lite 算法返回并回到

Wall Around 算法继续运行;本框架的输出是多架无人机进行绕墙飞行完毕之后绘制出的边界轮廓图。

4.1 行为树调度框架

本文所提出的方法与 SGBA^[19] 最大的不同就是使用行为树对无人机和算法进行调度和切换。图 3 所示为一棵控制单架无人机任务切换和算法调度的行为树。该行为树以一个选择结点为根结点,分为 4 棵子树。

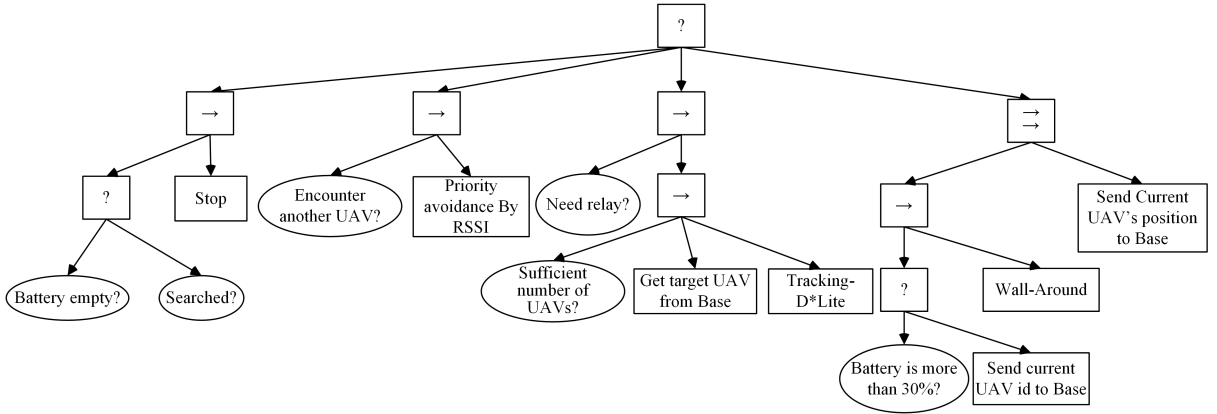


图 3 无人机控制行为树

Fig. 3 UAV control behavior tree

第一棵子树用于判断当前无人机是否需要继续飞行,当电量不足或者当前区域已被搜索过时,则停止飞行。

第二棵子树用于判断当前无人机是否需要周围无人机进行避撞,首先判断周围是否有其他无人机,若有其他无人机,则使用接收信号强度(RSSI)^[27]避免撞击。

第三棵子树用于无人机进行接力状态转换。当前无人机按照条件结点(Need relay?)询问基站探索系统中是否有电量不足或出现意外的无人机需要接力,若无,则跳过接力环节;若有,则按以下步骤执行:首先,条件结点(Sufficient number of UAVs?)用于判断基站当前是否具有足够的无人机,是则继续执行,然后执行结点(Get target UAV from Base)从基站获取目标无人机的位置,最后执行结点(Tracking-D* lite)命令无人机使用 Tracking-D* Lite 算法进行接力。

第四棵子树用于控制无人机对未知室内空间进行边界环绕以及无人机与基站的坐标信息传递。由于该子树使用并行结点作为根结点,因此并行结点下的两个子树同时执行。一方面,并行结点左子树中条件结点(Battery is more than 30%?)用于判断当前无人机的电量是否大于 30%,若是,则返回;若不是,则将当前无人机的编号发送给基站,用于无人机之间的接力。然后,令无人机使用 Wall-Around 算法进行边界环绕。另一方面,并行结点右子树用于获取当前无人机在边界环绕时的位置信息,并将其发送给基站。

本文所提出的基于行为树调度的多无人机未知室内空间协同探索方法利用图 3 所示的行为树对每一架无人机进行控制,并且通过与基站之间的相互通信,基站对每一架无人机进行协调和规划,在无人机接收到基站的指示之后,行为树对其做出相应的状态转换,从而达到单无人机的自主导航以及多无人机之间的协同探索目的。

4.2 无人机定位与感知

无人机的定位与感知功能是多无人机协调探索未知室内空间的基础,主要由搭载于机身上的摄像传感器和激光雷达

完成。无人机机身底部搭载一个单目相机,该相机使用基于光流的视觉里程计方法计算无人机在飞行过程中移动的距离和方向,可以达到辅助定位的效果。同时,无人机机身上配备了 6 个激光雷达用于测量前后左右上下 6 个方向上到障碍物的距离,6 个激光雷达用于感知周围的环境信息,获取障碍物和未知室内空间边界的位置,从而起到避障的效果。

4.3 Wall-Around 算法

本节中提出一种基于 Bug 算法改进的单架无人机地图边界环绕算法 Wall-Around。对于传统的 Bug 算法而言,无人机在了解目标点位置的基础上,不断朝目标点前进。当遇到障碍物时,则通过环绕障碍物的方式使无人机不断靠近目标点的方式进行导航。Wall-Around 算法与 Bug 算法的不同之处在于其没有一个明确的目标点,无人机只需要围绕地图边界进行环绕,最后回到出发点。对于无人机而言,可以采用类似于 TangenBug 算法^[28]的方式进行导航,通过视觉或者激光雷达等传感器感知周围一定距离内的障碍物信息,从而决定朝某个方向前进。

本算法中首先对整个未知室内空间进行二维平面上的场景建模,将该场景分为前、后、左、右 4 个固定方向。单架无人机朝前、后、左、右 4 个方向进行运动。某一时刻无人机的运动方向状态量可以分为主方向(Main Direction, MD)、下一运动方向(Next Direction, ND)、反方向(Opposite Direction, OD)和上一运动方向(Previous Direction, PD)。MD 表示当前无人机在环绕边界过程中指向边界的方向,ND 表示当前无人机在环绕边界过程中下一步将要前进的方向,OD 表示与 MD 相反的方向,PD 表示与 ND 相反的方向。若当前无人机 MD 已知,则可以通过逆时针计算规则得到无人机环绕边界飞行的其余 3 个运动方向状态量。逆时针计算规则是:MD 逆时针旋转 90 度得到 ND,ND 逆时针旋转 90 度得到 OD,OD 逆时针旋转 90 度得到 PD,PD 逆时针旋转 90 度得到 MD。例如,当 MD 为右时,则 ND 为后,OD 为左,PD 为前。

