软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(9):2678−2690 [doi: 10.13328/j.cnki.jos.005937]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

扫地机器人增强位姿融合的 Cartographer 算法及系统实现*

张 亮[1,3], 刘智宇[1], 曹晶瑛[2], 沈沛意[1], 蒋得志[1], 梅 林[1], 朱光明[1], 苗启广[1]

[1](西安电子科技大学 计算机科学与技术学院,陕西 西安 710071)

[2](北京石头世纪科技有限公司,北京 100039)

[3](上海宽带技术及应用工程研究中心,上海 200436)

通讯作者: 沈沛意, E-mail: pyshen@xidian.edu.cn

摘 要: Cartographer 是谷歌在 2016 年开源的一个可以在多传感器配置下实现低计算资源消耗的 SLAM 算法框架.针对原有 Cartographer 中位姿融合不准确、存在延迟的问题,首先设计了一种基于位姿增量的多传感器位姿融合方法;随后,针对扫地机器人 Player 平台,设计并实现了基于增强 Cartographer 算法的多模块 SLAM 系统;最后,通过 Cartographer 数据集的实验分析和真实场景的实际测试,验证了增强 Cartographer 算法的有效性以及 SLAM 系统在 Player 机器人平台上的可用性.

关键词: 扫地机器人;Cartographer;多传感器位姿融合;SLAM 系统

中图法分类号: TP242

中文引用格式: 张亮,刘智宇,曹晶瑛,沈沛意,蒋得志,梅林,朱光明,苗启广.扫地机器人增强位姿融合的 Cartographer 算法及系统实现.软件学报,2020,31(9):2678−2690. http://www.jos.org.cn/1000-9825/5937.htm

英文引用格式: Zhang L, Liu ZY, Cao JY, Shen PY, Jiang DZ, Mei L, Zhu GM, Miao QG. Cartographer algorithm and system implementation based on enhanced pose fusion of sweeping robot. Ruan Jian Xue Bao/Journal of Software, 2020,31(9): 2678−2690 (in Chinese). http://www.jos.org.cn/1000-9825/5937.htm

Cartographer Algorithm and System Implementation Based on Enhanced Pose Fusion of Sweeping Robot

ZHANG Liang[1,3], LIU Zhi-Yu[1], CAO Jing-Ying[2], SHEN Pei-Yi[1], JIANG De-Zhi[1], MEI Lin[1], ZHU Guang-Ming[1], MIAO Qi-Guang[1]

[1](School of Computer Science and Technology, Xidian University, Xi’an 710071, China)

[2](Beijing Roborock Technology Co., Ltd, Beijing 100039, China)

[3](Shanghai Broadband Network Technology and Application Engineering Research Center, Shanghai 200436, China)

Abstract: Cartographer is Google’s 2016 open source SLAM algorithm framework for low computational resource consumption in multi-sensor configurations. In this study, due to the inaccurate middle posture fusion and delay of the original Cartographer, a multi-sensor posture fusion method based on posture increment was designed. Subsequently, the multi-module SLAM system based on enhanced Cartographer algorithm was designed and implemented for the cleaning robot Player platform. Finally, the effectiveness of the enhanced Cartographer algorithm and the usability of the SLAM system on the Player robot platform were verified by the experimental analysis of the Cartographer data set and the actual test of the real scenario.

* 基金项目: 国家自然科学基金(61401324, 61305109); 陕西省重点研发计划(2018ZDXM-GY-36)

Foundation item: National Natural Science Foundation of China (61401324, 61305109); Shaanxi Province Key Research and Development Program (2018ZDXM-GY-36)

本文由“智能嵌入式系统”专题特约编辑王泉教授、吴中海教授、陈仪香教授、苗启广教授推荐.

收稿时间: 2019-06-27; 修改时间: 2019-08-18; 采用时间: 2019-11-02; jos 在线出版时间: 2020-01-13

CNKI 网络优先出版: 2020-01-14 11:26:53, http://kns.cnki.net/kcms/detail/11.2560.TP.20200114.1126.019.html

Key words: sweeping robot; Cartographer; multi-sensor pose fusion; SLAM system

同步定位与地图构建(simultaneous localization and mapping,简称 SLAM)产生于机器人领域[1],它指的是:机器人从未知环境的未知地点出发,在运动过程中,通过重复观测到的各传感器数据定位自身位置,再根据自身位置构建周围环境的增量式地图,从而达到同时定位和地图构建的目的.服务机器人为实现高智能化,在复杂的非结构场景完成任务,SLAM 技术是必要条件[2,3].

目前广泛应用于 SLAM 上的传感器分为摄像头和激光雷达两种,对应视觉 SLAM[4]和激光 SLAM.在实际使用中,单个传感器难以适应复杂多变的环境要求,需要惯性测量单元以及里程计等传感器辅助,增强环境的适应性.视觉 SLAM 的优点是具有语义信息、成本低,但易受光照影响,同时计算量较大.激光 SLAM 具有可靠性高、精度高、地图可用于路径规划等优点,是目前最主流的定位导航方法,缺点是无语义信息且成本较高.

SLAM 本质是个系统状态估计问题,即:在给定系统输入的条件下,估计出机器人的位姿和地图点的坐标,延伸出基于贝叶斯滤波[5]以及基于图优化[6]的两类 SLAM 算法.

基于贝叶斯滤波的方法是一种贝叶斯迭代状态估计理论,即先对机器人运动进行建模,构造出贴合物理场景的运动方程和观测方程,进行状态预测和测量更新.滤波类算法最大的问题是无法处理大尺度场景的建图.由于滤波类算法是基于递归计算的,下一时刻的估计值依赖于上一时刻的估计,因而在大尺度场景下,由于系统参数和传感器观测数据的不准确性,会造成误差慢慢累积,一旦当前时刻的估计出现偏差,无法修正该误差,最终将无法获得一致性的地图.基于图优化 SLAM 的出现,解决了大尺度场景建图的问题.图优化 SLAM 分为前端和后端两个模块:前端先从里程计获得初始的位姿,然后通过激光雷达数据连续帧匹配,求得激光约束后的位姿;后端采用回环检测,构造闭环约束,通过最小化观测和估计残差求得优化后的位姿[7].在前端中会产生由估计所引入的误差,随着运动范围的扩大,该误差会逐渐累积,导致最后产生错误的结果.为解决该问题,在后端使用闭环检测进行全局优化,通过判断机器人是否回到历史中的某一点,添加约束较强的闭环约束,消除误差累积.

