

工学硕士学位论文

基于多传感器融合的室外巡检机器人
同时定位与建图研究

马尔斯

哈尔滨理工大学

2023 年 3 月

国内图书分类号：TP242

工学硕士学位论文

基于多传感器融合的室外巡检机器人 同时定位与建图研究

硕 士 研究生：马尔斯

导 师：尤波

申请学位级别：学术硕士

学 科、专 业：控制科学与工程

所 在 单 位：自动化学院

答 辩 日 期：2023 年 3 月

授予学位单位：哈尔滨理工大学

Classified Index: TP242

Dissertation for the Master Degree in Engineering

**Research on Simultaneous Localization and
Mapping of Outdoor Inspection Robot
Based on Multi-Sensor Fusion**

Candidate:	Ma Ersi
Supervisor:	You Bo
Academic Degree Applied for:	Master of Engineering
Specialty:	Control Science and Engineering
Date of Oral Examination:	March, 2023
University:	Harbin University of Science and Technology

基于多传感器融合的室外巡检机器人

同时定位与建图研究

摘要

近年来，随着移动机器人的不断发展，室外巡检机器人逐渐应用于日常生活中。传统的室外移动机器人是基于全球卫星定位系统（Global Positioning System, GPS）或者惯性导航系统（Inertial Navigation System, INS）进行定位，然而当 GPS 信号受到非视距与多路径等因素影响时，传统依靠 GPS 定位的方法出现较大偏差，此外当移动机器人行驶到 GPS 拒止环境中时，会出现无法获取 GPS 定位数据的情况。因此人们逐渐使用以视觉相机或者激光雷达为主的同时定位与地图构建（Simultaneous Localization and Mapping, SLAM）技术来进行室外巡检机器人的定位与导航。

相较于视觉相机容易受光照强度变化的影响等缺点，激光雷达对光照强度不敏感，可以在不同室外环境中准确感知周围环境，从根本上保证激光 SLAM 系统在室外环境中的精度，因此以激光雷达为核心的 SLAM 系统成为大多数室外移动机器人公司的方案。此外由于单纯使用激光雷达的 SLAM 系统无法解决环境退化、采样频率低等问题，常采用惯性测量单元（Inertial Measurement Unit, IMU）应用于中小范围场景下解决该问题。然而由于惯性测量单元的测量误差会随着时间的增加而增加，并且在大范围场景下使用激光雷达与惯性测量单元的 SLAM 系统精度会有所下降。为了提升激光惯性 SLAM 系统的精度，所以常采用融合 GPS 数据的方法消除累计误差。

本文针对面向巡检机器人在科技园区室外环境中出现 GPS 中断情况下以激光雷达为核心的多传感器融合 SLAM 算法精度下降的问题进行研究。以激光雷达为核心，提出一种结合激光雷达、IMU 与 GPS 的多传感器融合 SLAM 方法。首先，为了搭建完整的多传感器融合算法，针对巡检机器人底盘与各个传感器建立了数据采样模型与坐标系。同时对激光雷达、IMU 与 GPS 之间的坐标系外参进行标定，并标定出 IMU 的内参，从而更精确地融合激光雷达、IMU 与 GPS 的传感器测量数据。最后使用软硬件相结合的方法，实现对激光雷达、IMU 与

GPS 的时间同步。

其次，针对基于传统激光 SLAM 系统其前端里程计在长时间运行时出现精度不佳的问题，并且为了巡检机器人行驶到 GPS 拒止环境中出现 GPS 数据中断时机器人定位更准确，提出一种使用迭代误差卡尔曼滤波（Iterative Error Kalman Filter, IESKF）紧耦合激光雷达与惯性测量单元的前端里程计，提升在室外场景中仅使用激光雷达与惯性测量单元进行机器人位姿估计时的精度。并且在 KITTI 数据集上与传统的前端里程计进行实验评估。证明了 IESKF 前端里程计在室外场景中的精度要优于传统的前端里程计。

然后，基于 IESKF 的激光惯性里程计的全局精度仍有提升的空间，并且为了进一步提高算法的定位精度和鲁棒性，提出使用因子图优化的方式融合激光雷达、惯性测量单元与 GPS 的方法。在因子图优化的 SLAM 系统中，为了减小图优化产生的计算负荷，采用关键帧与滑动窗口策略，在因子图中加入 IESKF 所得到的激光惯性里程计因子与 IMU 预积分因子，并且为了进一步减小累计误差，加入回环检测因子与 GPS 因子，在 GPS 数据出现波动的情况下，同时基于 GPS 状态与置信度筛选 GPS 数据，防止加入异常的 GPS 数据。

最后，为了验证本文提出的多传感器融合 SLAM 系统的精度与鲁棒性，通过开源数据集 KITTI 与自建的巡检机器人平台在 GPS 不中断与 GPS 中断的两种情况下进行测试，并且与其他开源的 SLAM 算法进行轨迹与位姿的评估。结果表明本文提出的 SLAM 系统在 GPS 不中断与 GPS 中断的两种情况下都有着更优秀的精度。

关键词 巡检机器人；多传感器融合；SLAM；IESKF；激光雷达

Research on Simultaneous Localization and Mapping of Outdoor Inspection Robot Based on Multi-Sensor Fusion

Abstract

In recent years, with the continuous development of mobile robots, outdoor inspection robots have gradually been used in daily life. Traditional outdoor mobile robots are positioned based on Global Positioning System (GPS) or Inertial Navigation System (INS). However, when GPS signals are affected by factors such as non-line-of-sight and multipath, there is a large deviation in the GPS positioning method. In addition, when the mobile robot travels into a GPS-denied environment, it will be impossible to obtain GPS positioning data. Therefore, people are gradually using Simultaneous Localization and Mapping (SLAM) technology based on visual cameras or lidar to locate and navigate outdoor inspection robots.

Compared with the shortcomings of visual cameras that are easily affected by changes in light intensity, lidar is not sensitive to light intensity and can accurately perceive the surrounding environment in different outdoor environments, fundamentally ensuring the accuracy of the laser SLAM system in outdoor environments. The SLAM system with lidar as the core has become the solution of most outdoor mobile robot companies. In addition, since the SLAM system using only lidar cannot solve the problems of environmental degradation and low sampling frequency, an Inertial Measurement unit (IMU) is often used to solve this problem in small and medium-scale scenarios. However, the measurement error of the inertial measurement unit will increase with time, and the accuracy of the SLAM system using lidar and inertial measurement unit will decrease in a large-scale scene. In order to improve the accuracy of the laser inertial SLAM system, the method of fusing GPS data is often used to eliminate the cumulative error.

This dissertation studies the problem of the accuracy degradation of the multi-sensor fusion SLAM algorithm with lidar as the core in the case of GPS interruption

in the outdoor environment of the science and technology park for inspection robots. Taking lidar as the core, a multi-sensor fusion SLAM method combining lidar, IMU and GPS is proposed. First, in order to build a complete multi-sensor fusion algorithm, a data sampling model and coordinate system are established for the inspection robot chassis and each sensor. At the same time, the external parameters of the coordinate system between the lidar, IMU and GPS are calibrated, and the internal parameters of the IMU are calibrated, so as to more accurately fuse the sensor measurement data of the lidar, IMU and GPS. Finally, the method of combining software and hardware is used to realize the time synchronization of lidar, IMU and GPS.

Secondly, aiming at the problem that the front-end odometer based on the traditional laser SLAM system has poor accuracy during long-term operation, and in order to make the robot's positioning more accurate when the inspection robot travels to a GPS-denied environment and the GPS data is interrupted, a method using Iterative Error Kalman Filter (IESKF) tightly couples the front-end odometry of lidar and inertial measurement unit to improve the accuracy of robot pose estimation using only lidar and inertial measurement unit in outdoor scenes. And experimental evaluation with traditional front-end odometry on KITTI dataset. It is proved that the accuracy of IESKF front-end odometry in outdoor scenes is better than that of traditional front-end odometry.

Then, there is still room for improvement in the global accuracy of the laser inertial odometry based on IESKF, and in order to further improve the positioning accuracy and robustness of the algorithm, a method of integrating laser radar, inertial measurement unit and GPS using factor graph optimization is proposed. In the factor graph optimized SLAM system, in order to reduce the calculation load generated by graph optimization, the key frame and sliding window strategy is adopted, and the laser inertial odometry factor and IMU pre-integration factor obtained by IESKF are added to the factor graph, and in order to further Reduce the cumulative error, add loopback detection factor and GPS factor, and screen GPS data based on GPS status and confidence at the same time when GPS data fluctuates to prevent adding abnormal GPS data.

Finally, in order to verify the accuracy and robustness of the multi-sensor fusion SLAM system proposed in this dissertation, the open source data set KITTI and the self-built inspection robot platform are tested in two cases of GPS uninterrupted and

GPS interrupted, and compared with other The open source SLAM algorithm evaluates trajectory and pose. The results show that the SLAM system proposed in this dissertation has better accuracy in both cases of GPS uninterrupted and GPS interrupted.

Keywords Inspection Robot, Multi-sensor Fusion, SLAM, IESKF, LiDAR

目 录

第 1 章 绪论.....	1
1.1 课题来源及研究背景意义	1
1.1.1 课题来源.....	1
1.1.2 课题研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.2.1 巡检机器人研究现状.....	2
1.2.2 多传感器融合 SLAM 研究现状	4
1.3 本文研究的主要内容.....	8
第 2 章 巡检机器人与传感器系统建模	10
2.1 巡检机器人与传感器坐标系定义	10
2.2 巡检机器人及其传感器数据采集模型	11
2.2.1 巡检机器人运动学模型.....	11
2.2.2 激光雷达点云成像模型.....	13
2.2.3 IMU 传感器模型	14
2.2.4 GPS 传感器模型	16
2.3 多传感器时空标定.....	18
2.3.1 多传感器外参标定.....	18
2.3.2 多传感器时间标定.....	19
2.4 本章小结	20
第 3 章 基于滤波的激光惯性里程计	21
3.1 激光点云数据预处理.....	21
3.1.1 激光点云运动补偿算法.....	21
3.1.2 激光点云体素滤波算法.....	23
3.2 传统的激光里程计.....	25
3.2.1 基于正态分布变换的激光里程计	25
3.2.2 基于点云特征匹配的激光里程计	27
3.3 基于 IESKF 的激光惯性里程计	31
3.3.1 IESKF 状态定义.....	31
3.3.2 IESKF 状态方程.....	33
3.3.3 IESKF 观测方程.....	35
3.3.4 IESKF 协方差与状态更新	37
3.4 基于 IESKF 的激光惯性里程计算法实验	38
3.5 本章小结	40
第 4 章 基于多传感器数据融合的定位与建图	41

4.1 激光惯性里程计因子	41
4.2 IMU 预积分因子	43
4.2.1 IMU 离散积分模型	43
4.2.2 IMU 预积分模型	43
4.3 GPS 因子	45
4.3.1 GPS 坐标变换	45
4.3.2 基于 GPS 状态与置信度筛选策略的 GPS 因子	47
4.4 回环因子	48
4.5 基于因子图的后端优化	50
4.6 本章小结	52
第 5 章 实验测试与评估	53
5.1 轨迹精度评估指标	53
5.2 KITTI 数据集测试与评估	54
5.2.1 KITTI 数据集介绍	54
5.2.2 KITTI 数据集测试与评估	55
5.3 巡检机器人实验平台测试与评估	60
5.3.1 巡检机器人实验平台介绍	60
5.3.2 巡检机器人实验测试与评估	61
5.4 本章小结	65
结论	66
参考文献	68

第1章 绪 论

1.1 课题来源及研究背景意义

1.1.1 课题来源

本课题来源于黑龙江省复杂智能系统与集成重点实验室与哈工大机器人（合肥）国际创新研究院的联合科技园区室外巡检机器人研究项目。

1.1.2 课题研究背景与意义

随着科技的发展，移动机器人已经广泛应用于工业自动化领域并逐渐走进人们的日常生活中，其中巡检机器人可以在各种行业和领域中应用，在制造业中，巡检机器人可以用于检测生产线上的设备和机器，提高设备的稳定性和可靠性，避免生产线因设备故障而停工，降低维护成本。在能源行业中，巡检机器人可以用于检测风力发电机、太阳能板等设备的状态，提高设备的稳定性和运行效率，减少故障率，提高发电量。在交通行业中，巡检机器人可以用于检测桥梁、隧道、高速公路等设施的状况，及时发现和修复潜在的安全隐患，保障交通的安全和畅通。在建筑领域中，巡检机器人可以用于检测建筑物的结构、电器设备、管道等，及时发现和修复潜在的安全隐患，保证建筑物的安全和稳定性。在医疗领域中，巡检机器人可以用于检测医院的卫生状况、医疗设备的状态等，帮助医院提高卫生管理和医疗服务的质量。在科技园区安保领域中，巡检机器人可以解决传统的视频安保系统由于各种客观条件的制约，存在视频监控的盲区的问题^[1]。

室外巡检机器人源自于如今快速发展的无人车系统，属于低速无人车的范畴。所以乘用无人车的系统与硬件结构都具有一定程度上的参考意义^[2]。但是由于应用的场景与功能的需求不同，室外巡检机器人的设计目的是降低安全巡检的日常成本，所以其相较于乘用无人车采用更经济的传感器，比如采用16线激光雷达而不是价格更昂贵的32线或64线的激光雷达，搭载消费级的IMU与GPS而不是工业级的IMU与GPS。

通常，室外巡检任务分为4个模块：建图定位、感知、控制与规划。其中建图服务于定位，而定位是准确感知机器人周围环境并且做出合理控制与规划的前提。对于只依靠GPS的传统室外巡检机器人，当其运动到树林、高度城市

化环境中或存在人为干扰时，GPS 信号容易受到非视距与多路径等因素影响^[3]，GPS 的信号状态将由常态下的固定解转换为精度更低的浮点解、单点解或伪距差分解^[4,5]，当运动到 GPS 拒止环境中时则会出现 GPS 中断的现象，此时则需要 SLAM 系统进行定位^[6,7]。

应用于无人车领域的 SLAM 技术，依据不同传感器的采集方式分为基于激光雷达的激光 SLAM 与基于相机的视觉 SLAM。由于相机廉价的价格与视觉图像丰富特征、明确的语义等优势，成为了 SLAM 领域的热门研究方向^[8]。

然而，在室外环境中，相机表现得不够稳定，尤其是对光照强度有着较高的敏感度，并且动态物体常常会造成视觉特征丢失，从而导致最终定位失败。近些年来，激光雷达主机厂不断优化激光雷达精度与降低其成本，这使得适合室外使用的多线激光雷达性价比不断提升，从而在预算有限的室外巡检机器人上搭载多线激光成为了可能。由于多线激光雷达输出频率较低，无法应对室外巡检机器人运动中发生意外颠簸或受雨滴的影响，形成非常有效的反射，导致激光雷达出现错误的读数，从而使得定位与建图误差增大或者失效。此外，IMU 在室外环境中不受环境因素影响，并且可以提供高精度的姿态测量与快速测量姿态与运动状态的功能。但是 IMU 测量数据具有累计误差，随着时间的推移测量值变得不准确与无法对周围物体环境进行感知等缺点。

综上所述，使用单一的传感器的 SLAM 系统难以满足室外巡检机器人的需求。因此构建一个在 GPS 信号出现中断时，仍有着较高精度的多传感器融合 SLAM 系统成为了本课题的研究重点。

1.2 国内外研究现状

1.2.1 巡检机器人研究现状

国外在巡检机器人的研究起步早于国内，开始于上世纪 80 年代。其中以美国与加拿大为代表，率先将其应用在电力行业中。加拿大 Hydro-Quikbec 研究所的 Serge 研究出一种可以远程操作的巡检机器人，并且通过远程巡视操作，可以获取各个输电线路的状态数据^[9]。后续通过不断地优化改进，名为 LineScout 的巡检机器人正式广泛应用于电力行业^[10]。此外，Moyo^[11]研制的智能巡检机器人也应用于高压电缆上进行检测与故障检测。2016 年，美国 FPL 公司研制的电力巡检机器人，在南佛罗里达正式应用于变电站。FPL 电力巡检机器人依靠高清相机、红外热成像仪、多线激光雷达等多种高精度传感器，可以

在变电站中进行自主巡检任务并且可以精确检测到肉眼无法发现的线缆故障，提前预警维修人员进行故障维修以免发生重大故障。

除了在电力行业中巡检机器人得到了广泛的应用，在民用领域巡检机器人也逐渐得到使用。美国加州 Knightscope 公司研制的安保巡检机器人，如图 1-1 所示，其搭载了高清相机、热成像仪与激光雷达等传感器，使其可以 7x24 小时全天候在住宅与园区进行室外巡检工作。日本机器人制造企业 Tmsuk 公司研制的 SECOM 室外巡检机器人，如图 1-2 所示。搭载高精度红外热成像相机，在夜间可以清晰显示热成像图。并且可以与搭载的高清相机同时传输视频画面至监控室，保证安保人员可以通过画面实时观察到室外情况。



图 1-1 K5 巡检机器人

Fig. 1-1 K5 inspection robot



图 1-2 SECOM 巡检机器人

Fig. 1-2 SECOM inspection robot

巡检机器人的国内的研究起步晚于国外，但是近几年来随着国家的飞速发展，关于巡检机器人的研究不断增多。尤其是在电力行业，巡检机器人率先得到研究与发展。2002 年，山东鲁能智能技术有限公司得到国家“863”计划的支持，并且在 2005 年成功研制出中国第一台电力巡检机器人。随后通过不断地迭代升级与更深入的研究，在长清 500KV 变电站成功投入使用^[12-14]。

同样的，巡检机器人的国内也应用于民用区域。主要包括中智科创、国自机器人、海康威视与哈工图南等。其中，中智科技最早研制出适合民用区域的巡检机器人，如 1-3 所示。其采用激光雷达，云台相机，GPS 等传感器进行室外园区或校园的巡检、安保工作工作，在周边环境发生异常时进行闪灯与语音报警，有效及时地警告当前出现的异常。另外哈工大机器人（合肥）国际创新研究院的智能装备所智能控制器项目组研制了一款部署于华中农业大学的室外巡检机器人，简称华农巡检机器人，如图 1-4 所示。其搭载激光雷达、GPS、

IMU 与视频监控相机与红外测温仪进行校园的巡检与测温工作。



图 1-3 K5 巡检机器人

Fig. 1-3 K5 inspection robot



图 1-4 华农巡检机器人

Fig. 1-4 HuaNong inspection robot

1.2.2 多传感器融合 SLAM 研究现状

传统的室外定位方式广泛地使用 GPS/INS（组合导航系统）。在一般情况下，GPS 的定位能够达到厘米级，然而在周边有较多城市高楼遮挡时，GPS 信号由于多径与多视距效应，常常会出现定位不准确的现象^[15]，并且，一般使用基于扩展卡尔曼滤波（Extended Kalman Filter, EKF）融合 GPS、IMU 进行定位的方案十分依赖于 GPS 数据，当 GPS 数据失效时，则会导致定位的结果出现较大的误差^[16]。所以，为了有效解决上述问题，如今常见的方案是结合融合 IMU、激光雷达（LiDAR）、GPS 等多个传感器的 SLAM 系统进行定位建图。

SLAM 即同时定位与建图^[17-19]，SLAM 是自主移动机器人进行导航与环境感知的关键技术。它解决了当工作空间的先验映射不可访问时的定位与建图问题。在国内外不断涌现出许多优秀的研究。根据 SLAM 所使用的主要传感器可以分为基于相机的视觉 SLAM^[20,21]与基于激光雷达的激光 SLAM^[22-24]。相比于相机，激光雷达可以直接获取周围一定半径内的三维结构信息，并且不会像摄像机一样对室外光照有着较高的敏感度。因此，在室外环境中，激光 SLAM 比视觉 SLAM 有着更好的鲁棒性与精度。此外，又因为激光雷达对周围环境的特征较为依赖，在快速运动中容易发生偏移，导致移动机器人丢失定位。为了解决上述问题，人们主要采用滤波或图优化的方法融合多个传感器。

本文将具体从以激光雷达为核心的多传感器融合 SLAM 所使用的两种方法：基于滤波与基于图优化两个方面进行阐述其发展现状。

（1）基于滤波方法且以激光雷达为核心的多传感器融合 SLAM

Gao^[25]使用 EKF 对 GPS 和 LiDAR 进行松耦合估计位姿，并且可以对 IMU 进行周期性地矫正。Shamsudin^[26]使用粒子滤波器（Rao-Blackwellized Particle Filters, RBPF）将 GPS 与激光雷达数据相结合，检测石油化工企业地图的一致性，并且使用 GPS 与激光雷达数据进行 Fast-SLAM 构建地图。Abdelaziz^[27]基于 EKF 松耦合 INS 与 LiDAR，并且与三维概率图的相对位姿进行匹配的 SLAM。另外对 GPS 进行断电 2 分钟测试，系统依旧保持较高的精度。Aboutaleb^[28]将 LiDAR 数据、全球导航卫星系统（Global Navigation Satellite System, GNSS）数据与三维简化惯性传感器通过 EKF 滤波器进行数据融合，并且使用 LO 来限制系统定位时出现的漂移。LIOM^[29]使用 CNN 分割网络去除动态物体并且通过 ESKF 对 LiDAR 与 IMU 进行融合，最后通过扫描匹配生成静态的全局地图。LINS^[30]是第一个使用 IESKF 紧耦合 LiDAR 与 IMU 来进行机器人运动估计的方法，其通过以机器人为中心的公式在迭代的过程中使用从激光雷达提取的特征来递归修正机器人估计的位姿，在保持计算速度的同时防止了滤波器在长时间运行中发生扩散。其效果图如图 1-5 所示。FAST-LIO^[31]通过 IESKF 将激光点云特征与 IMU 数据进行融合，为了降低激光点云特征带来的计算量，提出依赖维度的卡尔曼增益公式，其效果图如图 1-6 所示。FAST-LIO2^[32]在此基础上进行改进，取消了对激光雷达点云特征的提取，而采用将激光雷达原始点云信息与 IMU 进行 IESKF 滤波，因此可以更多的观察到环境的细微特征，提高准确性。同时通过增量的数据结构 ikd tree 进行点云的增量更新与动态平滑，减小计算量，增加实时性。Faster-LIO^[33]在前面两者基础上将 ikd tree 替换成增量体素 iVox 的点云数据结构，并且新的结构支持增量插入与并行逼近查询，得到高效、鲁棒的 LIO 框架。从而在快速运动或者噪声环境中依旧可以稳定地运行。

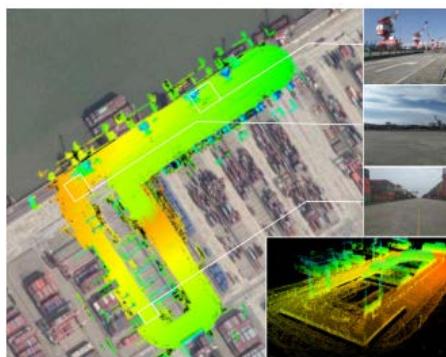
图 1-5 LINS 效果图^[30]

Fig. 1-5 The result of LINS

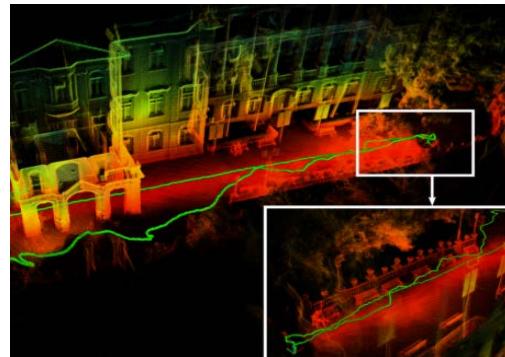
图 1-6 FAST-LIO 效果图^[31]

Fig. 1-6 The result of FAST-LIO

李文豪^[34]通过误差卡尔曼滤波器（Error State Kalman Filter, ESKF）将多线激光雷达、GPS 与 IMU 进行松耦合地融合在一起，得到更快的点云提取速度。并且系统同时兼顾了良好的鲁棒性与精度。白崇岳^[35]为了消除在融合时由于出现两个绝对位姿信息而导致融合的结果出现震荡现象而对 EKF 进行差分改进，避免出现这个现象。并且使用改进后的 EKF 对激光雷达、惯性测量单元与光电编码器进行融合，提升了无人驾驶轮椅轨迹定位的精度。张书亮^[36]使用 EKF 与蒙特卡洛方法（Monte Carlo method）依次对轮式里程计、惯性测量单元、超宽带与单线激光雷达进行数据融合，并且获得了更精确地定位精度与良好的鲁棒性。杨林^[37]提出一种前端为 IEKF 与后端为图优化组成的 SLAM 框架，在煤矿井下由于无 GPS 信号，故只搭载了激光雷达与惯性测量单元，实验表明该算法在矿井中取得了较高的精度与实时性。王馨阁^[38]提出一种改进 RBPF 算法将轮式里程计与激光雷达数据进行融合，加入自适应重采样机制对粒子的权重进行重行计算，解决粒子退化的问题，并且提升了机器人定位精度。李秀智^[39]提出一种结合月球车平动速度估计的粒子滤波 SLAM，在月球仿真环境下可以获得更好的实时性。王依人^[40]提出了一提出了一种粒子滤波 SLAM，利用激光雷达数据，修正了里程计的提议分布函数，优化了滤波所需的粒子数，引入了自适应重采样机制，解决粒子退化问题。