以此类推,当无人机任意运动方向状态量已知时,可以计算出所有其他状态量。

算法 1 给出了 Wall-Around 算法的核心伪代码,此算法在 Bug 算法的基础上去掉了目标点,使无人机只是单纯地绕墙移动。算法的第 1 行初始化无人机时,无人机通过搭载在机身上的激光雷达测量 4 个方向上到达障碍物的距离,选择距离最短的方向设为初始 MD,同时更新 ND,OD,PD,并且飞行到墙边。第 2—14 行的循环保证了无人机能够不断绕墙移动,并且能够有效地避开障碍物。其中,第 3—4 行表示当 MD 方向为障碍物(即边界墙壁)并且 ND 方向为可行空间时,令无人机朝 ND 方向前进。第 5—7 行表示当 MD 方向为可行空间时,令无人机朝 MD 方向前进,并将 MD 更新为 PD。第 8—10 行表示当 MD 和 ND 方向都为障碍物并且 OD 方向为可行空间时,令无人机朝 OD 方向前进,并将 MD 更新为 ND。第 11—13 行表示当 MD,ND 和 OD 方向都为障碍物时,令无人机朝 PD 方向前进,并将 MD 更新为 OD。第 14 行通过 MD 利用上述逆时针计算规则更新 ND,OD,PD。

算法 1 Wall-Around 算法

1. Move towards the closest distance to the wall and set this direction to MD
2. while keep flying do
3. if MD is obstacles and ND is free then
4. Go to ND
5. else if MD is free then
6. Go to MD
7. MD=PD
8. else if MD and ND are obstacles and OD is free then
9. Go to OD
10. MD=ND
11. else if MD,ND and OD are obstacles then
12. Go to PD
13. MD=OD
14. Update ND,OD,PD according to MD

4.4 Tracking-D* lite 算法

为解决多架无人机探索同一个未知室内空间的协同调度问题,需要设计一个在未知空间追踪目标的算法来解决无人机之间的接力问题。首先需要对整个未知室内空间进行二维平面上的场景建模,得到以无人机出发点为原点的平面直角坐标系,平面坐标系中每一个坐标值交点称为网格地图结点,并以坐标系原点为中心分为前、后、左、右 4 个方向,其中 x 正方向为右, x 负方向为左, y 正方向为前, y 的负方向为后。

对于二维网格地图而言,目前主流的路径规划算法如 A^* ^[29]、JPS^[30] 算法等都是完全已知地图先验信息的前提下进行两个定点之间的路径规划,对于在未知空间的路径规划如 D^* ^[31] 和 D^* Lite^[7] 算法只能在定点之间规划路径。而在追踪目标方面,I-ARA^{*}^[8] 算法虽然能很快地对移动目标点进行路径重规划,但是这是基于地图信息完全已知的前提。因此,对位于未知空间中的两架不断移动的无人机,在位置坐标已知的前提下,需要设计一种目标追踪算法来完成无人机之间的接力任务。本文提出的未知空间动态目标追踪算法(简称 Tracking-D* Lite)主要基于 D^* Lite 算法进行修改,由于 D^* Lite 算法是未知空间的路径规划算法,不能直接用于追踪,所以结合 I-ARA^{*} 算法的思想来进行未知空间目标的追踪。

在介绍 Tracking-D* Lite 算法之前,需要对算法中使用的符号进行说明: S 表示二维网格地图中的结点集合;集合 $Succ(s) \subseteq S$ 表示结点 $s \in S$ 的子结点,即从结点 s 扩展的结点;类似地,集合 $Pred(s) \subseteq S$ 表示结点 $s \in S$ 的父结点; $c(s, s')$ 表示结点 s 移动到结点 s' 的代价,由于本文中采用四联通的扩展方式,所以 $c(s, s') = 1$ 。本算法中使用的启发式函数为曼哈顿距离,由 $h(s, s_{goal})$ 表示,满足 $h(s_{goal}, s_{goal}) = 0$,并且必须满足三角不等式 $h(s, s_{goal}) \leq c(s', s) + h(s', s_{goal})$,其中 $s \in S, s' \in Succ(s)$ 。

Tracking-D* Lite 算法与 D^* Lite 算法的搜索方法类似,是从目标点开始扩展并达到起点位置,最后规划出一条路径。算法中涉及到 g 值和 rhs 值的计算, $g(s)$ 表示从目标点 s_{goal} 到当前结点 $s \in S$ 的最短距离, $rhs(s)$ 表示基于结点 s 的父结点计算出对 $g(s)$ 的预测值,目的是寻找更小代价的路径结点。式(1)是对每个结点 rhs 值的计算方法:当结点 s 为起始点时,设置其 rhs 值为 0;当结点 s 不为起始点时,设置其 rhs 值为单步代价加上所有父结点中最小值的 g 值。

$$rhs(s) = \begin{cases} 0, & s = s_{start} \\ \min_{s' \in Pred(s)} (g(s') + c(s', s)), & \text{otherwise} \end{cases} \quad (1)$$

对于结点 s 而言,当 $g(s) = rhs(s)$ 时,则认为结点 s 处于局部一致状态,表示能够找到一条从目标点 s_{goal} 到结点 s 的最短路径,并且当所有结点都处于局部一致状态时,意味着从目标点 s_{goal} 到任意一个可达结点都能找到一条最短路径;当 $g(s) > rhs(s)$ 时,则认为结点 s 处于局部过一致状态,表示能够找到从目标点 s_{goal} 到结点 s 更短的一条路径,主要体现在某一块区域由障碍物变为可通行区域;当 $g(s) < rhs(s)$ 时,则认为结点 s 处于局部欠一致状态,表示原本找到的从目标点 s_{goal} 到结点 s 的最短路径代价变大,需要重新计算最短路径,主要体现在某一块区域由可通行区域变为障碍物。

