

中图分类号: xxx

学校代码: 10216

UDC: xxx

密级: 公开

专业学位硕士学位论文

(应用研究型)

动态场景下基于视觉 SLAM 的机器人室内语 义建图系统

硕 士 研 究 生: sch
导 师: csh 教授
副 导 师: wx 高级工程师
申 请 学 位: 电子信息硕士
学 科 专 业: 电子信息
所 属 学 院: 电气工程学院
答 辩 日 期: 2023 年 5 月
授 予 学 位 单 位: 燕山大学

燕山大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《动态场景下基于视觉 SLAM 的机器人室内语义建图系统》，是本人在导师指导下，在燕山大学攻读硕士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签字：

日期：2023 年 6 月 1 日

燕山大学硕士学位论文使用授权书

《动态场景下基于视觉 SLAM 的机器人室内语义建图系统》系本人在燕山大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归燕山大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解燕山大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅。本人授权燕山大学，可以采用影印、缩印或其它复制手段保存论文，可以公布论文的全部或部分内容。

保密口，在 年解密后适用本授权书。

本学位论文属于

不保密口。

(请在以上相应方框内打“√”)

作者签名：

日期：2023 年 6 月 1 日

导师签名：

日期：2023 年 6 月 1 日

摘要

同时定位与地图构建（SLAM）是智能移动机器人在未知环境下进行状态估计的基础能力之一。然而，大多数视觉 SLAM 系统都是在静态场景假设的基础上构建，因此在动态环境中状态估计的准确性和鲁棒性会严重下降。并且，很多系统构建的点云地图较为稀疏且缺少直观语义信息，所以机器人很难以人类认知水平来理解周围环境并执行导航、路径规划等任务。

首先，为解决上述问题并进行验证，设计了一个移动机器人实验平台。硬件主要由 FreeRTOS 系统控制的麦克纳姆轮式移动平台、英伟达 Jetson 计算平台和 RGB-D 相机构成。软件主要是本文提出的语义建图系统，称之为 SG-SLAM。SG-SLAM 系统在基础框架 ORB-SLAM2 上新增了两个并行线程：目标检测线程和语义建图线程。平台软硬件系统之间的通信、可视化等功能由机器人操作系统（ROS）负责。平台通过合理的架构设计和工程创新，经实验证明系统的实时性能良好。

其次，针对动态场景下视觉 SLAM 系统跟踪性能下降的问题，提出了融合几何信息、语义信息的动态特征剔除算法。在 SG-SLAM 的跟踪线程中使用极线约束原理获取图像帧间的几何信息后，再根据采用 NCNN 框架、MobileNetV3-SSD 神经网络的目标检测线程获取的语义信息自动调整算法的经验阈值。所提算法在 TUM、Bonn RGB-D 两个数据集以及现实场景中的多组实验结果表明，在高动态环境下 SG-SLAM 的准确性和鲁棒性相比原框架分别至少提升 93% 和 90% 以上，是目前最准确和稳定的系统之一。

最后，针对稀疏点云地图缺少语义信息、难以执行导航任务等问题，提出了一种语义地图构建方法。根据关键帧深度图像和相机位姿生成 3D 点云后，使用体素滤波、统计离群点去除和欧式聚类分割方法对其进行处理。处理后的 3D 点云发布至 ROS 构建八叉树地图，同时又与 2D 语义信息联合计算获取 3D 语义对象。然后，利用 Kuhn-Munkres 算法对新获取的对象与语义对象数据库进行数据关联、更新。在公共数据集和现实场景中的实验表明，SG-SLAM 可以有效构建语义对象度量、八叉树和三维点云地图，这赋予了机器人执行非合作目标寻位、导航等任务的能力。

关键词：视觉 SLAM；动态场景；语义建图；极线约束；目标检测

目 录

摘 要	I
Abstract.....	II
第 1 章 绪 论	1
1.1 课题研究背景及意义	1
1.2 国内外研究现状	2
1.2.1 视觉 SLAM 研究进展	2
1.2.2 动态场景下视觉 SLAM 研究现状	4
1.2.3 语义建图研究现状	5
1.3 本文课题来源及研究内容	6
第 2 章 移动机器人系统设计	8
2.1 引言	8
2.2 硬件平台设计	9
2.2.1 麦克纳姆轮式移动平台	9
2.2.2 NVIDIA Jetson 计算平台	10
2.2.3 RGB-D 相机	11
2.3 ROS 机器人操作系统	12
2.4 视觉 SLAM 系统	12
2.4.1 基本原理	12
2.4.2 视觉里程计	17
2.4.3 后端优化	18
2.4.4 闭环检测	18
2.5 软件系统设计	19
2.5.1 ORB-SLAM2 框架分析	19
2.5.2 SG-SLAM 框架分析	20
2.6 本章小结	21
第 3 章 动态场景下的鲁棒视觉 SLAM	22
3.1 引言	22
3.2 动态场景处理算法原理	22
3.2.1 几何信息获取	22
3.2.2 语义信息获取	25
3.2.3 动态特征剔除融合算法	26
3.3 实验验证及分析	28

3.3.1 公共数据集与评价指标介绍	28
3.3.2 与 ORB-SLAM2 性能对比	30
3.3.3 消融实验	36
3.3.4 与近期同类先进工作对比	38
3.3.5 现实场景动态特征剔除效果	40
3.4 本章小结	42
第 4 章 多类型直观感知语义地图构建	43
4.1 引言	43
4.2 语义地图构建算法原理	43
4.2.1 2D 语义目标检测	43
4.2.2 3D 点云滤波和分割	44
4.2.3 语义对象度量地图构建算法	46
4.2.4 算法数据关联原理	49
4.2.5 八叉树地图构建原理	51
4.2.6 全局三维点云重建原理	52
4.2.7 消除动态因素对建图的影响	53
4.3 实验验证及分析	53
4.3.1 公共数据集介绍	53
4.3.2 语义对象度量地图构建	54
4.3.3 全局八叉树地图构建	55
4.3.5 三维点云重建地图构建	56
4.3.6 消除动态因素对比实验	57
4.3.7 现实场景语义地图构建	58
4.3.8 系统实时性分析	59
4.4 本章小结	60
结 论	62
参考文献	64
攻读硕士学位期间承担的科研任务与主要成果	69
致 谢	70

第1章 绪论

1.1 课题研究背景及意义

国家近年来在能源开发、航空航天、海洋勘探等高端装备制造领域发展迅速，重大智能制造装备的需求逐渐增加。习近平总书记提出，发展数字经济是把握新一轮科技革命和产业变革新机遇的战略选择^[1]。随着人工智能技术的兴起、数字经济的飞速发展，机器人在各个行业加速渗透，其作为数字经济时代最具标志性的工具，正在深刻改变着人类的生产和生活方式^[2]。从助力消防安全的宇树四足机器狗（如图 1-1）到解决农业生产难题的大疆农业无人机（如图 1-2），机器人如今不仅在消防、农业等领域大放异彩，在医疗、化工、交通等各行各业中同样占据了极为重要的位置。

超大型装备的智能加工制造技术创新能力急需提高，而机器人化智能制造方向则具备广阔的发展前景。在传统的大型复杂装备加工场景中，绝大多数生产制造工具根据工艺要求放置在生产流水线两侧的固定位置，即“铁打的机床，流水的工件”。这种制造模式在小型、简单的工件加工时非常有效，但是一旦涉及非标准化的大型复杂构件加工场景，便会面临作业精度低、效率低、一致性差以及严重依赖人工作业等诸多问题。因此需要一种新的生产制造模式，即“流水的机器，铁打的工件”。

机器人化智能制造为大型复杂构件的加工制造提供了新方案、新思路，多机器人协同合作、多传感器融合优化，促进了大型复杂构件制造模式的变革。机器人对未来制造业产业升级具有至关重要的作用，但当前阶段机器人智能加工技术仍面临极为严峻的挑战。随着制造模式发生转变，机器人的工作环境从具有可预知工件、



图 1-1 宇树四足机器狗



图 1-2 大疆农业无人机

固定位置、简单的结构化环境转化为具有非合作目标、动态场景、复杂的非结构化环境。因此，机器人如何在动态复杂的非结构化环境中准确进行自主定位与对目标寻位等问题便是需要解决的关键挑战之一。

实际上，这个挑战可以被简单的概括为 SLAM（Simultaneous Localization and Mapping，同时定位与地图构建）。由于激光雷达昂贵的成本以及获取信息较为单调等问题，相机视觉方案则成为 SLAM 领域令人青睐的选择。近年来，机器人多方面性能虽然有了长足的进步，但绝大多数机器人仍然无法执行诸如“去客厅的饮水机接一杯水然后送到卧室”这样在人类眼中很自然的简单指令。究其原因，想要使机器人能够理解并成功执行此类与非合作目标感知与捕获有关的高级任务，抛开其所必需的优异动力学控制性能外，以下与视觉 SLAM 相关的两点内容同样十分重要：

一是准确的状态估计。机器人学，本质上研究的是世界上运动物体的问题。尽管每种机器人的功能各异，然而在实际应用中，他们往往会面对一些共同的问题——状态估计和控制^[3]。控制确实十分重要，但机器人进行智能行为的首要前提是先估计自身与环境的状态。而且，准确的状态可以为后续的语义建图提供良好的位姿。因此，如何设计一个在动态场景下足够鲁棒、准确的视觉 SLAM 系统非常关键。

二是直观的语义交互。绝大多数视觉 SLAM 系统致力于构建一个关于机器人环境全局一致的度量地图^[4]。机器人虽然可以轻易的使用该度量地图进行自身位姿估计，但却很难将该地图同人类的意图、指令关联起来，甚至一些稀疏度量地图在人类直观上难以理解。只有机器人“知道”需要进行任务操作的目标对象位于何处，才可以方便导航至目标对象附近。解决该问题的方法是进行同步语义级别的建图，此处语义建图的概念是：当机器人在环境中漫游时，识别和记录含有对人类而言有意义的信号与标志^[4]。