虽然上述基于滤波的方法在定位速度上相较于图优化算法有显著的优势，但是在全局定位精度方面仍与图优化的方法有一定的差距。

（2）基于图优化方法且以激光雷达为核心的多传感器融合 SLAM

Kukko^[41]基于移动激光扫描（Mobile Laser Scanning, MLS）系统，从周围森林环境收集三维数据的点云，并且采用图优化方法融合 MLS 与 INS 的数据，对轨迹进行优化，从而可以准确地提取森林图与树木参数。Cartographer^[42]实现了基于图优化的二维激光雷达定位与建图，采用分支定界法加速子地图的搜索速度，在大范围室内地图构建中取得了优秀的定位结果。并且在后续的开源代码更新中使用图优化的方式融合了 GNSS、3D-LiDAR、IMU 与 ODO，进一步提高了算法的精度。Chang^[43]提出一种融入 IMU&ODO 预积分的 SLAM 系统，在前端使用三维概率图，提高在特征有限的环境中点云匹配的效果，在 GNSS 停止工作时，依旧可以保证系统轨迹的精度。Pierzchała^[44]使用基于图优化的 16 线激光雷达、相机、IMU 与 GPS 的 SLAM 系统，对木材结构、树间的相对距离进行评估。Zhang 提出的 LOAM^[45]对激光雷达点云提取面特征与边缘特征以减小匹配的计算量。整个框架采用前端高频低精度的激光雷达里程计进行位姿估计，后端采用低频高精度的图优化进行位姿估计，实现了低计算量、低漂移、

实时的 SLAM 系统，此外 LOAM 算法的精度在 KITTI^[46]数据集上常年位于榜单前列，其效果图如图 1-7 所示。随后，LOAM 的其他变体和更新版本被提出，包括 LeGO-LOAM^[47]、R-LOAM^[48]和 F-LOAM^[49]，这些基于 LOAM 改进算法的共同贡献是改进了 LOAM 算法的精度或处理时间。其中，LeGO-LOAM 是最具影响力的改进方案。对于车载导航系统，激光雷达与 GNSS/INS 的集成是实现系统冗余性和鲁棒性的关键。Shan 提出的 LIO-SAM^[50]在前端使用点云特征提取与关键帧的方式来减小计算量，在后端因子图优化中加入 IMU 预积分因子、GPS 因子、回环因子与激光里程计因子构建了一个多传感器紧耦合的融合和全局优化的 SLAM 框架，其效果图如图 1-8 所示。P3-LOAM^[51]将 LiDAR-SLAM 与 GNSS 的精确点定位（PPP）相结合并且根据 SVD 雅克比矩阵的误差传播模型来估计 SLAM 系统的协方差，并且在 GNSS 观测出现较大误差时，可以依靠激光 SLAM 来消除 PPP 的异常值，在城市峡谷环境下可以完成精度较高的定位。

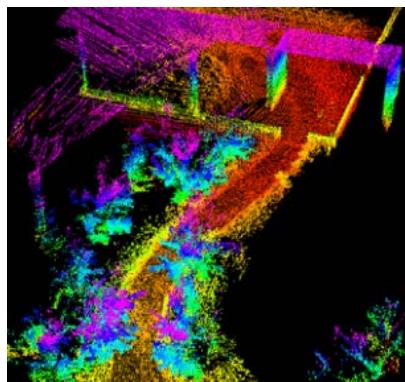


图 1-7 LOAM 效果图

Fig. 1-7 The result of LOAM

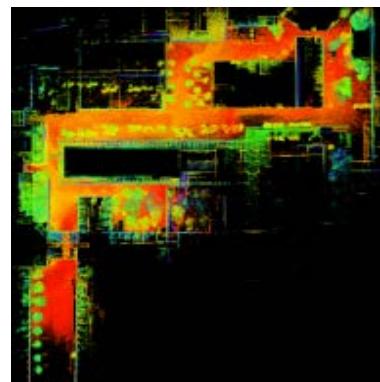


图 1-8 LIO-SAM 效果图

Fig. 1-8 The result of LIO-SAM

张福斌^[52]提出一种加入 MEMS IMU/里程计运动速度的多传感器融合 SLAM，将 MEMS IMU/里程计进行预积分用于对激光点云数据进行运动补偿，同时将 MEMS IMU/里程计的运动约束加入到后端优化中。张钊^[53]基于平面拟合法提出一种局部地图匹配机制与关键帧估计的方法，有解决了 LIO 方案中缺乏关键估计与局部地图中平面多向性问题。付林^[54]提出结合惯性/高度传感器的激光 SLAM 算法，在点云初始化时引入惯性/高度传感器的姿态，有效提高了高程退化环境中定位精度。牛国臣^[55]在传统激光 SLAM 算法中使用点云主成分析法对激光点云进度特征匹配，并且在后端图优化部分加入 GNSS 的位置数据，取得良好的局部地图精度与地图全局一致性。李帅鑫^[56]基于 LOAM 算法

进行改进，提出一种基于角度图像的快速点云分割方法，可以筛选出结构性特征突出的点云，降低了点云数据量，提高了激光里程计的计算效率。孙喜亮^[57]在前端对杆状与面状特征点云进行匹配，提高在城市化环境中定位精度。并且通过图优化融合 GNSS 角点位置约束，进一步提高全局地图构建的准确性。邱佳月^[58]提出一种基于先验信息的点云粗匹配的 SLAM 算法，其对动态点云进行识别并且有效地剔除。在大范围与小范围的定位与建图精度都有着比原算法更高的精度。刘振宇^[59]提出一种融合激光雷达与惯性测量单元融合的算法，在算法中使用帧与局部地图匹配所得的距离值和 IMU 数据构建出联合优化函数提高了原算法的精度。

然而基于图优化的 SLAM 系统，在前端大多采用 LOAM^[45]方案即基于激光点云特征所匹配得到的前端里程计或者是基于正态分布变换算法的前端里程计。然而上述两种方案的前端把激光雷达与 IMU 的数据分开处理，主要是为了快速得到位姿估计而牺牲了精度。因此，结合两种 SLAM 方法提高 SLAM 系统的整体精度，并且在 GPS 出现中断时仍可以保持较高的精度是本文的研究重点。

1.3 本文研究的主要内容

本文针对室外巡检机器人在科技园区室外环境中出现 GPS 中断情况下，以激光雷达为核心传感器的多传感器融合 SLAM 算法精度下降问题进行研究。在室外巡检机器人领域，由于多线激光雷达在室外环境中能够提供高精度的环境信息，因此得到广泛的使用。然而，只使用多线激光雷达的 SLAM 算法，往往会发生点云退化、数据输出频率低等问题。所以，一般都需要高频的惯性测量单元进行辅助。多线激光雷达与惯性测量单元融合的 SLAM 算法虽然可以实现许多场景下的高精度定位与建图，但是无法满足室外巡检机器人对定位的需求。因此为了在室外得到更精确的位姿估计，本文在多线激光雷达与惯性测量单元的基础上加入 GPS 传感器，并且将基于滤波与图优化方法融合上述三种传感器，搭建完整的多传感器融合 SLAM 系统，完成巡检机器人的位姿估计。本文具体研究内容如下：

(1) 为了更准确地融合多个传感器的数据，并控制巡检机器人进行室外导航建图，首先，对室外巡检机器人进行运动学建模，并且使用手柄对巡检机器人进行运动控制。其次为了更精确地融合多个传感器数据，定义巡检机器人底盘及其搭载的传感器的坐标系，并建立各个传感器数据采集模型，完成各个传感器的数据采集实验。最后对各个传感器进行外参标定与时间同步，完成多

传感器数据融合的前置条件。

(2) 针对基于传统前端里程计精度不佳的问题，并且为了巡检机器人在出现的 GPS 中断情况下，有着更精确的定位精度。提出一种基于滤波的激光惯性里程计，通过 IESKF 滤波方法紧耦合多线激光雷达与 IMU。使用以机器人为中心的公式并且采用依赖于状态维度而不是观测维度的卡尔曼增益公式，对激光特征点云与 IMU 数据进行 IESKF 迭代递归修正所估计得到的位姿。在 KITTI 数据集上进行验证，可以取得比传统的前端里程计更高的精度。

(3) 针对经过 IESKF 所得到的激光惯性里程计在全局精度上仍有不足，本文提出一种结合 IESKF 与图优化 SLAM 算法。SLAM 系统的前端采用基于 IESKF 的激光惯性里程计，后端采用因子图融合 IESKF 激光惯性里程计因子、IMU 预积分因子、回环因子与 GPS 因子。并且针对 GPS 数据出现较大波动的情况，提出一种基于 GPS 状态与置信度的筛选方式，防止在因子图中加入异常的 GPS 数据。

(4) 基于 Ubuntu18.04 操作系统下的 ROS (Robot Operating System) 平台，在公开的 KITTI 数据集与实际的巡检机器人实验平台上进行测试，对比分析了本文算法与开源算法在 GPS 不中断与中断情况下的轨迹与全局位姿误差。

第2章 巡检机器人与传感器系统建模

在本文中巡检机器人的底盘运动采用了两轮差速运动学模型，在底盘上搭载了 16 线激光雷达、IMU 与 GPS 传感器。本章节根据实际安装位姿进行传感器坐标系的定义，并且分别构建了巡检机器人与其搭载的传感器的运动学与数据采集模型，完成各个传感器的时空同步。

2.1 巡检机器人与传感器坐标系定义

巡检机器人在定位与建图的时候，需要估计出当前时刻的位姿，而其搭载的传感器所采集到的数据都是基于其自身坐标系，因此需要对各个坐标系进行定义。

(1) 世界坐标系 $O_w X_w Y_w Z_w$

世界坐标系在定位建图初始时刻与巡检机器人底盘坐标系是重合与固定不动的，并且作为参考坐标系用于描述巡检机器人与各个传感器坐标系的位姿。

(2) 巡检机器人底盘坐标系 $O_R X_R Y_R Z_R$

巡检机器人底盘坐标系是用于描述巡检机器人在三维空间中运动位姿的参考坐标系，其定义为 $O_R X_R Y_R Z_R$ 。如下图 2-1 所示，坐标系原点 O_R 为巡检机器人底盘、IMU 与 GPS 的坐标原点。 X_R 指向巡检机器人的正前方为巡检机器人底盘的 x 轴，同时为机器人的中轴线， Y_R 指向巡检机器人的左侧为巡检机器人底盘的 y 轴，同时平行于两个主动轮转轴连接线， Z_R 指向巡检机器人的上方为巡检机器人底盘的 z 轴，同时垂直于两个主动轮转轴连接线。

(3) IMU 坐标系 $O_R X_R Y_R Z_R$

为了简化模型的建立与减小计算的复杂度，在本文中将 IMU 安装于巡检机器人两个主动轮转轴连线的中心点正上方，如图 2-1 所示。则可以将 IMU 与巡检机器人的坐标系看成为同一个坐标系。

(4) GPS 坐标系 $O_G X_G Y_G Z_G$

并且本文采用基于实时动态（Real Time Kinematic, RTK）载波相位差分技术的 GPS 组合定位系统，其主天线安装在巡检机器人前方，辅天线安装在巡检机器人后方。GPS 输出的经纬度及姿态信息是主天线的数据，则 GPS 坐标系如图 2-1 所示的 $O_G X_G Y_G Z_G$ 。 O_G 为 GPS 主天线的几何中心，其 x 轴 X_G 与 X_R 共线， y 轴 Y_G 平行于 Y_R ， z 轴 Z_G 平行于 Z_R 。

(5) 激光雷达坐标系 $O_L X_L Y_L Z_L$

多线激光雷达对于周边环境感知的点云数据都是基于其坐标系而言，并且也是根据点云数据的变化来描述其相对位姿变化的参考坐标系。多线激光雷达坐标系 $O_L X_L Y_L Z_L$ 符合右手定则，如图 2-1 所示。 O_L 为激光雷达几何中心，并且与 O_R 的连线垂直于 $X_R O_R Y_R$ 平面。 X_L 平行于 Y_R 且与 Y_R 方向相反，指向激光雷达的正右侧。 Y_L 指向激光雷达的正前方且平行于 X_R 。

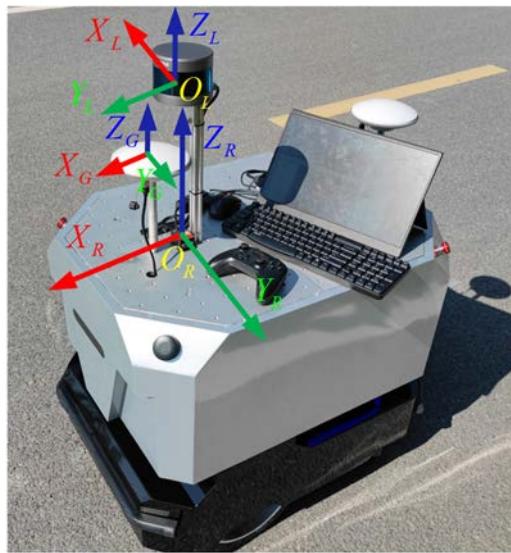


图 2-1 巡检机器人底盘与激光雷达坐标系

Fig. 2-1 Inspection robot body and lidar coordinate system

2.2 巡检机器人及其传感器数据采集模型

本文巡检机器人系统中使用手柄进行对底盘进行控制运动，并且安装了多线激光雷达、IMU 与 GPS 传感器用于环境感知与定位建图，下面介绍巡检机器人运动模型与各个传感器的数据采集模型。

2.2.1 巡检机器人运动学模型

巡检机器人运动模型的建立是对其底盘进行控制的基础，以便能够在科技园区内控制巡检机器人完成导航建图。本系统采用两轮差速移动底盘，运动学模型示意图如下图 2-2 所示。巡检机器人的运动设定在 $O_w X_w Y_w$ 坐标系中，位姿表示为： $p = [x, y, \theta]^T$ ， x, y 表示位置， θ 表示姿态的偏航角。

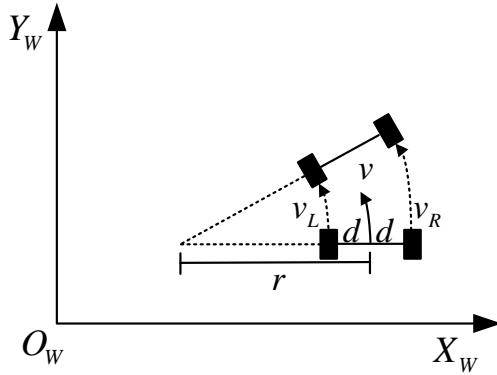


图 2-2 巡检机器人运动学模型
Fig. 2-2 Inspection robot kinematics model

根据图 2-1 巡检机器人运动学模型, v 表示巡检机器人运动速度, v_L 和 v_R 分别为左右轮线速度, d 为左右轮间距的一半, 因此可以计算运动角速度 ω 和运动转向半径 r , 如式(2-1)与(2-2)所示:

$$\omega = \frac{v_R - v_L}{2d} \quad (2-1)$$

$$r = \frac{v}{\omega} = \frac{2d \cdot (v_R + v_L)}{2 \cdot (v_R - v_L)} \quad (2-2)$$

差分移动机器人运动学如式(2-3)所示:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{v_L}{2} & \frac{v_R}{2} \\ -\frac{v_L}{2d} & \frac{v_R}{2d} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \quad (2-3)$$

式中 ω_L 表示左轮角速度, ω_R 表示右轮角速度。

巡检机器人在移动的时候, 我们可以将其连续的运动看成多段相连且是极短距离的离散运动, 并且假设巡检机器人在每段离散运动都是极短时间内进行的匀速直线运动。在这个假设前提下, 航迹推算如式(2-4)所示,

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dx + \epsilon_x \\ dy + \epsilon_y \\ d\theta + \epsilon_\theta \end{bmatrix} \quad (2-4)$$

式中 $[x' \ y' \ \theta']^T$ 为下一时刻机器人预测的位姿, $[dx \ dy \ d\theta]^T$ 为巡检机器人

姿态的变化量， $[\varepsilon_x \ \varepsilon_y \ \varepsilon\theta]^T$ 为巡检机器人姿态变化量的噪声。

本文通过北通斯巴达手柄，如图 2-3 所示，对底盘进行控制。通过无线蓝牙与工控机进行串口通信，在 ROS 系统上传输手柄数据以控制小车的运动。手柄控制节点的话题名为“/Joy”其消息的数据结构为“sensor_msgs/Joy”。



图 2-3 北通斯巴达手柄

Fig. 2-3 North pass spartan handle

2.2.2 激光雷达点云成像模型

激光雷达通过激光发射器发射激光，打到障碍物后激光雷达内置的接受器会接受到激光回波，根据激光测距中计算飞行时间（TOF）原理，就可以获取到障碍物与激光雷达之间的距离数据。从而可以计算出障碍物在激光雷达坐标系下的空间坐标如图 2-4 所示

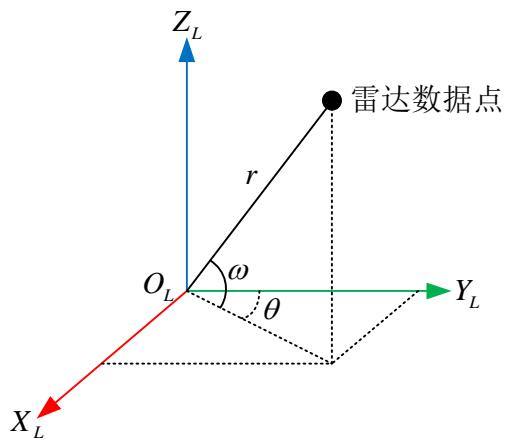


图 2-4 激光雷达单个数据点坐标

Fig. 2-4 Lidar single data point coordinates

激光雷达单个数据点坐标计算如式(2-5)所示:

$$\begin{cases} x = r * \cos(\omega) * \sin(\theta) \\ y = r * \cos(\omega) * \cos(\theta) \\ z = r * \sin(\theta) \end{cases} \quad (2-5)$$

式中 r 为激光雷达测得的距离值, ω 为激光的垂直角, θ 为激光的水平旋转度。

本文采用的是速腾聚创的 RS-16 多线机械激光雷达。通过设定工控机与激光雷达的 IP 地址完成对工控机与激光雷达之间的 IP 地址绑定, 并且设置数据通信串口号传输以太网数据, 将点云数据传送到工控机上, 如图 2-5 所示。

在工控机上接收到激光雷达的点云数据后, 通过 ROS 平台对激光雷达的数据进行解析, 发布 “sensor_msgs::PointCloud2” 数据格式的激光雷达话题名 “/PointCloud2”。具体激光雷达的点云可视化数据可以通过 ROS 平台的可视化界面 RVIZ 中订阅上述话题显示, 如图 2-6 所示。

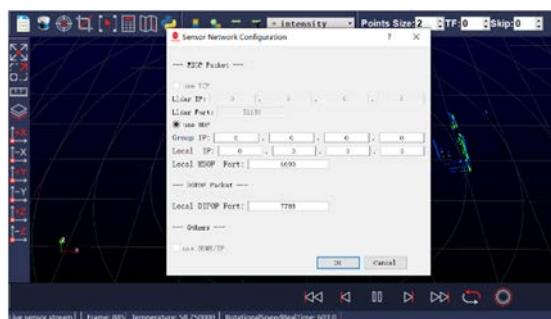


图 2-5 RS-16 以太网连接配置

Fig. 2-5 RS-16 Ethernet connection

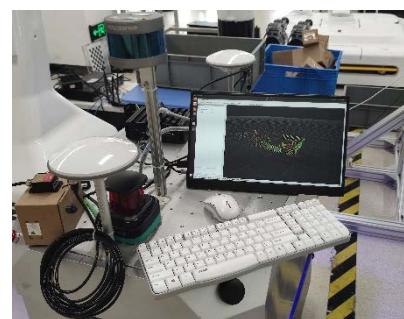


图 2-6 RS-16 点云数据采集

Fig. 2-6 RS-16 point cloud data

2.2.3 IMU 传感器模型

IMU 传感器由三轴正交的加速度计与陀螺仪组成。加速度计可以测量巡检机器人在 $O_RX_RY_RZ_R$ 坐标系下三轴的加速度, 陀螺仪可以测量巡检机器人在 $O_RX_RY_RZ_R$ 坐标系下的三轴角速度。

在 IMU 测量过程中还伴随着零偏与噪声。IMU 的零偏指的是传感器输出在没有物体运动时的基准值。IMU 测量过程中的零偏会导致 IMU 在长时间测量时发生测量数据随机偏移的现象, 通过标定 IMU 的零偏, 可以确保测量结果准确可靠。IMU 测量过程中的噪声会导致 IMU 输出精确性较差的位姿。通过校正 IMU 的输出噪声, 可以提高导航或控制系统的准确性和稳定性。

IMU 测量值是测量真值、测量噪声与传感器零偏之和。其测量模型如式

(2-6)与(2-7)所示。

$$\hat{a}_t = R_t^{RW} (a_t - g) + b_t^a + n_t^a \quad (2-6)$$

$$\hat{\omega}_t = \omega_t + b_t^\omega + n_t^\omega \quad (2-7)$$

式中 a_t 与 ω_t 分别表示 t 时刻测量得加速度与角速度的真值, \hat{a}_t 与 $\hat{\omega}_t$ 分别表示 t 时刻加速度与角速度的测量值, R_t^{RW} 表示 t 时刻世界坐标系 $O_w X_w Y_w Z_w$ 到机器人坐标系 $O_R X_R Y_R Z_R$ 的旋转矩阵, g 表示在世界坐标系 $O_w X_w Y_w Z_w$ 下的重力加速度, b_t^a 与 b_t^ω 分别表示加速度计与陀螺仪的零偏, 一般建模为随机游走, 其导数服从高斯分布, 如式(2-8)所示。 n_t^a 与 n_t^ω 分别表示 t 时刻加速度与角速度测量噪声, 一般建模为高斯白噪声如式(2-9)所示。

$$\dot{b}_t^a \sim \mathcal{N}(0, \sigma_{b_a}^2), \dot{b}_t^\omega \sim \mathcal{N}(0, \sigma_{b_\omega}^2) \quad (2-8)$$

$$n_t^a \sim \mathcal{N}(0, \sigma_a^2), n_t^\omega \sim \mathcal{N}(0, \sigma_\omega^2) \quad (2-9)$$

本文采用的是维特智能的 IMU, 型号为 HWT905, 如图 2-7 所示。通过设定工控机与 IMU 的端口号进行串口数据的通信, 随后打开上位机完成对 IMU 数据输出频率、加速度计等参数的设置与校准, 如图 2-8 所示。并且基于 ROS 平台对 IMU 的数据进行解析, 发布 “sensor_msgs::Imu” 数据格式的 IMU 话题名 “wit_imu”。



图 2-7 维特智能 HWT905

Fig. 2-7 Witte Smart HWT905



图 2-8 IMU 校准界面

Fig. 2-8 IMU calibration interface

Allan 方差法是基于时域的误差分析方法, 并且在惯性导航领域中经常用于标定 IMU 的随机误差。采集 IMU 数据 2 个小时, 基于 Allan 方差法标定 IMU 内参, 结果如表 2-1 所示。

表 2-1 IMU 内部参数

Table 2-1 Internal parameters of IMU

误差名	高斯白噪声数值	高斯随机游走数值
角速度平均值	1.1492191673391677e-03rad/s	1.5149990825893628e-04rad/s
X 轴角速度	1.8475973764858021e-03rad/s	1.7310573973194319e-04rad/s
Y 轴角速度	1.3135309584124579e-03rad/s	1.3234809358384975e-04rad/s
Z 轴角速度	1.2215404924960235e-03rad/s	3.4175390897891535e-04rad/s
加速度平均值	2.8083711101519865e-02m/s ²	3.1803364980091817e-02m/s ²
X 轴加速度	1.9301353318947560e-02m/s ²	1.2250586546553851e-03m/s ²
Y 轴加速度	2.9444151604296801e-02m/s ²	4.0946924359851260e-02m/s ²
Z 轴加速度	3.5505628381315222e-02m/s ²	4.2212584033870353e-02m/s ²

2.2.4 GPS 传感器模型

全球定位系统（GPS）是全球卫星导航系统（GNSS）中的一种，主要提供定位信息服务并且常常用于军用与民用定位导航领域。GPS 输出的经纬度坐标主要是接受至少 4 个卫星的信号并且计算其运转轨迹从而得到伪距定位信息。由于在计算 GPS 卫星与载体接收机之间距离观测值时，存在卫星与接收机之间的钟差，所以在计算距离观测值时需要考虑钟差，则所计算的并非是真实 GPS 卫星与地面的距离即伪距 S ，如式(2-10)所示。

$$\begin{aligned} S &= \tau * c = c * (t_a - t_s) + c * \Delta t \\ &= \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + c * \Delta t, i \in [1, n] \end{aligned} \quad (2-10)$$

式中， τ 为 GPS 卫星信号发送与载体接受信号的时间差， c 为光速， t_a 为载体接收到信号的时间， t_s 为 GPS 卫星信号发送时间， Δt 为 GPS 卫星钟差与载体接收机种差所得到的时间差， x 、 y 、 z 为载体接收机的定位坐标， x_i 、 y_i 、 z_i 为第 i 个卫星坐标。通过建立多组伪距计算方程组，可解得载体的定位坐标。

本论文采用的 GPS 定位系统是基于 RTK 动态高精度定位技术。相较于传统的 GPS 定位模式，RTK 定位精度可以达到厘米级，但是其依旧依靠于 GPS 信号，在高度城市化、树木或隧道地段仍然会出现多路径效应或信号失效等现象。所以在一些复杂环境中无法单独使用 GPS 进行室外巡检机器人的定位，需要融合 SLAM 算法进行定位建图。