Tracking-D* Lite 算法需要维持一个优先队列 U 用来保存局部不一致状态的结点,这些结点都需要通过一定规则排序之后再选择部分结点进行扩展,进而转化为局部一致状态。队列 U 的优先级计算方式如式(2)所示:

$$k(s) = \begin{bmatrix} k_1(s) \\ k_2(s) \end{bmatrix} = \begin{bmatrix} \min(g(s), rhs(s)) + h(s, s_{goal}) + km \\ \min(g(s), rhs(s)) \end{bmatrix} \quad (2)$$

其中, $k_1(s) = \min(g(s), rhs(s)) + h(s, s_{goal})$, $k_2(s) = \min(g(s), rhs(s))$, $h(s, s_{goal})$ 为启发式函数; km 表示每一次起点改变之后的启发式补偿值,用于保持后续搜索中优先队列 U 中键值的严格升序。

优先队列 U 中的键值优先级顺序的计算方法如式(3)所示, $k(s)$ 值越小表示优先级越高, $k(s)$ 值越大表示优先级越低。

$$k(s) \leq k(s') \Rightarrow \begin{cases} k_1(s) \leq k_1(s') \\ \text{or} \\ k_1(s) = k_1(s') \text{ and } k_2(s) \leq k_2(s') \end{cases} \quad (3)$$

Tracking-D* Lite 算法在 D^* Lite 算法的基础上加上了 I-ARA^{*} 算法中 Delete 队列的思想,用于存储当次搜索过程中不属于以本次搜索起点为根结点的搜索树中的结点,重复利用以用于下一次搜索。Tracking-D* Lite 算法步骤如下所述,流程图如图 4 所示。

(1) 初始化移动目标点和起点位置。

(2)感知起点可视距离内的环境信息,初次计算起点到目标点的最短路径。

(3)移动起始点和目标点位置,若起始点追上目标点,则结束程序;否则,更新环境信息。

(4)删除与本次搜索无关的结点:从前一个目标点开始,删除搜索树中不属于以本次根结点为起点的结点,并放入 Delete 队列。

(5)重用 Delete 队列中的部分结点:对于所有 Delete 结点,若其邻居结点属于本次搜索树(而不在优先队列 U 中的结点),则对该结点重新扩展并加入优先队列 U 中;结束上述步骤后清空 Delete 队列。

(6)重新搜索起点到目标点的最短路径。

(7)重复上述步骤(3)–(6)。

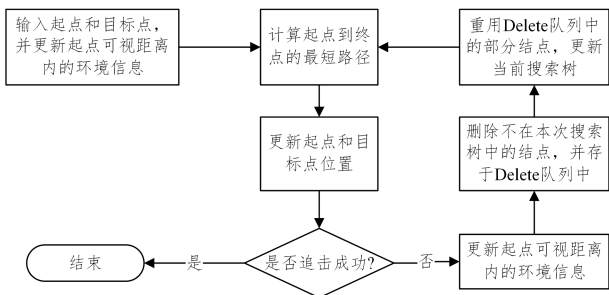


图 4 Tracking-D* Lite 算法流程图

Fig. 4 Flow chart of Tracking-D* Lite algorithm

算法 2 为 Tracking-D* Lite 的主要伪代码。第 3 行 *Initialize()* 将所有结点的 g 值和 rhs 值按照式(1)赋值,并将每个结点的父结点都设为空。初始化之后只有目标点处于局部不一致状态,因此将其加入优先队列 U 中。在 *Initialize()* 执行之后,第 4 行调用 *ComputerShortestPath()* 函数,该函数用于搜索一条从目标点到起始点的最短路径(即类似于 A* 算法),这个过程中将局部过一致的结点的 g 值设为其 rhs 值并扩展其周围的邻居结点;对于局部欠一致的结点,将其 g 值设为无穷大,然后将其重新加入优先队列 U 中,此过程相当于归结到局部过一致状态。第 6–10 行更新起点和目标点位置并计算不同起点之间的启发式差值 k_m 用于保证优先队列中键值的前后一致性,其中 *GetNextGoal()* 用于获取移动目标点的下一个位置。第 12 行设置新目标点的父结点为空值。第 13 行无人机探索附近的新环境并更新地图信息。第 15–21 行删除搜索树中根结点不是本次搜索起点的成员并将其放入 *Delete* 队列,然后重用部分被删除的结点。其中, D 表示 *Delete* 队列,函数 *ReuseDeletedNodes()* 重用 *Delete* 队列中的部分结点并对部分结点重新加入优先队列 U 中。第 21–24 行对地图结点发生改变的边更新代价,并更新每个结点的 rhs 值和优先队列 U , *UpdateNode()* 函数用于更新每一个结点的 g 值和 rhs 值。最后,在第 25 行再次调用 *ComputerShortestPath()* 函数进行重规划,得到新的路径,直到起点追上目标点为止。

算法 2 Tracking-D* Lite 算法

1. $s_{last_start} = s_{start}$
2. $s_{last_goal} = s_{goal}$
3. *Initialize()*
4. *ComputerShorestPath()*
5. while $s_{start} \neq s_{goal}$ do

6. $s_{start} = \operatorname{argmin}_s'_{\text{succ}(s_{start})} (c(s_{start}, s') + g(s'))$
7. Move to s_{start}
8. $k_m = k_m + h(s_{last_start}, s_{start})$
9. $s_{last_start} = s_{start}$
10. $s_{goal} = \text{GetNextGoal}()$
11. $s_{last_goal} = s_{goal}$
12. $\text{parent}(s_{goal}) = \emptyset$
13. Scan graph for changed edge cost
14. if any edge costs changed or $s_{goal} \neq s_{last_goal}$ then
15. if $s_{goal} \neq s_{last_goal}$ then
16. for all node s in the search tree rooted at s_{last_start} but not rooted at s_{start} do
17. $U.\text{remove}(s)$
18. $rhs(s) = g(s) = \infty$
19. $\text{parent}(s) = ?$
20. $D.\text{insert}(s)$
21. *ReuseDeletedNodes()*
22. for all directed edges (u, v) with changed edge costs do
23. Update the edge cost (u, v)
24. *UpdateNode(u)*
25. *ComputerShorestPath()*