图优化的 SLAM 核心是构建位姿图,图中每个节点表示不同时刻机器人的位姿,边由各个节点间的变换关系构成.其中:节点是优化变量;边是误差项,随着时间的累积,这个误差会越来越大.图构建完成后,需要调整机器人位姿去满足边构成的约束,最终目的是求解机器人位姿使目标函数即误差平方函数最小,可表示为

$$F(x)=\sum_{i,j\in C} e(x_i,x_j,z_{i,j})\Omega_{i,j}e(x_i,x_j,z_{i,j}) \tag{1}$$

其中,x_i,x_j 表示机器人位姿,$z_{i,j}$ 表示测量值,$\Omega_{i,j}$ 即权重矩阵.

本文的研究内容包括:

(1) 以应用日趋广泛的扫地机器人为研究对象,依托于低成本硬件嵌入式扫地机器人,以 Cartographer 算法[8]为核心,在 Player 系统平台上,融合激光雷达、惯性导航仪以及里程计等多种传感器数据来感知环境、构建地图,并在各种实际场景进行实验;

(2) 针对 Cartographer 中,位姿融合采用 UKF(unscented kalman filter)滤波方法带来计算量较大且提供下一时刻位姿估计不够准确的问题,本文提出了一个基于位姿增量的多传感器位姿融合方法的增强 Cartographer 算法;

(3) 基于扫地机器人 Player 平台,实现了一个基于增强 Cartographer 算法的 SLAM 系统,系统分为 6 个模块:系统配置模块完成初始配置,其中,接口通信模块完成与平台上其他系统的交互,Player 传感器数据交互模块完成与 Player 平台的交互,数据管理及分发模块完成与算法交互,算法模块完成建图,数据处理算法模块进行数据处理.

1 增强位姿融合的 Cartographer 算法

本节在 Cartographer 算法的基础上设计了基于增强型的位姿融合方法:首先给出增强型位姿融合的 SLAM 算法框架,该算法包含局部 SLAM(前端)和全局 SLAM(后端)两部分;然后,对局部 SLAM 的组成和基于位姿增量的多传感器位姿融合进行分析设计;最后分析了全局优化过程中基于分支定界法加速回环检测的算法.

1.1 Cartographer算法架构

Cartographer 是一个实时的室内建图算法,能生成分辨率 r=5cm 的栅格地图[9,10].在前端将最新的激光雷达扫描数据在相邻的子图上(整个地图的一小块)完成扫描匹配,得到一个在短时间内准确的最佳插入位置(位姿)后,将扫描插入到子图中.扫描匹配中,位姿估计的误差会在整个地图中随时间逐渐累积,在后端中,通过回环检测加约束进行优化消除误差.

图 1 为 Cartographer 算法框架图.整个框架分为数据源、局部 SLAM 以及全局 SLAM 以及数据融合等部分,其中,红虚线框表示本文设计的位姿融合方法所处位置,红框表示融合的数据.

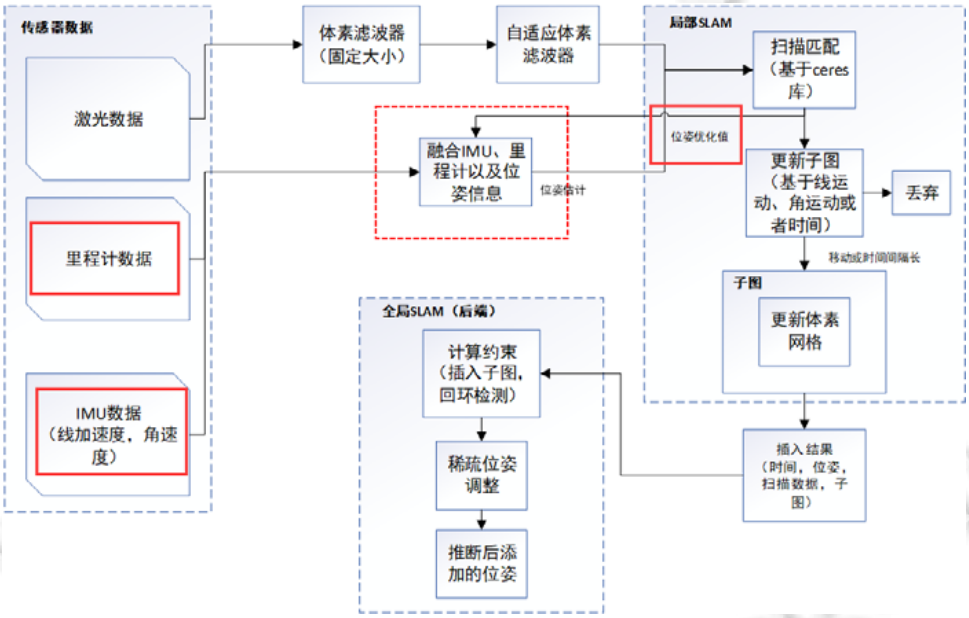


Fig.1 Cartographer algorithm framework diagram

图 1 Cartographer 算法框架图

Cartographer 是基于图优化的 SLAM,采用误差累积小、计算成本低的 Scan to Map 匹配方式,而不是会造成误差快速累积以及计算成本高的 Scan to Scan 匹配方式[11,12].整个算法分为局部 SLAM 和全局 SLAM 两部分,两部分均会对激光雷达观测的位姿(即扫描)进行优化.

- 在局部 SLAM 中,使用非线性优化将每个连续的扫描与子图进行匹配,优化将扫描与子图对齐,也就是扫描匹配,生成较好的子图.在这部分中,设计了基于位姿增量的多传感器位姿融合方法,更快速、更准确地为下一时刻提供估计位姿;
- 而在全局 SLAM 中,将局部 SLAM 累积的误差消除,将不同的子图以最匹配的位姿融合.

1.2 局部SLAM的组成与位姿优化

一帧扫描即一次激光雷达采集的数据,子图由多个扫描构成,构建子图是一个重复对齐扫描和子图坐标帧的迭代过程.多个扫描构成的子图由概率网格表示,将给定分辨率的离散网格点映射为值,这些值是网格点被占用的可能.对于每个网格点,定义对应的像素以包含最接近这个网格点的所有点.