因此，能够在未知场景下实时根据传感器数据对自身进行定位、对周围环境进行语义建图的方法是机器人执行高级任务的必要条件，具有十分重要的研究意义。

1.2 国内外研究现状

1.2.1 视觉 SLAM 研究进展

SLAM 是移动机器人构建周围环境的地图，同时使用该地图来估计其自身位姿的过程^[5]。概率 SLAM 理念起源于 1986 年在旧金山举办的 IEEE 机器人和自动化会

议过程中的一系列讨论，然后开始逐步发展。然而一系列里程碑式的工作表明，机器人由于自身位姿的一般误差，在对地图中地标位置进行估计时，地标之间必然具有相关性，且记录相关性的状态向量会随着观测呈平方级指数增长^[6-8]。而且，这些工作没有研究地图的收敛性。所以，定位问题的计算复杂性和建图问题的收敛性问题使得 SLAM 发展受阻。而后，1995 年机器人研究国际研讨会上 Durrant-Whyte 首次提到了“SLAM”的缩略词和问题结构、收敛结果^[9]。Sebastian Thrun 在 1998 年的论文中将 SLAM 问题使用概率方法描述为一个最大似然估计问题^[10]。这个时期以后，SLAM 的研究工作如雨后春笋般涌现。Cesar Cadena 在 2016 年的 SLAM 发展综述论文中将这段时期称为 SLAM 发展中的古典时期^[11]（classical age）。

接下来，从 2004 到 2015 年左右称之为算法分析时期（algorithmic-analysis age）。Durrant-Whyte 在 2006 年将 SLAM 问题的概率公式、结构、解决方法和最新进展汇总到两篇综述论文^[5,12]后，这段时期主要研究 SLAM 的基本特性，比如可观性、收敛性和一致性问题^[13]。对于稀疏性（sparsity）的深入了解，成为了解决古典时期计算复杂性的关键。许多 SLAM 相关的开源程序库也在这一时期开发。虽然在基本理论和算法层面进行了深入的探索，但在此时期的 SLAM 在高速运动、动态场景等复杂环境下很容易跟踪失败，同时也缺少对环境的高级理解。因此，Cesar Cadena 断言自 2016 年之后，SLAM 进入了新的鲁棒感知时代（robust-perception age）。

视觉 SLAM 是使用单目、多目或深度相机等传感器进行数据输入的 SLAM 分支。从鲁棒感知时代以来，同样涌现出了许多优秀的视觉 SLAM 系统。现挑选具有代表性特点的算法，总结如表 1-1。

表 1-1 鲁棒感知时代的视觉 SLAM 算法

名称	支持传感器类型	估计方法	回环检测	时间
SVO ^[14]	单目、双目、鱼眼	Local BA	否	2016
ORB-SLAM2 ^[15]	单目、双目、RGB-D	Local BA	是	2017
MSCKF ^{[16][17]}	单目（IMU）、双目（IMU）	EKF	否	2018
Vins-mono ^[18]	单目（IMU）	Local BA	是	2018
VI-DSO ^[19]	单目（IMU）	Local BA	否	2018
BASALT ^[20]	双目（IMU）、鱼眼	Local BA	是	2019
DSM ^[21]	单目	Local BA	否	2020
ORB-SLAM3 ^[22]	单目（IMU）、双目（IMU）、鱼眼、RGB-D	Local BA	是	2021
Kimera-multi ^[23]	双目（IMU）	Local BA	是	2022

1.2.2 动态场景下视觉 SLAM 研究现状

目前大多数视觉 SLAM 系统都假设工作场景是静态和刚性的。当这些系统在动态场景中工作时，由于静态场景假设而产生的错误数据关联会严重削弱系统的准确性和稳定性。动态对象在场景中的存在使所有特征分为两类：静态特征和动态特征。如何检测和拒绝动态特征是解决问题的关键。以往的研究工作可分为三类：几何信息法、语义信息法和几何信息与语义信息相融合的方法。

几何信息法，其主要思想是假设只有静态特征才能满足算法的几何约束。Kundu 等人的工作提供了一个令人印象深刻的早期单目动态目标检测系统^[24]。系统创建两个几何约束，基于多视图几何^[25]来检测动态对象。其中最重要的是由基础矩阵定义的极线约束。其思想是，当前图像中的一个静态特征点必须位于与前一幅图像中相同的特征点相对应的极线上。如果一个特征点与相应极线的距离超过了设置的经验阈值，则认为它是动态的。该论文利用惯性传感器辅助计算了基础矩阵。在纯视觉系统中，基础矩阵可以通过基于 RANSAC^[26]的七点法来计算。Kundu 等人的算法具有速度快、场景泛化能力强等优点。然而，它缺乏对场景的高等级理解，因此该算法的经验阈值难以确定，结果精度不高。此外，一些工作使用直接法对场景进行运动检测^[27-30]。直接法速度更快，可以利用更多的图像信息。然而，由于基于灰度不变性假设，因此此类算法在复杂环境中的鲁棒性较差。

语义信息法，其主要思想是暴力剔除使用深度学习技术获得的先验动态区域内的特征。Zhang 等人采用 YOLO^[31]目标检测方法获取工作场景中动态对象的先验语义信息，然后基于语义信息剔除动态特征点，以提高系统的跟踪精度^[32]。然而，YOLO 通过边界检测框提取语义信息的方式会导致部分静态特征点被错误地视为异常值而被剔除。同样，Xiao 等人提出的 Dynamic-SLAM 也存在同样的暴力剔除边界检测框内所有特征点的问题^[33]。Liu 和 Miura 采用了一种语义分割的方法来检测动态对象，并剔除关键帧中的异常值^[34]。语义分割方法在一定程度上解决了目标检测方法边界检测框导致的动态特征错误识别问题。然而，语义分割方法严重依赖于神经网络的质量，因此很难同时满足速度和准确性的要求。

近年来，人们对几何信息和语义信息的融合方法进行了大量的研究。对于 RGB-D 相机，Bescos 等人使用 Mask R-CNN^[35]的语义分割结果结合多视图几何约束来检测动态对象和剔除异常值^[36]。Yu 等人使用基于光流的移动一致性检查方法检

测所有特征点^[37]，同时在一个独立的线程中使用 SegNet^[38]对图像进行语义分割。如果移动一致性检查方法检测到人类对象范围内超过一定百分比的动态点，则位于对象内部的所有特征点都将被直接剔除。Wu 等人使用 YOLO 检测场景中的先验动态对象，然后将其与 Depth-RANSAC 方法相结合，剔除动态对象范围内的特征点^[39]。Chang 等人通过 YOLACT 语义分割方法对图像帧进行分割，然后去除先验动态对象内部的异常值^[40]。然后，再引入几何约束进一步滤除未检出的动态特征点。

上述结合方法在提高精度方面取得了较好的效果。然而，这些方法的思想在很大程度上都依赖于语义信息，而在较小程度上依赖于几何信息。因此，它们或多或少都有以下缺点：

(1) 无法正确处理先验动态信息之外的动态特性。例如，椅子默认情况下是静态物体，但在被人移动时是动态的；移动的猫出现在场景中，而神经网络模型没有对猫类对象进行训练；检测算法的低召回率问题。

(2) 先验动态对象保持静止但仍然暴力地剔除其范围内的特征点，导致可用的关联数据减少，进而导致算法准确性、鲁棒性降低。例如，图像帧中一个坐着一动不动的人、睡着不动的狗等仍然被认为是动态的对象。

(3) 实时性能弱，难以满足后续任务需求。由于语义分割网络复杂或系统架构不合理等因素，系统的处理帧率较低，平均每帧处理时间较长。

针对上述问题，本文提出了一种结合几何信息和语义信息的可以有效剔除动态特征的算法。与目前大多数严重依赖深度学习技术的工作不同，本文所提算法主要依赖几何信息，然后引入深度学习技术获取的先验语义信息来辅助它。这种思维上的转变使本文所提算法避免了过于依赖深度学习的诸多缺点。

1.2.3 语义建图研究现状

目前许多视觉 SLAM 只提供一个满足移动机器人定位和导航基本功能的度量地图，如 ORB-SLAM2 构造的稀疏特征点云地图。如果移动机器人要以人类概念层面上感知周围环境，就有必要在此类度量地图中加入语义信息，形成语义度量地图。语义度量地图可以帮助机器人按照人类的规则行事，执行高级任务，并在概念层面上与人类进行交流。

在早期的研究中，Mozos 等人使用隐马尔可夫模型将度量地图划分为不同的功能位置^[41]（如房间、走廊和门道）。Nieto-Granda 等人在 ROS^[42]上部署了基于饶-黑

化粒子（Rao-Blackwellized）滤波技术的建图模块，并使用高斯模型将地图划分为标记的语义区域^[43]。随后，深度学习的兴起极大地促进了目标检测和语义分割算法的发展。Sünderhauf 等人使用 SSD^[44]检测每个彩色关键帧中的语义对象，然后使用自适应三维无监督分割方法为每个对象分配一个三维点云^[45]。这项工作通过 ICP-like 匹配分数的数据关联机制，来决定是否在语义地图中创建新的对象，还是将它们与现有的对象关联融合。Zhang 等人通过 RGB-D SLAM 系统中的 YOLO 目标检测模块和定位模块获取了工作场景的语义地图^[32]。

总之，许多工作只停止在使用 SLAM 来帮助进行语义建图，而没有充分利用所获得的语义信息来帮助 SLAM 进行目标导航跟踪。比如 Yu 等人提出的语义建图系统 DS-SLAM 采用语义分割信息构建语义地图^[37]。然而，DS-SLAM 只是简单地将语义标签附加到度量地图上以进行可视化显示，缺乏以数学形式描述的对象位置坐标，限制了系统执行高级任务规划的能力。