以下是未知空间动态目标追踪算法的例子,图 5 所示为一个 5×3 大小的网格地图,机器人(起点)位置在 $B1$,目标点位置在 $D3$,图 5 中 Map 为整个地图的真实环境,机器人可以感知周围一个单位距离的地图环境信息;Heuristics 为机器人在可到达网格点处的启发值,启发值使用的是曼哈顿距离。

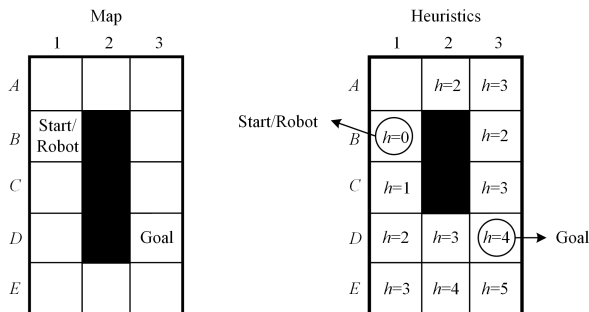


图 5 Tracking-D* Lite 算法例子(第一部分)

Fig. 5 Example of Tracking-D* Lite(Part 1)

图 6 所示为未知空间动态目标追踪算法第一次运行 *ComputerShortestPath()* 函数的过程。本算法的初次搜索迭代步骤与 D* Lite 算法相同,都是从目标点开始搜索并扩展。如图 6 所示,黄色结点代表处于优先队列中的结点,红色方框结点代表下一次要扩展的结点,结点之间的箭头指向表示父结点指向子结点,每个结点都有 g 值、 rhs 值以及优先队列键值。第一次搜索在初始化阶段从目标点开始扩展,将目标点 rhs 值按照式(1)设置为无穷大和零,并根据式(2)计算结点的优先级键值。本次算法中演示使用四联通的扩展方式,通过选择优先队列中优先级最高的结点,扩展该结点的邻居结点并放入优先队列中,接着将扩展过的结点从优先级队列中删除,最后设置该结点的 g 值为 rhs 值。第一次运行完 *ComputerShortestPath()* 函数之后的路径如图 6 中 First path 所示。

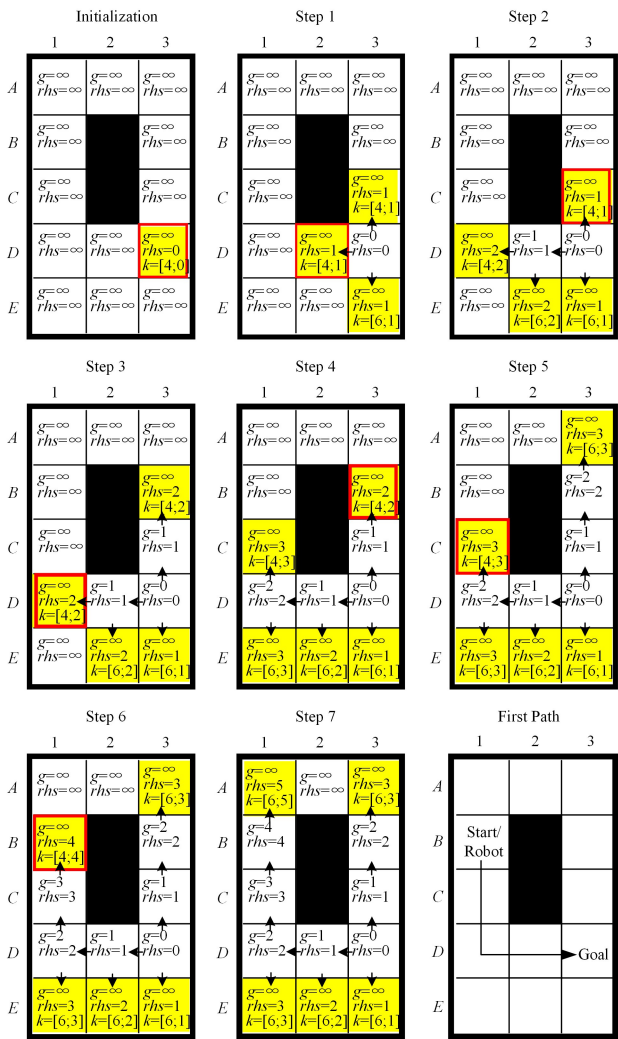


图6 Tracking-D* Lite算法例子(第二部分)(电子版为彩图)

Fig. 6 Example of Tracking-D* Lite(Part 2)

在搜索出第一条路径之后,如图7所示,机器人按照规划处的路径移动一个单位距离到达C1,目标点随机一个单位距离到达C3,机器人感知到周围环境信息,发现结点D2为新障碍物,更新地图信息并设置新的启发值,同时更新 $k_m = 1$ 。