在将一个扫描插入子图之前,使用基于 Ceres[13]的扫描匹配器对插入的扫描位姿相对于当前匹配的子图进行局部优化.扫描器的目标是在子图中找到使插入扫描中扫描点概率最大的扫描位姿,可以将此问题转变为一个非线性最小二乘问题:

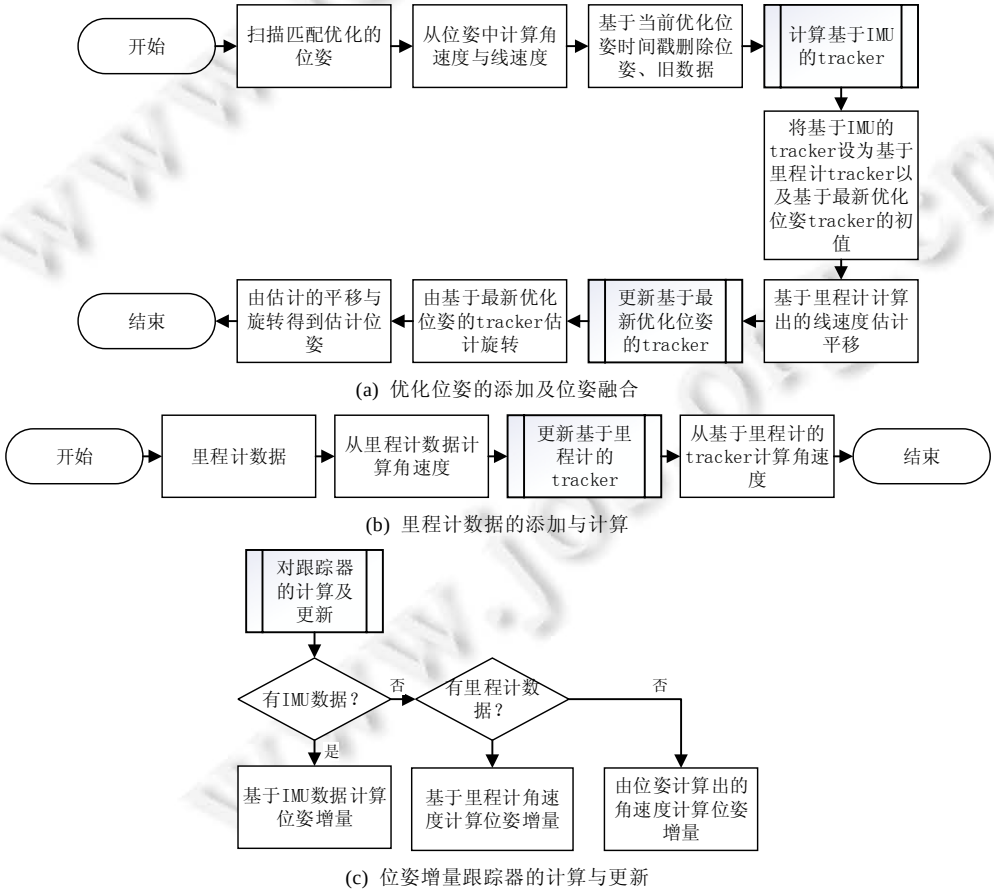
$$\arg\min_{\xi}\sum_{k=1}^{K}(1-M_{smooth}(T_{\xi}h_k))^2 \tag{2}$$

其中,T_{ξ}表示扫描转换到对应子图帧中的位姿变换,位姿变换将扫描点 h_k 从扫描转换到子图帧中,平滑函数 $M_{smooth}:R^2\rightarrow R$ 将每个扫描点概率值平滑映射到局部子图中,目标是使插入扫描的所有扫描点概率值最大,构建非线性最小二乘目标函数.

1.3 基于位姿增量的多传感器位姿融合

局部 SLAM 的目标是使用除了激光雷达之外的传感器数据来预测下一个扫描应该被插入到子图的位置.在原本的位姿融合估计部分中,Cartographer 算法利用 IMU(inertial measurement unit)构建预测模型,扫描匹配器与里程计构建观测模型,采取无损滤波器进行运动预测.当扫描匹配器估计的位姿或者里程计数据到来时,利用观测值与系统状态变量均值的矢量差修正状态变量(只有 IMU 数据到来时,只进行预测不进行更新)及其协方差.这种方法中存在的问题有:对各数据源的时间戳对准不够精确;IMU 位姿的融合过程只有在扫描匹配器估计的位姿或里程计数据到来时才能做状态更新,这样会造成位姿估计较为滞后;没有通过线速度而是直接使用里程计数据估计相邻时刻位姿增量,精度较低.

针对基于 UKF 方法[14]中位姿融合存在的问题,本文设计了一种基于位姿增量的多传感器位姿融合方法.图 2 给出了该方法的具体描述.



(a) 优化位姿的添加及位姿融合

(b) 里程计数据的添加与计算

(c) 位姿增量跟踪器的计算与更新

Fig.2 Data addition and pose fusion

图 2 数据添加及位姿融合

该方法基于时间戳对齐数据,可以在 IMU 以及里程计等辅助传感器的帮助下提供更精确的下一时刻包含

旋转和平移的位姿估计.在位姿融合中,因为IMU和里程计的采集速率比激光雷达传感器高很多,因而在一定时间段内可能没有激光扫描匹配估计的位姿数据,这时需要通过IMU和里程计数据进行推断;同时,因为这两个传感器存在相对激光扫描较大的累积误差,所以在有估计位姿的时候对两者进行一些更新和初始化消除累积误差;同时,在估计位姿的旋转量时,可以在没有 IMU 数据时利用之前的位姿或者里程计估算出的速度进行模拟,在一定程度上可以保持位姿数据的稳定性.本方法将所有数据进行缓存,放在双端队列中,便于新数据的添加和旧数据的删除.算法不断轮询处理来自扫描匹配器优化的位姿、IMU 数据以及里程计数据.在获取最后估计位姿以及处理 IMU 和里程计两种传感器数据期间,位姿的变化由 3 个跟踪器表示,基于 IMU 数据,对这 3 个跟踪器进行更新.最后获取融合完成的位姿,提供给扫描匹配器作为初值.

对各数据源的数据进行了严格的时间对齐后,能够为后续的非线性优化提供一个较为准确的估计初值.同时,在使用该位姿融合方法后,需要在算法外部配合激光雷达的传感器发布速率,同时添加一个发布局部位姿优化的模块,将局部优化的位姿类似于传感器数据一样作为一个外部数据源.

1.4 基于回环检测的全局优化

由于扫描仅与时间相近的子图匹配,其包含扫描数量少,局部扫描匹配方法会逐渐累积误差.在短时间的连续扫描中,累积的误差很小,在局部便可进行优化;而长期建图累积的误差,则需要通过检测回环通过稀疏位姿调整[15]完成全局优化来消除.