1.3 本文课题来源及研究内容

本课题项目、基金来源：国家重点研发计划《汽车级高精度组合导航传感器系统开发及应用》(2021YFB3202303)；河北省 2022 年省级研究生创新资助项目《多重感知下移动机器人非合作目标捕获系统控制方法研究》(CXZZBS2022145)。

本文将研究动态场景下基于视觉 SLAM 的机器人室内语义建图系统，并提出基于几何与语义信息的动态特征点剔除算法和语义建图算法。本文的研究思路与内容，如图 1-3 所示。本文的章节结构安排如下：

第 1 章为绪论。首先介绍本文的课题研究背景及其意义，然后从视觉 SLAM 的历史进展、动态场景下的视觉 SLAM 和语义建图算法等三个方面分别介绍与论文课题相关的研究现状。最后，概括论文的主要研究内容。

第 2 章为移动机器人系统设计。首先介绍移动机器人的硬件平台设计情况，包括三部分：一是搭载计算平台和传感器的麦克纳姆轮式移动平台，二是运行操作系统、处理输入数据、执行软件算法的英伟达计算平台，三是采集图像和深度数据的 RGB-D 相机。然后，介绍负责通信和可视化的 ROS 系统。最后，介绍视觉 SLAM 系统基本原理与模块，再给出经典视觉 SLAM 框架 ORB-SLAM2 和本文基于其改进的 SG-SLAM 的两个架构分析。

第 3 章为动态场景下的鲁棒视觉 SLAM。提出了一种融合几何与语义信息的动

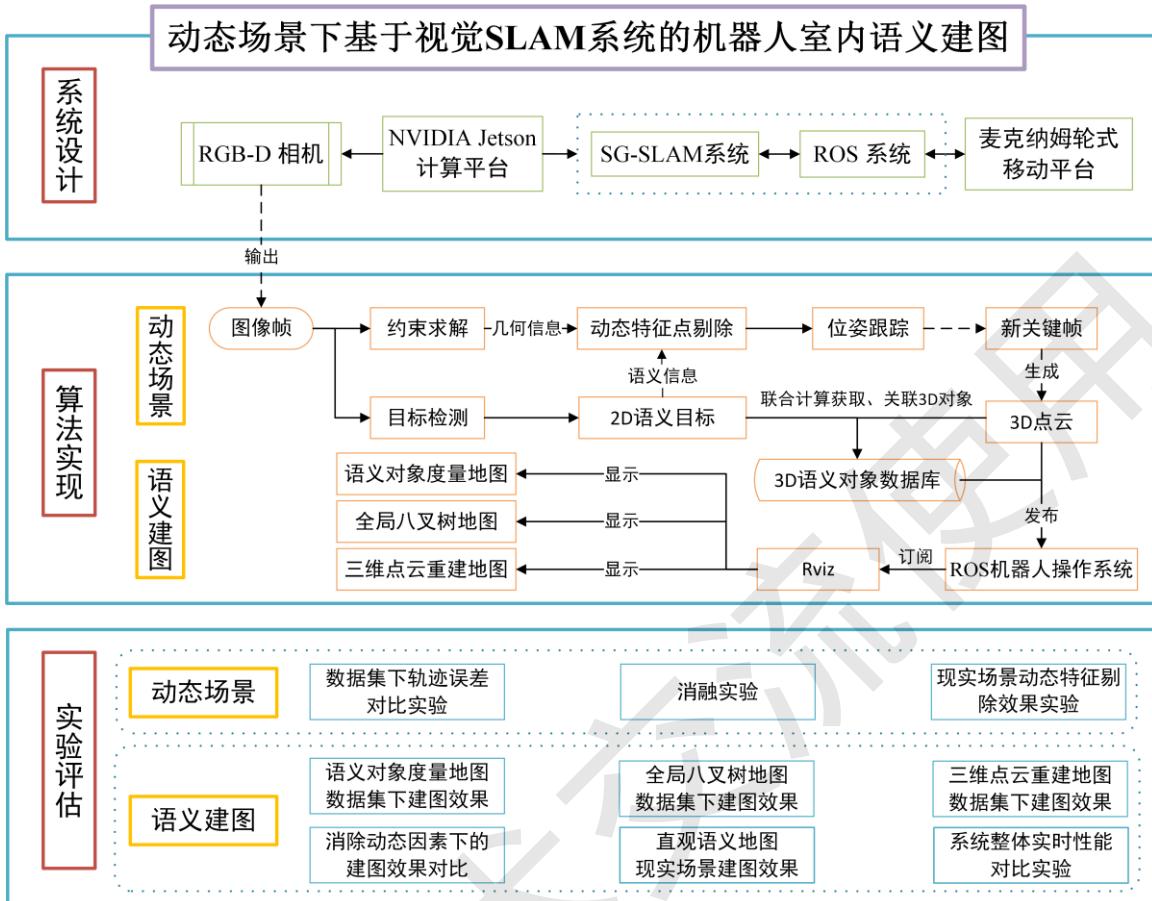


图 1-3 本文研究思路与内容

态特征剔除算法。首先详细讨论该算法的理论原理，其中包括对极几何约束、动态对象语义信息的获取方式以及该算法的具体实现方式。然后，介绍实验所用公开数据集和评价指标等相关情况。最后，将 SG-SLAM 系统在公开数据集与现实场景中进行多项定量、定性实验，验证所提算法的有效性、准确性、泛化性。

第 4 章为多类型直观感知语义地图构建。提出了一种语义度量地图生成方法。首先介绍语义地图构建算法的理论原理，包括以深度学习技术进行目标检测获取 2D 语义目标信息、3D 点云的滤波和分割处理、获取语义对象点云团信息的求解算法以及语义对象之间的数据关联方法。其次，介绍构建全局八叉树地图和全局三维点云重建的原理，说明如何消除动态因素对地图构建的影响。另外，介绍实验所用公开数据集。然后，将 SG-SLAM 系统在数据集和现实场景中进行多项定性实验进行效果展示，验证所提算法的有效性。最后，测试系统整体的实时性能并进行分析。

文章末尾将对全文内容和贡献进行总结，并根据系统目前的缺点来对未来的工
作进行展望。

第2章 移动机器人系统设计

2.1 引言

机器人是一项硬件和软件相结合的综合工程，其各个功能实现需要软硬件系统多方面的协调与配合。本文采用的机器人软硬件平台总体架构，情况如图 2-1 所示。

本章将首先介绍机器人的硬件平台，其中包括负责全向行走的麦克纳姆轮式移动平台、负责执行软件程序的英伟达 Jetson AGX Xavier 计算平台以及负责输入图像和深度数据的 RGB-D 传感器。然后，将介绍负责各系统之间通信和可视化显示功能的 ROS 机器人操作系统的相关情况。接下来，从概率学的角度阐述 SLAM 系统的基本理论原理，并介绍目前视觉 SLAM 系统的主要功能模块。最后，在介绍经典视觉 SLAM 框架 ORB-SLAM2 系统架构后，分析本文在其基础上所提出的 SG-SLAM 系统的整体架构情况。