在第二次搜索中,需要对删除与本次搜索无关的结点,即从前一个目标点开始,使用深度优先搜索删除搜索树中根结点不是本次搜索起点的成员,将其并放入 Delete 队列。

如图8所示,LastState 为上一次搜索之后遗留下的

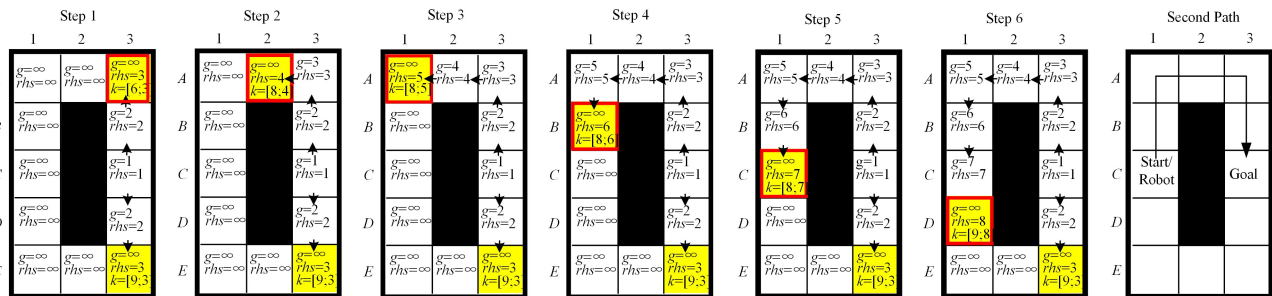


图9 Tracking-D* Lite算法例子(第五部分)(电子版为彩图)

Fig. 9 Example of Tracking-D* Lite(Part 5)

本文提出的 Tracking-D* Lite 算法能够在未知区域对移动的目标点进行追踪。算法的主要思想是对每一次移动之后的目标点进行路径重规划,在重规划的过程中会对上一次搜索

状态,其中包含一棵以 D3 为根结点的搜索树(因为 D3 是第一次搜索过程中的目标点),而第二次搜索的目标是 C3,所以需要删除搜索进行修剪。如图8的 Delete 图所示,灰色的网格代表已经被放入 Delete 队列中的结点,这些结点都是以第一次搜索中 D3 为根结点的子结点组成。我们可以发现,在 Delete 图中,以 C3 为根结点的子树被保留了下来,所以在删除结点的这个过程中保留了上一次搜索结果的部分状态信息。然后需要对 Delete 队列中的部分结点进行重新使用,对于所有 Delete 结点,若其邻居结点属于本次搜索树中(而非优先队列中的结点),则对该结点重新扩展并加入优先队列。如图8中 Reuse DeleteNode 所示,将结点 D3 加入了优先队列。

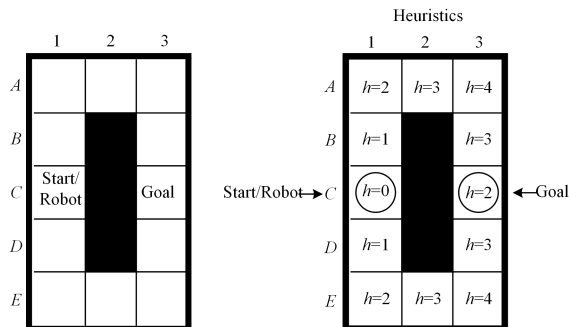


图7 Tracking-D* Lite算法例子(第三部分)

Fig. 7 Example of Tracking-D* Lite(Part 3)

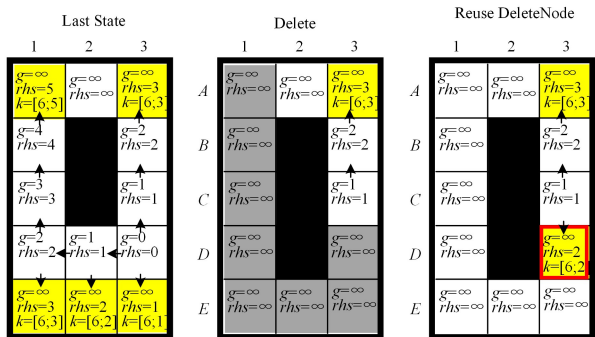


图8 Tracking-D* Lite算法例子(第四部分)(电子版为彩图)

Fig. 8 Example of Tracking-D* Lite(Part 4)

如图9所示,第二次搜索过程与前面的搜索过程类似,调用 ComputerShortestPath() 函数直到搜索结束,搜索出第二条路径。

的结点扩展次数,从而提升搜索速度和效率并达到追踪移动目标的效果。

5 原型系统实现与实验验证

基于所提出的算法,本文在 ROS 平台设计并实现了一套基于行为树调度的多无人机协同探索未知室内空间的系统,对 Tracking-D * Lite 算法进行性能实验,并通过 Gazebo 仿真搭建了未知室内空间仿真场景,用多无人机协同探索以验证系统可信性。最后,将本方法与 SGBA 方法进行实验对比,进一步验证本文所提出方法对探索未知室内空间的有效性。

5.1 原型系统的设计与实现

5.1.1 原型系统设计

本文综合所提出的算法,进行了多无人机协同探索原型系统的设计,总体上包括 4 个部分:硬件层、软件层、总控制软件和实验验证层。

硬件层主要涉及到本文算法的相关传感器,包括激光雷达、基于光流的视觉里程计、无线网络传感器,能够为软件层提供必要的数据库支持并与总控制软件相连接。在 Gazebo 仿真环境中,激光雷达和视觉里程计可通过 hector_quadrotor 无人机模型模拟获得数据,该模型集成了激光雷达、里程计等功能,为完成感知、测距、坐标计算以及避障提供基础,从而支撑上层模块的运行。

软件层主要负责未知室内空间的探索,主要包括感知定位模块、地图边界环绕模块、接力追踪模块、电量控制模块,同时基站上的总控制软件通过与无人机通讯共同完成地图边界绘制模块。

感知定位模块用于无人机在未知空间中的障碍物感知、障碍物测距和定位。该模块利用激光雷达获得前、后、左、右、上、下 6 个方向上到障碍物的距离,若得到的某一方向上的障碍物距离无人机小于等于阈值 α (α 设为 1m,即无人机可视距离),则判定该方向上存在障碍物并且无人机处于该障碍物附近。感知定位模块的定位功能通过基于光流法的单目视觉里程计实现,在每架无人机机身下搭载一个单目相机再通过基于光流的视觉里程计方法,可以计算出无人机相对于起点移动的距离和方向,从而得到当前无人机的坐标位置。