在算法中,被插入扫描的相对位姿存储在内存中,用于在回环闭合时进行优化.扫描匹配器一直在后端运行,如果找到一个高于一定阈值的匹配,就将相应的相对位姿添加到优化约束中.与扫描匹配一样,全局优化也被转化为一个非线性最小二乘问题,这样,在有其他数据时能够较容易地添加残差,残差即约束与真实值之差.约束被添加后,每隔一段时间,通过优化计算全局位姿最优解.

在回环检测中,最优的匹配是精确到像素的匹配.原始算法以一定的角度步长遍历搜索窗口内的每个像素,但在目前的搜索量级下,运行速度非常慢.

Cartographer 中采用基于 DFS(depth first search)的分支定界法进行回环的检测,这种方法首次在混合整数线性规划中提出,在 Land 和 Clausen 的论文中[16,17]有相关概述.该方法的主要思想是:将可能性子集表示为一棵树中的节点,其中:根节点表示所有可能的解,每个节点的子节点是其父节点的分区,一起表示同一组的可能性;叶节点表示单独个体,每个表示一个可行解.只要内部节点 c 的得分是其所有子节点的得分上限,这样的算法得到的解与原始算法相同.在这种情况下,每当节点有界时,在该子树中不存在比目前已知解更好的解.算法获得目标解的效率取决于搜索树被修剪的部分是否占大多数,而这又依赖于一个好的上界以及一个好的当前解,DFS 可以快速评估大量叶节点获得较好的当前解来降低计算量.同时,因为不希望将较差的匹配作为闭环的约束,在匹配中引入一个分数阈值,低于这个阈值则不将其作为解.引入分数阈值后,在一定程度上减少了对节点选择以及寻找启发式解的依赖.DFS 算法中,访问一个节点的子节点顺序是先计算每个子节点的上界,然后访问具有最大上界也就是最有可能的子节点.

2 基于扫地机器人 Player 平台的 SLAM 系统实现

本节中,在扫地机器人 Player 平台[18,19]上实现了基于增强 Cartographer 算法的 SLAM 系统.整个系统分为系统配置、Player 传感器数据交互、接口通信、数据管理及分发、数据处理以及增强的 Cartographer 算法这 6 个模块.

2.1 基于增强Cartographer算法的SLAM系统设计

2.1.1 基于 Player 平台的 SLAM 整体设计

Cartographer 中原有的接口是基于 ROS[20]的,ROS 可以提供硬件抽象、设备驱动程序、库、可视化器、消息传递以及包管理等功能.但因为 ROS 集成各种功能,它显得过于庞大而冗余,无法部署在成本较低的嵌入式设备上.而本文中所使用的 Player 平台是一个轻量级的中间件,具有小而快的优点,同时最大化实用性和适用性,能

部署在低配置的嵌入式设备上.

本文中实现的 SLAM 系统以驱动的方式合入到整个 Player 平台中,易于修改替换,通过订阅相应传感器接收其数据进行处理.整个 SLAM 系统分为 6 个模块,如图 3 所示:系统配置模块中完成初始化,Player 传感器数据交互模块接收平台的传感器数据,接口通信模块对外提供接口与其他系统交互,数据管理及分发模块负责与算法进行数据交互,数据处理算法模块完成数据处理矫正.在数据流向中,Player 传感器数据交互模块接收来自各传感器的数据传递给数据管理及分发模块,数据管理及分发模块调用数据处理算法模块对各种传感器数据处理后传递给算法,Player 传感器数据交互模块通过数据管理及分发模块获取算法由激光雷达数据计算出的局部优化位姿,再经过数据管理及分发模块传到算法中进行位姿融合.

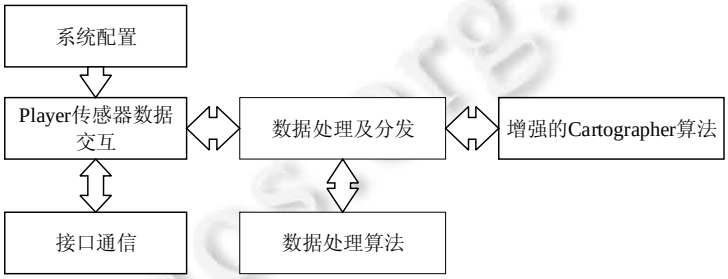


Fig.3 Cartographer-based SLAM system

图 3 基于 Cartographer 的 SLAM 系统

2.1.2 基于多传感器的系统配置

系统配置模块中,整个系统首先需要向 Player 平台注册,以独特的名字标识,同时说明本系统需要的传感器;然后对传感器进行订阅,加载配置文件,完成算法初始化;最后注册传感器数据处理的回调函数.

系统使用 Lua[21]进行配置文件的加载,其小巧而速度快的特点适合应用于嵌入式系统的配置文件.在系统停止建图处理完所有传感器数据后,结束当前轨迹,完成对当前优化图中子图位姿、扫描位姿以及扫描在子图坐标系中匹配位姿的最终优化.最后,将当前优化好的地图信息通过 Protobuf[22]进行序列化保存到文件中,可在 PC 端对保存的地图信息进行可视化.

2.2 基于数据交互的通信模型

2.2.1 Player 传感器数据交互

Player 传感器数据交互模块主要完成传感器数据及局部优化位姿的发送.

在 Player 传感器数据交互模块中,首先根据系统配置模块加载的参数调用数据管理及分发模块中的轨迹构建函数,并完成位姿融合函数的初始化.在建图过程中,会不断接收到来自 Player 平台的各种传感器数据,Player 传感器数据交互模块检查其采样频率是否高于配置中的设置:高于则丢弃;否则将其发送到数据管理及分发模块.同时,按照激光数据的采样频率从数据管理及分发模块中获取局部优化后的位姿,将其再转发到数据管理及分发模块,作为位姿融合的一个数据源.Player 传感器数据交互模块作为一个中间件模块,转发来自接口通信模块或数据管理及分发模块的数据.本模块也是在系统启动后唯一与 Player 平台交互的模块,作为整个 SLAM 系统的数据发布源.

在建图终止后,通过数据管理及分发模块结束轨迹完成最终全局优化以及序列化地图信息保存到文件中.

2.2.2 数据管理及分发

数据管理及分发模块与 Player 传感器数据交互模块相同,也是一个中间件模块.模块首先调用算法中的轨迹添加函数进行配置,并初始化数据处理算法模块.在建图过程中,循环调用数据处理算法模块对来自 Player 传感器数据交互模块的传感器数据进行处理,将处理后的数据传递给算法.同时,将从 Player 传感器数据交互模块发送的局部优化位姿传递给算法,算法将数据融合后得到下一时刻估计的位姿.

2.2.3 接口通信

SLAM 系统提供的接口分为输入和输出两部分.