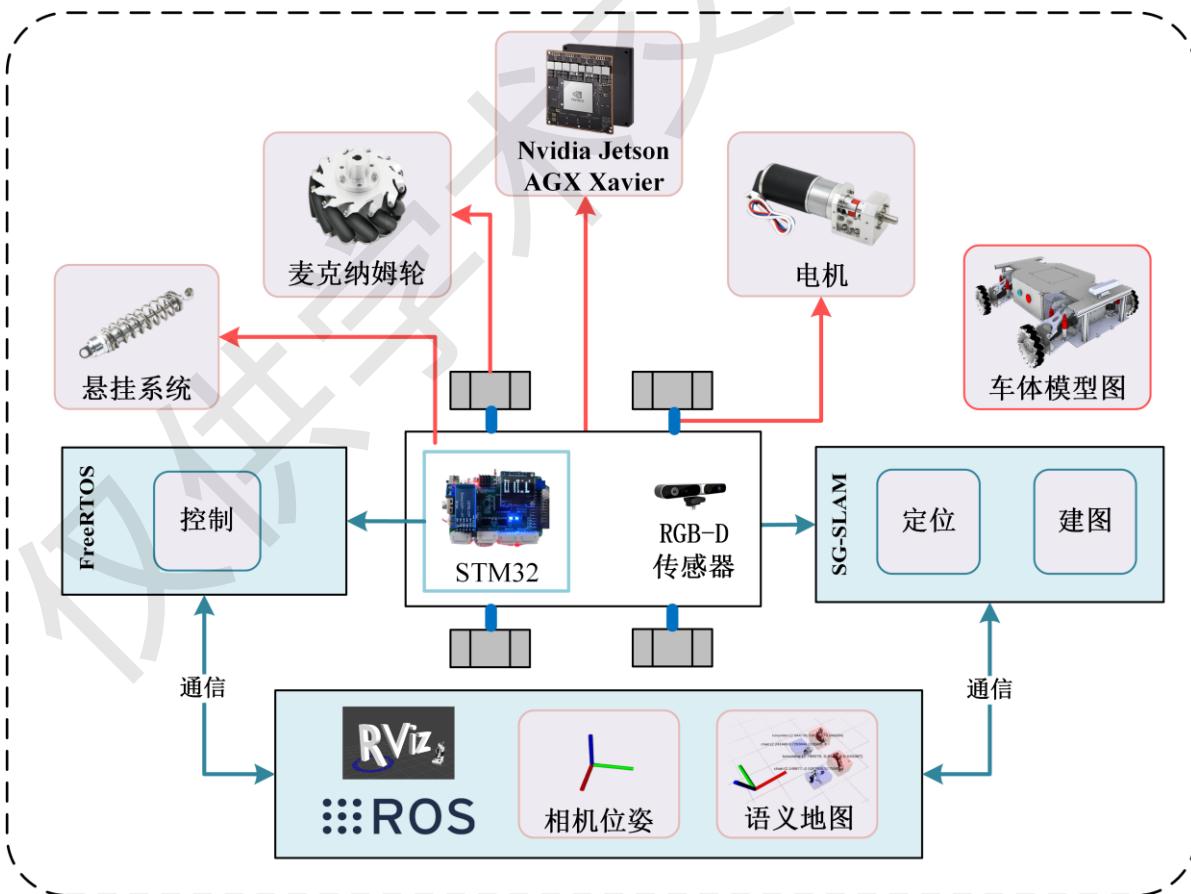


图 2-1 机器人软硬件平台总体架构图

2.2 硬件平台设计

2.2.1 麦克纳姆轮式移动平台

移动平台是机器人进行移动的主体，也是搭载计算平台和各类传感器的载体。轮式移动是机器人出现最早也是应用最广泛的移动方式，其机械结构简单、稳定，可以在很多地形中提供良好的运动性能。按照运动空间分类，轮式移动又可分为非全向移动和全向移动两种方式。全向移动方式相比前者有更强的灵活性、操纵性和高效性，因此在仓储物流、工业自动化等空间紧凑的应用场景更具优势。其中，麦克纳姆轮便是全向移动方式的一个经典模型。

一般来说，底盘不能直接进行侧向移动是因为普通车轮只有切向一个自由度。麦克纳姆轮的设计原理是在车轮外环添加一组斜向 45 度角的辊子（如图 2-1 中麦克纳姆轮所示），使得其转动时可同步产生侧向和切向两个自由度的推力。通过四只电机以不同组合方式驱动两对辊子呈镜像排列的车轮，底盘便可以实现前行、斜行、平移、原地旋转等各种全向移动方式的运动。底盘运动学和动力学运动模型解算现已集成在以 STM32 为处理器的 FreeRTOS 系统的电机驱动程序中，并通过 ROS 系统与其他算法系统进行通信。然而，由于车轮辊子之间安装工艺存在缝隙以及车轮横截面并非理想圆形等多种原因，麦克纳姆轮驱动方式在移动时会发生振动。为了尽可能缓冲振动所带来的冲击力，需要在底盘与车轮之间安装悬挂系统。

因为诸如大型复杂构件加工的自动化场景一般空间紧凑、场地非结构化，所以机器人移动方案选定为由麦克纳姆轮驱动的全向移动方式。本论文采用的麦克纳姆轮式移动平台硬件实物如图 2-2 所示。

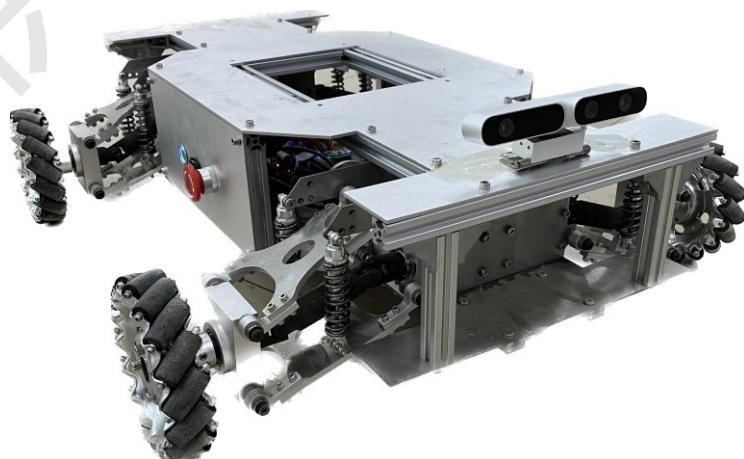


图 2-2 麦克纳姆轮式机器人移动平台

2.2.2 NVIDIA Jetson 计算平台

计算平台是机器人运行操作系统、处理传感器输入数据流、执行各种软件算法的关键硬件设备。英伟达近年推出的 Jetson AGX Xavier 系列计算设备（开实物如图 2-3 所示），其 GPU 支持 CUDA 技术且开发生态丰富，能够支持实时的深度学习算法推理等高性能任务。官方默认安装 Ubuntu 18.04 LTS 操作系统为初始开发环境，非常适合机器人 SLAM 和路径规划等任务。而且，Jetson AGX Xavier 开发主板接口丰富，可以方便的与显示、输入以及 RGB-D 相机等外部设备连接。



图 2-3 英伟达 Jetson AGX Xavier 设备实物图

一方面，本文研究的动态特征剔除、语义建图等 SLAM 算法程序需要处理大量的数据，因此需要一个高性能设备满足系统的计算能力。另一方面，依靠锂电池为主要能源的移动平台为保证足够的续航能力，所以对硬件设备的功耗十分敏感。而且机器人移动平台本身不适宜搭载太大的计算设备。这就需要计算平台在多方面要求下达到一个不错的平衡。最终，相比于其他工作站方案而言，Jetson AGX Xavier 计算平台性能足够强、功耗较低、尺寸小。因此选定其做为移动机器人的综合计算平台。其硬件主要规格参数如表 2-1 所示。

表 2-1 英伟达 Jetson AGX Xavier 开发套件规格参数

规格项	参数
CPU	8 核 ARM® v8.2 最高频率 2.2GHz 64 位 CPU
GPU	64 个 Tensor Core 的 512 核 NVIDIA Volta™ GPU
AI 性能	32 TOPs, 2 个 NVidia v1.0 深度学习加速器
功耗	10W 到 30W
尺寸	105 毫米×105 毫米×65 毫米
系统	Ubuntu 18.04
接口	2 个 USB-C 3.1、HDMI 2.0、RJ45 以太网、UART、CAN 等

2.2.3 RGB-D 相机

目前常用的视觉传感器主要有单目、双目和 RGB-D 相机三种。单目相机由于缺乏深度信息，所以无法直接得到环境的真实尺寸。随着时间推移，相机位姿状态估计的误差会逐渐积累，进而导致估计的相机位姿发生尺度漂移现象。双目相机具有预先确定的相机位置和基线长度，通过计算像素间的视差来估计深度值。但是双目相机的标定较为困难，而且立体匹配算法运算量大，很难进行实时的稠密建图。

RGB-D 相机又被称为深度相机，是一种可以同时采集 RGB 彩色图像及各像素对应深度的传感器。本文机器人平台采用的深度相机型号为乐视 LeTMC-520，硬件实物如图 2-4 所示。该相机测量深度的原理是通过特制光束发射器向探测范围内的目标发射一束光线，然后根据反射回的结构光图案信息计算距离。距离测量完毕之后，相机自身便可完成彩色像素和深度值的匹配，然后同时输出互相匹配的 RGB 彩色图像和深度图像。



图 2-4 RGB-D 相机设备实物图

一方面，由于深度相机可以直接测量深度信息，因此其相对单目相机而言不存在尺度漂移现象。另一方面，深度相机相对于双目相机的优势是可以直接通过物理手段获取深度信息，因此实时性能良好。然而，深度相机同样存在一些缺点：一是容易受到阳光或其他红外传感器的光线干扰，所以一般只建议在室内使用；二是因为很难接收到透明材质表面的反射光，所以很难测量到玻璃、水等物体的深度值。本文机器人设计应用于室内，且为满足系统实时性和稠密建图的要求，因此最终选择 RGB-D 相机作为视觉传感方案。表 2-2 为所选相机的相关技术指标。

表 2-2 乐视 LeTMC-520 深度相机规格参数

规格项	参数
深度数据工作范围	0.6 到 8 米
分辨率@帧率	640×480@30fps
视场角	H 66.10° × V 40.2°
设备尺寸	164.85×48.25×40（单位：毫米）

2.3 ROS 机器人操作系统

ROS (Robot Operating System) 是一种应用非常广泛的开源机器人领域的元操作系统^[42]，它提供了一系列用于构建机器人应用程序的工具和库。ROS 最早诞生于机器人公司 Willow Garage 和美国斯坦福大学的人工智能实验室进行合作的机器人项目，其设计目标是提供一个灵活的、可复用的机器人软件平台，所以它采用了多语言支持、松耦合软件框架、分布式通信机制等特性。

严格意义上来说，ROS 不是传统意义上进行进程管理和任务调度的操作系统，而是构建在传统操作系统上的一种结构化的通信层软件框架。ROS 将执行某种特定任务的进程称为节点（node），不同的节点可以分布在不同的计算机上进行分布式计算。节点之间可以基于消息、服务和参数等多种方式进行通信。这种机制可以使机器人系统不同功能分解为多个独立的节点，大大提高了系统的灵活性、鲁棒性和功能可扩展性。

如 2.1 节的图 2-1 所示，本文使用 ROS 以及其所提供的 Rviz 可视化工具为机器人控制系统和视觉 SLAM 系统提供通信和显示功能。

2.4 视觉 SLAM 系统

2.4.1 基本原理

SLAM 的数学概率模型基于贝叶斯理论（bayesian theorem），思想是将机器人状态和周围环境的地图视为随机变量，然后根据观测量和控制量确定这些随机变量的概率分布来进行状态估计^[46]。这是因为由概率论可知，只要已知随机变量的概率分布函数就可以求得该随机变量落在任一区间内的概率。所以只要求解出机器人待估计状态量和周围环境地图点集的概率分布函数，便可以有完整描述出这些随机变量统计规律性质的能力。具体来说，即将 SLAM 看作在已知某些含噪声的观测量和控制量的条件下确定状态量和地图点集的条件概率分布问题。