地图边界环绕模块与感知定位模块、接力追踪模块、地图边界绘制模块相连,与地图边界绘制模块同时运行,采用 Wall-Around 算法控制无人机围绕边界进行飞行,从而获取未知室内空间的边界信息,将未知室内空间的边界信息发送给位于基站的总控制软件实时绘制无人机的飞行轨迹。

接力追踪模块与感知定位模块、地图边界环绕模块相连,采用 Tracking-D * Lite 算法控制两架无人机之间的接力,在已知起点和目标点位置的情况下,解决在未知空间中对移动目标点的接力追踪。

地图边界绘制模块与地图边界环绕模块相连,与地图边界环绕模块同时运行,从地图边界环绕模块接收每架无人机的飞行轨迹,得到每一架无人机在地图边界飞行时的部分边界轮廓图,然后依次通过每一架无人机将边界轮廓图进行拼接,最终得到整个未知室内空间的边界轮廓图。

电量控制模块与地图边界环绕模块、接力追踪模块相连接。该模块一方面负责无人机剩余电量的实时监控,随无人机

启动和降落而开启和关闭;另一方面负责触发从地图边界环绕模块转换到接力追踪模块的电量条件,当无人机电量剩余小于等于 30% 时,电量控制模块通知基站总控制软件派出下一架无人机触发,从而进入接力追踪模块。

总控制软件包括控制行为树控制调度、显示无人机实时轨迹和地图轨迹拼接,主要完成对所有无人机的调度和绘制最终未知边界轮廓图。

实验验证层建立在 Gazebo 仿真平台上。通过该平台搭建一个未知室内空间仿真,使用多架无人机协同探索,最终利用 Rviz 绘制出整个空间的边界轮廓图,用于系统效果验证。

5.1.2 原型系统实现

本文基于系统架构设计进行了具体的系统实现,原型系统中主要包含两种算法插件包:一个 Simulation 类和一个 UAV 类。两种算法插件包分类是 Wall-Around 算法和 Tracking-D * Lite 算法的实现,分别继承自 BasePlanner 类,为 Simulation 类提供规划器。Simulation 类依赖于 UAV 类,主要完成对无人机的控制、算法的调用以及最后边界轮廓图的实施绘制。BasePlanner 类主要为不同的算法提供同一接口实现,使用 getPlannerName() 作为同一接口,在 Wall-Around 算法和 Tracking-D * Lite 算法中分别实现,主要是为无人机提供下一步将要前进的方向和位置。

UAV 类为每一架无人机的实现类,主要包括控制无人机前进方向、旋转角度和订阅无人机当前坐标位置、速度,同时加载不同的算法插件,达到导航的目的。

Simulation 类实现总控制软件的功能,主要负责调度无人机,为不同的无人机搭载合适的算法,通过订阅无人机发出的位置信息完成对地图边界轮廓图的绘制。

5.2 Tracking-D * Lite 算法对比实验

本实验的目的是验证 Tracking-D * Lite 算法性能,在不同无人机视距下主要对追击点的追击时间、追击移动步数和、结点扩展次数和目标点未停止时追击成功率这 4 个指标进行比较。本实验将 Tracking-D * Lite 算法和 Repeated-D * Lite 算法(重复调用 D * lite 算法)进行比较,在 Ubuntu 16.04 系统下,使用 C++ 语言开发实验。本实验采用大小为 60 * 60 的二维网格地图进行实验,一共进行 64 组实验,每一组实验都随机给定一个追击点和目标点并且位置不同,目标点的移动路径不同。对比实验中目标点的移动速率和移动路径相同,实验结果数据包括追击时间、追击点移动步数、结点扩展次数以及在目标点未停止时追击的成功率。如表 1 所列,实验中对两种算法在机器人视野分别为 2、4 和 6 的情况进行了测试,可以看出 Tracking-D * Lite 算法的平均搜索时间、平均移动步数、平均扩展结点数以及成功率都好于 Repeated-D * Lite 算法。从平均追踪时间来看,Tracking-D * Lite 相比于 Repeated-D * Lite 能够以更少的时间追踪到目标点,并且随着无人机视距的增加,两种算法消耗的时间都会增加,但是 Tracking-D * Lite 的用时远比 Repeated-D * Lite 用时少并且用时增长幅度更小。对于平均移动距离而言,同一个算法下不同视距的结果相差不大,但是 Tracking-D * Lite 比 Repeated-D * Lite 移动的距离更短,这是 Tracking-D * Lite 的优势所在。与平均移动距离的结果类似,对于平均扩展结点次数,Tracking-D * Lite 比 Repeated-D * Lite 要少很多,并且在

不同视距下的节点扩展次数相差不大。最后,对于无人机追击到未停止目标的成功率,从单个算法的角度来说,两种算法的追击成功率都随着视距的增加而增大,但是 Tracking-D * Lite 比 Repeated-D * Lite 效果更好。

表 1 Tracking-D * Lite 与 Repeated-D * Lite 的对比

Table 1 Comparison of Tracking-D * Lite and Repeated-D * Lite

算法	视距	平均追击时间/ms	平均移动距离	平均扩展次数	追击成功率/%
Tracking-D * Lite	2	2863.727	61.250	29596.735	83
	4	2997.994	64.530	27582.955	85
	6	3370.850	66.440	26755.190	93
Repeated-D * Lite	2	6864.102	88.797	78877.591	19
	4	11421.131	81.950	85398.840	37
	6	16354.322	84.850	80229.755	40

5.3 仿真对比实验

本仿真实验用于验证研究内容的合理性和有效性。实验内容为搭建一个仿真室内空间,使用多架四旋翼无人机模型通过基于行为树调度的多无人机未知区域协同探索方法完成对整个空间的搜索,并绘制出地图轮廓图。同时,将本方法与 SGBA^[19]方法进行对比,实验对比的指标为在探索相同的未知室内空间时的单架无人机的探索覆盖区域百分比和所有无人机的总覆盖区域百分比。