- 输入部分中包含:设置暂停地图,地图将暂停更新,停止对传感器数据的处理,重设置后继续;重定位功能,在搬动机器人后,从当前已有地图上重定位出其位姿,用当前一帧或几帧激光雷达扫描数据在地图中找到最佳匹配,类似于 Cartographer 中对闭环的检测;加载地图,导入之前已经建好的地图,对当前地图信息进行覆盖;设置机器人位姿,将当前机器人位姿设置为参数值,比如在重定位后,将得到的位姿设置为正确值.
- 在输出部分中包含以下接口:获取当前完整地图、已使用地图以及增量更新地图;获取机器人位姿;保存地图,通过 Protobuf 将当前地图序列化保存为文件.

在初始时,地图空间按照预定义尺寸建图(如 1000×1000,精度为 5cm),地图中心为坐标原点.获取完整地图是获得整个预定义好的地图,其中,网格表示等同于栅格地图,未知表示为–1,非障碍表示为 0,障碍表示为 1.已使用地图表示整体地图的已使用区域,返回整体地图中已知区域的最大矩形.增量更新地图则是每帧激光雷达扫描数据对地图的更新量.

2.3 数据矫正及适配算法的研究

2.3.1 激光雷达数据矫正

机器人建图主要依靠激光雷达传感器采集的点云数据,通过基于激光雷达的 SLAM 算法实现建图与定位,而建图与定位的准确度非常依赖于激光雷达提供扫描数据的准确度.

在机器人运动中,有两种会对传感器扫描数据准确度产生明显影响的特殊情况:一种是传感器发射激光束的可旋转部分与机器人具有相反的旋转方向,且旋转的角速度相等时,激光雷达传感器自身并不能探测到其采集的扫描数据是对同一个位置的多次扫描;另一种情况是机器人的前进方向垂直于一面墙,激光雷达在前进中不断进行扫描,在扫描过程中,机器人从远处逐渐靠近墙体,此时激光雷达采集到与机器人直面的墙面扫描数据将不再与机器人的前进方向所垂直,其他方向的墙面数据也与实际情况不同,只是误差相比垂直情况较小.

只有激光雷达的旋转部分进行旋转,而机器人的其他部分没有产生相对旋转部分的运动时,传感器得到的扫描数据才最接近于真实环境的情况,这时的误差只有传感器自身所带来的测量误差.而在其他情况下,传感器采集的扫描数据不仅仅包含激光雷达旋转部分的运动,还包括机器人除旋转部分外其他部分所产生的运动[23].因而需要考虑激光雷达在这些运动情况下采集的扫描数据,根据扫描周期进行矫正,使扫描数据更符合所扫描的真实场景情况,减小误差,是进行高质量建图的基础.

2.3.2 IMU 和里程计数据转换及处理算法

在 Player 中,由平台传输过来的 IMU 数据和里程计数据同时得到,耦合在同一个数据结构里,原始数据无法直接提供给算法使用,需要进行转换处理后传递给 Cartographer.

其中,IMU 由陀螺仪和加速度计两部分构成:陀螺仪提供各轴的角速度,加速度计提供三轴的加速度大小.其中,重力加速度是地表垂直方向的加速度.Cartographer 中要求陀螺仪提供的是角速度,但在 Player 中提供的是角度,这就需要对数据进行转化.IMU 的采集频率(30Hz)较高,角速度可以通过角度差除以时间差得到,时间差通过记录每次数据中附带的时间戳获取.这样,第一次的 IMU 数据并不发送给算法,只是进行一个记录.

而加速度计部分,需要将其由四元数转化为欧拉角给算法使用.四元数和欧拉角都可以表示机器人的位姿:四元数可以避免万向节锁现象,提供平滑插值,但是表示不直观;欧拉角表示很直观,只用到 3 个数,但是会造成万向节锁现象[24].实际使用中,直接利用 Eigen 库构造四元数和欧拉角,调用库函数完成转换.

里程计中的数据——机器人前进速度以及角速度，是通过对左右轮的速度计算得到,通过机器人在两个相邻时刻的位置变换,根据几何关系得到偏航角等于机器人绕一段圆弧运动角度得到角速度.

3 实验测试和分析评估

本节首先在 PC 端对增强的 Cartographer 建图质量进行测试,基于数据集对进行定量误差分析;然后基于扫

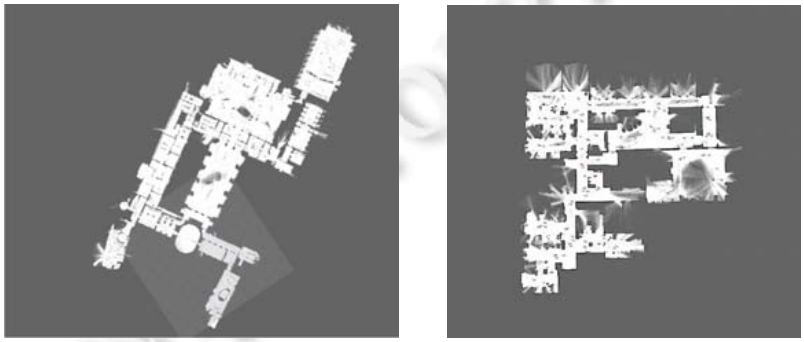
地机器人平台,在环境复杂度以及尺度不同的实际场景使用不同组合的传感器,对 SLAM 系统进行评估测试.

3.1 基于标准数据集的测试与分析

PC 端测试平台使用的系统为 Ubuntu 16.04.4,其内核为 4.4.0-139-generic x86_64.

在测试中,基于 ROS kinetic 平台进行数据发送.

图 4 表示两个由 2D 激光雷达背包和 Revo LDS 激光雷达[25]数据建图得到的栅格地图,其中,德意志博物馆数据集包含来自 IMU 的数据、水平和竖直激光雷达的数据,整个数据集时间长 1913s,数据集覆盖范围约为 128m*270m,地图轨迹比较复杂.Revo LDS 数据集则只包含低成本激光雷达采集的点云数据,数据集时间长度为 990s,覆盖范围约为 33m*32m.两个数据集包含大量的闭环,能有效检测算法的回环性能.