整个状态估计的直观过程可如图 2-5 所示，现做出如下定义：

狭义情况下的状态量表示机器人自身的状态信息，如位置、旋转姿态、速度、加速度等。在广义情况下，由于地图点集和机器人自身状态量在估计处理上并无本质不同，所以可以将地图点集与机器人自身状态量共同处理、不加区分，并统称为状态量。观测量表示机器人在某状态下通过传感器获取的与地图点集有关的数据。

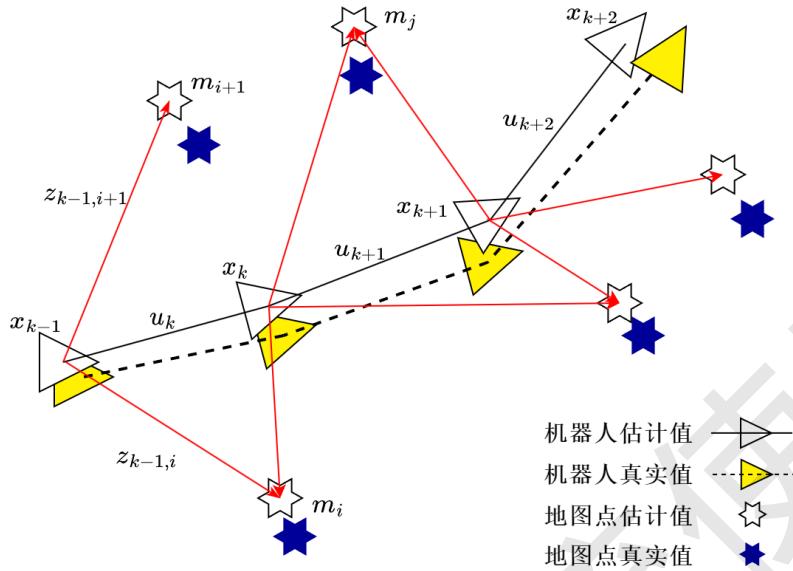


图 2-5 SLAM 概率估计模型过程示意图

控制量表示机器人从运动传感器获取到的控制输入信息，如速度、加速度、轮速计数据等。

x_k 表示在 k 时刻的状态量。 m_i 表示第 i 个地图点的位置，其简写 m 表示所有地图点位置的集合。 $z_{k,i}$ 表示机器人在 k 时刻采集的第 i 个地图点的观测量，其简写 z_k 表示机器人在 k 时刻采集的所有地图点的观测量。 u_k 表示可以在 $k-1$ 时刻将机器人自 x_{k-1} 驱动至 x_k 的控制量。默认的，观测量 z_k 在控制量 u_k 执行之后获取。

此外， $x_{1:k}$ 表示从 1 时刻到 k 时刻所有状态量的集合， $z_{1:k}$ 表示从 1 时刻到 k 时刻所有观测量的集合， $u_{1:k}$ 表示从 1 时刻到 k 时刻所有控制量的集合。

现在，从概率学的角度 SLAM 问题有两种主要的表示形式。第一种是只关心当前时刻状态量和地图点集的条件概率，称之为在线 SLAM

$$P(x_k, m | z_{1:k}, u_{1:k}) \quad (2-1)$$

第二种是求解所有时刻状态量和地图点集的条件概率，称之为完全 SLAM

$$P(x_{1:k}, m | z_{1:k}, u_{1:k}) \quad (2-2)$$

这两种问题有着极为密切的联系：在线 SLAM 可以视为将完全 SLAM 所有历史时刻的机器人状态量和地图点集进行边缘化^[47]（marginalization）的结果，即

$$P(x_k, m | z_{1:k}, u_{1:k}) = \int \cdots \int \int P(x_{1:k}, m | z_{1:k}, u_{1:k}) dx_1 dx_2 \cdots dx_{k-1} \quad (2-3)$$

在这里，边缘化的作用是将完全 SLAM 所有历史时刻的机器人状态量和地图点

集作为先验信息融合并固定下来。

为更清晰描述两种 SLAM 问题的形式，前文所述状态量均为狭义情况下的机器人自身状态量。后文为方便起见，将机器人自身状态量和地图点集在不引起混淆的情况下不加区分，二者统一使用符号 x 代替，并简称为状态量。尽管式（2-1）和式（2-2）两者在形式上仅存在一些细微差异，却使得它们在解决方法上产生了很大的不同。下面介绍这两种不同问题形式的算法解决思路：

一、在线 SLAM 的求解——贝叶斯滤波法。

式（2-1）的实际含义可以理解为：表示了从 1 时刻到 k 时刻的观测量 $z_{1:k}$ 和控制量 $u_{1:k}$ 集合为已知条件时，机器人当前时刻状态量的后验概率密度。如果某状态量的概率密度函数值相比其他位置要大，那么事件在该微小区间范围内的发生概率相比其他位置要大。直观来说，就是可以相信机器人和周围环境处于最大概率密度值时对应状态量的可能性是最大的。所以式（2-1）也可以称之为状态量的置信度（belief），记做 $bel(x_k)$ 。那么，在置信度最大值附近的微小区间便可作为机器人当前状态量的最终估计值。

从数学角度来说，这是一个最大后验估计问题（maximum a posteriori estimation）。然而直接求解该后验概率的最大值十分困难，所以不得不将该最大后验概率问题继续进行转化，以简化问题的求解。

首先，根据贝叶斯法则对式（2-1）进行展开，可以得到

$$bel(x_k) = \frac{P(z_k | x_k, z_{1:k-1}, u_{1:k}) \cdot P(x_k | z_{1:k-1}, u_{1:k})}{P(z_k | z_{1:k-1}, u_{1:k})} \quad (2-4)$$

然后，假设观测量 $z_{1:k}$ 与控制量 $u_{1:k}$ 之间都是互相独立的（即它们的对应噪声是不相关的），当前时刻的观测量只取决于当前时刻的状态量。所以式（2-4）右侧分子第一项可变换为 $P(z_k | x_k)$ 。另外，由于式（2-4）分母的取值与状态量 x_k 无关，故可以将之看作归一化常量 η 。此处归一化常量的意义是保证该置信度全域积分值仍然为 1 的合法性。则置信度又可表示为

$$bel(x_k) = \eta \cdot P(z_k | x_k) \cdot P(x_k | z_{1:k-1}, u_{1:k}) \quad (2-5)$$

现在，将等式右侧第一项式子 $P(z_k | x_k)$ 称之为似然函数（likelihood function）。第二项式子 $P(x_k | z_{1:k-1}, u_{1:k})$ 称之为先验概率（prior probability），可记作 $\overline{bel}(x_k)$ 。其中，似然函数即表示在给定状态量下，获取到该观测量的可能性（也即似然程度）。直观来看，就是把观测量看作已知结果而状态量看作是导致该结果的原因，即由结

果反推原因的可能性大小。先验概率则表示在未进行观测量之前，根据以往经验或分析推断出的当前时刻状态量的概率分布。比如，我们可以根据 $k-1$ 及以前时刻的已知数据列写状态转移方程来在理论上推测 k 时刻的状态量。

于是，通过贝叶斯法则证明了后验概率 $bel(x_k)$ 与似然函数和先验概率的乘积成正比。换句话说，现在将求解最大置信度的问题转化为了间接求解最大化似然函数与先验概率的乘积问题。

在继续推理之前，提出一个非常重要的假设——系统满足马尔科夫性（markov property）。该假设在此的含义：如果当前时刻的状态量已经足够预测未来的状态，那么任何过去时刻的状态量都不会对未来产生影响。换句话说，如果已经给定了当前时刻状态量时，那么未来状态量便与过去所有历史时刻状态量是条件独立的。

接下来，根据一阶马尔科夫性质的假设继续将式（2-5）的先验概率 $\overline{bel}(x_k)$ 在前一时刻 x_{k-1} 处边缘化展开，即

$$\overline{bel}(x_k) = \int P(x_k | x_{k-1}, u_k) \cdot P(x_{k-1} | z_{1:k-1}, u_{1:k-1}) dx_{k-1} \quad (2-6)$$

观察式（2-6）可知，被积函数第二项其实就是 $k-1$ 时刻的置信度，即

$$bel(x_{k-1}) = P(x_{k-1} | z_{1:k-1}, u_{1:k-1}) \quad (2-7)$$

稍加整理便可得到贝叶斯滤波算法的一般形式

$$\begin{cases} \overline{bel}(x_k) = \int P(x_k | x_{k-1}, u_k) \cdot bel(x_{k-1}) dx_{k-1} \\ bel(x_k) = \eta \cdot P(z_k | x_k) \overline{bel}(x_k) \end{cases} \quad (2-8)$$

因为本文研究的是没有明确控制量的纯视觉 SLAM 系统，式（2-8）可简化为

$$\begin{cases} \overline{bel}(x_k) = \int P(x_k | x_{k-1}) \cdot bel(x_{k-1}) dx_{k-1} \\ bel(x_k) = \eta \cdot P(z_k | x_k) \overline{bel}(x_k) \end{cases} \quad (2-9)$$

不难看出，贝叶斯滤波是一个两步迭代式算法。第一个式子称为运动预测模型，它表明利用运动模型和前一时刻的置信度来预测当前状态的置信度。第二个式子称为观测更新模型，它表明利用观测数据进一步对预测置信度进行修正更新。总的来说，机器人依靠某种运动模型对自身状态进行预测的过程中，又通过不断观测外界来更新预测量以降低自身状态的不确定性。为了使算法真正迭代起来，还需要给定一个初始置信度值 $bel(x_0)$ 。例如，如果没有相关的先验知识则可以直接假设其是均匀分布（即看作一个常数）。