安装配置 GPS 定位系统时，为了避免较高的障碍物遮挡，选择将 GPS 定位基站与基站电台安装在智能科技园 C1 栋电梯房房顶，移动站与其天线安装在巡检机器人的内部。其安装位置与 RTK 原理示意图如图 2-9 所示。RTK 使用载体移动站与固定 GPS 基站用于降低载体移动站定位误差并且 RTK 是实时使用差分法来处理载体移动站与固定 GPS 基站卫星的载波信号。所以在采用 RTK 定位方法的 GPS 系统中，固定 GPS 基站将其收到的载波信号通过与其相连接的固定电台发送给载体的移动站，而移动站通过移动电台接收电台信号同时接收卫星的载波信号。接着对两个信号进行差分解算出 GPS 固定基站相对于载体移动站的相对位置关系。最后固定 GPS 基站会把矫正数据发送给载体移动站，并且载体移动站对其进行修正，以测量到更加精确地定位坐标。

GPS 的定位数据通过串口形式进行传输，其定位数据为 NMEA 结构，根据 NMEA-0183 协议进行对其解析，如图 2-10 所示。并且将解析出来的数据通过 ROS 转换成 “sensor_msgs::NavSatFix” 格式的话题，话题名为 “gps_fix”。

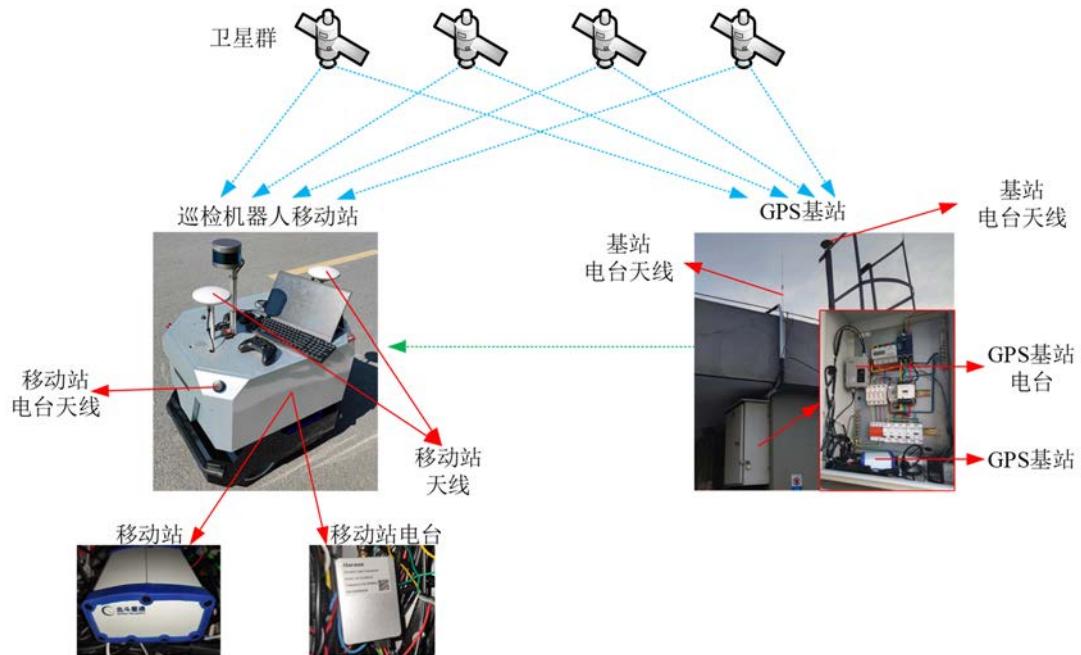


图 2-9 GPS 安装位置与 RTK 原理示意图

Fig. 2-9 Schematic diagram of GPS installation location and RTK principle

```

[16:06:07.860]收->◆$GPGGA,080608,30,3146,68645854,N,11716,35294305,E,1,28,0,8,25,3515,M,-4,4808,M,*4C
[16:06:07.910]收->◆$GPGGA,080608,40,3146,68645276,N,11716,35293913,E,1,28,0,8,25,4207,M,-4,4808,M,*4B
[16:06:08.015]收->◆$GPGGA,080608,50,3146,68644657,N
[16:06:08.046]收->◆,11716,35293570,E,1,28,0,8,25,4861,M,-4,4808,M,*4C
[16:06:08.114]收->◆$GPGGA,080608,60,3146,68643938,N
[16:06:08.145]收->◆,11716,35293226,E,1,28,0,8,25,5532,M,-4,4808,M,*4B
[16:06:08.211]收->◆$GPGGA,080608,70,3146,68643669,N,11716,35292770,E,1,28,0,7,25,6234,M,-4,4808,M,*40
[16:06:08.322]收->◆$GPGGA,080608,80,3146,68643541,N,11716,35292263,E,1,28,0,7,25,6895,M,-4,4808,M,*40
[16:06:08.411]收->◆$GPGGA,080608,90,3146,68643358,N,11716,35291678,E,1,28,0,7,25,7554,M,-4,4808,M,*43
[16:06:08.476]发->◇bestposa
[16:06:08.518]收->◆$GPGGA,080609,00,3146,68643108,N,11716,35291123,E,1,28,0,7,25,8216,M,-4,4808,M,*4B
$command.bestposa.response: OK*53
[16:06:08.615]收
->◆#BESTPOS,COM1,0,58,0,FINE,2216,201987,100,216816,28,18,SOL_COMPUTED,SINGLE,31,77810714761,117.27254845439,25,8911,-
4,4808,WGS84,1,6388,1,5853,3,6417,"",0,0.000,0.000,40,28,23,23,0,06,02,53*d69755e0
$GPGGA,080609,10,3146,6864
[16:06:08.659]收->◆2886,N,11716,35290726,E,1,28,0,7,25,8911,M,-4,4808,M,*4A
[16:06:08.717]收->◆$GPGGA,080609,20,3146,68642819,N,11716,35290357,E,1,28,0,7,25,9446,M,-4,4808,M,*43
[16:06:08.810]收->◆$GPGGA,080609,30,3146,68642809,N,11716,35289867,E,1,28,0,7,25,9924,M,-4,4808,M,*4A

```

图 2-10 GPS 数据解析

Fig. 2-10 GPS data analysis

2.3 多传感器时空标定

2.3.1 多传感器外参标定

本文采用“手眼标定法”对 16 线激光雷达与 IMU，激光雷达与 GPS 进行外参标定，并且采用眼在手上模式即激光雷达作为“眼”安装在载体上，IMU 与 GPS 作为“手”。先对 IMU 与激光雷达进行外参标定，运动关系如式(2-11)所示：

$$T_{R-t_2}^W T_{L-t_2}^{R-t_2} T_O^{L-t_2} = T_{R-t_1}^W T_{L-t_1}^{R-t_1} T_O^{L-t_1} \quad (2-11)$$

式中 T 为位姿变换矩阵，并且规定形式如 T_A^B 表示 A 物体到 B 坐标系下的变换矩阵， t_1, t_2 表示两个不同时刻。 W 、 R 、 L 、 O 分别表示世界坐标系、IMU 坐标系、激光雷达坐标系、标定物体坐标系。对式(2-11)等号两边同时左乘 $T_{R-t_1}^{W-1}$ 与右乘 $T_O^{L-t_2-1}$ ，则可得式(2-12)。

$$T_{R-t_2}^{R-t_1} T_{L-t_2}^{R-t_2} = T_{L-t_1}^{R-t_1} T_{L-t_2}^{L-t_1} \quad (2-12)$$

又因为 IMU 与激光雷达是固定连接，他们之间存在一个静态的坐标变换矩阵即 $T_{L-t_2}^{R-t_2} = T_{L-t_1}^{R-t_1} = T_L^R$ 。 $T_{R-t_2}^{R-t_1}$ 与 $T_{L-t_2}^{L-t_1}$ 分别表示 IMU 与激光雷达在 t_1 与 t_2 时刻

的变换矩阵。IMU 的坐标变换矩阵 $T_{R-t_1}^R$ 可以根据 2.1 节的航迹推算得到，激光雷达坐标变换矩阵 $T_{L-t_2}^L$ 可以根据点云匹配得到，然后根据式(2-12)求得 IMU 与激光雷达的外参变换矩阵 T_L^R ，如式(2-13)所示。同理可求得 GPS 与激光雷达的外参变换矩阵 T_L^G ，如式(2-14)所示。

$$T_L^R = \begin{bmatrix} 0.959766 & -0.276478 & -0.0490861 & -0.0239772 \\ 0.277476 & 0.96062 & 0.0147037 & 0.0150389 \\ 0.0430878 & -0.0277323 & 0.998686 & 0.335897 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-13)$$

$$T_L^G = \begin{bmatrix} 0.958709 & -0.252458 & -0.0505167 & -0.0149853 \\ 0.270461 & 0.965524 & 0.01495273 & 0.0319782 \\ 0.0435524 & -0.027292 & 0.999872 & 0.1828651 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-14)$$

2.3.2 多传感器时间标定

多传感器的时间同步分为硬件时间同步与软件时间同步。本文同时采用硬件时间同步与软件时间同步。因为多线激光雷达、IMU 与 GPS 内部均有不同的时钟源，所以可以使用硬件时间同步使它们在相同时刻触发。然而即使激光雷达、IMU 与 GPS 进行了硬件时间同步，在各个传感器运行一段时间后依然会由于钟飘现象出现时间差，则需要软件时间同步。首先进行硬件同步，使用“PPS+NMEA”的时钟同步方法。GPS 系统可以从卫星上获得原子钟的时钟信号，所以将 GPS 的时钟信号作为系统的时钟源。通过 GPS 系统将 PPS 脉冲信号与 NMEA 时间数据输出至激光雷达，则可完成硬件的时间同步。

然后对 IMU、GPS 与激光雷达进行软件时间同步。激光雷达与 GPS 的数据输出频率分别为 10Hz 与 1Hz，而 IMU 的数据输出频率为 200Hz，明显快于激光雷达与 GPS。本文使用的 ROS 平台可以高效得解决数据输出不同的现象。因为 ROS 消息传输时，会循环读取在消息队列里的数据，并且传感器当前值为队列中对头的位置，解析出其中的时间戳。在 IMU 数据队列中查找时间戳与 GPS、激光雷达相近的两帧数据并对其等比插值，则可得到时间戳相同的 IMU、激光雷达与 GPS 数据。当传感器数据出现卡帧或掉帧时，对差值时间间隔内的数据进行判断与筛选。若激光雷达与 GPS 前后两帧数据的时间差分别大于正确差值时，则要丢弃该数据，具体操作如下：

(1) 若当前帧激光雷达与 GPS 数据的时间戳比 IMU 数据的时间戳小，即

插入时刻前没有 IMU 数据，则舍弃此帧 GPS 与激光雷达数据并且读取下一帧 GPS 与激光雷达数据的时间戳。

(2) 若当前帧激光雷达与 GPS 数据的时间戳比 IMU 数据的时间戳小，下一帧数据的时间戳也比插入时刻早，那么丢弃当前帧的 GPS 数据。

(3) 若当前帧 IMU 数据的时间戳位于两帧激光雷达与 GPS 数据的时间戳中间，则判断两帧数据的时间差是否过大。若时间差过大，则说明出现掉帧或时间戳错误的情况，丢弃当前激光雷达与 GPS 数据。

若各个传感器数据的时间戳无上述情况，则继续多传感器的软件时间同步。

2.4 本章小结

本章首先对本论文中需要的世界坐标系、巡检机器人底盘坐标系、IMU 坐标系、激光雷达坐标系和 GPS 坐标系进行定义。然后重点建立了巡检机器人的运动学建模、多线激光雷达点云成像模型、IMU 与 GPS 的数据采集模型。完成了传感器的内参标定与数据采集实验。最后对各个传感器之间的外参进行标定与完成多传感器时间同步。

第3章 基于滤波的激光惯性里程计

对于 SLAM 系统而言，前端里程计是十分关键的部分，它可以直接影响定位与建图模块是否正常工作。并且由于多线激光雷达与 IMU 自身携带传感器噪声，所以仅根据对激光雷达点云配准来估计相邻帧之间的相对位姿会对定位与建图模块带来较大的误差。尤其是在室外 GPS 信号出现中断时，室外巡检机器人主要依靠多线激光雷达与 IMU 进行定位。本文提出一种对多线激光雷达与 IMU 进行 IESKF 紧耦合的前端激光惯性里程计。首先对激光点云数据进行预处理，其次对激光点云进行特征提取，最后使用 IESKF 递归修正估计的状态。

3.1 激光点云数据预处理

3.1.1 激光点云运动补偿算法

本文采用的机械式的 16 线激光雷达固连在巡检机器人上。在激光雷达内部，激光发射器通过电机进行绕中心轴旋转 360° 来获取周围场景的点云信息。随着巡检机器人的移动，激光雷达与其扫描的点云在一个扫描周期内也进行了相应的运动，使得此帧扫描的起始时刻与终止时刻的坐标系发生了变化。从而导致激光雷达采样回来的点云发生了畸变。点云运动补偿就是消除激光雷达在运动中发生的畸变现在，如图 3-1 所示，正方形为巡检机器人，测量物体的轮廓为黑色粗线，红色虚线为激光雷达所测量的点云数据。

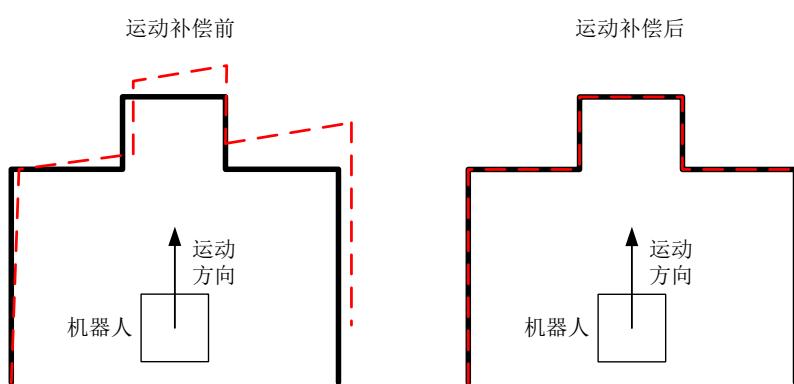


图 3-1 激光点云运动补偿前后对比

Fig. 3-1 Comparison of before and after laser point cloud motion compensation

由于本文采用的是多线激光雷达，在相同水平角位置一帧激光雷达会发射多束激光线束，并且附带激光线束的 ID。如图 3-2 所示，在时刻 t_k 激光雷达开始当前帧的扫描，在时刻 t_{k+1} 激光雷达完成当前帧的扫描并立即开始下一帧的扫描。水平虚线箭头表示将时刻 t_k 的激光线束所扫描获取的点云投影到时刻 t_{k+1} 上， P_k 表示当前帧扫描所得的点云， P_{k+1} 表示下一帧扫描所得的点云， \bar{P}_k 表示将点云 P_k 通过位姿转换到时刻 t_{k+1} 下。

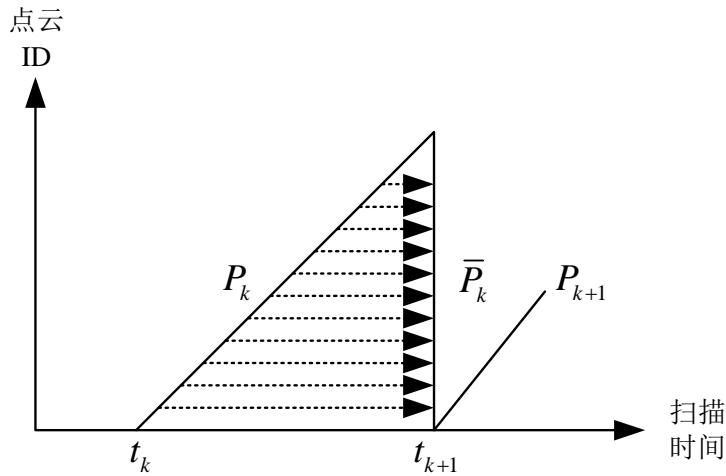


图 3-2 激光雷达点云映射过程
Fig. 3-2 LiDAR point cloud mapping process

本文使用 IMU 数据进行上述的位姿转换，完成对激光雷达点云的运动补偿。IMU 数据输出频率为 200Hz，远快于激光雷达点云数据输出频率 10Hz，在两帧激光雷达之间可以测量得到 20 帧 IMU 数据，并且可以根据 IMU 测量得到的角速度与加速度完成对激光点云的旋转与平移操作。

第一，计算时刻 t_k 与 t_{k+1} 之间，所有在世界坐标系下的 IMU 数据的三轴角度如公式(3-1)所示：

$$a_{t_{i+1}} = a_{t_i} + \omega_{t_i} * \Delta t \quad (3-1)$$

式中 $a_{t_i} = [\varphi_{t_i}, \theta_{t_i}, \psi_{t_i}]^T$ 与 $a_{t_{i+1}}$ 表示时刻 t_i 与 t_{i+1} 的世界坐标系下 IMU 数据的三轴角度， ω_{t_i} 表示时刻 t_i 的世界坐标系下 IMU 数据的三轴角速度， $\Delta t = t_{i+1} - t_i$ 表示时刻 t_i 与 t_{i+1} 的时间间隔， $t_k \leq t_i < t_{i+1} \leq t_{k+1}$ 。

第二，计算点云的三轴角度与相对于时刻 t_k 的三轴位移，如式(3-2)所示：

$$\begin{cases} a_{t_j} = \frac{a_{t_{i+1}}(t_{i+1}-t_j) + a_{t_i}(t_j-t_i)}{t_{i+1}-t_i} \\ p_{t_j} = \frac{(t_j-t_k)p_{t_k}}{t_{k+1}-t_k} \end{cases} \quad (3-2)$$

式中 $p_{t_j} = [p_{t_j}(x), p_{t_j}(y), p_{t_j}(z)]^T$ 表示激光点云从时刻 t_k 到时刻 t_j 的相对位移变换。 p_{t_k} 表示时刻 t_k 到 t_{k+1} 的平移矩阵， $t_k \leq t_i \leq t_j < t_{i+1} \leq t_{k+1}$ 。

第三，计算点云相对于世界坐标系的位姿变换矩阵 R_{t_j} ，如式(3-3)所示：

$$R_{t_j} = \begin{bmatrix} R & p_{t_j} \\ 0 & 1 \end{bmatrix} \quad (3-3)$$

式中旋转矩阵 $R = R_z(\varphi_{t_j})R_y(\theta_{t_j})R_x(\psi_{t_j})$ ，表示 IMU 所测得欧拉角转换所得。

并且计算时刻 t_j 激光点云相对于时刻 t_k 的位姿变换矩阵 $R_{t_k}^{t_j}$ ，如式(3-4)所示：

$$R_{t_k}^{t_j} = R_{t_k}^{-T} R_{t_j} \quad (3-4)$$

式中 R_{t_k} 表示激光点云在时刻 t_k 相对于世界坐标系下的位姿变换矩阵。

第四，将时刻 t_j 下的点云坐标 $[x_j \ y_j \ z_j]^T$ 转换到时刻 t_k 的坐标系下，得到坐标 $[x_k \ y_k \ z_k]^T$ 如公式(3-5)所示。

$$\begin{bmatrix} x_k \\ y_k \\ z_k \\ 1 \end{bmatrix} = R_{t_k}^{t_j T} \begin{bmatrix} x_j \\ y_j \\ z_j \\ 1 \end{bmatrix} \quad (3-5)$$

3.1.2 激光点云体素滤波算法

16 线激光雷达采集的点云数据量较大，在不考虑数据采集异常点云的情况下，在采集周期内，每帧可以采集到约 30000 个激光点。而在实际环境中，激光雷达采集的点云数据会包含数据异常点如噪声点、离群点等。这些异常点云会影响后续激光惯性里程计的精度。所以需要对激光点云进行滤波，在确保不丢失点云几何特征的前提下，降低激光点云的数据量并且提高计算效率。本文对原始激光点云进行体素滤波，体素滤波示意图如图 3-3 所示，其中蓝色点为原始点云，红色点为所有小栅格内蓝色点云完成体素滤波后的点云重心。

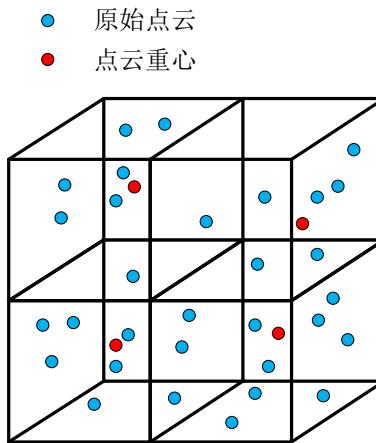


图 3-3 体素滤波示意图

Fig. 3-3 Schematic diagram of voxel filtering

第一，根据激光点云数据集合找到在激光雷达坐标系三轴上最大值与最小值，表示为： x_{\max} 、 x_{\min} 、 y_{\max} 、 y_{\min} 、 z_{\max} 、 z_{\min} 。

第二，计算可以包围所有激光点云的最小长方体，长方体在激光雷达坐标系三轴上的边长 l_x 、 l_y 、 l_z 如式(3-6)所示：

$$\begin{cases} l_x = x_{\max} - x_{\min} \\ l_y = y_{\max} - y_{\min} \\ l_z = z_{\max} - z_{\min} \end{cases} \quad (3-6)$$

第三，人为设定体素栅格的大小 r_x 、 r_y 、 r_z ，结合式(3-6)所求得的 l_x 、 l_y 、 l_z ，并且计算出体素栅格的数量，如式(3-7)所示：

$$\begin{cases} n_x = l_x / r_x \\ n_y = l_y / r_y \\ n_z = l_z / r_z \\ N = n_x * n_y * n_z \end{cases} \quad (3-7)$$

式中 n_x 、 n_y 、 n_z 分别表示体素栅格在激光雷达坐标系三轴上的数量， N 表示体素栅格的数量。

第四，对所有体素栅格进行编号同时计算其索引*i*、*j*、*k*，如式(3-8)所示：

$$\begin{cases} i = \lfloor (x - x_{\min}) / r_x \rfloor \\ j = \lfloor (y - y_{\min}) / r_y \rfloor \\ k = \lfloor (z - z_{\min}) / r_z \rfloor \end{cases} \quad (3-8)$$

式中 x 、 y 、 z 为当前体素栅格在激光雷达坐标系下的坐标， $\lfloor \cdot \rfloor$ 表示向下取整。

第五，计算各个体素栅格的激光点云重心用于整体代替体素栅格中所有的点云数据，如式(3-9)所示：

$$g_{ijk} = \frac{1}{m} \sum_{i=1}^m p_i \quad (3-9)$$

式中 g_{ijk} 表示索引为 i 、 j 、 k 体素栅格的重心， p_i 表示体素栅格中的激光点云， m 表示当前索引的体素栅格中所有激光点云的数量。

如图 3-4 所示，为原始的激光雷达点云数据，图 3-5 为完成体素滤波后的点云数据。经过素滤波后的点云数量更少且拥有与原始点云想同的结构特征。

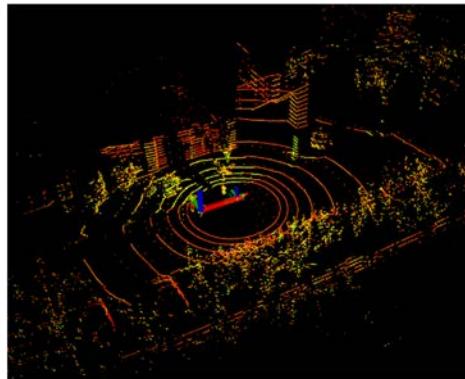


图 3-4 激光雷达原始点云

Fig. 3-4 LiDAR raw point cloud

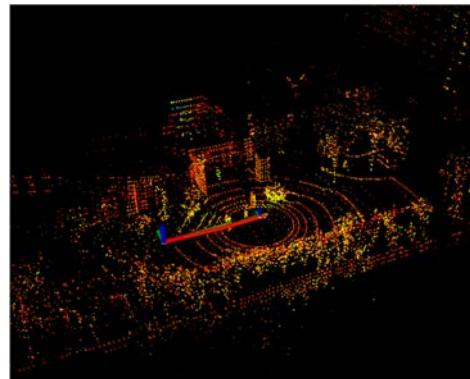


图 3-5 激光雷达体素点云

Fig. 3-5 LiDAR voxel point cloud

3.2 传统的激光里程计

3.2.1 基于正态分布变换的激光里程计

基于迭代最近点 (Iterative Closest Point, ICP) 的点云匹配算法对点云位姿初值较为敏感，迭代过程中容易出现局部最优的情况。相较于 ICP 点云匹配算法，基于正态分布变换算法 (Normal Distribution Transform, NDT) 的点云匹配算法不考虑点与点之间的欧式距离，而是将点云转换为多维度变量的正态

分布。在 KITTI 数据集上，基于 NDT 点云匹配算法的精度要优于基于 ICP 的点云匹配算法^[60]，所以本文在 3.4 节实验部分将基于 IESKF 的前端里程计与基于 NDT 的前端里程计进行比较。

基于 NDT 的点云匹配算法首先将激光点云所在的空间划分为立体的空间栅格，则栅格内点云匹配源点云集合为 $S = \{s_1, s_2, \dots, s_n\}$ ，点云匹配目标点云集合为 $G = \{g_1, g_2, \dots, g_n\}$ 。

第一，根据式(3-10)与(3-11)计算空间栅格内点云的均值与协方差：

$$\mu = \frac{1}{n} \sum_{i=1}^n s_i \quad (3-10)$$

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^n (s_i - \mu)(s_i - \mu)^T \quad (3-11)$$