(a) 德意志博物馆数据集 (b) Revo LDS 数据集

Fig.4 Dataset grid map

图 4 数据集栅格地图

验证算法的建图质量评估公式如下:

$$\varepsilon(\delta)=\frac{1}{N}\sum_{i,j}trans(\delta_{i,j}\Theta\delta_{i,j}^{*})^{2}+rot(\delta_{i,j}\Theta\delta_{i,j}^{*})^{2} \tag{3}$$

文献[26]中提出的方法能够评估 SLAM 算法的准确度,这种方法基于机器人位姿间的相对位移,其中,N 是相对关系的数量,$trans(\cdot)$和 $rot(\cdot)$分别用于分离和加权平移和旋转分量,δ和δ^{*}分别表示估计值和真实值(ground truth),$\delta_{i,j}=x_i\Theta x_j$ 表示从节点 x_i 到节点 x_j 的相对变换关系,Θ表示标准运动合成的逆运算符.

表 1 是增强 Cartographer 算法在两个数据集测试后,利用上述公式将结果与 ground truth 比较分析后所得.

Table 1 Dataset test quantitative analysis

表 1 数据集测试定量分析

德意志博物馆			
	原 Cartographer	改进后的 Cartographer	GMapping
绝对平移误差(m)	1.272±1.305	**0.591±0.728**	2.442±4.671
均方平移误差(m^2)	1.330±2.065	**0.819±1.602**	2.419±2.457
绝对旋转误差(deg)	2.879±4.395	**1.433±2.724**	3.623±5.026
均方旋转误差(deg^2)	2.362±4.605	**1.712±2.038**	8.991±12.065
Revo LDS			
	原 Cartographer	改进后的 Cartographer	GMapping
绝对平移误差(m)	0.040±0.046	**0.037±0.054**	0.147±0.206
均方平移误差(m^2)	0.039±0.029	**0.004±0.018**	0.164±0.438
绝对旋转误差(deg)	0.747±0.997	**0.542±0.466**	0.871±0.836
均方旋转误差(deg^2)	**0.420±0.512**	0.511±0.911	1.455±1.753

表 1 中包含的误差分为绝对平移和绝对旋转以及均方平移和均方旋转等 4 种,能够较全面地评估算法.基于两个数据集,在原算法以及 GMapping 也完成相同测试.从表中的对比数据可以看出:改进后的算法在具有多传感器数据效果有一定的提升,说明改进后的位姿融合对多传感器数据建图的质量具有较明显效果,其中,绝对

平移误差降低了 53%,绝对旋转误差降低了 50%;而在只有一种传感器数据时,结果没有得到有效改善.

3.2 基于扫地机器人的实际场景测试分析

扫地机器人为米家扫地机器人二代,机器人平台使用的系统为 Ubuntu 14.04.3,其内核为 3.4.39 armv7l,基于 Player 平台进行传感器数据采集和发送.配备的是自研 LDS 激光雷达,最大和最小探测距离分别为 6m 和 15cm,发布频率为 5Hz,配有 SLAM 中使用的 IMU 和里程计传感器.

基于扫地机器人平台,在两种完全不同的场景对算法进行测试,检测其效果.两个场景如图 5 所示.

Fig.5 Laboratory and long corridor scene

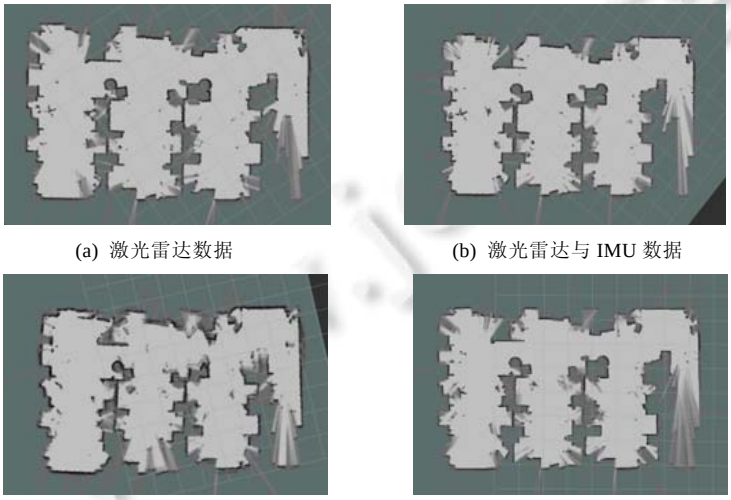
图 5 实验室和长走廊场景

其中,实验室场景范围约为 15m×8m,这个场景信息丰富,能产生较多的回环,局部地图不会产生重复,场景大小与复杂度比较符合家用扫地机器人实际应用的场景.长走廊场景范围约为 12m×63m,这个场景较大而信息简单,在走廊部分局部信息重复,用于检测评估算法在特殊场景的建图质量.

扫地机器人在真实场景中的测试,因为现有条件无法测得 ground truth,因而使用扫描匹配中约束构建的分数进行评估,评估指标分为最大值、最小值以及平均值这 3 项.

3.2.1 实验室场景测试与分析

图 6 是扫地机器人在实验室场景扫描建图,保存地图信息后,在 PC 端通过 rviz 软件可视化的 2D 栅格地图.



(a) 激光雷达数据 (b) 激光雷达与 IMU 数据

(c) 激光雷达与里程计数 (d) 3 种传感器数据

Fig.6 Laboratory scene 2D raster map

图 6 实验室 2D 栅格地图

图 6(a)是只含有激光雷达数据的地图,图 6(b)是含有激光雷达与 IMU 数据的地图,图 6(c)是含有激光雷达

与里程计数据的地图,图 6(d)是含有所有传感器数据的地图.从栅格地图显示的结果可以看出:图 6(c)(含有激光雷达和里程计数据)的结果中建图质量最差,其余 3 图结果中无较大差异.

表 2 是在实验室场景对各种传感器组合的建图测试,因为对真实场景扫描不可能像数据集一样做到每次一致,因而匹配分数是在进行 5 次测试后取 Mean 的中位值为代表.

Table 2 Matching scores of sensor combinations in the laboratory scene

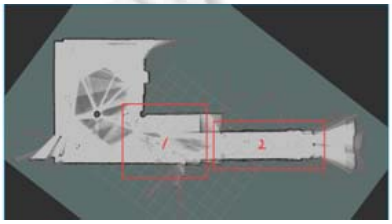
表 2 实验室场景各传感器组合的匹配分数

	Min	Max	Mean
激光雷达数据	0.651	**0.836**	0.714
激光雷达与 IMU 数据	0.650	0.788	0.704
激光雷达与里程计数据	0.650	0.790	0.685
激光雷达、IMU 与里程计数据	**0.651**	0.824	**0.729**

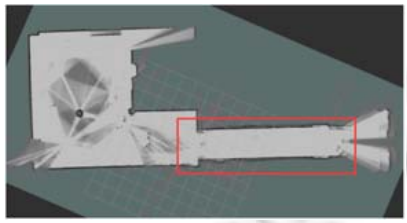
从表中可以得到:在含有 3 种传感器数据时建图质量是最高的,拥有最高的平均匹配分数,IMU 和里程计数据对激光雷达数据起到了较好的辅助作用.只有激光数据时的建图质量高于有两种传感器的建图质量,其中,只包含里程计数据作为辅助数据的建图质量是最低的.说明里程计数据对建图的干扰相比 IMU 数据稍大,在除激光雷达外只有一种传感器的情况下,这种传感器数据因为误差较大,没有其余传感器进行辅助矫正,对建图起到负面影响.因而在实际应用中,需要使用所有传感器数据才能得到最佳的建图质量.