贝叶斯滤波并没有给定后验概率分布的具体形式，它算作滤波类算法的通用框架，而后在其基础上衍生了很多算法。例如，基于参数滤波的卡尔曼滤波和信息滤波，还有基于非参数滤波的直方图滤波和粒子滤波。其中，在 SLAM 里应用最为广泛的便是假设运动预测模型、观测更新模型、初始置信度 $bel(x_0)$ 均符合高斯分布的卡尔曼滤波系列算法，如扩展卡尔曼滤波^[48]（Extended Kalman Filter）法、无迹卡尔曼滤波^[49,50]（Unscented Kalman Filter）法等。

综上所述，滤波类算法实时处理每一时刻的数据，然后把历史时刻获得的信息通过边缘化分解到概率分布中，并只针对当前时刻进行状态估计。虽然滤波类算法具有很高的计算效率，但也存在很多不能忽视的缺陷。比如卡尔曼滤波将历史信息融入进矩参数和协方差矩阵，然而当地图规模逐渐增大时算法复杂度会呈平方型指数增长。还有，由于贝叶斯滤波类算法基于马尔科夫性质的假设，这无疑限制了滤波类算法的情况处理范围（比如当前时刻的状态不仅与上一时刻状态有关，确实还和很久之前的某时刻状态有关）。

二、完全 SLAM 的求解——优化法

与基于马尔科夫性质假设的贝叶斯滤波类算法不同，优化类算法将更大范围内历史状态量也都纳入处理范围。因为本文研究的纯视觉 SLAM 系统并无明确控制量，后文为方便起见将之省略简化。

考虑式（2-2）继续利用贝叶斯法则展开，分母与状态量无关故视作归一化常量

$$P(x_{1:k} | z_{1:k}) = \eta \cdot P(x_{1:k}) \cdot P(z_{1:k} | x_{1:k}) \quad (2-10)$$

式（2-10）右侧由先验概率与似然函数乘积组成。如果没有其他手段获取到各状态量的先验值，那么同样将先验项融入常量不再考虑。于是，最大化后验估计问题在这里转为了最大化似然估计的问题。

与式（2-5）处理手法类似，假设各个时刻的观测量 z_k 只取决于对应时刻的状态量 x_k 。则问题可表示为

$$P(x_{1:k} | z_{1:k}) = \lambda \cdot \prod_k P(z_k | x_k) \quad (2-11)$$

高斯分布是概率推理中表现不确定性最有效的方式之一，其在滤波类方法获得了巨大的成功，在优化类方法中同样如此。假设观测模型符合高斯分布，则每个观测量可以建模为 $z_k = h_k(x_k) + \varepsilon_k$ 。其中， $h_k(\cdot)$ 是传感器的观测模型函数， ε_k 是符合

均值为 0，信息矩阵为 Ω_k 的随机测量噪声。因为最大似然估计等同于最小化负对数似然估计，所以式 (2-11) 最终化为由众多误差项之和组成的非线性最小二乘问题

$$\arg \min_{x_{1:k}} \left(-\sum_k \log p(z_k | x_k) \right) = \arg \min_{x_{1:k}} \sum_k \|h_k(x_k) - z_k\|_{\Omega_k}^2 \quad (2-12)$$

求解式 (2-12) 便可得到系统所有待估计的状态量。非线性最小二乘问题有许多种不同的解法，不同解法各有其优缺点，需要根据系统需要酌情加以选择。现在流行的图优化则是将上述非线性优化问题与图论结合得到的理论，这样结合有许多益处。比如可以直观的把非线性优化问题展现出来，还可以利用图论中相关的性质对问题进行更好的优化等。这一部分将在 2.4.3 后端优化一节中加以论述。

历史上由于优化类算法存在的计算量大和存储空间占用问题，并未引起重视。不过，随着对海塞矩阵 (Hessian) 稀疏性、图优化等方面的认识逐渐深入以及计算机硬件性能的飞速增长，优化类算法目前已经成为 SLAM 领域的主流方向。本文所提出的 SG-SLAM 系统的基础框架 ORB-SLAM2 便采用了图优化的方式进行状态量的求解。

2.4.2 视觉里程计

在 SLAM 中，视觉里程计 (visual odometry) 是根据相邻图像帧之间的信息估计相机位姿运动的算法，也被称为前端。主流的视觉里程计可以大致分为两大类：特征法与直接法。本文主要使用的是特征法。具体地说，基于特征的视觉里程计的计算过程可以分为以下几步：

一、特征提取

在每次到来的图像帧中，提取一定数量的特征点。这些特征点是在相机视角发生少量变化时依旧保持不变的具有代表性的点。特征点由描述特征在图像中位置的关键点和只要外观相似即应该有相似描述的描述子两部分组成。比如人工设计的 SIFT 特征^[51]和 ORB 特征^[52]，利用深度学习提取的 SuperPoint 特征^[53]等。

二、特征匹配

将前一帧和当前帧中的所有特征点进行匹配，找出它们之间的对应关系。这里的匹配是通过计算各个特征点描述子之间的汉明距离来完成的。两特征点的对应描述子之间的汉明距离越小，说明两个特征点越有可能是相同的特征。

三、位姿估计

一般情况下，单目相机要先进行初始化。即通过已匹配特征点之间的视图几何约束关系，列写约束方程式并求解恢复出基础矩阵（fundamental matrix）或单应矩阵（homography matrix）（当拍摄场景退化为平面场景时）。分解基础矩阵或单应矩阵便可得到两图像帧之间的旋转矩阵和平移向量，然后再通过三角化生成地图点。因为双目或 RGB-D 相机都具有物理测量深度的能力，因此无需进行初始化便可直接生成地图点。创建地图之后就可以使用 PnP（Perspective-n-Point）或 ICP（Iterative Closest Point）等方法求解后续的图像帧位姿。

四、构建局部地图和优化

部分视觉里程计还会利用部分历史位姿、地图点等信息，构建一个局部地图并进行小规模的状态优化。

2.4.3 后端优化

小规模优化的视觉里程计不可避免的会产生误差的累积漂移，所以后端优化模块的主要任务是接收不同时刻前端传来的各种数据，然后与历史数据一起进行联合优化，减少前端误差的累积漂移，以提高整个系统的精确度和鲁棒性。简言之，后端相比视觉里程计来说就是构建求解一个更大规模的状态优化问题。

后端优化从历史角度来说一直是 SLAM 研究的核心问题，这一部分实际上就是 2.4.1 基本原理一节所论述的优化类算法内容。该模块从本质上进行的是非线性优化问题的求解，有很多种求解的方法，如梯度下降法、高斯牛顿法、列文伯格-马夸尔特（Levenberg-Marquardt）法、狗腿（Dogleg）法等，应根据不同的场景恰当选择。

2.4.4 闭环检测

虽然普通的后端优化已经通过历史信息尽可能消除了视觉里程计累积的误差漂移，但全局的误差漂移仍然不可避免的存在。闭环检测模块具有识别机器人是否在历史上曾经到访过该场景的能力。一旦确认机器人历史上确实曾经到访该场景，模块接下来便会计算当前时刻帧和历史该场景帧的相对位姿信息。然后启动全局优化来纠正所有的位姿轨迹和地图信息，以提高精确度并形成全局一致的地图。

闭环检测模块赋予了 SLAM 系统进行长期数据关联的能力，它在消除系统前后端的累积漂移误差上起着至关重要的作用。

2.5 软件系统设计

2.5.1 ORB-SLAM2 框架分析

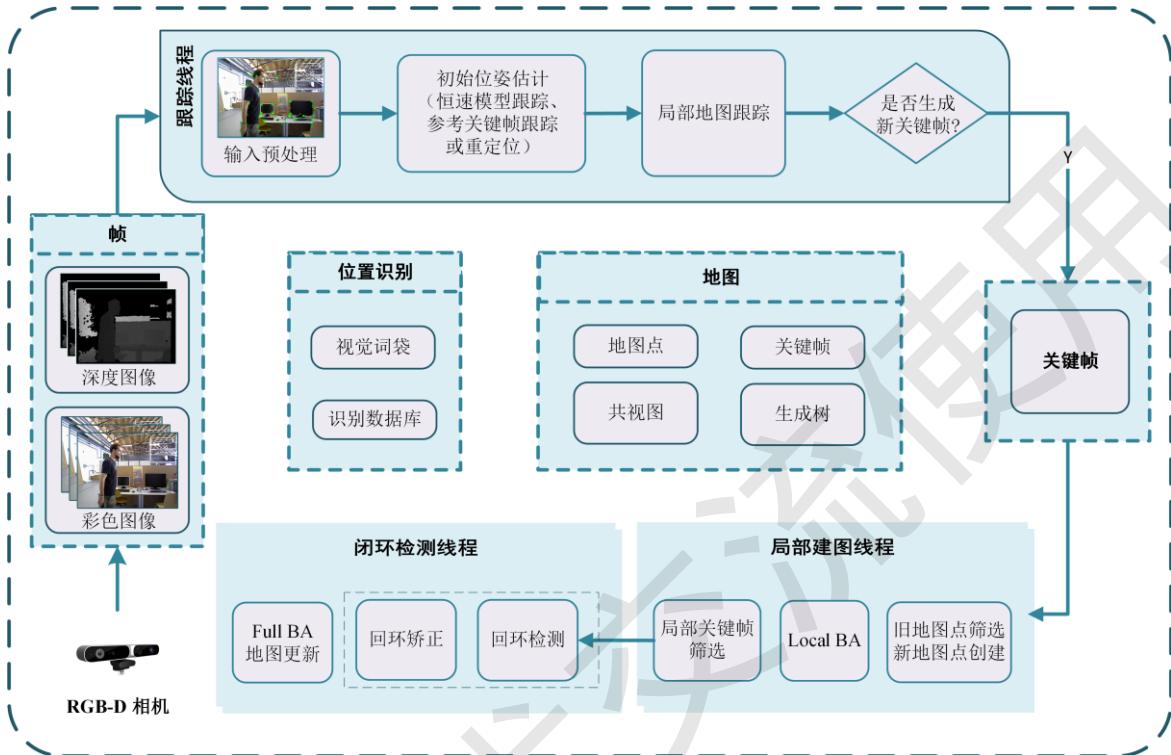


图 2-6 ORB-SLAM2 (双目/RGB-D) 系统框架图

ORB-SLAM2 系统原生支持单目、双目、RGB-D 等相机输入。该系统除了在图像数据输入预处理、闭环检测计算相对位姿（单目相机为 $Sim3$ ，双目/RGB-D 相机为 $SE(3)$ ）等一些原理略有区分的功能之外，框架大体部分均保持了一致。如图 2-6 所示，该系统共分为三大主要线程，分别是跟踪、局部建图和闭环检测线程（大致对应视觉里程计、后端优化和闭环检测模块）。系统框架主要工作流程描述如下：