式中 n 为栅格内点云的数量。

第二，初始化位姿变换矩阵 T ，并且根据式(3-12)将点云匹配目标点云通过变换矩阵 T 转换到点云匹配源点云的坐标系下。

$$g'_i = Rg_i + t \quad (3-12)$$

式中 R 式变换矩阵 T 中的旋转矩阵， t 是变换矩阵 T 中的平移向量。

第三，根据式(3-13)计算激光各点云的联合概率密度 Ψ ，并且为了简化问题，对联合概率密度取对数，如式(3-14)所示。

$$\Psi = \prod_{i=1}^{n_s} \frac{1}{\sqrt{2\pi} \sqrt{|\Sigma|}} \exp\left(-\frac{(g'_i - \mu)^T \Sigma^{-1} (g'_i - \mu)}{2}\right) \quad (3-13)$$

$$\ln \Psi = \sum_{i=1}^{n_s} \left(-\frac{(g'_i - \mu)^T \Sigma^{-1} (g'_i - \mu)}{2} + \ln\left(\frac{1}{\sqrt{2\pi} \sqrt{|\Sigma|}}\right) \right) \quad (3-14)$$

然后，为了求解联合概率密度的最大值，使(3-14)取最大值。去除常数项 $\ln\left(\left(\sqrt{2\pi} \sqrt{|\Sigma|}\right)^{-1}\right)$ ，则 $(g'_i - \mu)^T \Sigma^{-1} (g'_i - \mu)$ 越小，联合概率密度的对数就越大。

令 $e_i = g'_i - \mu$ 与 $m_i = \sum_{i=1}^{n_s} e_i^T \Sigma^{-1} e_i$ ，由式(3-12)可知，变换矩阵 T 越小，则 m_i 越小。

所以需要经过不断迭代求解出 ΔT ，使得式(3-15)最小。

$$m_i(T + \Delta T) = \sum_{i=1}^{n_s} e_i^T (T + \Delta T) \Sigma^{-1} e_i (T + \Delta T) \quad (3-15)$$

如式(3-16)所示，对 $e_i^T(T + \Delta T)$ 进行泰勒展开：

$$\begin{cases} e_i(T + \Delta T) \approx e_i(T) + \frac{de_i}{dT} \Delta T = e_i + J_i \Delta T \\ m_i(T + \Delta T) = e_i^T \Sigma^{-1} e_i + 2e_i^T \Sigma^{-1} J_i \Delta T + \Delta T^T J_i^T \Sigma^{-1} J_i \Delta T \end{cases} \quad (3-16)$$

式中 J_i 为 e_i 的雅克比矩阵。

令 $a_i^T = e_i^T \Sigma^{-1} J_i$ 与 $H_i = J_i^T \Sigma^{-1} J_i$ 得式(3-17)。则将求解式(3-15)最小转化为求解式(3-17)最小。对式(3-17)求导，令其导数等于零，如式(3-18)所示，求得 ΔT 。

$$\Delta m_i(T) = m_i(T + \Delta T) - m_i(T) = 2a_i^T \Delta T + \Delta T^T H_i \Delta T \quad (3-17)$$

$$\frac{d\Delta m_i(T)}{d\Delta T} = 2a_i + 2H_i \Delta T = 0 \quad (3-18)$$

第四，在求解出变换矩阵后，转至第二步继续迭代求解，直到收敛。则可求解出两帧激光点云之间的位姿变换矩阵。

3.2.2 基于点云特征匹配的激光里程计

(1) 激光点云特征提取算法

在完成体素滤波后，激光点云在数量上有所减少，但是如果使用传统的 ICP 算法对体素滤波后的点云进行匹配，仍然会占据较多的内存资源，并且匹配时间依旧较长，LOAM 算法提出了基于点云特征的匹配算法，减小点云匹配的计算量，同时使用莱文贝格-马夸特方法 (Levenberg-Marquardt algorithm, L-M)，减少点云匹配时间。16 线的激光雷达在垂直方向上有 16 条激光扫描线束，因此对一周扫描时间内每束激光束进行特征提取。曲率是判断激光雷达感知周围环境几何面的弯曲程度指标，计算各个激光扫描线束上每个激光点与其前后各 5 个激光点的曲率。激光点 i 的曲率 c ，如式(3-19)所示：

$$c = \frac{1}{|S| \cdot \|X_{(k,i)}^L\|} \left\| \sum_{j \in S, j \neq i} (X_{(k,i)}^L - X_{(k,j)}^L) \right\| \quad (3-19)$$

式中 S 表示以激光点 i 为中心周围连续激光点的集合， $X_{(k,i)}^L$ 与 $X_{(k,j)}^L$ 分别表示在激光雷达坐标系 $O_L X_L Y_L Z_L$ 下第 k 帧激光雷达扫描激光束中激光点 i 与其为中心周围的激光点 j 。

接着对所有激光点的曲率值进行排序，根据曲率大小排序为曲率大于阈值的边缘点与曲率小于阈值的平面点。由于激光点云集合中激光点的分布是不均匀的，为了使激光点云特征点分布均匀，将每条激光扫描线束上分为 4 个相同

的子区域，同时限制每个子区域中提取的特征点至多为 2 个边缘点与 4 个平面点。在激光点云特征点提取时，会出现两种含有歧义的特征点。第一种激光点云特征点如图 3-6 所示，当激光雷达的激光线束与障碍物表面的夹角过小则计算获取到的激光点 A 与激光雷达的距离时会出现误差，所以只考虑障碍物表面与激光线束之间夹角够大的激光点 B 为特征点。第二种激光点云特征点如图 3-7 所示，激光点 B 所在障碍物表面被另一个物体遮挡，遮挡部分使用虚线表示。在计算激光点 B 的曲率时，会将激光点 A 包括进点云集合中，从而计算得到的曲率值偏大，经过判断后将激光点 B 认定为边缘点。然而实际上激光点 B 应该认定为平面点，此时出现了误判，则需要判断相邻激光点的距离信息是否差距过大进而对其剔除。

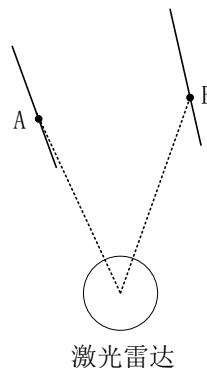


图 3-6 物体平面平行激光线束

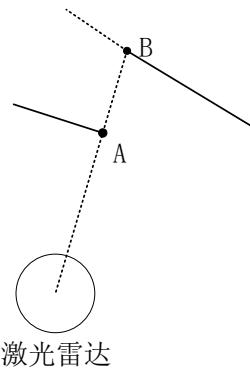


图 3-7 物体平面遮挡激光线束

Fig. 3-6 Object plane parallel laser beam

Fig. 3-7 Object plane blocks laser beam

图 3-8 所示，为完成点云特征提取之后的点云数据，其中绿色的为边缘点，紫色的为平面点。

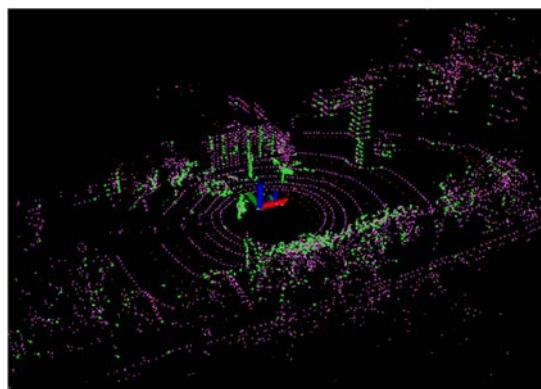


图 3-8 激光雷达特征点云

Fig. 3-8 LiDAR feature point cloud

(2) 激光点云特征匹配算法

上一节中对激光点云进行了边缘特征与平面特征的提取，接下来需要对边缘特征与平面特征的激光点云进行点云匹配，并且计算出相邻两帧的相对位姿变换。

假设第 k 帧激光点云为 P_k ，第 k 帧边缘特征点云集合为 e_k ，第 k 帧平面特征点云集合为 s_k ，第 $k+1$ 帧激光点云为 P_{k+1} ，第 $k+1$ 帧边缘特征点云集合为 e_{k+1} ，第 $k+1$ 帧平面特征点云集合为 s_{k+1} 。

由于激光雷达在第 k 帧到第 $k+1$ 帧这段时间内是在运动的，所以为了一帧中的所有特征点云都可以得到对应的姿态变换信息，将当前的所有特征点云投影到每一帧的起始时刻，投影至每一帧起始时刻的边缘特征点云与平面特征点云分别表示为： \tilde{e}_{k+1} 与 \tilde{s}_{k+1} 。此时投影后的特征点云与前一帧特征点云都处于第 $k+1$ 帧起始时刻的坐标系下。

边缘特征点云就是在三维空间中线或面的端点，如图 3-9 所示。求解点到直线的最短距离就是求解第 k 帧边缘特征点云到第 $k+1$ 帧边缘特征点云之间最优变换。首先在集合 \tilde{e}_{k+1} 选取特征点 i ，其坐标为 $\tilde{X}_{(k+1,i)}^{L_e}$ ，使用数据结构 kd-tree 在集合 e_k 中找到与特征点 i 距离最近的特征点 j ，其坐标为 $X_{(k,j)}^{L_e}$ ，并且在 j 所在激光束的相邻线束上找到距离 j 最近的特征点 l ，其坐标为 $X_{(k,l)}^{L_e}$ 。 i 、 j 、 l 三点不共线，特征点 i 到直线的最短距离 d_e 求解如式(3-20)所示。

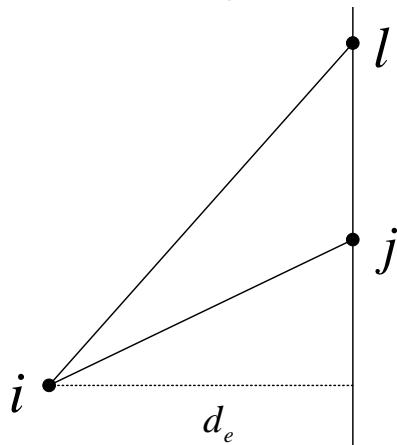


图 3-9 边缘点匹配示意图

Fig. 3-9 Schematic diagram of edge point matching

$$d_e = \frac{\left| \left(\tilde{X}_{(k+1,i)}^{L_e} - X_{(k+1,i)}^{L_e} \right) \times \left(\tilde{X}_{(k+1,i)}^{L_e} - X_{(k,l)}^{L_e} \right) \right|}{\left| X_{(k,j)}^{L_e} - X_{(k,l)}^{L_e} \right|} \quad (3-20)$$

平面特征点是三维空间中线或面上的点，如图 3-10 所示。求解点到平面的最短距离就是求解第 k 帧平面特征点云到第 $k+1$ 帧平面特征点云之间最优变换。平面特征点云与边缘特征点云的选取方式相似。

从集合 \tilde{s}_{k+1} 中选取特征点 i ，其坐标为 $\tilde{X}_{(k+1,i)}^{L_s}$ ，使用数据结构 kd-tree 在集合 s_k 中找到与特征点 i 距离最近的特征点 j ，其坐标为 $X_{(k,j)}^{L_s}$ ，并且找到在特征点 i 所在激光束的次近点 l 与特征点 j 相邻扫描线上的最近特征点 m ，其坐标分别为 $X_{(k,l)}^{L_s}$ 与 $X_{(k,m)}^{L_s}$ 。 j 、 l 、 m 不共线且共面。点到平面最短距离 d_s 如式(3-21)所示。

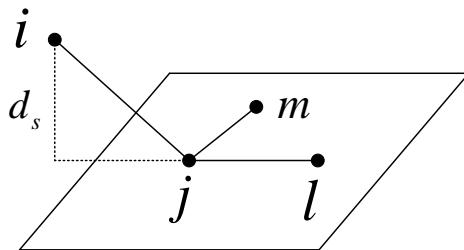


图 3-10 平面点匹配示意图

Fig. 3-10 Schematic diagram of plane point matching

$$d_s = \frac{\left| (\tilde{X}_{(k+1,i)}^{L_s} - X_{(k,j)}^{L_s}) \cdot ((X_{(k,j)}^{L_s} - X_{(k,l)}^{L_s}) \times (X_{(k,j)}^{L_s} - X_{(k,m)}^{L_s})) \right|}{\left| (X_{(k,j)}^{L_s} - X_{(k,l)}^{L_s}) \times (X_{(k,j)}^{L_s} - X_{(k,m)}^{L_s}) \right|} \quad (3-21)$$

在获取到直线的最短距离 d_e 与平面最短距离 d_s 之后，只需要求得 d_e 与 d_s 的最小值就可以获得连续两帧激光特征点云之间的最优位姿变换。根据 3.1.1 获得位姿变换矩阵，设定激光特征点云在第 $k+1$ 帧内的位姿变换矩阵为 $T_{k+1}^L = [r_\alpha \ r_\beta \ r_\gamma \ t_x \ t_y \ t_z]$ ，在式(3-11)与式(3-12)中，待求变量实则为旋转参数 $r = [r_\alpha \ r_\beta \ r_\gamma]$ 与平移参数 $t = [t_x \ t_y \ t_z]$ ，则将式(3-20)与式(3-21)改写成约束方程，如式(3-22)所示：

$$\begin{cases} f_e(r, t) = d_e \\ f_s(r, t) = d_s \end{cases} \quad (3-22)$$

位姿变换公式如式(3-23)所示。

$$X_{(k+1,i)}^L = r\tilde{X}_{(k+1,i)}^{L_s} + t \quad (3-23)$$

根据式(3-20)与式(3-23)，推导出边缘点约束公式，如式(3-24)所示。

$$f_e\left(X_{(k+1,i)}^{L_e}, T_{k+1}^{L_e}\right) = d_e \quad (3-24)$$

根据式(3-21)与式(3-23), 推导出平面点约束公式, 如式(3-25)所示。

$$f_s\left(X_{(k+1,i)}^{L_s}, T_{k+1}^{L_s}\right) = d_s \quad (3-25)$$

为了方便推导, 将式(3-24)与式(3-26)统一为一个约束方程, 如式(3-26)所示:

$$f\left(T_{k+1}^L\right) = d \quad (3-26)$$

式中 d 表示求解的距离。

为了使距离 d 最小, 通过 L-M 法, 对上述约束方程进行优化求解, 如式(3-27)所示:

$$T_{k+1}^L \leftarrow T_{k+1}^L - \left(J^T J + \lambda \text{diag}(J^T J) \right)^{-1} J^T d \quad (3-27)$$

式中 $J = \frac{\partial f}{\partial T_{k+1}^L}$ 是约束方程 f 对于 T_{k+1}^L 的雅克比矩阵。

3.3 基于 IESKF 的激光惯性里程计

为了稳定的位姿估计, 采用以巡检机器人为中心的迭代误差状态卡尔曼滤波器, 在 SLAM 前端前端部分实现激光雷达和 IMU 的传感器融合。本文使用 FAST-LIO^[31]中 IESKF 更新方式。观测协方差矩阵采用 LINS^[30]中通过链式法则求偏导的方法, 此形式与 FAST-LIO 中的基于一阶泰勒展开的观测协方差公式等价。此外, 为了算法代码的工程化, 使用 3.1.1 节中介绍点云补偿方法将点云补偿至 $k+1$ 时刻, 此方法等价于 FAST-LIO 中用于将点云补偿至 $k+1$ 时刻的反向传播环节。

3.3.1 IESKF 状态定义

在描述 IESKF 滤波过程之前, 先进行符号的规定: 符号 $\dot{[•]}$ 表示对变量进行微分, 符号 $[•]_x$ 表示变量的反对称阵, 符号 $[•]^y$ 表示变量的反对称阵对应的向量, 符号 \bullet 表示后验状态变量, 符号 \circlearrowleft 表示先验状态变量, 符号 \oplus 与 \ominus 分别表示广义的加号与减号, \otimes 表示四元数与以自然常数 e 为底的指数的乘积。

在基于 IESKF 传感器融合系统中, 需要对位置位移 p 、速度 v 、姿态位姿

R 、IMU 加速度计的零偏 b_a 、IMU 陀螺仪的零偏 b_ω 与重力加速度 g ，进行状态估计，则定义系统状态变量 x 定义如式(3-28)所示。

$$x = [R^T \quad v^T \quad p^T \quad b_\omega^T \quad b_a^T \quad g^T]^T \quad (3-28)$$

并且由于 IESKF 式对状态变量的误差进行估计，则误差状态变量 δx 定义如式(3-29)所示：

$$\delta x = [\delta\theta^T \quad \delta p^T \quad \delta v^T \quad \delta b_\omega^T \quad \delta b_a^T \quad \delta g^T]^T \quad (3-29)$$

式中 $\delta\theta$ 表示位姿角度的误差， δp 表示位置误差， δv 表示速度误差， δb_ω 表示角速度零偏误差， δb_a 表示加速度零偏的误差， δg 表示重力加速度误差，其误差为 0。

对系统状态变量 x 进行离散化向前传播，如式(3-30)所示：

$$\begin{cases} \tilde{x}_{i+1} = \tilde{x}_i \oplus [T \cdot f(\tilde{x}_i, u_i, \omega_i)] \\ \tilde{x}_k = \hat{x}_k \end{cases} \quad (3-30)$$

式中 \tilde{x}_{i+1} 与 \tilde{x}_k 分别表示在激光雷达帧 k 与 $k+1$ 帧之间接收到 IMU 数据的 i 与 $i+1$ 时刻的先验系统状态变量。 $T = t_{i+1} - t_i$ ，表示 IMU 采样周期， $f(\tilde{x}_i, u_i, \omega_i)$ 为系统状态转移矩阵， u_i 与 ω_i 表示 i 时刻 IMU 的测量值与其测量的噪声。

u_i 、 ω_i 与 $f(\tilde{x}_i, u_i, \omega_i)$ 分别如式(3-31)、(3-32)与(3-33)所示：

$$u_i = [\omega_i^T \quad a_i^T]^T \quad (3-31)$$

$$\omega_i = [n_{\omega,i}^T \quad n_{a,i}^T \quad n_{b_\omega,i}^T \quad n_{b_a,i}^T]^T \quad (3-32)$$

$$f(\tilde{x}_i, u_i, \omega_i) = \begin{bmatrix} \omega_i - b_{\omega,i} - n_{b_\omega,i} \\ v + \frac{1}{2} \left(R_i (a_i - b_{a,i} - n_{b_a,i}) + g_i \right) \\ R_i (a_i - b_{a,i} - n_{b_a,i}) + g_i \\ n_{b_\omega,i} \\ n_{b_a,i} \\ 0_{3 \times 1} \end{bmatrix} \quad (3-33)$$

式中 a 与 ω 分别表示 i 时刻 IMU 测量的角速度与加速度。 $n_{\omega,i}$ 与 $n_{a,i}$ 分别表示 i 时

刻 IMU 测量的角速度与加速度的白噪声, $n_{b_\omega,i}$ 与 $n_{b_a,i}$ 分别表示 i 时刻 IMU 陀螺仪与加速度计的零偏白噪声, R_i 表示 i 时刻位姿估计的旋转矩阵。

3.3.2 IESKF 状态方程

对接受到激光雷达帧第 k 帧与第 $k+1$ 帧之间最新收到的一帧 IMU 数据进行解算, 所得的 IMU 误差方程如式(3-34)所示:

$$\begin{cases} \dot{\delta\theta} = -[\omega_t - b_{\omega_t}]_x \delta\theta - \delta b_\omega - n_\omega \\ \dot{\delta p} = \delta v \\ \dot{\delta v} = -R_t [a_t - b_{a_t}]_x \delta\theta + R_t \delta b_a - n_a + \delta g \\ \dot{\delta b_\omega} = n_{b_\omega} \\ \dot{\delta b_a} = n_{b_a} \\ \dot{\delta g} = 0 \end{cases} \quad (3-34)$$

式中 $\dot{\delta p}$ 表示位置误差量的微分, $\dot{\delta v}$ 表示速度误差的微分, $\dot{\delta g}$ 表示重力加速度的误差, R_t 表示 t 时刻的旋转矩阵, a_t 表示 t 时刻的加速度, b_{a_t} 表示 a_t 的零偏, ω_t 表示 t 时刻的角速度, b_{ω_t} 表示 ω_t 的零偏, n_a 表示加速度的白噪声, ω_t 表示 t 时刻的角速度, n_ω 表示角速度的白噪声, n_{b_a} 与 n_{b_ω} 分别表示加速度计的白噪声与角速度计的白噪声。

运动噪声变量 $\omega = [n_\omega^T \quad n_a^T \quad n_{b_\omega}^T \quad n_{b_a}^T \quad 0]^T$ 。则 IESKF 的状态方程如式(3-35)所示:

$$\dot{\delta x} = A_t \delta x + B_t \omega \quad (3-35)$$

$$A_t = \begin{bmatrix} -[\omega_t - b_{\omega_t}]_x & 0 & 0 & -I_3 & 0 & 0 \\ 0 & 0 & I_3 & 0 & 0 & 0 \\ -R_t [a_t - b_{a_t}]_x & 0 & 0 & 0 & -R_t & I_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3-36)$$

$$B_t = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -R_t & 0 & 0 & 0 \\ 0 & -I_3 & 0 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & I_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3-37)$$

式中 A_t 与 B_t 分别表示状态误差变量 δx 的误差状态转移矩阵和运动噪声变量 ω 的噪声矩阵，如式(3-36)与式(3-37)所示。

因为上述的状态方程推导是基于连续时间的，构建 IESKF 滤波器需要把前面推导的状态方程应用于离散时间，所以需要按照采样时间 T 进行离散化。使用矩阵指数法将状态矩阵进行离散化如式(3-38)所示：

$$F_i = e^{A_t T} = I + A_t T + \frac{(A_t T)^2}{2!} + \frac{(A_t T)^3}{3!} + \dots \quad (3-38)$$

为了方便计算，忽略数值无穷小项并且取式(3-38)前两项，得到离散状态矩阵的近似值，如式(3-39)所示：

$$F_i = I + A_t T_t = \begin{bmatrix} I_3 - [\omega_i - b_{\omega_i}]_x T & 0 & 0 & -I_3 T & 0 & 0 \\ 0 & I_3 & I_3 T & 0 & 0 & 0 \\ -R_i [a_i - b_{a_i}]_x T & 0 & I_3 & 0 & -R_i T & I_3 T \\ 0 & 0 & 0 & I_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_3 \end{bmatrix} \quad (3-39)$$

根据 $\exp(x)$ 一阶近似泰勒展开可得 $\exp[-[\omega_i - b_{\omega_i}]_x T] = I_3 - [\omega_i - b_{\omega_i}]_x T$ ， 并且带入式(3-39)得到最终的离散状态矩阵，如式(3-40)所示：

$$F_i = \begin{bmatrix} \exp[-[\omega_i - b_{\omega_i}]_x T] & 0 & 0 & -I_3 T & 0 & 0 \\ 0 & I_3 & I_3 T & 0 & 0 & 0 \\ -R_i [a_i - b_{a_i}]_x T & 0 & I_3 & 0 & -R_i T & I_3 T \\ 0 & 0 & 0 & I_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_3 \end{bmatrix} \quad (3-40)$$

计算过程噪声项 $B_i Q B_i^T$, 式中噪声协方差矩阵 Q 定义如式(3-41)所示。

$$\begin{aligned} Q &= \omega \omega^T \\ &= \begin{bmatrix} n_\omega^2 & 0 & 0 & 0 \\ 0 & n_a^2 & 0 & 0 \\ 0 & 0 & n_{b_\omega}^2 & 0 \\ 0 & 0 & 0 & n_{b_a}^2 \end{bmatrix} \end{aligned} \quad (3-41)$$

运动噪声变量 ω 的离散噪声矩阵 B_i 的第一项应为 $-A(\omega_i - b_{\omega_i})T$, 当 $(\omega_i - b_{\omega_i})T$ 为小量时 $-A(\omega_i - b_{\omega_i})T \approx I_3 T$, 则 B_i 简化后的计算公式如式(3-42)所示:

$$B_i = \begin{bmatrix} I_3 T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & R_i T & 0 & 0 \\ 0 & 0 & I_3 T & 0 \\ 0 & 0 & 0 & I_3 T \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3-42)$$

将式(3-40)、式(3-41)、式(3-42)与运动噪声变量 ω , 分别带入离散状态方程进行状态预测与协方差预测方程进行协方差预测, 求得离散误差状态变量 δx_{i+1} 与预测的协方差 \check{P}_{i+1} , 如式(3-43)与(3-44)所示:

$$\delta x_{i+1} = F_i \delta x_i + B_i \omega_{i+1} \quad (3-43)$$

$$\begin{cases} \check{P}_{i+1} = F_i \check{P}_i F_i^T + B_i Q B_i^T \\ \check{P}_0 = \hat{P}_k \end{cases} \quad (3-44)$$

式中 \hat{P}_k 为激光雷达第 k 帧的后验协方差。

3.3.3 IESKF 观测方程

对于基于 EKF 的滤波器来说, 状态更新主要是对观测方程 y 、观测矩阵 H 以及增益矩阵 K 进行计算。观测方程 y 如式(3-45)所示:

$$y_{k+1} = f(x_{k+1}^k) - H_{k+1} \delta x_{k+1} + n_k \quad (3-45)$$

式中 $f(\bullet)$ 表示观测函数, x_{k+1}^k 表示从第 $k+1$ 帧第 k 帧到位姿变换的局部状态变量, n_k 表示第 k 帧激光雷达的观测噪声。

观测函数由观测匹配残差构成, 即由 3.2.2 节所得到的边缘特征点云与平面特征点云的残差构成如式(3-46)所示。

$$f(x_{k+1}^k) = \begin{cases} \frac{\left| (\tilde{X}_{(k+1,i)}^{L_e} - X_{(k+1,i)}^{L_e}) \times (\tilde{X}_{(k+1,i)}^{L_e} - X_{(k,l)}^{L_e}) \right|}{\left| X_{(k,j)}^{L_e} - X_{(k,l)}^{L_e} \right|} \\ \frac{\left| (\tilde{X}_{(k+1,i)}^{L_s} - X_{(k,j)}^{L_s}) \left((X_{(k,j)}^{L_s} - X_{(k,l)}^{L_s}) \times (X_{(k,j)}^{L_s} - X_{(k,m)}^{L_s}) \right) \right|}{\left| (X_{(k,j)}^{L_s} - X_{(k,l)}^{L_s}) \times (X_{(k,j)}^{L_s} - X_{(k,m)}^{L_s}) \right|} \end{cases} \quad (3-46)$$