3.2.2 长走廊场景测试与分析

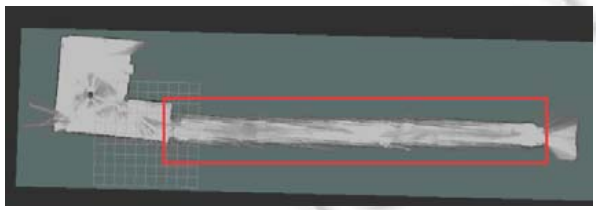
图 7 是在长走廊场景基于不同传感器建图得到的栅格地图.



(a) 激光雷达数据



(b) 激光雷达与 IMU 数据



(c) 激光雷达与里程计数据

(d) 3 种传感器数据

Fig.7 2D raster maps of long corridors with different combinations of sensors

图 7 长走廊不同组合传感器的 2D 栅格地图

图 7(a)是只含有激光雷达传感器数据的地图,可以看到,图中出现了比较明显的两处错误(两个标出的红框 1 和红框 2):红框 1 出现局部地图匹配错误,红框 2 中尺度与实际场景差异较大.图 7(b)是含有激光雷达和 IMU

数据的地图,红框部分的地图与实际环境的尺度差异仍然存在,IMU 数据并不能处理尺度问题.图 7(c)是含有激光雷达和里程计数据的地图,因为从里程计数据得到较为准确的线速度,红框部分的尺度问题得到解决,但还是存在匹配错误问题.图 7(d)是包含所有传感器数据的地图,局部地图的匹配以及走廊部分的尺度不一致有明显改善.

在长走廊场景的图 7(a)~图 7(c)中出现明显建图错误,是因为此场景的局部环境基本重复无变化,信息较为简单.在此场景中仅依靠激光雷达传感器数据,无法探测到场景实际已发生改变.加入 IMU 数据后也无法帮助正确建图,仅能帮助改善局部的地图匹配;而在里程计数据的辅助下虽然能正确感知环境的改变,但无法解决局部匹配错误的问题.只有在 3 种传感器的共同作用下,才能得到一个理想的建图结果.

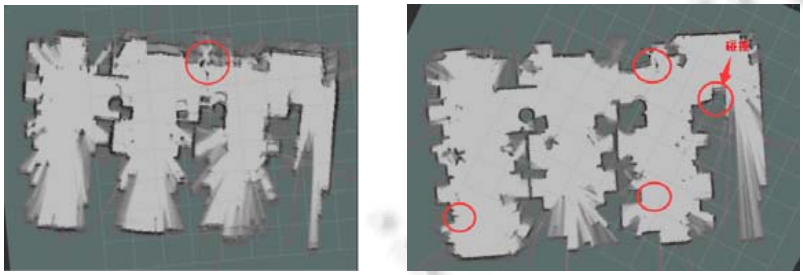
在这个场景的测试说明:扫地机器人在一些特殊场景的建图,对 IMU 和里程计等传感器数据有较大的依赖.IMU 数据能提供角度增量,得到较为准确的位姿估计;里程计数据能提供较为准确的线速度进行位置估计,在激光雷达数据的基础上得到较好的地图.

3.2.3 机器人碰撞的数据处理

在扫地机器人实际应用的场景中,碰撞不可避免,有时为了清扫障碍物边缘的灰尘,往往还会主动碰撞障碍物.在测试中发现:产生碰撞会对里程计数据产生较大的干扰,最后生成的栅格地图具有明显的误差.而在没有里程计数据的情况下,只有激光雷达数据或者激光和 IMU 数据都不会因为发生碰撞而干扰建图.而在前文的测试中得到,里程计对于探测场景变换以及提升建图质量都是不可缺少的,这时需要在发生碰撞时对里程计数据进行处理.

在扫地机器人上,前置有 3 个感知碰撞的传感器,在 SLAM 模块中,加入对这部分传感器的订阅.当发生碰撞,也就是接收到感知碰撞传感器的数据时,SLAM 模块停止发送里程计数据,在碰撞结束后(一定时间内不再接收到感知碰撞传感器的数据),再继续发送.算法中需要清空缓存队列中的里程计数据,防止之后利用这部分数据进行角速度与线速度的计算;同时,在位姿融合部分保存之前通过里程计数据计算的线速度与角速度,需要加入时间戳,在发生碰撞期间不再使用这些值,这样能避免里程计数据不准确造成的错误影响.

图 8 是在实验室场景产生碰撞后的栅格地图(红圈处为发生碰撞处,图 8(a)中发生 1 次,图 8(b)中为 4 次).



(a) 无里程计数据处理 (b) 有里程计数据处理

Fig.8 Comparison diagram of collision generated in the lab scene

图 8 实验室场景产生碰撞时对比图

图 8(a)中,在红圈处仅发生一次碰撞,此测试没有加入里程计数据的处理;图 8(b)中,在红圈处发生了 4 次碰撞,这个测试中加入了对里程计数据的处理.可以看出:在加入里程计数据处理后,即使碰撞多次,建图质量仍未下降;而在没有里程计数据处理的情况下,仅发生一次碰撞,也会使地图产生较大的误差.

表 3 是在上述两种情况下的匹配分数对比图.从表中可以看出:相较于有里程计数据处理的测试,未加入处理的测试中,建图匹配分数有明显的降低.另外,当碰撞发生时,未加入对里程计数据处理的测试中,建图匹配分数有明显的下降;对比有里程计数据处理的测试,建图匹配分数并未下降.说明在碰撞中对里程计数据的处理是不可缺少的,否则会极大影响建图质量.

Table 3 Match scores with or without odometer data processing collision

表 3 有无里程计数据处理发生碰撞的匹配分数

	Min	Max	Mean
有里程计数据处理	**0.651**	**0.867**	**0.723**
无里程计数据处理	0.650	0.698	0.676

4 总 结

本文主要对当前的激光 SLAM 算法进行了分析和研究,在 Cartographer 的基础上设计了基于位姿增量的多传感器位姿融合算法,在 Player 平台设计实现了基于增强 Cartographer 算法的 SLAM 系统,并在实际场景生成 2D 栅格地图.