跟踪线程实时处理相机输入的图像帧（主要是提取 ORB 特征），根据情况选择恒速运动、参考关键帧或重定位模型估计出当前帧的位姿。然后继续通过小规模的局部地图优化该位姿，并判断是否生成新关键帧。局部建图线程队列接收到新关键帧后，首先剔除以前添加的低质量地图点，然后根据新关键帧生成一些新的地图点。如果队列中所有新关键帧都处理完毕，那么就融合重复地图点、进行 Local BA 优化并删除冗余的相邻关键帧。最后，将新关键帧送入闭环检测线程。闭环检测线程队列接收到新关键帧后，首先判断该帧是否发生闭环。如果成功闭环，则计算当前帧与闭环帧之间的相对位姿，并进行本质图优化。最后进行全局 BA 优化和地图更新。

2.5.2 SG-SLAM 框架分析

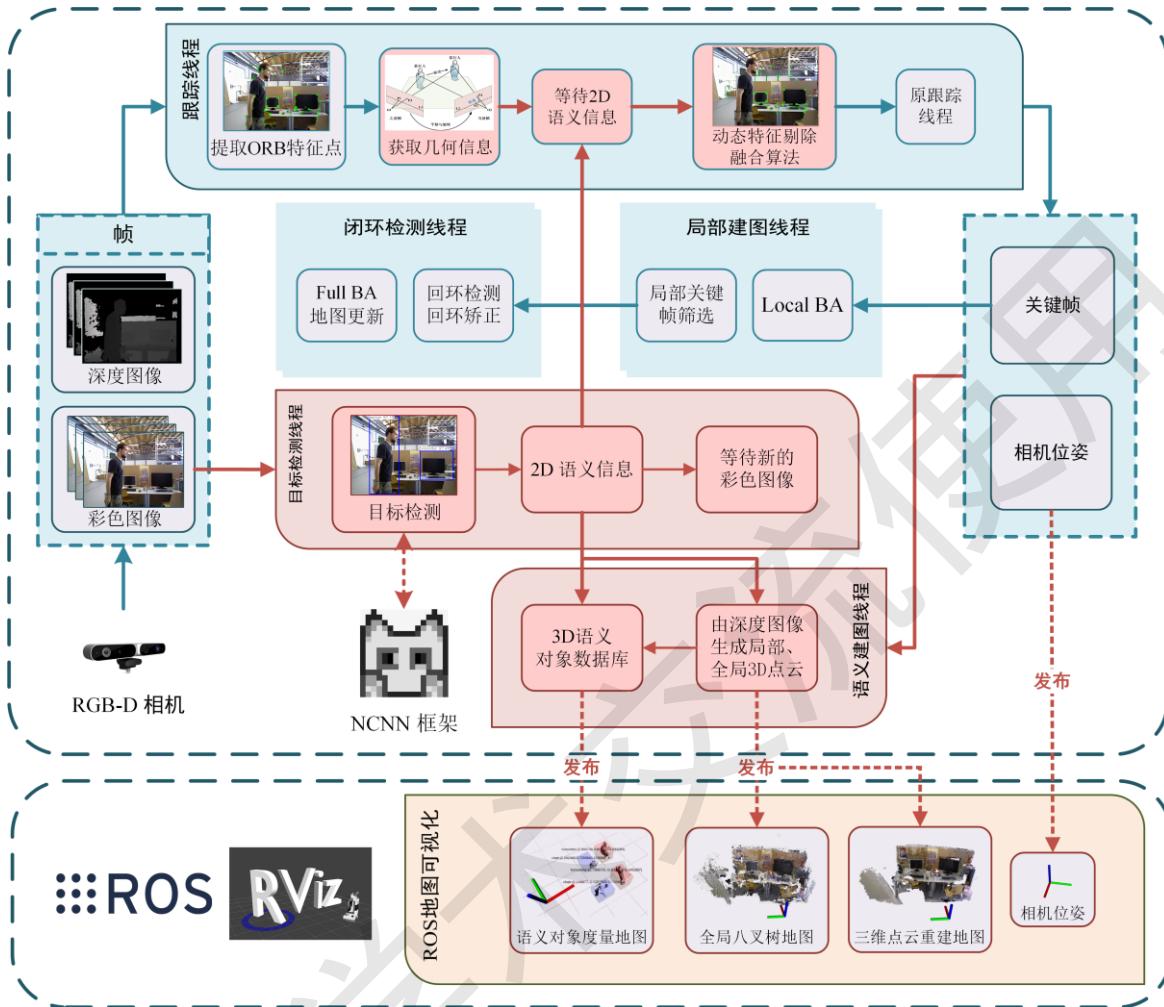


图 2-7 本文所提出的 SG-SLAM 系统框架图

为了提升机器人在动态场景中的状态估计精度以及创建直观的语义地图，本文提出了基于 RGB-D 相机的 SG-SLAM 软件系统。这是一个基于 ORB-SLAM2 框架的实时 RGB-D 语义视觉 SLAM 系统，其系统架构如图 2-7 所示。图中水绿色背景呈现的是 ORB-SLAM2 原有模块（详细可见图 2-6），红色、黄色背景下呈现的是本文新增或修改的相关功能。其主要工作流程描述如下：

RGB-D 相机采集图像帧后，同时送入跟踪线程与目标检测线程。跟踪线程先从图像帧计算获取 ORB 特征点和几何信息后进入阻塞状态，等待 2D 语义信息到来。目标检测线程基于 NCNN 神经网络前向推理框架^[54]对彩色图像进行目标检测，获取 2D 语义信息。获取完成之后，跟踪线程通过融合了几何信息与语义信息的动态特征剔除算法策略，来消除动态对象对系统在准确性和鲁棒性方面的负面影响。

为了降低系统计算量，语义建图线程部分只接收被设置为关键帧的深度图像数据来生成局部 3D 点云信息。动态对象会影响地图构建的效果，因此如果目标检测线程检测到动态对象，那么生成局部 3D 点云信息时会剔除位于 2D 动态语义对象检测框范围内的点云。一方面，使用局部 3D 点云信息和之前目标检测线程获取的 2D 语义信息计算获取对应的 3D 语义对象。接下来再使用 Kuhn-Munkres 算法将获取到的语义对象与 3D 语义对象数据库已有对象进行数据关联和对象融合。如果融合对象的数量小于原本获取的对象数量，那便将这些新对象放入 3D 语义对象数据库。另一方面，将所有关键帧的局部 3D 点云进行拼接、滤波，便可得到全局 3D 点云。

SG-SLAM 系统的语义地图可视化部分基于 ROS 系统接口。在系统运行时，会将 3D 语义对象数据库中带有坐标的语义对象和相机位姿通过消息机制发布至 Rviz 中展示。同时语义建图线程中生成的局部点云及对应位姿、全局 3D 点云也会通过话题进行发布。octomap_server 建图服务功能包根据局部 3D 点云和对应位姿，进行全局八叉树地图（OctoMap）^[55]的增量式构建，然后在 Rviz 展示。三维点云重建地图效果同样可以直接在 Rviz 中订阅全局 3D 点云话题进行展示。

2.6 本章小结

本章主要对移动机器人整体系统设计进行研究。首先介绍本文的硬件实验平台设计与选型，包括负责搭载各种硬件和传感器的麦克纳姆轮式移动车体、负责运行系统和算法程序的英伟达计算平台，以及负责输入图像数据的 RGB-D 相机。然后介绍了实现地图可视化，以及硬件控制系统和定位建图系统之间的通信等功能的 ROS 机器人操作系统。接下来，分别详细推导了基于滤波和优化的 SLAM 系统概率数学原理，并介绍了主流视觉 SLAM 系统的三大组成模块。最后，在梳理经典视觉 SLAM 系统 ORB-SLAM2 主要流程的基础上，继续深入分析了本文所提出的 SG-SLAM 系统框架结构。

第3章 动态场景下的鲁棒视觉 SLAM

3.1 引言

在第2章中已经介绍了移动机器人的硬件系统设计与软件系统架构等整体情况。生产生活场景中常常会出现动态物体，这些移动的对象会给默认静态场景假设的经典视觉SLAM系统的定位带来许多错误关联信息，从而降低系统的精确度和鲁棒性。本章提出一个动态特征点检测与剔除算法，可以有效提高SLAM系统在动态场景下的状态估计性能，减少轨迹误差。首先，介绍图像帧之间的视图几何约束原理，该原理算法可以从几何信息上判断特征点是否为动态点，此类算法缺点是经验阈值很难进行准确设定。然后，介绍使用目标检测方法获取语义信息剔除动态点的算法原理。接下来，论述本文所提出的融合上述几何信息和语义信息的动态特征剔除算法，并给出该算法的具体实现。最后，介绍检验算法有效性的公开数据集，并进行四组实验分析验证算法的准确性、融合有效性、先进性、泛化性及实用性。