为了方便计算使用扰动近似, 并且对观测函数 $f(\bullet)$ 进行偏导数链式法则求解, 进而求解出观测量的协方差矩阵 H_{k+1} , 如式(3-47)所示。

$$\begin{aligned} H_{k+1} &= \frac{\partial f}{\partial \tilde{X}_{(k+1,i)}^L} \cdot \frac{\partial \tilde{X}_{(k+1,i)}^L}{\partial \delta x} \\ &= \left\{ \frac{\left[(\tilde{X}_{(k+1,i)}^{L_e} - X_{(k+1,i)}^{L_e}) \times (\tilde{X}_{(k+1,i)}^{L_e} - X_{(k,l)}^{L_e}) \right]^T}{\left| (\tilde{X}_{(k+1,i)}^{L_e} - X_{(k+1,i)}^{L_e}) \times (\tilde{X}_{(k+1,i)}^{L_e} - X_{(k,l)}^{L_e}) \right|} \right\} \cdot \left[R_{k+1}^k \left(X_{(k+1,i)}^{L_e} \right)_x, I \right] \quad (3-47) \\ &\quad \left\{ \frac{\left[(X_{(k,j)}^{L_s} - X_{(k,l)}^{L_s}) \times (X_{(k,j)}^{L_s} - X_{(k,m)}^{L_s}) \right]^T}{\left| (X_{(k,j)}^{L_s} - X_{(k,l)}^{L_s}) \times (X_{(k,j)}^{L_s} - X_{(k,m)}^{L_s}) \right|} \right\} \cdot \left[R_{k+1}^k \left(X_{(k+1,i)}^{L_s} \right)_x, I \right] \end{aligned}$$

在传统的迭代误差扩展卡尔曼滤波器的迭代过程中不会传递协方差矩阵, 但是使用如下公式可以在迭代过程中对协方差矩阵进行更新, 以提高系统的精度, 真实第 $k+1$ 帧的误差状态 δx_{k+1} 与第 α 次迭代计算出的误差状态 δx_{k+1}^α 关系如式(3-48)所示。

$$\delta x_{k+1} = x \ominus \check{x}_{k+1} = \check{x}_{k+1}^\alpha \oplus \delta x_{k+1}^\alpha \ominus \check{x}_{k+1} \quad (3-48)$$

对第 i 次迭代计算出状态 \check{x}_{k+1}^α 在 $\delta x_{k+1}^\alpha = 0$ 处进行一阶泰勒展开得到式(3-49):

$$\delta x_{k+1} = \check{x}_{k+1}^\alpha \ominus \check{x}_{k+1} + J_{k+1}^\alpha \delta x_{k+1}^\alpha \quad (3-49)$$

式中 J_{k+1}^α 是 δx_{k+1} 在 $\delta x_{k+1}^\alpha = 0$ 处的雅克比矩阵，如式(3-50)所示：

$$J_{k+1}^\alpha = \begin{bmatrix} A_{k+1} (\delta \theta_{k+1})^{-T} & 0_{3 \times 15} \\ 0_{15 \times 3} & I_{15 \times 15} \end{bmatrix} \quad (3-50)$$

则增益矩阵 K_{k+1} 如式(3-51)所示：

$$K_{k+1} = \check{P}_{k+1} H_{k+1}^T (H_{k+1} \check{P}_{k+1} H_{k+1}^T + L_{k+1})^{-1} \quad (3-51)$$

式中 L_k 为激光雷达测量数据的协方差矩阵。

则先验状态量迭代公式如式(3-52)所示。

$$\check{x}_{k+1}^{\alpha+1} \ominus \check{x}_{k+1}^\alpha = -K_{k+1} f_{k+1}^\alpha - (I - K_{k+1} H_{k+1}) (J_{k+1}^\alpha)^{-1} (\check{x}_{k+1}^\alpha \ominus \check{x}_{k+1}) \quad (3-52)$$

式(3-51)为传统的协方差更新矩阵， L_{k+1} 是 m 维的对角阵且只与激光雷达有关，独立于其他变量， m 为激光特征点云的数量。 H_{k+1} 为 $m*18$ 的矩阵， \check{P}_{k+1} 18 维的方阵，则 $H_{k+1} \check{P}_{k+1} H_{k+1}^T + L_{k+1}$ 是一个 m 维的方阵，而即使经过了体素滤波与特征提取，激光点云特征数量仍然较大， m 的数量级为千，对此规模的矩阵求逆较为困难。本文使用 FAST-LIO^[31] 中提出的增益矩阵 K_{k+1} 的求解公式，如式(3-53)所示：

$$K_{k+1} = (\check{P}_{k+1}^{-1} + H_{k+1}^T L_{k+1}^{-1} H_{k+1})^{-1} H_{k+1}^T L_{k+1}^{-1} \quad (3-53)$$

式中 $H_{k+1}^T L_{k+1}^{-1} H_{k+1}$ 为 18 维的方阵，则 $\check{P}_{k+1}^{-1} + H_{k+1}^T L_{k+1}^{-1} H_{k+1}$ 同为 18 维的方阵。因此，对方阵求逆计算较为简单。另外对 m 维对角阵 L_{k+1}^{-1} 只需根据对角阵求逆公式直接求得。

3.3.4 IESKF 协方差与状态更新

判断式(3-52)是否小于阈值 δ ，若大于阈值，则重新迭代更新直至小于阈值，输出状态估计量与后验协方差 \hat{P}_{k+1} ，如式(3-54)与(3-55)所示。

$$\hat{x}_{k+1} = \check{x}_k^{\alpha+1} \quad (3-54)$$

$$\hat{P}_{k+1} = (I - K_{k+1} H_{k+1}) \check{P}_{k+1} \quad (3-55)$$

本文所使用的 IESKF 激光惯性里程计算法流程图如图 3-11 所示，通过下图流程则可以完成 IESKF 激光惯性里程计的构建。

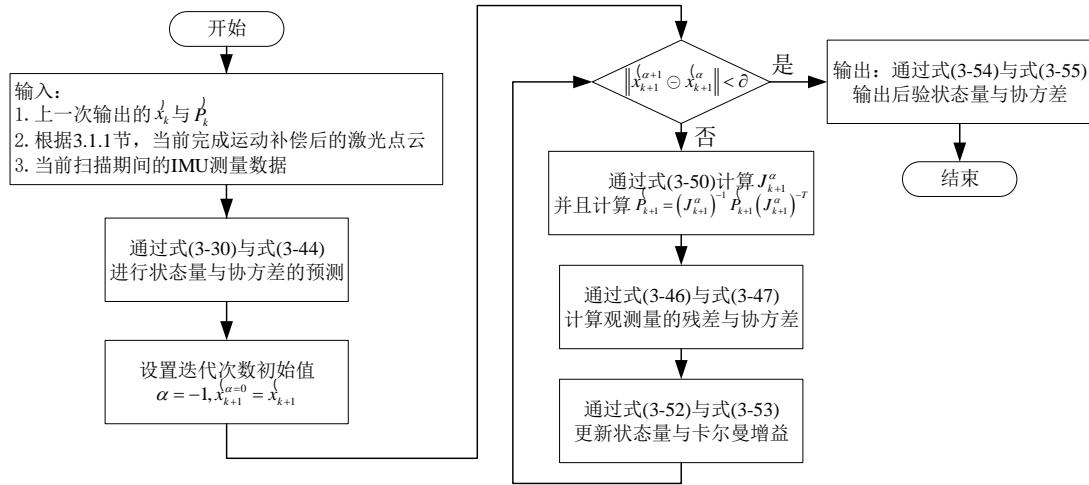


图 3-11 IESKF 算法流程图

Fig. 3-11 IESKF algorithm flow chart

3.4 基于 IESKF 的激光惯性里程计算法实验

本节在 KITTI 数据集上进行对 IMU 与激光雷达里程计进行 IESKF 滤波，得到的轨迹与传统的 NDT 前端里程计通过 EVO 轨迹评估工具进行对比，EVO 评估工具与 KITTI 数据集将在第 5 章详细介绍，IESKF 前端里程计与 NDT 前端里程计与地面真值 Ground_Truth 的轨迹对比如图 3-12 至图 3-15 所示。

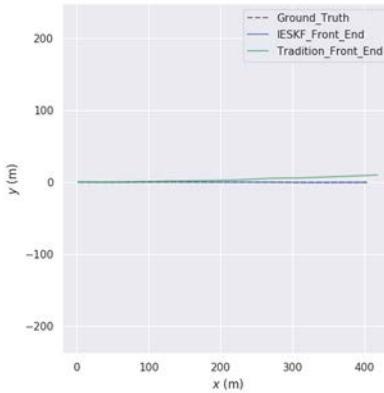


图 3-12 KITTI04 IESKF 前端、NDT 前端与真值轨迹对比

Fig. 3-12 Comparison of KITTI04 IESKF front-end, NDT front-end, and true trajectory

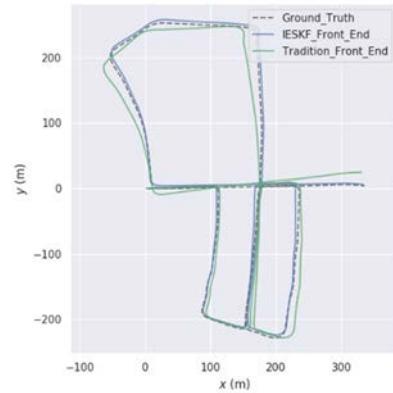


图 3-13 KITTI05 IESKF 前端、NDT 前端与真值轨迹对比

Fig. 3-13 Comparison of KITTI05 IESKF front-end, NDT front-end, and true trajectory

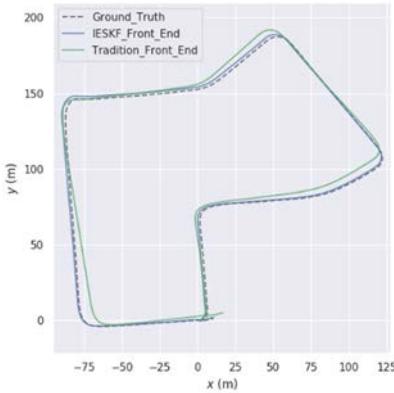


图 3-14 KITTI07 IESKF 前端、NDT 前端与真值轨迹对比

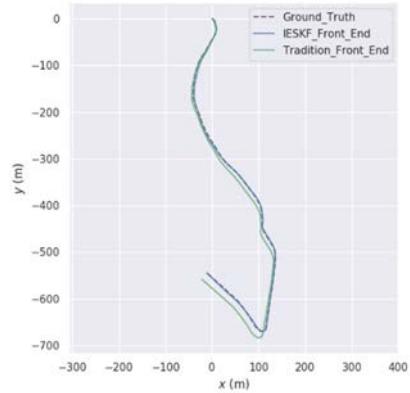


图 3-15 KITTI10 IESKF 前端、NDT 前端与真值轨迹对比

Fig. 3-14 Comparison of KITTI07 IESKF front-end, NDT front-end, and true trajectory

Fig. 3-15 Comparison of KITTI10 IESKF front-end, NDT front-end, and true trajectory

接着，我们对误差进一步的分析，并计算了绝对轨迹误差（Absolute Trajectory Error, ATE）与相对位姿误差（Relative Pose Error, RPE），用以评估 IESKF 前端里程计所生成的轨迹与地面真值轨迹之间的一致性，在第 5 章将具体介绍上诉两种评价指标的计算公式及其相关统计量。其中地面真值使用的是 RTK-GPS 数据。使用开源轨迹评估工具 EVO 对 IESKF、传统的前端里程计与地面真值进行定量比较。

表 3-1 与表 3-2 分别是 IESKF 在 KITTI 数据集上与真值之间的绝对轨迹误差与相对位姿误差，从表 3-1 中可以得出，使用的 KITTI 数据集中 4 个序列的绝对轨迹误差都小于传统的前端激光惯性里程计。在误差最大值方面分别提升了 76.39%、72.48%、81.76% 与 61.8%。在误差平均值方面分别提升了 88.02%、55.11%、68.67% 与 45.22%。在均方根误差方面分别提升了 87.41%、55.95%、69% 与 47.96%。在误差最小值方面分别提高了 89.11%、2.06%、68.23% 与 40.39%。

如表 3-2 所示，在相对位姿误差方面表现得没有传统的激光里程计好，但是也取得了良好的局部精度，在最常用的统计指标均方根误差（RMSE）方面只比传统的激光里程计分别多了 0.085096m、0.083944m、0.075717m 与 0.125565m。

表 3-1 在 KITTI 数据集上 IESKF 与真值之间的绝对轨迹误差

Table 3-1 Absolute trajectory error between IESKF and ground truth on KITTI

序列	方法	Max (m)	Mean (m)	RMSE (m)	Min (m)
04	IESKF-LIO	1.911283	0.603406	0.664415	0.101882
	Tradition-LIO	8.096145	5.038647	5.279502	0.935909
05	IESKF-LIO	12.451245	6.439309	6.965593	0.752063
	Tradition-LIO	45.247761	12.517785	15.515361	0.767579
07	IESKF-LIO	4.474666	2.855653	3.090045	0.413016
	Tradition-LIO	24.530172	7.769106	9.863228	1.300140
10	IESKF-LIO	5.499427	3.041901	3.257945	0.910566
	Tradition-LIO	14.394668	5.553384	6.260657	1.527527

表 3-2 在 KITTI 数据集上 IESKF 与真值之间的相对位姿误差

Table 3-2 Relative pose error between IESKF and ground truth on KITTI

序列	方法	Max (m)	Mean (m)	RMSE (m)	Min (m)
04	IESKF-LIO	1.526497	0.091896	0.187569	0.004584
	Tradition-LIO	0.223433	0.090565	0.102473	0.003998
05	IESKF-LIO	1.092161	0.082818	0.144966	0.000432
	Tradition-LIO	0.250936	0.054175	0.061022	0.002444
07	IESKF-LIO	1.187079	0.058630	0.122798	0.000712
	Tradition-LIO	0.139759	0.040197	0.047081	0.002619
10	IESKF-LIO	1.405954	0.100633	0.180268	0.002105
	Tradition-LIO	0.155423	0.047906	0.054703	0.002981

3.5 本章小结

本章从激光点云数据预处理开始介绍，分别介绍了激光点云的运动补偿与体素滤波算法及其作用。并且对传统的前端激光里程计进行了说明并且分析如何通过激光点云信息与优化问题进行结合，并且构建出一个机器人状态估计的问题。分析了激光里程计构建的过程，具体介绍了基于 IESKF 改进的激光惯性里程计，通过多传感器紧耦合的方式并且充分结合多线激光雷达与 IMU 的测量特性将其测量数据用于巡检机器人的状态估计。通过实验证明，本章算法相较于传统的前端里程计可以更准确地估计机器人的状态，具有较高的定位精度。

第4章 基于多传感器数据融合的定位与建图

在第3章中对激光惯性里程计进行 IESKF 改进，并且取得良好的精度，但是在全局精度上仍然可以提高，所以本文将使用因子图对 SLAM 系统进一步优化。本文搭建的 SLAM 算法如图 4-1 所示，在前端采用基于 IESKF 的激光惯性里程计并且加入后端因子图中，提升 SLAM 系统整体的定位精度。由于 IESKF 数据输出频率较快，将所有数据实时地加入后端因子图优化会导致需要的计算资源增多，所以本文采用关键帧与滑动窗口的方式对激光惯性里程计进行位姿稀疏化，以减小后端因子图的计算量。并且在因子图中加入 IMU 预积分因子、回环检测因子与 GPS 因子。经过因子图优化处理后，可获得优化后的激光雷达关键帧的全局位姿，将关键帧通过全局位姿信息转换到世界坐标系下，拼接关键帧点云地图就能获得全局地图。此外，本文提出一种基于 GPS 状态与置信度进行 GPS 数据的筛选策略，防止在后端因子图加入异常的 GPS 因子。

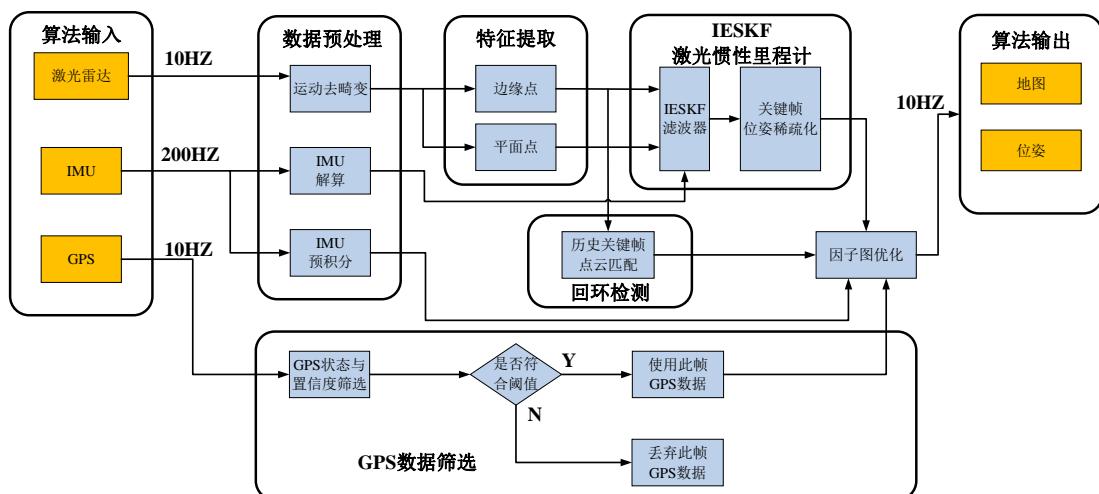


图 4-1 结合 IESKF 与因子图的 SLAM 框架

Fig. 4-1 SLAM framework combining IESKF and factor graph

4.1 激光惯性里程计因子

在因子图优化的过程中，IMU 数据与激光点云数据之间存在多余的信息，若将每一帧的 IESKF 里程计数据加入到后端因子图优化中，对定位精度的提升较小且占据大量计算资源，反而会影响系统的定位精度。因此在因子图优化中

使用关键帧与滑动窗口策略，降低后端因子图优化时使用的计算资源。关键帧需要选取在一段时间内具有代表性激光惯性里程计数据，同时可以降低数据量。而在滑动窗口中则只需要优化所有在窗口中的关键帧数据而舍弃普通的数据帧。

本文的普通帧为 IESKF 观测到激光点云数据时的激光帧，关键帧为当 IESKF 激光惯性里程计估计的位置变换超过 1m 或者姿态变换超过 5°，同时将相邻关键帧构造局部地图。根据当前关键帧的位姿，提取 i 个距离最近关键帧，构造相邻关键帧集合 $\{F_{i-k}, \dots, F_k\}$ ，并且将相邻关键帧集合对应的位姿将相邻关键帧集合转换至当前关键帧 F_k 坐标系下，转换完成后的相邻关键帧点云合并为一张局部地图。当后续新的关键帧点云加入至局部地图中时，删除局部地图中距离较远的关键帧点云。为了更准确地获得两帧关键帧的位姿变换关系，本文采用 ICP 配准算法，将当前关键帧与局部地图进行匹配，得到位姿变换关系，获得得第 k 帧与第 $k+1$ 帧激光雷达关键帧之间的残差方程，如式(4-1)所示：

$$\begin{aligned} r_L = & \begin{bmatrix} \Delta t - R_k^T(t_{k+1} - t_k) \\ \log(\Delta R^T R_k^T R_{k+1}) \end{bmatrix} \end{aligned} \quad (4-1)$$

式中 R 与 t 分别表示姿态旋转量与平移量， ΔR 与 Δt 分别表示第 k 帧与第 $k+1$ 帧的姿态旋转变换量与平移量。

基于滑动窗口的因子图，如图 4-2 所示，其后端优化通过增量平滑算法，在因子图中每次加入新的观测时不必重新计算整个因子图的雅克比矩阵。在动态大场景下本文采用固定步长的滑动窗口，从而保证在后端因子图优化时进一步减少所使用的计算资源。

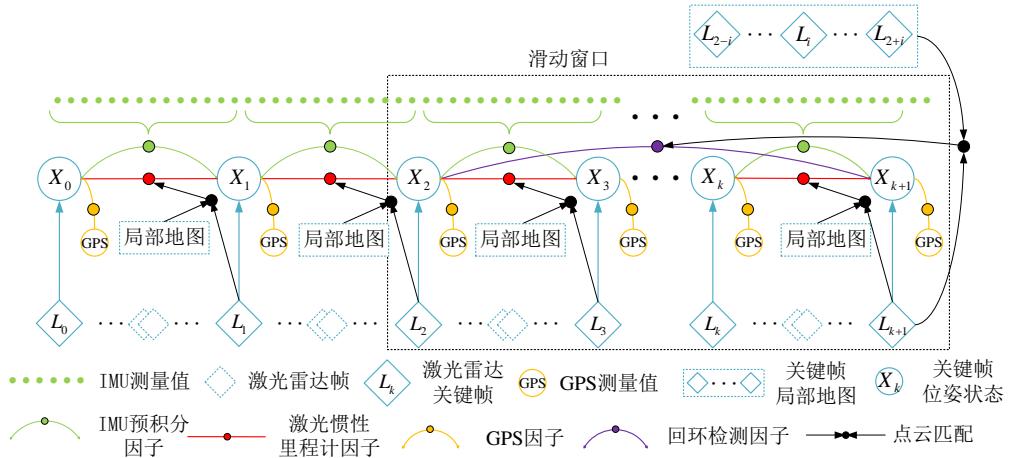


图 4-2 滑动窗口因子图模型

Fig. 4-2 Sliding window factor graph model

4.2 IMU 预积分因子

IMU 数据输出频率远高于其他传感器的数据输出频率，以多线激光雷达为例，一般 IMU 数据输出频率为激光雷达的 5~50 倍之间，所以在 IMU 数据在相邻两帧激光雷达数据输出间隔中，持续输出了 5~50 帧数据。

在使用以激光雷达为主的 SLAM 系统中，IMU 预积分可以将这 5~50 帧数据进行整合从而将两帧激光雷达数据之间 IMU 相对观测量转化为因子图中节点间的约束边。

4.2.1 IMU 离散积分模型

对式(2-6)与(2-7)所示的 IMU 测量模型中的加速度与角速度进行积分得到 IMU 的运动位姿，如式(4-2)所示：

$$\begin{cases} R(t + \Delta t) = R(t) \text{Exp} \left(\int_t^{t+\Delta t} \omega(\tau) d\tau \right) \\ v(t + \Delta t) = v(t) + \int_t^{t+\Delta t} a(\tau) d\tau \\ p(t + \Delta t) = p(t) + \int_t^{t+\Delta t} v(\tau) d\tau + \iint_t^{t+\Delta t} a(\tau) d\tau^2 \end{cases} \quad (4-2)$$

式中 R 、 v 、 p 、 Δt 分别表示 IMU 的姿态变换矩阵、速度、位置平移与采样时间。

对式(4-2)进行离散化且将式(2-6)与式(2-7)带入得到式(4-3)。

$$\begin{cases} R(t + \Delta t) = R(t) \text{Exp} \left((\hat{\omega}_t - b_t^\omega - n_t^\omega) \Delta t \right) \\ v(t + \Delta t) = v(t) + g \Delta t + R_w^R (\hat{a}_t - b_t^a - n_t^a) \Delta t \\ p(t + \Delta t) = p(t) + v(t) \Delta t + 0.5 g \Delta t^2 + 0.5 R_w^R (\hat{a}_t - b_t^a - n_t^a) \Delta t^2 \end{cases} \quad (4-3)$$

4.2.2 IMU 预积分模型

在 IMU 与激光雷达数据已经完成时空同步的前提下，假设在 2 个激光雷达关键帧 i 、 j 之间的 IMU 测量帧为 k ，则对激光雷达关键帧 i 、 j 之间的 IMU 数据积分如式(4-4)所示：

$$\left\{ \begin{array}{l} R_j = R_i \prod_{k=i}^{j-1} \text{Exp}\left(\left(\hat{\omega}_t - b_t^\omega - n_t^\omega\right) \Delta t\right) \\ v_j = v_i + g \Delta t_{ij} + \sum_{k=i}^{j-1} R_k \left(\hat{a}_t - b_t^a - n_t^a\right) \Delta t \\ p_j = p_i + \sum_{k=i}^{j-1} \left[v_k \Delta t + 0.5 g \Delta t^2 + 0.5 R_k \left(\hat{a}_t - b_t^a - n_t^a\right) \Delta t^2 \right] \end{array} \right. \quad (4-4)$$

式中 $\Delta t_{ij} = \sum_{k=i}^{j-1} \Delta t$ 。通过式(4-4)，可以在两个关键帧之间对 IMU 进行运动估计，

但是一旦在优化过程中运动状态发生改变，则需要重新对两个关键帧之间的运动状态进行估计。为了减少计算量，分离式(4-4)中的测量项，如式(4-5)所示：

$$\left\{ \begin{array}{l} \Delta R_{ij} \doteq R_i^T R_j \\ = \prod_{k=i}^{j-1} \text{Exp}\left(\left(\hat{\omega}_t - b_t^\omega - n_t^\omega\right) \Delta t\right) \\ \Delta v_{ij} \doteq R_i^T (v_j - v_i - g \Delta t_{ij}) \\ = \sum_{k=i}^{j-1} \Delta R_{ik} \left(\hat{a}_t - b_t^a - n_t^a\right) \Delta t \\ \Delta p_{ij} \doteq R_i^T (p_j - p_i - v_i \Delta t_{ij} - 0.5 g \Delta t_{ij}^2) \\ = \sum_{k=i}^{j-1} \left[v_{ik} \Delta t + 0.5 \Delta R_{ik} \left(\hat{a}_t - b_t^a - n_t^a\right) \Delta t^2 \right] \end{array} \right. \quad (4-5)$$