5 展 望

提高扫地机器人智能化,是未来服务机器人应用的一个重要方向,之后的研究还可以从以下几个方面进行.(1) 目前使用的传感器无法探测语义信息[27],从而限制机器人的智能化.将来可以研究加入视觉传感器,在提供更精确位姿的同时,能够与深度学习结合,在不同场景使用不同的清扫方式,增强机器人的智能化程度;(2) 目前,在扫地机器人上实现的是二维环境下的 SLAM 算法,在今后的实际发展中,三维环境的 SLAM 算法具有更广泛的应用场景;(3)究在复杂场景对动态物体的检测以及进行轨迹的预测,实现规避,让机器人更安全、更方便.

References:

[1] Leonard JJ, Durrant-Whyte HF. Mobile robot localization by tracking geometric beacons. IEEE Robotics and Automation, 1991, 7(3):376−382.

[2] Liang Z, Peiyi S, Guangming Z, *et al*. A fast robot identification and mapping algorithm based on kinect sensor. Sensors, 2015, 15(8):19937−19967.

[3] Zhang L, Shen P, Ding J, *et al*. An improved RGB-D SLAM algorithm based on Kinect sensor. In: Proc. of the IEEE Int'l Conf. on Advanced Intelligent Mechatronics. IEEE, 2015.

[4] Nister D, Narodisky O, Bergen J. Visual odometry for ground vehicle applications. Journal of Field Robotics, 2006,23(1):3−20.

[5] Alin A, Butz MV. Improved tracking and behavior anticipation by combining street map information with bayesian-filtering. In: Proc. of the Int'l IEEE Conf. on Intelligent Transportation Systems. IEEE, 2014.

[6] Grisetti G, KuMmerle R, Stachniss C, *et al*. A tutorial on graph-based SLAM. Intelligent Transportation Systems Magazine IEEE, 2010,2(4):31−43.

[7] Ho KL, Newman P. Loop closure detection in SLAM by combining visual and spatial appearance. Robotics and Autonomous Systems, 2006,54(9):740−749.

[8] Hess W, Kohler D, Rapp H, *et al*. Real-time loop closure in 2D LIDAR SLAM. In: Proc. of the 2016 IEEE Int'l Conf. on Robotics and Automation (ICRA). IEEE, 2016.

[9] Isbell JR. Six theorems about injective metric spaces. Commentarii Mathematici Helvetici, 1964,39:65−76. [doi: 10.1007/BF02566944]

[10] Lee YC, Yu W, Lim JH, *et al*. Sonar grid map based localization for autonomous mobile robots. In: Proc. of the 2008 IEEE/ASME Int'l Conf. on Mechtronic and Embedded Systems and Applicationbs. 2008. 558−563.

[11] Lu F, Milios E. Globally consistent range scan alignment for environment mapping. Autonomous Robots, 1997,4(4):333−349.

[12] Martín F, Triebel R, Moreno L, Siegwart R. Two different tools for three-dimensional mapping: DE-based scan matching and feature-based loop detection. Robotica, 2014,32(1):19−41.

[13] Agarwal S, Mierle K, *et al*. Ceres solver. http://ceres-solver.org

[14] Julier S, Uhlmann JK. A new extension of the Kalman filter to nonlinear systems. In: Proc. of the Int'l Symp. on Aerospace/Defense Sensing, Simul. and Controls. Signal Processing, Sensor Fusion, and Target Recognition VI. 3. 1997. 182.

[15] Konolige K, Grisetti G, Kümmerle R, *et al*. Sparse pose adjustment for 2D mapping. In: Proc. of the IROS. 2010.

[16] Land AH, Doig AG. An automatic method of solving discrete programming problems. Econometrica, 1960,28(3):497−520.

[17] Clausen J. Branch and bound algorithms-principles and examples. Department of Computer Science, University of Copenhagen, 1999. 1–30.

[18] Gerkey B, Vaughan R, Howard A. The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proc. of the Int'l Conf. on Advanced Robotics. 2003. 317–323.

[19] Collet THJ, MacDonald BA, Gerkey B. Player 2.0: Toward a practical robot programming framework. In: Proc. of the Australasian Conf. on Robotics and Automation (ACRA). 2005.

[20] ROS. http://wiki.ros.org/

[21] Schuytema P, Manyen M, Wrote; Tian J, Trans. Game Development with Lua. Beijing: China Machine Press, 2013 (in Chinese).

[22] Frequently Asked Questions|Protocol Buffers. Google Developers, 2016.

[23] Bai W, Li G, Han L. Correction algorithm of LIDAR data for mobile robots. 2017.

[24] Hoag D. Apollo guidance and navigation-considerations of apollo IMU gimbal lock—MIT instrumentation laboratory document E-1344. 1963.

[25] Konolige K, Augenbraun J, Donaldson N, Fiebig C, Shah P. A low-cost laser distance sensor. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA 2008). IEEE, 2008. 3002–3008.

[26] Kümmerle R, Steder B, Dornhege C, Ruhnke M, Grisetti G, Stachniss C, Kleiner A. On measuring the accuracy of SLAM algorithms. Autonomous Robots, 2009,27(4):387–407.

[27] Zhang L, Wei L, Shen P, *et al*. Semantic SLAM based on object detection and improved octomap. IEEE Access, 2018:1–1.

附中文参考文献:

[21] Schuytema P, Manyen M,著;田剑,译.Lua 游戏开发实践指南.北京:机械工业出版社,2013.

张亮(1975－),男,博士,副教授,CCF 专业会员,主要研究领域为嵌入式多核系统,机器人语义 SLAM,深度学习和计算机视觉,三维场景语义分割,手势识别,图像处理.

刘智宇(1993－),男,硕士生,主要研究领域为机器人同步定位与建图.

曹晶瑛(1984－),男,学士,主要研究领域为分布计算,可信计算与信息安全.

沈沛意(1975－),男,博士,教授,CCF 专业会员,主要研究领域为计算机视觉,语义结构图,大数据.

蒋得志(1994－),男,硕士,主要研究领域为语义 SLAM.

梅林(1975－),男,博士,研究员,主要研究领域为计算机视觉,人工智能,物联网应用,大数据处理.

朱光明(1975－),男,博士,讲师,CCF 专业会员,主要研究领域为深度学习,手势识别.

苗启广(1972－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为计算机视觉,机器学习,大数据分析.