仿真验证实验在 Ubuntu 系统下使用 C++ 语言开发,基于 ROS, Gazebo 以及 Rviz 平台进行仿真模拟,主要使用的是 hector_quadrotor 无人机模型工具包。

5.3.1 仿真实验

本方法的仿真实验基于行为树对无人机和算法进行调度和切换。其中某一实验场景如图 10 所示,为 4 队 8 架无人机对仿真场景进行探索。首先集成 Wall-Around 算法,Wall-Around 算法主要是在网格地图中运行的,需要在仿真环境中对网格地图的坐标进行替换,主要的实现方式是对原点进行偏移,使得网格二维地图的原点与仿真中的三维原点重合。利用无人机模型前、后、左、右 4 个方向的激光雷达来对无人机周围环境进行感知,从而进行有效的避障和路径规划。

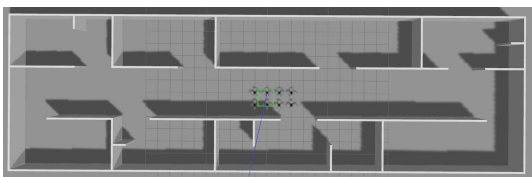


图 10 仿真实验场景

Fig. 10 Simulation experiment scene

接着将 Tracking-D * Lite 算法集成到仿真空间中,若当前探索无人机出现电量不足或意外情况时,基站会发出指令通知下一架无人机通过行为树调度使用 Tracking-D * Lite 算法接力前一架无人机,在到达前一架无人机停止的地点后,继续使用 Wall-Around 算法绕墙,最后进行轨迹提取和融合,并将所有无人机的轨迹进行拼接。如图 11 所示,绿色、蓝色、红色和黄色轨迹分别为上一架无人机探索室内空间边界实时显示的轮廓图,其余天蓝色轨迹为剩余无人机分别对前序无人机进行接力的轨迹。图 12 为四队无人机对整个未知室内空间探索完毕之后绘制的边界轮廓图,不同的颜色代表不同的无人机探索的地图边界。



图 11 Rviz 实时显示轨迹(电子版为彩图)

Fig. 11 Real-time display of trajectories in Rviz

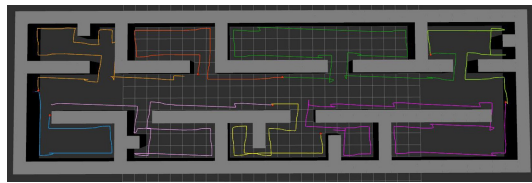


图 12 边界轮廓图(电子版为彩图)

Fig. 12 Boundary contour map

5.3.2 对比实验

为对比本论文提出的基于行为树的多无人机未知室内空间探索方法与 SGBA^[19]方法在探索同一未知室内空间的性能差异,在实验设计上,设置了 10 个大小、边界不同的室内空间,分别使用 2,3,4,6,8 架无人机对未知室内空间各进行了 10 次探索。实验对比指标包括单架无人机的探索覆盖区域百分比和所有无人机总覆盖区域百分比。

图 13 所示为本文方法和 SGBA 方法在无人机数量不同时单架无人机的探索覆盖区域百分比变化情况,可以看到 SGBA 算法单架无人机覆盖百分比在不同无人机数量的情况下都处于 10% 以下,而本方法中单架无人机探索覆盖百分比随无人机数量的增加而减少,但是总体都高于 SGBA 方法,因为单架无人机探索区域会因总无人机数量的增多而减小,另外由于 SGBA 方法需要进行返航,所以它的单架无人机探索覆盖百分比比较小。本文方法和 SGBA 方法在无人机数量不同时所有无人机总探索覆盖区域百分比变化情况如图 14 所示,可以看出随着无人机数量的增多,两种方法的覆盖面积百分比都逐渐增大,但是本方法总是优于 SGBA 方法,能够在相同的无人机数量上探索更多的未知空间。

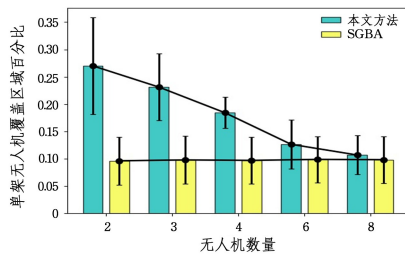


图 13 单架无人机探索覆盖区域百分比

Fig. 13 Percentage of coverage areaper UAV

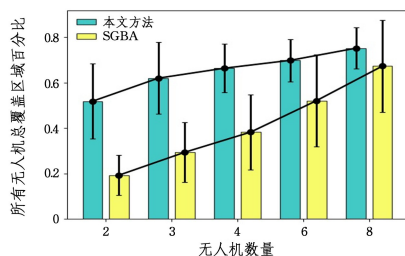


图 14 所有无人机总探索覆盖区域百分比

Fig. 14 Percentage of total exploration coverage area

综上所述,本论文所提出的基于行为树的多无人机未知室内空间探索方法能够有效地探索未知室内空间,行为树对无人机和算法进行调度与切换能够高效地协调探索过程的状态转化。因此,本方法能够胜任对未知室内区域的探索任务。

结束语 为了更加简洁高效地解决无人机在未知室内环境中的探索问题,本文提出了基于行为树调度的多无人机未知区域协同探索方法。该方法基于行为树框架,对无人机和算法进行调度和切换,在无人机上使用 Wall-Around 算法和 Tracking-D* Lite 算法,利用 Wall-Around 算法对未知室内区域的边界进行探索,使用 Tracking-D* Lite 算法对两架无人机之间进行接力协作,最后完成对整个未知室内环境的探索任务。本文所提出的 Tracking-D* Lite 算法能够有效地在未知区域对移动目标进行追踪,并且通过行为树对 Wall-Around 算法和 Tracking-D* Lite 算法进行合理的切换,在仿真实验中取得了较好的效果。