式中 $\Delta R_{ij} \doteq R_i^T R_j$ 表示将 ΔR_{ij} 定义为 $R_i^T R_j$ ，同理可得将 Δv_{ij} 定义为 $R_i^T (v_j - v_i - g \Delta t_{ij})$ 、将 Δp_{ij} 定义为 $R_i^T (p_j - p_i - v_i \Delta t_{ij} - 0.5 g \Delta t_{ij}^2)$ 。

此时噪声项为扰动，对式(4-5)进行扰动分析，可得 IMU 预积分的测量模型，如式(4-6)所示：

$$\left\{ \begin{array}{l} \Delta \tilde{R}_{ij} = R_i^T R_j \text{Exp}(\delta \theta_{ij}) \\ \Delta \tilde{v}_{ij} = R_i^T (v_j - v_i - g \Delta t_{ij}) + \delta v_{ij} \\ \Delta \tilde{p}_{ij} = R_i^T (p_j - p_i - v_i \Delta t_{ij} - 0.5 g \Delta t_{ij}^2) + \delta p_{ij} \end{array} \right. \quad (4-6)$$

式中 $\Delta \tilde{R}_{ij}$ 、 $\Delta \tilde{v}_{ij}$ 、 $\Delta \tilde{p}_{ij}$ 分别表示 IMU 预积分中姿态、速度与位置的测量值， $\delta \theta_{ij}$ 、 δv_{ij} 、 δp_{ij} 分别表示 IMU 预积分中的旋转、速度与平移噪声。

则 IMU 预积分的残差方程如式(4-7)所示。

$$\left\{ \begin{array}{l} r_{\Delta R_{ij}} = \text{Log} \left(\left(\Delta \tilde{R}_{ij} \left(\bar{b}_i^{\omega} \right) \text{Exp} \left(\frac{\partial \Delta \bar{R}_{ij}}{\partial b^{\omega}} \delta b^{\omega} \right) \right)^T R_i^T R_j \right) \\ r_{\Delta v_{ij}} = R_i^T \left(v_j - v_i - g \Delta t_{ij} \right) - \left[\Delta \tilde{v}_{ij} \left(b_i^{\omega}, b_i^a \right) + \frac{\partial \Delta \bar{v}_{ij}}{\partial b^{\omega}} \delta b^{\omega} + \frac{\partial \Delta \bar{v}_{ij}}{\partial b^a} \delta b^a \right] \\ r_{\Delta p_{ij}} = R_i^T \left(p_j - p_i - v_i \Delta t_{ij} - 0.5 g \Delta t_{ij}^2 \right) - \left[\Delta \tilde{p}_{ij} \left(\bar{b}_i^{\omega}, \bar{b}_i^a \right) + \frac{\partial \Delta \bar{p}_{ij}}{\partial b^{\omega}} \delta b^{\omega} + \frac{\partial \Delta \bar{p}_{ij}}{\partial b^a} \delta b^a \right] \end{array} \right. \quad (4-7)$$

式中 $r_{\Delta R_{ij}}$ 、 $r_{\Delta v_{ij}}$ 、 $r_{\Delta p_{ij}}$ 分别表示 IMU 预积分的姿态、速度与位置的残差项， \bar{b}_i^{ω} 、 \bar{b}_i^a 分别表示在两个关键帧之间的零偏，为了方便计算，在此处假设为常量， δb^{ω} 、 δb^a 分别表示陀螺仪与加速度计的零偏变化量， $\partial \Delta \bar{R}_{ij} / \partial b^{\omega}$ 、 $\partial \Delta \bar{v}_{ij} / \partial b^{\omega}$ 、 $\partial \Delta \bar{p}_{ij} / \partial b^{\omega}$ 表示 IMU 预积分项对于陀螺仪零偏的雅克比矩阵， $\partial \Delta \bar{v}_{ij} / \partial b^a$ 、 $\partial \Delta \bar{p}_{ij} / \partial b^a$ 表示 IMU 预积分项对于加速度计零偏的雅克比矩阵。

4.3 GPS 因子

使用激光雷达与 IMU 的 SLAM 系统依旧存在累计误差，当经过某些无回环的路线时，回环检测模块无法修正累计误差。所以本文使用 GPS 数据作为全局约束对累计误差进行修正。

4.3.1 GPS 坐标变换

在室外巡检机器人定位中使用的 GPS 数据主要涉及 3 个坐标系，分别为世界大地坐标系（World Geodetic System-1984, WGS-84）、地心地固坐标系（Earth-Centered & Earth-Fixed, ECEF）与东北天坐标（East North Up, ENU）。

WGS-84 坐标系是为 GPS 全球定位系统使用而建立的坐标系统。通过遍布世界的卫星观测站观测到的坐标建立，坐标系的原点为地球的质心、 $X_{\text{WGS-84}}$ 轴指向本初子午线与赤道线的交点， $Z_{\text{WGS-84}}$ 轴指向于北极点， $Y_{\text{WGS-84}}$ 轴垂直于 XOZ 平面，组成右手坐标系。ECEF 坐标系是一种以地心为原点的地固坐标系，是一种笛卡儿坐标系。WGS-84 坐标系与 ECEF 坐标系重合，如图 4-3 所示， λ 表示经度， φ 表示维度，高度 h 为基准地球椭球面法线到 GPS 用户点的距离， a 为地球的长半轴， b 为地球的短半轴。

为了获得 GPS 在东北天坐标系下的数据，需要将 WGS-84 坐标系下的经纬高数据转换到 ENU 坐标系下。

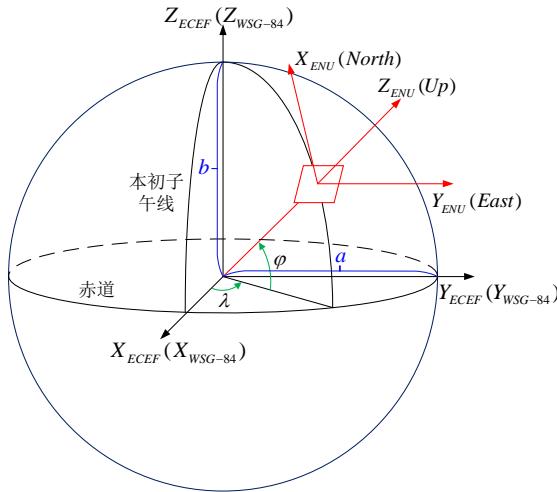


图 4-3 WSG-84、ECEF 与 ENU 坐标系
Fig. 4-3 WSG-84, ECEF and ENU coordinate systems

首先将 WGS-84 坐标系下的数据转换到 ECEF 坐标系下，如式(4-8)所示：

$$\begin{cases} x_{ECEF} = (N + h) \cos(\varphi) \cos(\lambda) \\ y_{ECEF} = (N + h) \cos(\varphi) \sin(\lambda) \\ z_{ECEF} = (N(1 - e^2) + h) \sin(\varphi) \end{cases} \quad (4-8)$$

式中 e 表示地球椭球第一偏心率，如式(4-9)所示：

$$e = \sqrt{a^2 - b^2} / a \quad (4-9)$$

式中 N 表示卯酉圈半径，如式(4-10)所示：

$$N = a / \sqrt{1 - e^2 \sin(\varphi)^2} \quad (4-10)$$

其次，将 ECEF 坐标系下的数据转换到 ENU 坐标系下，如式(4-11)所示：

$$\begin{bmatrix} x_{ENU} \\ y_{ENU} \\ z_{ENU} \end{bmatrix} = \begin{bmatrix} -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\lambda) \cdot \sin(\varphi) & -\sin(\lambda) \cdot \sin(\varphi) & \cos(\varphi) \\ \cos(\lambda) \cdot \cos(\varphi) & \sin(\lambda) \cdot \cos(\varphi) & \sin(\varphi) \end{bmatrix} \begin{bmatrix} x_{ECEF} - x_0 \\ y_{ECEF} - y_0 \\ z_{ECEF} - z_0 \end{bmatrix} \quad (4-11)$$

式中 (x_0, y_0, z_0) 为 ENU 坐标系原点在 ECEF 坐标系下的坐标，WGS-84 坐标系下的坐标为 (λ, φ, h) 。

4.3.2 基于 GPS 状态与置信度筛选策略的 GPS 因子

本文使用的 GPS 频率为 1Hz，在低速运行时数据量较为充足，采取基于关键帧加入 GPS 因子。由于 GPS 数据在高度城市化环境中会出现较大波动，为了避免在因子图中加入异常的 GPS 数据，本文使用 GPS 状态与置信度的方法筛选出可用的 GPS 数据，并将其作为 GPS 因子加入因子图中。

GPS 数据的状态一般固定解、浮点解、单点解与差分解。固定解或窄巷固定解精度为厘米级，浮点解精度波动较大，在厘米级与米级之间，差分解与单点解精度同为米级。为了在因子图中加入精度的更高的 GPS 因子，本文只采用状态为固定解与窄巷固定解的 GPS 数据。

首先，将巡检机器人移动至空旷场地中，使用手柄进行控制其移动，并且记录下当 GPS 状态为固定解时多组置信度，求其最大值并且设定为固定解置信度阈值。其次，将巡检机器人移动至楼宇之间，重复上述步骤记录下多组 GPS 状态的最大值作为窄巷固定解时的置信度，并且设定为窄巷固定解置信度阈值。最后，在 SLAM 系统运行时，每收到一帧 GPS 数据，就对其进行状态进行判断，根据不同的 GPS 状态选择相应的置信度阈值与当前的置信度进行比较，并且将小于置信度阈值的 GPS 数据加入因子图中。

本文所使用基于 GPS 状态与置信度筛选策略的流程图如图 4-4 所示。

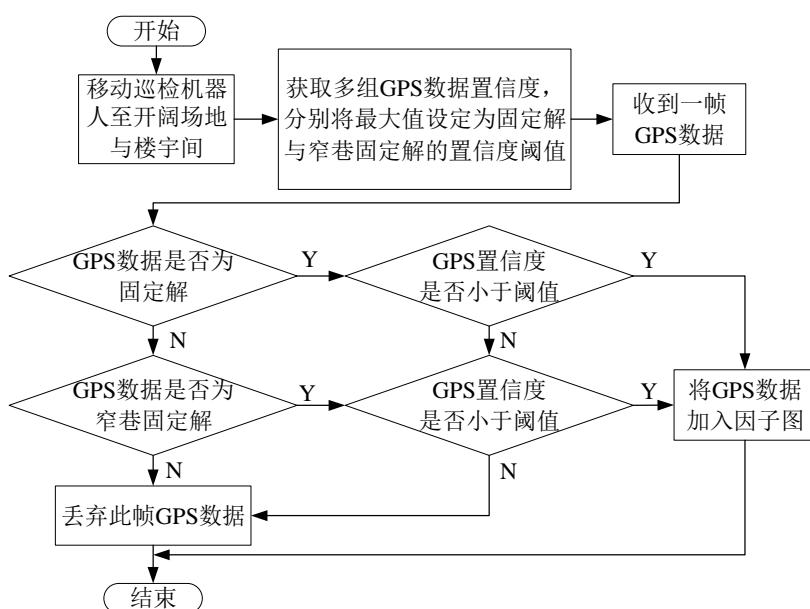


图 4-4 基于 GPS 状态与置信度筛选策略流程图

Fig. 4-4 Flow chart of screening strategies based on GPS status and confidence

此外，我们在实际场景中进行了测试，在智能科技园中对出现波动的 GPS 数据进行筛选，如图 4-5 与 4-6 所示，分别为筛选前的 GPS 轨迹与筛选后的 GPS 轨迹。可以从图中看到，GPS 数据筛选前波动较大，而筛选后波动减小。

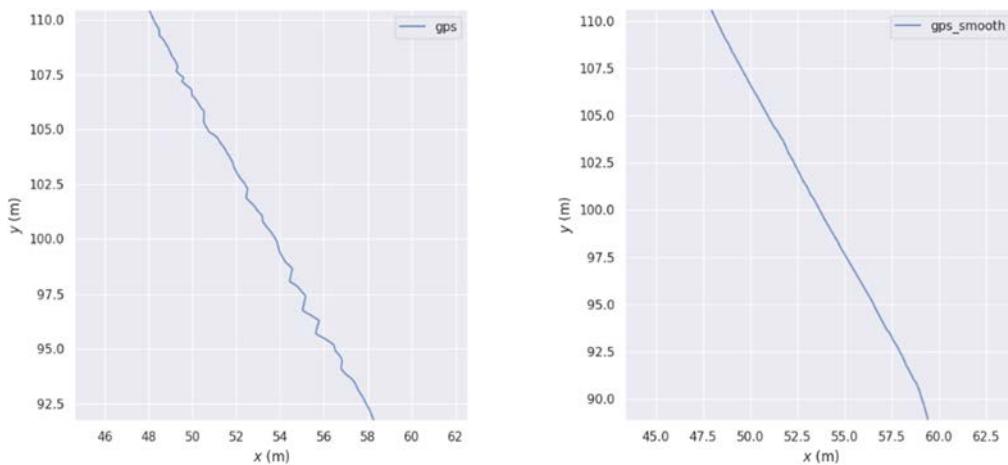


图 4-5 GPS 数据筛选前

Fig. 4-5 Before GPS data filtering

图 4-6 GPS 数据筛选后

Fig. 4-6 After GPS data filtering

在完成 GPS 坐标转换与 GPS 数据筛选后，在后端因子图优化中加入 GPS 因子。由于 GPS 数据在高度信息上存在较大的噪声，所以本文不使用 GPS 的 Z 轴数据。则 GPS 因子的残差如式(4-12)所示， t_k 表示 k 时刻状态点 x_k 的平移量， t_k^{GPS} 表示 k 时刻 GPS 的位置。

$$r_{GPS} = t_k^{GPS} - t_k \quad (4-12)$$

4.4 回环因子

在 SLAM 中，回环检测是重要且独立的模块，用于检测机器人在移动过程中是否经过了之前已经访问过的地点。它通过比较当前传感器数据和先前保存的地图数据来确定机器人是否回到了之前的位置，如果回到了，则需要更新地图和机器人的位置估计。

回环检测的主要目的是解决 SLAM 中的漂移问题。SLAM 算法通常使用增量式方法，即根据机器人传感器数据逐步构建地图，并同时估计机器人的运动和位置。这种方法容易受到测量误差的影响，导致位置估计漂移，从而导致地图不准确。回环检测可以检测这种漂移，并利用已经访问过的地点的信息来校正机器人的位置估计和地图。

通常，回环检测需要在大型环境中使用，因为只有在机器人移动的距离足够长时，才能产生明显的回环效应。如图 4-7 所示，为回环检测示意图。其中每个蓝色圆为激光雷达关键帧的位姿，黑线为激光惯性里程计与 IMU 预积分约束，箭头所指向的机器人运动方向，红色圆为检测到回环的关键帧位姿状态节点，图中为 X_1 与 X_8 为回环节点。并且使用回环约束通过因子图优化对节点位姿进行修正，得到精度较高的位姿状态与轨迹。

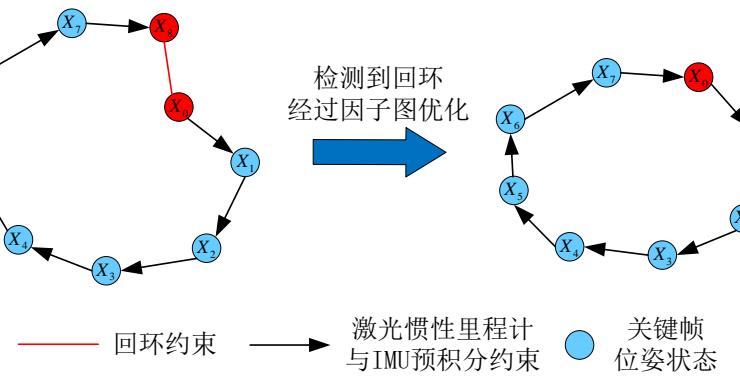


图 4-7 回环检测示意图

Fig. 4-7 Schematic diagram of loop closure detection

本文采用基于关键帧的欧式距离回环检测，首先将激光点云转换至世界坐标系下，根据当前关键帧的位置，设定一个距离为 d 的关键帧搜索范围，将范围内的关键帧位置根据设定的时间间隔再次进行筛选，找到距离相近且时间检测较长的历史关键帧。并且将找到的历史关键帧周围 25 帧范围内的特征点云所构成的局部特征点云地图与当前关键帧进行 ICP 点云匹配，求解出在世界坐标系下的相对位姿变换关系 T_{icp} ，则可以计算出当前关键帧点云与回环点云的相对位姿变换矩阵 $\Delta T_{loop,i}$ ，如式(4-13)所示：

$$\Delta T_{loop,i} = (T_{icp} T_{loop})^{-1} T_i \quad (4-13)$$

式中 T_i 与 T_{loop} 分别为当前点云与回环点的位姿。

则回环因子的残差公式如式(4-14)所示：

$$r_{loop} = \begin{bmatrix} \Delta t_{loop,i} - R_{loop}^T (t_i - t_{loop}) \\ \log(\Delta R_{loop,i}^T R_{loop}^T R_i) \end{bmatrix} \quad (4-14)$$

式中 $\Delta t_{loop,i}$ 、 $\Delta R_{loop,i}^T$ 与 t_i 、 R_i 分别表示 $\Delta T_{loop,i}$ 与 T_i 的平移量与旋转量。

4.5 基于因子图的后端优化

虽然第3章采用的IESKF紧耦合激光惯性里程计取得了较高的局部精度，但是在全局精度上还有待提高，所以本文使用因子图的表达方式，构建多因子融合定位的框架，如图4-2所示，用以提高SLAM系统的全局无偏估计的一致性。本章在4.1至4.4节中分别介绍了激光惯性里程计因子、IMU预积分因子、GPS因子与回环因子。本节将对上述因子使用因子图进行后端优化。

因子图是一种概率图模型，在多传感器融合的SLAM全局优化中是将系统的状态量与观测量通过构建因子图模型关联数据，然后基于贝叶斯理论实验全局估计，则一个广义的因子图最大后验估计 X^{MAP} 如式(4-15)所示：

$$\begin{aligned} X^{MAP} &= \arg \max_X (P(X)) \\ &= \arg \max_X \prod_i f_i(x_i) \end{aligned} \quad (4-15)$$

式中 X 为所有带估计状态量， $P(X)$ 为当前帧所有状态的联合概率分布函数， x_i 与 f_i 分别表示第*i*个状态量节点与因子节点。

本文所构建的因子图模型如式(4-16)所示：

$$G = (X, F, E) \quad (4-16)$$

式中 X 表示所有变量节点的集合，如图4-2中蓝色圆为各个关键帧的状态变量。 F 表示为所有因子节点的集合，如图4-2中各个颜色的实心原点。 E 表示变量节点与因子节点的连接边集合，如图中所有蓝色圆与实心点的连接线。

假定各个因子模型都符合高斯分布，则式(4-15)等价于求解一个非线性最小二乘问题，如式(4-17)所示：

$$X^{MAP} = \arg \min_X \sum_i \|h_i(X_i) - z_i\|_{\Sigma_i}^2 \quad (4-17)$$

式中 $h_i(X_i)$ 表示第*i*个状态量各个传感器的观测函数， z_i 表示*i*个状态量各个传感器的测量值， Σ_i 表示各个传感器的测量方差。

为了减小计算量，只使用滑动窗口中的各个状态量节点、因子节点与连接边，在一个滑动窗口中有*k*个状态量，则状态量的集合为 $X = \{X_1, \dots, X_k\}$ ，观测量的集合为 $Z = \{Z_1, \dots, Z_k\}$ 。在滑动窗口初始时刻因子图的贝叶斯估计如式(4-20)所示：

$$f(X_{0:k} | Z_{0:k}) = \prod_{i=1}^k \frac{f(Z_i | X_i) f(X_i | X_{i-1})}{f(Z_i | Z_{1:i-1})} f(X_0) \quad (4-18)$$

式中 $f(X_i | X_{i-1})$ 表示每一关键帧状态量的先验概率密度， $f(X_{0,k} | Z_{0,k})$ 表示结合观测量的后验概率密度。

由于式(4-18)右侧与似然概率密度成 $f(Z_i | X_i)$ 与先验概率密度 $f(X_i | X_{i-1})$ 正比，则得到式(4-19)：

$$\begin{aligned} \frac{f(Z_i | X_i) f(X_i | X_{i-1})}{f(Z_i | Z_{1:i-1})} f(X_0) &\propto f(Z_i | X_i) f(X_i | X_{i-1}) \\ &= f(E_{Z,i}^{\text{type}_i}) f(E_{X,i}) \end{aligned} \quad (4-19)$$

式中 $f(E_{Z,i}^{\text{type}_i})$ 与 $f(E_{X,i})$ 分别表示各个传感器观测量与激光里程计观测量的概率密度。

根据最大后验概率发展，将使得后验概率最大的状态量作为估计量，则式(4-19)右侧改写为式(4-20)与(4-21)：

$$\begin{aligned} f(E_{Z,i}^{\text{type}_i} | \mu_Z^{\text{type}_i}, \Sigma_Z^{\text{type}_i}) \\ = \frac{1}{(2\pi)^{\frac{k}{2}} (\Sigma_Z^{\text{type}_i})^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (E_{Z,i}^{\text{type}_i} - \mu_Z^{\text{type}_i})^T \Sigma_Z^{\text{type}_i - 1} (E_{Z,i}^{\text{type}_i} - \mu_Z^{\text{type}_i}) \right\} \end{aligned} \quad (4-20)$$

$$f(E_{X,i} | \mu_X, \Sigma_X) = \frac{1}{(2\pi)^{\frac{k}{2}} (\Sigma_X)^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (E_{X,i} - \mu_X)^T \Sigma_X^{-1} (E_{X,i} - \mu_X) \right\} \quad (4-21)$$

式中 $\mu_Z^{\text{type}_i}$ 为传感器的测量噪声

所以，为了最大化式(4-20)与(4-21)，得到如式(4-22)所示的2个目标函数。

$$\begin{cases} J_{Z,i} = \arg \min_Z \left\{ (E_{Z,i}^{\text{type}_i} - \mu_Z^{\text{type}_i})^T \Sigma_Z^{\text{type}_i - 1} (E_{Z,i}^{\text{type}_i} - \mu_Z^{\text{type}_i}) \right\} \\ J_{X,i} = \arg \min_X \left\{ (E_{X,i} - \mu_X)^T \Sigma_X^{-1} (E_{X,i} - \mu_X) \right\} \end{cases} \quad (4-22)$$

根据高斯牛顿迭代法则进行求解，如式(4-23)所示：

$$\arg \min_{\Delta X} \| J(\hat{X}) \Delta X - r(\hat{X}) \|^2 \quad (4-23)$$

式中 $\Delta X = \{\Delta X_1, \dots, \Delta X_k\}$ ， $J(\hat{X})$ 表示滑动窗口中所有因子节点的雅克比矩阵，如式(4-24)所示：

$$J(\hat{X}) = \begin{bmatrix} \frac{\partial J_{X,2}}{\partial X_1} & \frac{\partial J_{X,2}}{\partial X_2} \\ \vdots & \ddots \\ \frac{\partial J_{Z,2}}{\partial X_2} \\ & \frac{\partial J_{X,k}}{\partial X_{k-1}} & \frac{\partial J_{X,k}}{\partial X_k} \\ & \frac{\partial J_{Z,k}}{\partial X_k} \end{bmatrix} \quad (4-24)$$

式中 $r(\hat{X}) = \{r_1, \dots, r_k\}$ 为当前关键帧估计状态量 \hat{X} 的观测残差。

4.6 本章小结

本章主要在第 3 章基于 IESKF 滤波的激光惯性里程计的基础上，使用因子图优化融合 IESKF 前端里程计、IMU、GPS 与回环检测模块，对全局位姿进行优化估计。首先介绍了本文 SLAM 系统的整体框架，并且介绍了基于关键帧的激光惯性里程计，在两个关键帧之间使用 IMU 预积分进行约束。其次为了提高全局一致性，加入 GPS 对位姿进行全局约束，同时加入回环检测模块对同一场景下的位姿进行矫正。最后基于因子图构建最小二乘问题，融合多个传感器，完成定位与建图的框架。

第5章 实验测试与评估

本章将对第4章提出的基于多传感器数据融合的SLAM算法在公开数据集KITTI与实际巡检机器人底盘采集的数据集上进行测试，SLAM算法流程如图4-1所示，分析其在不同场景中各项定位指标。对于定位鲁棒性，本章将模拟部分实际情况，在GPS进行不同时间的中断的情况下，验证融合算法对传感器失效的鲁棒性。最后，将对所提出SLAM算法与其他开源算法进行比较分析。

5.1 轨迹精度评估指标

为了评估所提出的SLAM系统所估计出的位姿与GPS真值位姿之间的误差，使用评估全局一致性的绝对轨迹误差（Absolute Trajectory Error, ATE）和评价局部一致性的相对位姿误差（Relative Pose Error, RPE）来评估SLAM系统的性能。