参 考 文 献

- [1] FLOREANO D, WOOD R J. Science, technology and the future of small autonomous drones[J]. *Nature*, 2015, 521(7553): 460-466.
- [2] CADENA C, CARLONE L, CARRILLO H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age [J]. *IEEE Transactions on Robotics*, 2016, 32(6): 1309-1332.
- [3] FLÓREZ-PUGA G, GÓMEZ-MARTÍN M, DÍAZ-AGUDO B, et al. Dynamic expansion of behaviour trees[C]// *Proceedings of Artificial Intelligence and Interactive Digital Entertainment Conference*. AAAI Press, 2008: 36-41.
- [4] LIU R F, WANG J S, ZHANG Y L, et al. Research progress and application of behavioral tree technology [J]. *Computer and Modernization*, 2020(2): 76-82, 88.
- [5] ROBERTSON G, WATSON I. Building behavior trees from observations in real-time strategy games[C]// *2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, 2015: 1-7.
- [6] MCGUIRE K N, DE CROON G C H E, TUYLS K. A comparative study of bug algorithms for robot navigation. *Robotics and Autonomous Systems*, 2019, 121: 103261.
- [7] KOENIG S, LIKHACHEV M. Fast replanning for navigation in unknown terrain [J]. *IEEE Transactions on Robotics*, 2005, 21(3): 354-363.
- [8] SUN X, URAS T, KOENIG S, et al. Incremental ARA* - an incremental anytime search algorithm for moving-target search [C]// *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling*, 2012: 243-251.
- [9] HE R, PRENTICE S, ROY N. Planning in information space for a quadrotor helicopter in a GPS-denied environment[C]// *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008.
- [10] PRENTICE S, ROY N. The belief roadmap: Efficient planning in linear pomdps by factoring the covariance[C]// *Proc. ISRR*. 2007.
- [11] PRAVITRA C, CHOWDHARY G, JOHNSON E. A compact exploration strategy for indoor flight vehicles[C]// *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011: 3572-3577.
- [12] STIRLING T, ROBERTS J, ZUFFEREY J C, et al. Indoor navigation with a swarm of flying robots[C]// *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012: 4641-4647.
- [13] GRZONKA S, GRISETTI G, BURGARD W. A fully autonomous indoor quadrotor [J]. *IEEE Transactions on Robotics*, 2012, 28(1): 90-100.
- [14] BI Y, QIN H, SHAN M, et al. An autonomous quadrotor for indoor exploration with laser scanner and depth camera [C]// *IEEE International Conference on Control and Automation (ICCA)*. 2016: 50-55.
- [15] SAMPEDRO C, RODRIGUEZ-RAMOS A, BAVLE H, et al. A Fully-Autonomous Aerial Robot for Search and Rescue Applications in Indoor Environments using Learning-Based Techniques [J]. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2019, 95(2): 601-627.
- [16] PHAM H X, LA H M, FEIL-SEIFER D, et al. Autonomous uav navigation using reinforcement learning [J]. *arXiv:1801.05086*, 2018.
- [17] WALKER O, VANEGAS F, GONZALEZ F, et al. A deep reinforcement learning framework for uav navigation in indoor environments [C]// *2019 IEEE Aerospace Conference*. IEEE, 2019: 1-14.
- [18] WALKER O, VANEGAS F, GONZALEZ F, et al. Multi-UAV Target-Finding in Simulated Indoor Environments using Deep Reinforcement Learning [C]// *2020 IEEE Aerospace Conference*. IEEE, 2020: 1-9.
- [19] MCGUIRE K N, DE WAGTER C, TUYLS K, et al. Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment [J]. *Science Robotics*, 2019, 4(35): eaaw9710.
- [20] COLLEDANCHISE M, ÖGREN P. Behavior trees in robotics and AI: An introduction [M]. CRC Press, 2018.
- [21] COLLEDANCHISE M, ÖGREN P. How behavior trees modularize robustness and safety in hybrid systems [C]// *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014: 1482-1488.
- [22] GHZOULI R, BERGER T, JOHNSEN EB, et al. Behavior trees in action: a study of robotics applications [C]// *Proceedings of the 13th ACM SIGPLAN International Conference on Software Language Engineering*. 2020: 196-209.
- [23] IOVINO M, SCUKINS E, STYRUD J, et al. A survey of behavior trees in robotics and ai [J]. *arXiv:2005.05842*, 2020.
- [24] MARZINOTTO A, COLLEDANCHISE M, SMITH C, et al. Towards a unified behavior trees framework for robot control [C]// *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014: 5420-5427.
- [25] JONES S, STUDLEY M, HAUERT S, et al. Evolving behaviour trees for swarm robotics [M]// *Distributed Autonomous Robotic Systems*. Cham: Springer, 2018: 487-501.
- [26] KUCKLING J, LIGOT A, BOZHINOSKI D, et al. Behavior trees as a control architecture in the automatic modular design of robot swarms [C]// *International Conference on Swarm Intelligence*. Cham: Springer, 2018: 30-43.

[27] XU J, LIU W, LANG F, et al. Distance measurement model based on RSSI in WSN[J]. *Wireless Sensor Network*, 2010, 2(8):606.

[28] KAMON I, RIVLIN E, RIMON E. A new range-sensor based globally convergent navigation algorithm for mobile robots [C]//*Proceedings of IEEE International Conference on Robotics and Automation*. 1996:429-435.

[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 and Cybernetics*, 1968, 4(2): 100-107.

[30] HARABOR D, GRASTIEN A. Online graph pruning for path-finding on grid maps[C]//*Proceedings of the AAAI Conference on Artificial Intelligence*. 2011.

[31] STENTZ A. The focussed d^* algorithm for real-time replanning

[C]//*IJCAI*. 1995:1652-1659.



SHI Dian-xi, born in 1966, Ph.D, professor, Ph.D supervisor. His main research interests include distributed object middleware technology, adaptive software technology, artificial intelligence and robot operation systems.



ZHANG Yong-jun, born in 1966, Ph.D, professor. His main research interests include artificial intelligence, multi-agent cooperation, machine learning and feature recognition.