ATE是对两条轨迹进行全局一致性的评估，其中一条轨迹选用RTK-GPS数据作为地面真值（Ground Truth, GT），则可以直观的表示出所评估算法的全局精度。ATE的评价公式如式(5-1):

$$E_{ATE} = T p_k^{SLAM} - p_k^{GT} \quad (5-1)$$

式中 T 表示将SLAM系统的轨迹对齐于地面真值的变换矩阵， p_k^{SLAM} 与 p_k^{GT} 分别表示第 k 个SLAM系统与地面真值的位姿。

在ATE评价指标中，包括ATE最大值（Max）、最小值（Min）、平均值（Mean）、中位数（Median）、标准差（Std）与均方根误差（RMSE）进行统计，其中最常用的是RMSE，其计算公式如式(5-2)所示：

$$RMSE_{ATE} = \sqrt{\frac{1}{N} \sum_{k=1}^N \|E_{ATE}\|^2} \quad (5-2)$$

式中 N 为评估位姿的总数。

RPE是对两条轨迹进行局部一致性的评估，与ATE相同，将地面真值作为参照进行计算有一定时间间隔的两个绝对位姿之间的误差。如式(5-3)所示：

$$E_{RPE} = (R_k^{-1} R_{k+n}) (t_k^{-1} t_{k+n}) \quad (5-3)$$

式中 R_k 与 t_k 分别表示第 k 个SLAM系统与地面真值的位姿的旋转矩阵与平移向

量， n 表示间隔的帧数。

则均方根误差（RMSE）计算公式如式(5-4)所示。

$$RMSE_{RPE} = \sqrt{\frac{1}{N-n} \sum_{k=1}^{N-n} \|E_{RPE}\|^2} \quad (5-4)$$

基于上述 2 种评价指标，本文采用开源评价工具 EVO 进行位姿误差评价。

5.2 KITTI 数据集测试与评估

5.2.1 KITTI 数据集介绍

KITTI 数据集是自动驾驶领域广泛使用的公开数据集，基于大众帕萨特搭载多个传感器用于满足自动驾驶领域中评估各种算法，包括 2 个百万像素的灰度相机，2 个百万像素的彩色相机，一个 64 线机械式激光雷达与一套惯性 GPS 组合导航系统，如图 5-1 所示。并且完成硬件同步，同步后的传感器采集频率为 10Hz。此外 GPS-RTK 的定位信息作为地面真值，并且给出了未进行硬件同步的 raw data 数据集。在不同的场景下进行实时数据采集，包括城市、乡村、校园等场景。

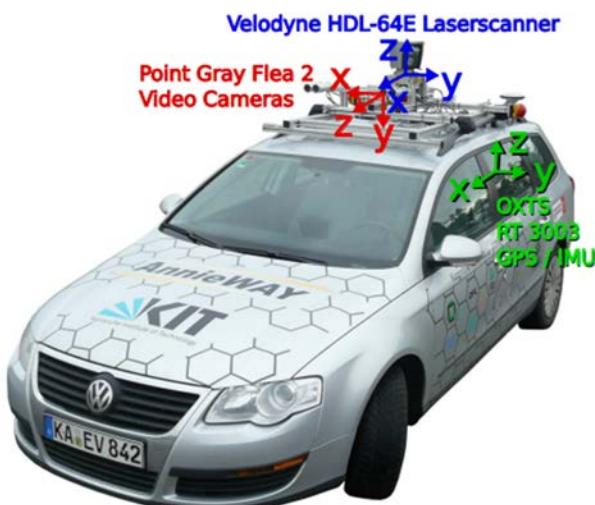


图 5-1 KITTI 数据集的采集平台

Fig. 5-1 KITTI data collection platform

本文主要使用 KITTI 数据集中 64 线激光雷达与惯性 GPS 组合导航系统，为了适配巡检机器人实验平台所搭载的 16 线激光雷达，本文只使用 64 线激光

束中的 16 束，将 64 线激光雷达当做 16 线激光雷达使用。KITTI 采集平台所搭载的传感器及其相关型号与精度如表 5-1 所示。

表 5-1 KITTI 数据集所使用的传感器参数

Table 5-1 Sensor parameters used in the KITTI dataset

传感器类型	传感器型号	传感器频率	传感器精度
64 线激光雷达	Velodyne HDL-64E	10Hz	0.02m
惯性 GPS 导航系统	OXTS RT3003	100Hz	0.02m/0.1°

5.2.2 KITTI 数据集测试与评估

本文将所提出的融合滤波与图优化的 SLAM 算法分别在 KITTI 数据集 04、05、07、10 序列进行实验，并且与开源的多传感器融合 SLAM 算法通过 EVO 轨迹评估工具所生成的轨迹与 ATE 统计指标进行对比分析。其中，滤波算法采用融合了多线激光雷达与 IMU 的 FAST-LIO，图优化算法采用融合多线激光雷达、IMU 与 GPS 的 LIO-SAM 进行对比。

同时，对 GPS 数据进行中断处理以此来评估在 GPS 突然失效时算法的鲁棒性。由于 FAST-LIO 并没有融入 GPS，则在 GPS 中断与否不影响其精度，故在 GPS 中断实验中，不在呈现其 ATE 统计指标。

(1) GPS 无中断实验

首先进行 GPS 无中断实验，在 04、05、07 与 10 序列上的轨迹由 EVO 轨迹评估工具得到，分别如图 5-2、图 5-3、图 5-4 与图 5-5 所示。

在图 5-2 中可以由 A、B 图看到本文的算法与 LIO-SAM 算法的轨迹与真值的轨迹相差不大，相较于 FAST-LIO 的轨迹，两者都更贴近真值的轨迹。

在图 5-3 中可以由 A、B 图看到本文的算法在大多数路段中的轨迹相较于上述两种算法的轨迹更贴近真值的轨迹。在 A 图中 $x \in [160, 180]$, $y \in [-20, -50]$ 的路段本文算法的轨迹与 LIO-SAM 的轨迹相较于真值的轨迹相差不大。在 B 图中除了 $x \in [170, 180]$, $y \in [120, 130]$ 的路段本文算法的轨迹与 LIO-SAM 相差不大之外，其余路段都优于上述两种算法。

在图 5-4 中 A 图所示，在 $x \in [-88, -87]$, $y \in [134, 135]$ 路段本文的算法优于 FAST-LIO 算法，但是略差于 LIO-SAM 算法。而其余路段本文算法的轨迹更贴近真值轨迹。在 B 图中本文的算法优于上诉两种开源算法。

在图 5-5 中 A、B 图所示，本文算法的轨迹优于上诉两种开源算法，其轨迹更贴近真值的轨迹。

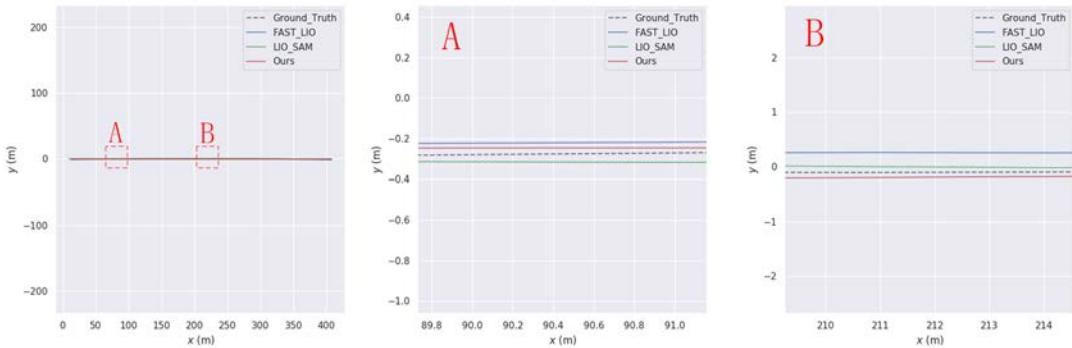


图 5-2 KITTI 数据集 04 序列轨迹对比图

Fig. 5-2 KITTI data set 04 sequence trajectory comparison chart

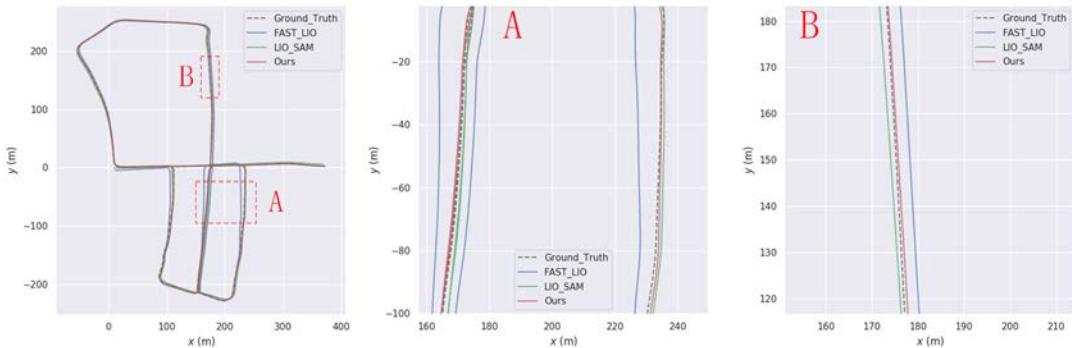


图 5-3 KITTI 数据集 05 序列轨迹对比图

Fig. 5-3 KITTI data set 05 sequence trajectory comparison chart

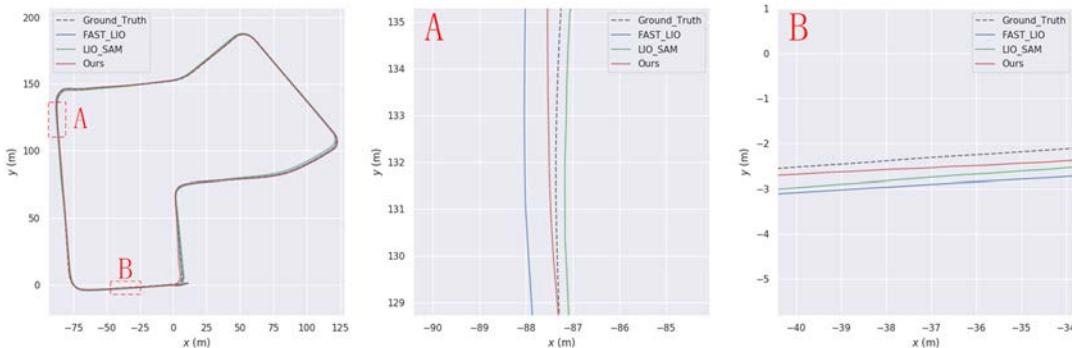


图 5-4 KITTI 数据集 07 序列轨迹对比图

Fig. 5-4 KITTI data set 07 sequence trajectory comparison chart

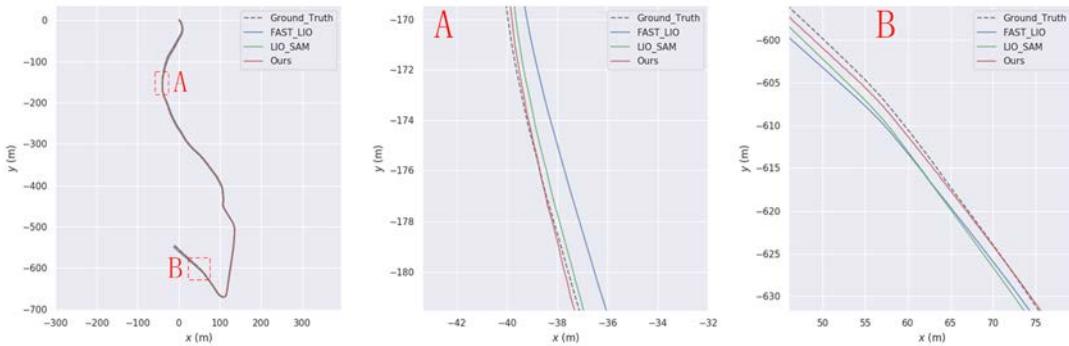


图 5-5 KITTI 数据集 10 序列轨迹对比图

Fig. 5-5 KITTI data set 10 sequence trajectory comparison chart

本文的算法与 FAST-LIO、LIO-SAM 的 ATE 统计数据如表 5-2 所示。

在 04 序列上本文的算法与 LIO-SAM 相比而言，在误差最大值（Max）与误差最小值（Min）方面 LIO-SAM 优于本算法分别比本文多了 0.154304 米与 0.037178 米，但是在误差平均值（Mean）、均方根误差（RMSE）方面略优于 LIO-SAM，分别提高了 19.97% 与 3.76%。此外本文算法与 FAST-LIO 相比，在误差最小值方面表现不如 FAST-LIO，在其他统计指标方面分别提升 71.72%、70% 与 71.69%。

在 05 序列上本文的算法在误差最大值方面表现不及 LIO-SAM，在误差最小值方面与 LIO-SAM 相差无几，而在误差平均值与均方根误差方面比 LIO-SAM 分别提升了 8.13% 与 6.43%。此外相比于 FAST-LIO，本文在各项统计指标皆优于 FAST-LIO，分别提升了 87.05%、89.82%、89.68% 与 92.73%。

在 07 序列上本文的算法在误差最大值方面与 LIO-SAM 相差无几，且优于 FAST-LIO，在其他统计指标上均优于 LIO-SAM 与 FAST-LIO，在误差平均值方面分别提升了 10.09% 与 76.64%，在均方根误差方面分别提升了 9.02% 与 75.98%，在误差最小值方面分别提升了 45.91% 与 82.48%。

在 10 序列上本文的算法在误差最小值方面优于 FAST-LIO 且提升了 71.81%，但是不及 LIO-SAM 且多了 0.122694 米。在误差最大值方面相较于 LIO-SAM 与 FAST-LIO 分别提升了 50.23% 与 52.33%，在误差平均值方面分别提升了 25.74% 与 60.91%，均方根误差值方面分别提升了 16.94% 与 59.44%。

表 5-2 KITTI 数据集上各个 SLAM 算法的 ATE 统计数据
Table 5-2 ATE statistics for each SLAM algorithm on the KITTI dataset

序列	方法	Max (m)	Mean (m)	RMSE (m)	Min (m)
04	Ours	0.550191	0.178998	0.195172	0.151904
	LIO-SAM	0.395887	0.223665	0.232748	0.114726
	FAST-LIO	1.945428	0.596698	0.689333	0.097367
05	Ours	1.572958	0.555240	0.623827	0.054046
	LIO-SAM	1.140164	0.604368	0.666729	0.050848
	FAST-LIO	12.143285	5.453287	6.046867	0.743559
07	Ours	1.333466	0.555258	0.605827	0.079949
	LIO-SAM	1.323349	0.617591	0.665901	0.147816
	FAST-LIO	4.248161	2.376953	2.522683	0.456325
10	Ours	2.751671	1.146823	1.293698	0.262802
	LIO-SAM	5.528663	1.544244	1.914991	0.140108
	FAST-LIO	5.772416	2.933960	3.189745	0.497090

(2) GPS 中断实验

由于 FAST-LIO 没有融入 GPS，则在本部分实验不进行比较。本文在 KITTI 数据集上，从第 20 秒开始对 GPS 数据进行中断处理，其中 KITTI04 序列中断约 10 秒，KITTI05 序列中断约 268 秒，KITTI07 序列中断约 95 秒，KITTI10 序列中断约 107 秒，用于模拟现实情况中 GPS 信号突然中断或行驶到 GPS 拒止环境下的情况。在 04、05、07 与 10 序列上的轨迹分别如图 5-6、图 5-7、图 5-8 与图 5-9 所示。可以由各图中的 A、B 图看到本文的算法在 GPS 中断情况下的轨迹更贴近真值，优于 LIO-SAM 的轨迹。

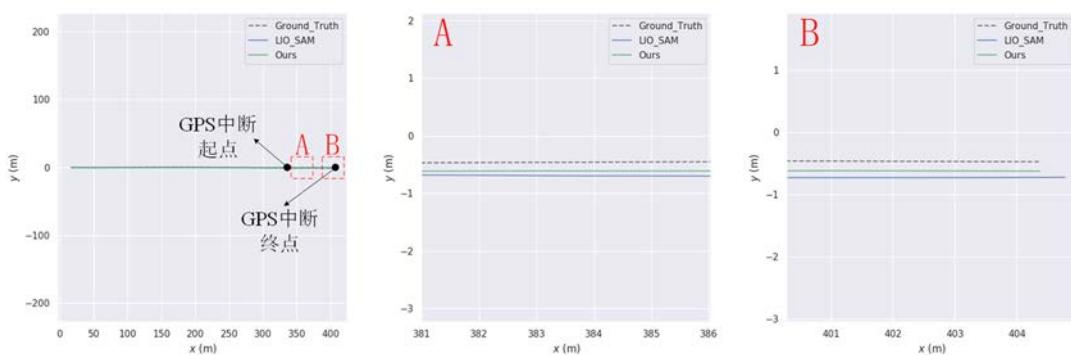


图 5-6 KITTI 数据集 04 序列轨迹对比图

Fig. 5-6 KITTI data set 04 sequence trajectory comparison chart

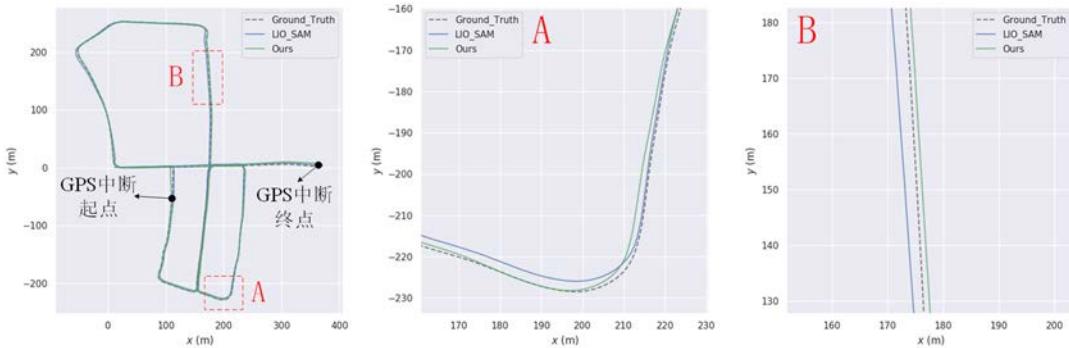


图 5-7 KITTI 数据集 05 序列轨迹对比图

Fig. 5-7 KITTI data set 05 sequence trajectory comparison chart

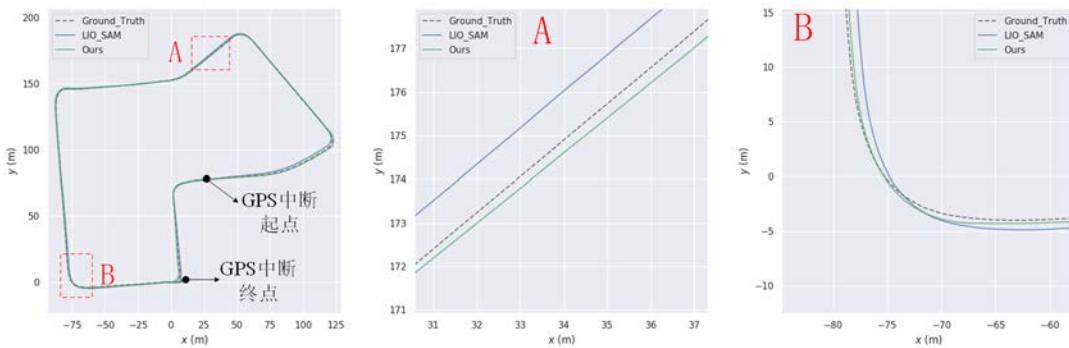


图 5-8 KITTI 数据集 07 序列轨迹对比图

Fig. 5-8 KITTI data set 07 sequence trajectory comparison chart

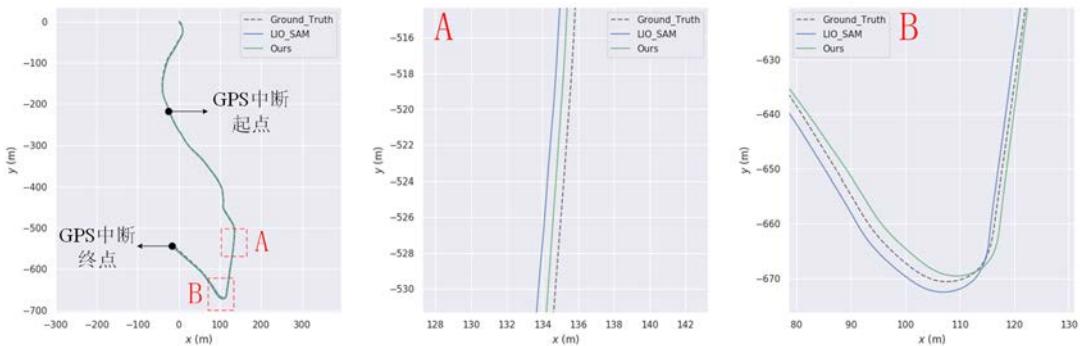


图 5-9 KITTI 数据集 10 序列轨迹对比图

Fig. 5-9 KITTI data set 10 sequence trajectory comparison chart

在 GPS 中断情况下，本文的算法与 FAST-LIO、LIO-SAM 的 ATE 统计数据如表 5-2 所示。

在 04 序列上，本文的算法与 LIO-SAM 相比而言，在误差最大值与误差最小值方面表现不及 LIO-SAM，分别多了 0.209414 米与 0.014831 米，在其余方面略优于 LIO-SAM，分别提升 17.1% 与 10.28%。

在 05 序列上，本文的算法在误差最小值方面略不及 LIO-SAM，多了 0.126808 米。在其余的各个指标上均优于 LIO-SAM，分别提升了 40.92%、29.01% 与 28.2%。

在 07 序列上，本文的算法在误差最小值方面表现略有不及 LIO-SAM，在其余的统计指标上均优于 LIO-SAM 分别提升了 43.13%、13.85% 与 17.06%。

在 10 序列上，本文的算法在误差最小值方面表现略有不及 LIO-SAM，多了 0.209469 米，在其余指标方面均优于 LIO-SAM，分别提升了 50.96%、21.14% 与 29.75%。

表 5-2 KITTI 数据集上 GPS 中断时各个 SLAM 算法的 ATE 统计数据

Table 5-2 ATE statistics for each SLAM algorithm on the KITTI dataset

序列	方法	Max (m)	Mean (m)	RMSE (m)	Min (m)
04	Ours	0.614576	0.190359	0.214301	0.115945
	LIO-SAM	0.405162	0.229615	0.238865	0.101114
05	Ours	2.936086	1.242616	1.392347	0.177911
	LIO-SAM	4.969321	1.750427	1.939206	0.051103
07	Ours	1.272413	0.667964	0.723489	0.139385
	LIO-SAM	2.237385	0.775331	0.872355	0.135207
10	Ours	2.978839	1.422265	1.524391	0.331625
	LIO-SAM	6.073832	1.803528	2.169900	0.122156

5.3 巡检机器人实验平台测试与评估

5.3.1 巡检机器人实验平台介绍

在完成公开数据集 KITTI 的实验后，本文通过搭载 1.6GHz Intel i5-8250U 的 CPU、16G 内存条、Ubuntu18.04 操作系统与 ROS 的工控机，连接第 2 章所介绍的传感器与室外巡检机器人底盘进行搭建实验平台，所使用的传感器参数如表 5-3 所示，工控机裸板如图 5-10 所示。



图 5-10 工控机裸板

Fig. 5-10 Industrial computer bare board

表 5-3 巡检机器人实验平台所使用的传感器参数

Table 5-3 Sensor parameters used in the inspection robot experimental platform

传感器类型	传感器型号	传感器型号	传感器频率	传感器精度
16 线激光雷达	速腾聚创	RS-LiDAR-16	10Hz	0.02m
IMU	维特智能	HWT905	200Hz	0.061°
GPS 基站	北斗星通	NC502-D	1Hz	水平 1cm 垂直 3m
GPS 移动站	北斗星通	NC507-S	1Hz	水平 1cm 垂直 3m

5.3.2 巡检机器人实验测试与评估

该实验是为了评估我们的算法在户外环境下应用于移动自主机器人在 GPS 数据出现中断时，移动自主机器人的定位精度。在智能科技园区内进行低速大范围数据采集，行驶平均速度约为 2.5m/s，在接近 $280\text{m} \times 280\text{m}$ 的正方形园区内进行导航建图，完成约 7.84 万平方米的地图构建，本文算法所建的点云地图在卫星图的对比如图 5-10 所示。由于卫星图与地面建筑存在角度偏差，所以在卫星图中高楼建筑略有倾斜，而点云地图中的高楼为垂直方式呈现，所以在卫星图以高楼建筑物的地基或者底楼层与所构建点云地图进行对比，可以看到点云地图与卫星图的场景基本吻合。



图 5-11 本文算法构建的点云地图与卫星图对比

Fig. 5-11 The comparison between the point cloud map constructed by the algorithm in this dissertation and the satellite image

接着，本文将所提出的融合滤波与图优化的 SLAM 算法在智能科技园区内与开源的多传感器融合 SLAM 算法 FAST-LIO 与 LIO-SAM 通过 evo 轨迹评估工具所生成的轨迹与 ATE 统计指标进行对比分析。同时，对 GPS 数据进行中断处理以此来评估在 GPS 突然失效时算法的鲁棒性。与 5.2.2 节相同，在 GPS 中断实验中，不呈现 FAST-LIO 的 ATE 统计指标。

(1) GPS 无中断实验

首先进行 GPS 无中断实验，在智能科技园区内的轨迹如图 5-12 所示，本文算法的轨迹最贴近真值的轨迹，LIO-SAM 在某些路段的轨迹与本文算法相差无几，而在某些路段轨迹精度不如本文的算法。例如在图 5-12 的 A 图中， $x \in [-90, -40], y \in [250, 280]$ 的路段中本文的算法最接近真值的轨迹，其次分别是 LIO-SAM 与 FAST-LIO。而在 A 图其余路段，LIO-SAM 与本文算法的轨迹极为接近，同时都优于 FAST-LIO。在 B 图中， $x \in [-160, -120], y \in [-40, 20]$ 与 $x \in [-140, -80], y \in [50, 80]$ 的路段，本文算法与 LIO-SAM 的轨迹几乎重合且接近真值轨迹，但是其余路段则是本文的算法优于 LIO-SAM 与 FAST-LIO。

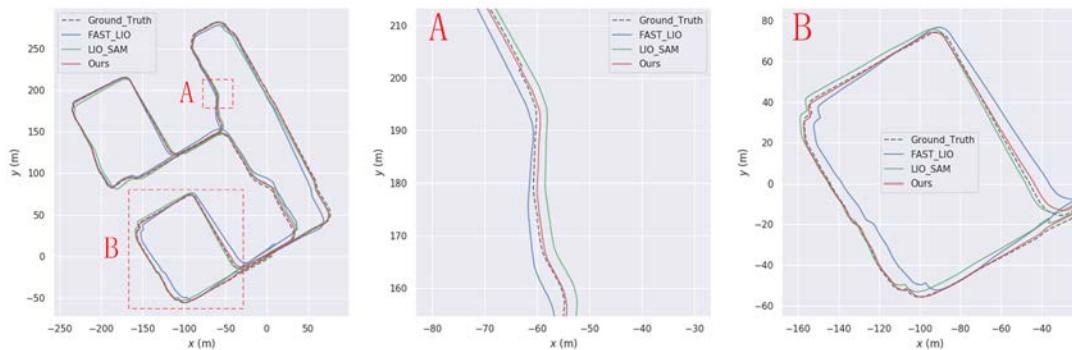


图 5-12 智能科技园轨迹对比图

Fig. 5-12 Smart Technology Park track comparison chart

在智能科技园区内，本文的算法与 FAST-LIO、LIO-SAM 的 ATE 统计数据如表 5-4 所示。本文的算法相较于 LIO-SAM 与 FAST-LIO，在各个统计指标上都优于上述两个开源算法。由于 FAST-LIO 并没有融入 GPS 数据，除了误差最小以外的各个指标上表现是最差的。在误差最大值方面分别提升了 51.03% 与 64.64%，在误差平均值方面分别提升了 21.68% 与 55.5%，在均方根误差方面分别提升了 24.98% 与 58.58%，在误差最小值方面分别提升了 71.30% 与 68.92%。可见本文的算法精度较高。

表 5-4 智能科技园各个 SLAM 算法的 ATE 统计数据

Table 5-4 ATE statistics for each SLAM algorithm on Smart Technology Park

方法	Max (m)	Mean (m)	RMSE (m)	Min (m)
Ours	7.827892	3.801397	4.189983	0.245139
LIO-SAM	15.985649	4.853936	5.585050	0.854267
FAST-LIO	22.136492	8.540459	10.115659	0.788699

(2) GPS 中断实验

我们对 GPS 数据进行中断处理，从第 200 秒开始，进行中断 50 秒，100 秒与 200 秒。与 5.2.2 节相同，FAST-LIO 不参与评估。LIO-SAM 与本文算法的轨迹如图 5-13 所示。其中中断 50 秒时，A 图区域发生了偏移。中断 100 秒时，A 图与 B 图区域同时发生了偏移。中断 200 秒时，在 A 图、B 图与 C 图区域中同时发生了偏移。可以分别从 A 图、B 图与 C 图中看到，本文算法的轨迹相较于 LIO-SAM 的轨迹更接近真值，在 GPS 中断时本文算法的鲁棒性更好。

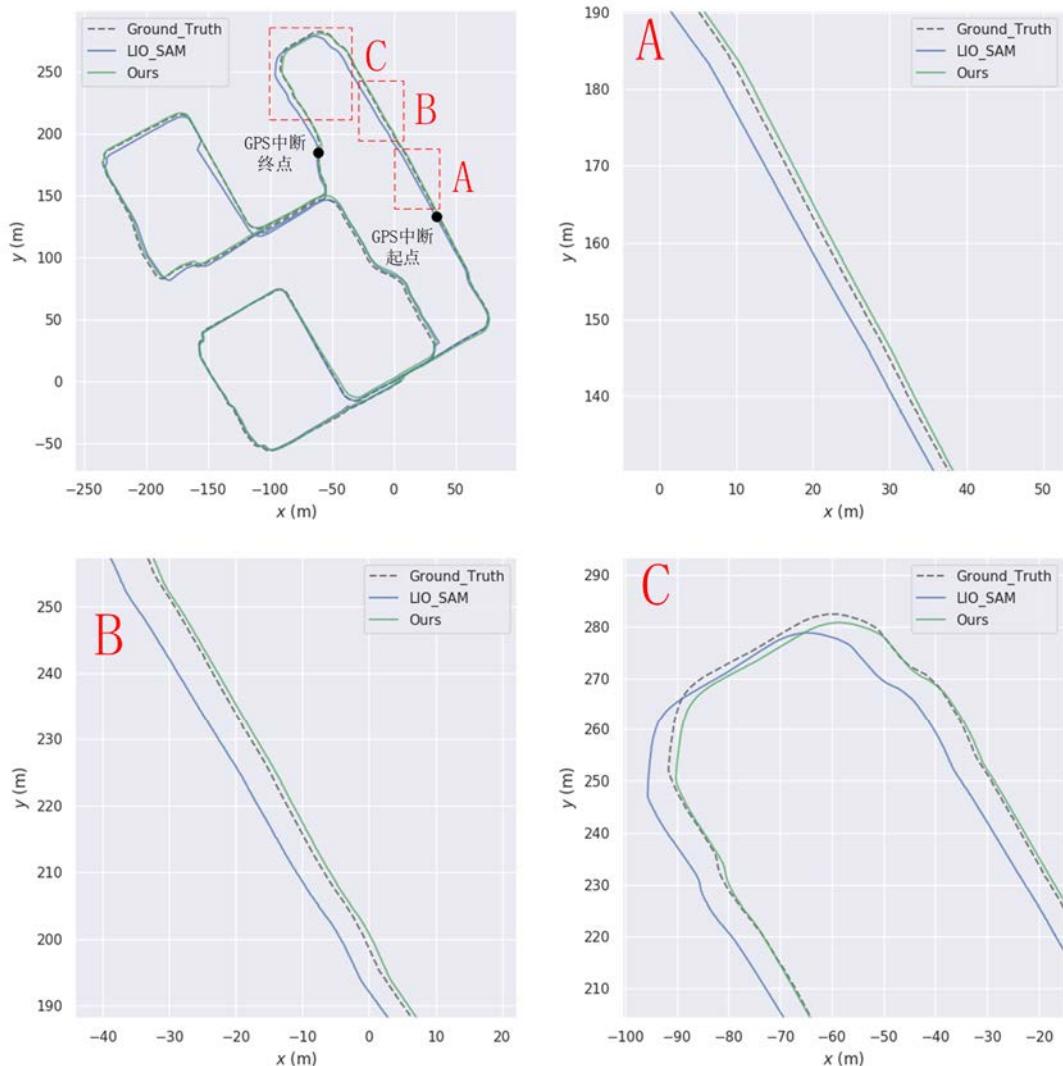


图 5-13 智能科技园轨迹对比图

Fig. 5-13 Smart Technology Park track comparison chart

在智能科技园区内，在 GPS 中断的情况下，本文的算法与 LIO-SAM 的 ATE 统计数据如表 5-5 所示。在 GPS 中断不同时间的情况下，误差最小值无变换，误差最大值变换较小，其原因是在整个智能科技园的轨迹中，所产生的误差最小值与误差最大值的位姿并不在 GPS 中断时所产生的轨迹上，所以在后续对 ATE 统计指标分析时，不再对误差最大值与误差最小值进行分析。

在 GPS 中断 50 秒的情况下，本文的算法与 LIO-SAM 相比而言，在误差平均值与均方根误差方面优于 LIO-SAM，分别提高了 19.61% 与 30.53%。在 GPS

中断 100 秒的情况下，本文的算法在误差平均值与均方根误差方面相较于 LIO-SAM 分别提高了 17.33% 与 29.11%。在 GPS 中断 200 秒的情况下，本文的算法同样优于 LIO-SAM，分别提高了 12.05% 与 23.14%。从提升率可以看到本文的算法在 GPS 数据突然中断 50 秒时可以保持相对较好的精度提升，在后续较长时间的 GPS 数据中断时本文的算法的提升率逐渐减小，但是轨迹整体的精度依旧高于 LIO-SAM。

表 5-5 智能科技园各个 SLAM 算法的 ATE 统计数据

Table 5-5 ATE statistics for each SLAM algorithm on Smart Technology Park

中断时间 (s)	方法	Max (m)	Mean (m)	RMSE (m)	Min (m)
50	Ours	7.955014	4.338442	4.700273	0.245139
	LIO-SAM	15.671642	5.396973	6.765650	0.854267
100	Ours	7.819960	4.595851	4.917642	0.245139
	LIO-SAM	15.863218	5.559095	6.937252	0.854267
200	Ours	7.927294	4.952018	5.397438	0.245139
	LIO-SAM	15.90683	5.630523	7.022866	0.854267

5.4 本章小结

本章通过公开数据集 KITTI 与巡检机器人实验平台，对本文所提出的多传感器融合 SLAM 系统进行了测试与评估。本章首先在 KITTI 数据集中，对 GPS 不中断与 GPS 中断的情况下进行测试评估。其次使用巡检机器人实验平台在智能科技园区内进行 GPS 不中断与 GPS 中断的情况下进行测试评估。实验结果表明本文提出的前端使用 IESKF 的激光惯性里程计，后端使用图优化算法融合激光惯性里程计因子、IMU 因子与回环因子的 SLAM 系统在 GPS 不中断与 GPS 中断的情况下皆优于只使用滤波或图优化算法的 SLAM 系统，同时说明本文的算法在 GPS 是否中断的情况下都具有更好的精度。

结 论

本文针对面向室外巡检机器人在科技园区室外环境中出现 GPS 中断情况下，以激光雷达为核心的多传感器融合的 SLAM 算法精度下降问题进行研究。提出了一种结合激光雷达、惯性测量单元与 GPS 的多传感器融合 SLAM 系统。首先采用“手眼标定”方法分别标定了激光雷达与 IMU、激光雷达与 GPS 之间的外参并且采用 Allan 方差法标定了 IMU 的内参。其次为了减少传统激光惯性里程计的累计误差，采用 IESKF 紧耦合激光雷达与 IMU，完成激光惯性里程计的构建。并且为了进一步提升机器人位姿估计的全局精度，使用因子图优化的方式，融合激光惯性里程计因子、IMU 预积分因子、GPS 因子与回环因子，构建了基于因子图优化的多传感器融合 SLAM 系统。最后基于开源数据集 KITTI 与巡检机器人实验平台进行测试与评估。本文主要工作总结如下：

(1) 定义了机器人底盘与各个传感器的坐标系，建立激光雷达、IMU 与 GPS 传感器模型并且获得其原始数据。基于“手眼标定”方法对激光雷达与 IMU、激光雷达与 GPS 之间坐标系的外参分别进行标定，其中 GPS 的数据为经过 RTK 解算得到的定位数据，并且通过 Allan 方差法对 IMU 内参进行标定。最后采用 GPS 中的 PPS 脉冲对激光雷达与 IMU 进行硬件时间同步，并且通过 ROS 操作系统完成软件时间同步。

(2) 针对传统激光 SLAM 系统前端里程计精度不足的问题，并且为了巡检机器人行驶到 GPS 拒止环境中出现 GPS 数据中断时有着更精确的定位。提出了一种使用 IESKF 紧耦合激光雷达与惯性测量单元的前端激光惯性里程计，提升在大范围场景中仅使用激光雷达与惯性测量单元进行位姿估计精度。首先对原始激光点云进行运动补偿与体素滤波，其次对处理后的激光点云进行特征提取，然后构建基于激光特征点云与 IMU 数据的迭代卡尔曼滤波器进行机器人的位姿估计。最后在公开数据集 KITTI 上与传统前端里程计进行轨迹精度的对比，传统前端里程计采用在 KITTI 上表现更好的 NDT 前端里程计。实验表明本文使用的 IESKF 前端里程计的精度优于传统前端里程计。

(3) 基于 IESKF 的激光惯性里程计在全局下的位姿轨迹精度有所欠缺，为了进一步提高算法的全局定位精度，使用因子图优化的方式融合激光雷达、IMU 与 GPS，在因子图中加入 IESKF 激光惯性里程计因子，为了减小图优化时计算负荷，采用关键帧与滑动窗口策略。为了对两个关键帧的位姿进行约束，根据 IMU 离散模型推导了在两个关键帧之间的 IMU 预积分模型，并且在因子

图中加入预积分因子。为了进一步减小累计误差，加入基于关键帧的欧式回环检测因子与 GPS 因子。并且在 GPS 数据出现波动的情况下，提出了一种基于 GPS 状态与置信度的筛选方式，防止加入异常的 GPS 数据，在室外出现 GPS 波动的路段中，可以有效地剔除异常的 GPS 数据。

(4) 为了验证本文提出的多传感器融合的定位与建图系统的精度与鲁棒性，在开源数据集 KITTI 与自建的巡检机器人平台上进行实验与评估。在 GPS 不中断与 GPS 中断两种情况下与基于滤波和基于图优化的 SLAM 算法进行轨迹与位姿的评估。结果表明本文提出的 SLAM 系统在 GPS 不中断与 GPS 中断的两种情况下都有着更高的精度与鲁棒性。

在本文的研究基础上，可以对下述方面进行研究：

(1) 随着激光雷达成本进一步地降低，在基于多传感器融合的 SLAM 系统中可以使用精度更高、频率更快的固态激光雷达代替本文使用的机械激光雷达，可以提升 SLAM 系统整体的精度。

(2) 根据不同的应用场景，可以将更多的传感器融入 SLAM 系统中，后续在室内场景中可以融入 UWB 定位技术与深度相机等传感器提升 SLAM 系统的精度。

参考文献

- [1] 黄山, 吴振升, 任志刚, 等. 电力智能巡检机器人研究综述[J]. 电测与仪表, 2020, 57(02): 26-38.
- [2] 张钟元. 室外物流配送机器人的多传感器融合同步定位与建图技术[D]. 上海: 上海交通大学, 2019: 1-2.
- [3] Li W, Cui X, Lu M. A Robust Graph Optimization Realization of Tightly Coupled GNSS/INS Integrated Navigation System for Urban Vehicles[J]. Tsinghua Science and Technology, 2018, 23(6): 724-732.
- [4] 赵磊. GPS/Galileo 组合模糊度固定精密单点定位方法研究[D]. 武汉: 武汉大学, 2017: 1-3.
- [5] Li X, Zhang W, Zhang K, et al. GPS Satellite Differential Code Bias Estimation with Current Eleven Low Earth Orbit Satellites[J]. Journal of Geodesy, 2021, 95(7): 1-18.
- [6] Pan Z, Chen H, Li S, et al. Clustermap Building and Relocalization in Urban Environments for Unmanned Vehicles[J]. Sensors, 2019, 19(19): 4252.
- [7] Chiang K W, Tsai G J, Li Y H, et al. Navigation Engine Design for Automated Driving Using INS/GNSS/3D LiDAR-SLAM and Integrity Assessment[J]. Remote Sensing, 2020, 12(10): 1564-1588.
- [8] 冯明驰, 刘景林, 李成南, 等. 一种多焦距动态立体视觉 SLAM[J]. 仪器仪表学报, 2021, 42(11): 200-209.
- [9] Montambault S, Pouliot N. The HQ LineROVer: Contributing to Innovation in Transmission Line Maintenance[C]. 2003 IEEE 10th International Conference on Transmission and Distribution Construction, Operation and Live-Line Maintenance, 2003. 2003 IEEE ESMO, Orlando, FL, USA, 2003: 33-40.
- [10] Toussaint K, Pouliot N, Montambault S. Transmission Line Maintenance Robots Capable of Crossing Obstacles: State-of-The-Art Review and Challenges Ahead[J]. Journal of Field Robotics, 2009, 26(5): 477-499.
- [11] Moyo T, Plessis F D . The Use of The Travelling Salesman Problem To Optimise Power Line Inspections[C]. 2013 6th Robotics and Mechatronics Conference (RobMech), Durban, South Africa, 2013: 99-104.
- [12] 林超, 戴昊, 薛志成, 等. 变电站智能巡检机器人的应用综述[J]. 自动化应用,

2018(12): 73-75.

- [13] Lu S, Qian Q, Zhang B, et al. Development of A Mobile Robot for Substation Equipment Inspection[J]. Automation of Electric Power Systems, 2006, 30(13): 94-98.
- [14] Guo R, Li B, Sun Y, et al. A Patrol Robot for Electric Power Substation[C]. 2009 International Conference on Mechatronics and Automation, Changchun, China, 2009: 55-59.
- [15] Wen W, Bai X, Kan Y C, et al. Tightly Coupled GNSS/INS Integration via Factor Graph and Aided by Fish-Eye Camera[J]. IEEE Transactions on Vehicular Technology, 2019, 68(11): 10651-10662.
- [16] Schmid K, Lutz P, Tomić T, et al. Autonomous Vision-Based Micro Air Vehicle for Indoor and Outdoor Navigation[J]. Journal of Field Robotics, 2014, 31(4): 537-570.
- [17] Qin T, Li P, Shen S. Vins-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.
- [18] Borges G A, Aldon M J. Line Extraction in 2D Range Images for Mobile Robotics[J]. Journal of Intelligent and Robotic Systems, 2004, 40(3): 267-297.
- [19] Leonard J J, Durrant-Whyte H F. Mobile Robot Localization by Tracking Geometric Beacons[J]. IEEE Transactions on Robotics and Automation, 1991, 7(3): 376-382.
- [20] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [21] Mur-Artal R, Tardós J D. Orb-slam2: An Open-Source Slam System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE transactions on robotics, 2017, 33(5): 1255-1262.
- [22] Pierzchała M, Giguère P, Astrup R. Mapping Forests Using An Unmanned Ground Vehicle with 3D LiDAR and Graph-SLAM[J]. Computers and Electronics in Agriculture, 2018, 145: 217-225.
- [23] Geneva P, Eckenhoff K, Yang Y, et al. Lips: Lidar-Inertial 3D Plane Slam[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018: 123-130.

- [24] Chen X, Milioto A, Palazzolo E, et al. Suma++: Efficient Lidar-Based Semantic Slam[C]. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019: 4530-4537.
- [25] Gao Y, Liu S, Atia M M, et al. INS/GPS/LiDAR Integrated Navigation System for Urban and Indoor Environments Using Hybrid Scan Matching Algorithm[J]. Sensors, 2015, 15(9): 23286-23302.
- [26] Shamsudin A U, Ohno K, Hamada R, et al. Consistent Map Building in Petrochemical Complexes for Firefighter Robots Using SLAM Based on GPS and LIDAR[J]. Robomech Journal, 2018, 5(1): 1-13.
- [27] Abdelaziz N, El-Rabbany A. An Integrated INS/LiDAR SLAM Navigation System for GNSS-Challenging Environments[J]. Sensors, 2022, 22(12): 4327.
- [28] Aboutaleb A, El-Wakeel A S, Elghamrawy H, et al. Lidar/RISS/GNSS Dynamic Integration for Land Vehicle Robust Positioning in Challenging GNSS Environments[J]. Remote Sensing, 2020, 12(14): 2323.
- [29] Zhao S, Fang Z, Li H L, et al. A Robust Laser-Inertial Odometry and Mapping Method for Large-Scale Highway Environments[C]. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019: 1285-1292.
- [30] Qin C, Ye H, Pranata C E, et al. LINS: A Lidar-Inertial State Estimator for Robust and Efficient Navigation[C]. 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020: 8899-8906.
- [31] Xu W, Zhang F. FAST-LIO: A Fast, Robust Lidar-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 3317-3324.
- [32] Xu W, Cai Y, He D, et al. FAST-LIO2: Fast Direct Lidar-Inertial Odometry[J]. IEEE Transactions on Robotics, 2022.
- [33] Bai C, Xiao T, Chen Y, et al. Faster-LIO: Lightweight Tightly Coupled Lidar-Inertial Odometry Using Parallel Sparse Incremental Voxels[J]. IEEE Robotics and Automation Letters, 2022, 7(2): 4861-4868.
- [34] 李文豪. 基于 ESKF 与图优化的轻量级多传感器融合 SLAM 系统[J]. 科学技术创新, 2021(08): 15-18.
- [35] 白崇岳, 王建军, 程霄霄, 等. 融合激光 SLAM 实现无人驾驶轮椅空间定位优化[J]. 激光与光电子学进展, 2022, 59(02): 485-493.

- [36] 张书亮, 谭向全, 吴清文. 基于多传感器融合技术的室内移动机器人定位研究[J]. 传感器与微系统, 2021, 40(08): 53-56.
- [37] 杨林, 马宏伟, 王岩. 煤矿井下移动机器人基于激光惯性的融合 SLAM 算法 [J/OL]. 煤炭学报: 1-12[2022-10-06]. DOI: 10.13225/j.cnki.jccs.2022.0506.
- [38] 王馨阁, 田宗强, 陈文涛, 等. 基于改进 RBPF 的激光 SLAM 算法研究[J]. 安徽工程大学学报, 2020, 35(06): 18-25.
- [39] 李秀智, 居鹤华. 一种基于粒子滤波的月球车同时定位与地图创建方法[J]. 宇航学报, 2009, 30(05): 1891-1895.
- [40] 王依人, 邓国庆, 刘勇, 等. 基于激光雷达传感器的 RBPF-SLAM 系统优化设计[J]. 传感器与微系统, 2017, 36(09): 77-80.
- [41] Kukko A, Kaijaluoto R, Kaartinen H, et al. Graph SLAM Correction for Single Scanner MLS Forest Data under Boreal Forest Canopy[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2017, 132: 199-209.
- [42] Hess W, Kohler D, Rapp H, et al. Real-Time Loop Closure in 2D LIDAR SLAM[C], 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016: 1271-1278.
- [43] Chang L, Niu X, Liu T. GNSS/IMU/ODO/LiDAR-SLAM Integrated Navigation System Using IMU/ODO Pre-Integration[J]. Sensors, 2020, 20(17): 4702.
- [44] Pierzchała M, Giguère P, Astrup R. Mapping Forests Using an Unmanned Ground Vehicle with 3D LiDAR and Graph-SLAM[J]. Computers and Electronics in Agriculture, 2018, 145: 217-225.
- [45] Zhang J, Singh S. Low-Drift and Real-Time Lidar Odometry and Mapping[J]. Autonomous Robots, 2017, 41(2): 401-416.
- [46] Geiger A, Lenz P, Stiller C, et al. Vision Meets Robotics: The KITTI Dataset[J]. The International Journal of Robotics Research, 2013, 32(11): 1231-1237.
- [47] Shan T, Englot B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018: 4758-4765.
- [48] Oelsch M, Karimi M, Steinbach E. R-LOAM: Improving LiDAR Odometry and Mapping with Point-to-Mesh Features of A Known 3D Reference Object[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 2068-2075.
- [49] Wang H, Wang C, Chen C L, et al. F-LOAM: Fast Lidar Odometry and

- Mapping[C]. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021: 4390-4396.
- [50] Shan T, Englot B, Meyers D, et al. LIO-SAM: Tightly-Coupled Lidar Inertial Odometry via Smoothing and Mapping[C]. 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), Las Vegas, NV, USA, 2020: 5135-5142.
- [51] Li T, Pei L, Xiang Y, et al. P3-LOAM: PPP/Lidar Loosely Coupled Slam with Accurate Covariance Estimation and Robust RAIM in Urban Canyon Environment[J]. IEEE Sensors Journal, 2020, 21(5): 6660-6671.
- [52] 张福斌, 王凯, 廖伟飞, 等. 激光雷达/MEMS IMU/里程计紧组合导航算法[J]. 仪器仪表学报, 2022, 43(07): 139-148.
- [53] 张钊, 张轶, 黄瑞. 带关键帧和可靠平面表示的激光定位算法[J]. 西北工业大学学报, 2020, 38(06): 1339-1344.
- [54] 付林, 郑佳楠, 何洪磊, 等. 面向大型室内场景的无人机三维激光雷达解耦 SLAM 方法[J]. 电子测量技术, 2022, 45(13): 96-103.
- [55] 牛国臣, 王瑜. 基于多约束因子图优化的无人车定位与建图方法[J]. 北京航空航天大学学报, 2021, 47(02): 306-314.
- [56] 李帅鑫, 李广云, 王力, 等. LiDAR/IMU 紧耦合的实时定位方法[J]. 自动化学报, 2021, 47(06): 1377-1389.
- [57] 孙喜亮, 关宏灿, 苏艳军, 等. 面向高精度城市测绘的激光紧耦合 SLAM 方法 [J]. 测绘学报, 2021, 50(11): 1585-1593.
- [58] 邱佳月, 赖际舟, 方玮, 等. 一种基于动态剔除和场景匹配的 Robust SLAM 方法[J]. 仪器仪表学报, 2022, 43(03): 249-257.
- [59] 刘振宇, 惠泽宇, 郭旭, 等. 基于滑动窗口优化的激光雷达惯性测量单元紧耦合同时定位与建图算法[J]. 科学技术与工程, 2022, 22(21): 9167-9175.
- [60] Kim H, Song S, Myung H. GP-ICP: Ground Plane ICP for Mobile Robots[J]. IEEE Access, 2019, 7: 76599-76610.