3.2 动态场景处理算法原理

3.2.1 几何信息获取

对极几何约束表示相机在运动过程中两个不同位置发生的成像信息之间存在的一种特殊的几何关系。图3-1便是这种几何关系的形象表示。

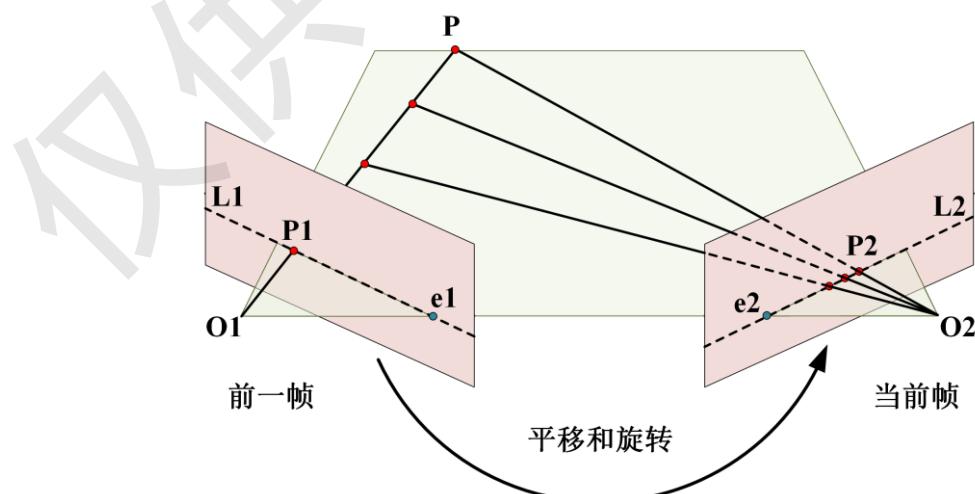


图3-1 对极几何约束示意图

如图 3-1 所示，左右的粉色平行四边形代表着相机在两个位置的成像平面。 O_1 和 O_2 分别代表当时相机的光心位置，它们两者之间的连线 O_1O_2 称为基线。基线与两个成像平面之间的交点 e_1 和 e_2 称之为极点（Epipoles）。点 P 是一个空间中的三维地图点， P_1 和 P_2 分别是三维点 P 在前一帧和当前帧上的投影像素点。点 P 、点 O_1 和点 O_2 三者一起形成的平面（图中绿色平面）称为极平面（Epipolar plane）。极平面分别和两个成像平面之间相交的虚线 L_1 和 L_2 称之为极线（Epipolar line）。

从前一帧的相机位置来看，如果没有 P 点确切的深度信息，那么从 O_1 点起始的射线 O_1P 上所有位置都有可能是三维地图点 P 的真实位置。很明显，这些可能的位置在下一帧（即当前帧）上的投影全都落在了成像平面的极线 L_2 上。

通过这些点线之间的代数几何关系，便可以推导出下面一个代数恒等式

$$P_2^T \cdot F \cdot P_1 = 0 \quad (3-1)$$

式（3-1）中的 F 即代表两图像帧之间的基础矩阵，该式子就被称为对极几何约束。对极几何约束式给出了匹配的特征像素点之间的位置约束关系。所以，既可以使用已知的正确匹配特征像素点来求解图像帧之间的位姿变换（可从基础矩阵 F 中分解得到），又可以在已知基础矩阵的条件下推断特征像素点匹配是否正确。

SG-SLAM 使用上述对极几何约束原理中特征像素点与其对应极线之间的距离来推断当前帧中特征点是否为动态点，因此也称之为极线约束。通过极线约束判断动态特征的方法流程十分简单直接。首先，在相邻两张彩色图像帧中建立起 ORB 特征点的匹配关系。其次，根据对极几何约束完成基础矩阵的求解。最后，计算当前帧中各个特征点与其相对应极线之间的距离信息。该距离的值越大，说明该点为动态特征点的概率就越大，如果该距离超过某个阈值，那么就认为其是动态特征点。

一方面，要求解出两张图像帧之间准确的基础矩阵，就需要特征点之间具有正确的数据关联。然而问题是，求解基础矩阵的最终目的就是检测特征点之间的数据关联是否正确，这就成了一个经典的先有鸡还是先有蛋的问题。

另一方面，原基础框架 ORB-SLAM2 系统跟踪线程在特征点匹配时，一般采取视觉词袋法（参考关键帧跟踪）或投影区域搜索法（恒速模型跟踪）完成特征点描述子之间的匹配。但是它们并不能识别出特征点所属区域的物体是否已经发生移动，继续使用这些方法显然无法消除动态环境对系统的负面影响。

鉴于上述两种情况，为了得到一个相邻图像帧之间的相对准确的基础矩阵，SG-SLAM 先使用金字塔的迭代 Lucas-Kanade 光流方法来计算当前帧中已提取的特

征点在之前帧中的对应匹配点集。接下来，我们受到 DS-SLAM^[37]与双目立体视觉匹配中绝对差异之和（SAD）算法思想的启发，再将位于图像帧边缘以及像素外观差异过大的特征匹配点对去除。而且，如果在上一帧中检测到了潜在的动态对象，那么同样将这些位于动态对象检测框内的潜在动态特征移除。然后，使用基于随机采样一致（RANSAC）算法的七点法来计算两帧之间的基础矩阵 F 。一般情况下，动态对象相比于整体背景的比例是相对较小的。所以随机采样一致算法可以进一步减少动态区域的错误数据关联对基础矩阵的准确求解产生的影响。

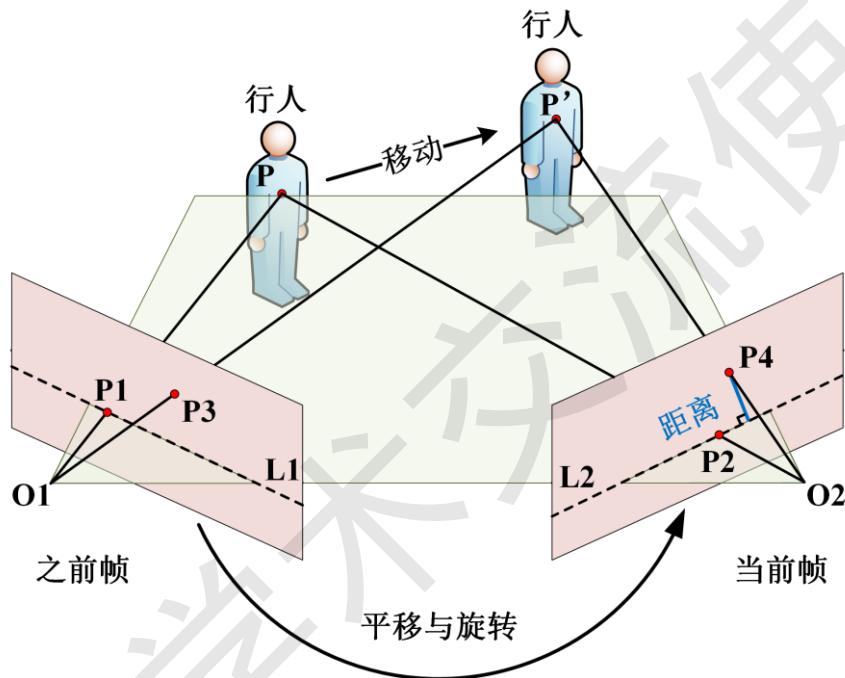


图 3-2 极线约束判断动态特征点示意图

根据针孔相机模型，如图 3-2 所示，相机在运动过程中从两个不同位置观测到同一个三维地图点 P 。 P_1 和 P_2 对应的齐次坐标形式表示如下：

$$P_1 = [x_1 \ y_1 \ 1], P_2 = [x_2 \ y_2 \ 1] \quad (3-2)$$

此处的 x 和 y 分别表示特征点在图像像素坐标系统中的坐标值。然后，当前帧中的极线 L_2 可以由基础矩阵（表示为 F ）计算，方程式如下：

$$L_2 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = F \cdot P_1 = F \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \quad (3-3)$$

此处的 X ， Y 和 Z 表示线向量。根据式 (3-1)，极线约束可表示为下式：

$$P_2^T \cdot F \cdot P_1 = P_2^T \cdot L_2 = 0 \quad (3-4)$$

然后，定义特征点 $P_i (i=2,4)$ 与对应极线的距离为偏移距离，用符号 d 表示。

偏移距离可以由以下公式计算：

$$d_i = \frac{P_i^T \cdot F \cdot P_1}{\sqrt{X^2 + Y^2}} \quad (3-5)$$

如果点 P 为静态空间点，联合方程式 (3-4) (3-5)，则 P_2 点的偏移距离为：

$$d_2 = \frac{P_2^T \cdot F \cdot P_1}{\sqrt{X^2 + Y^2}} = 0 \quad (3-6)$$

方程式 (3-6) 说明在理想情况下，当前帧中的特征点 P_2 点恰好落在极线 L_2 上。不过在现实情况下，由于受到各类噪声等因素影响，偏移距离一般大于零但低于一个经验阈值 ε 。

如果点 P 不是静态空间点，例如当相机由之前帧移动到当前帧时，点 P 所代表的三维地图点也移动到 P' 处。这种情况下，与 P_1 点相匹配的则是由 P' 映射至当前帧中的 P_4 特征点。如果三维地图点 P 的移动未发生退化^[24]，那么一般情况下 P_4 的偏移距离大于阈值 ε 。反过来说，可以由偏移距离与经验阈值 ε 比较来判断特征点是否为动态特征点。然而，经验阈值 ε 十分难以确定。因为阈值一旦设置的过大，那么将会有许多实际的动态特征漏检。而阈值设置的太小，又会将许多在允许误差范围内的静态点错检成动态特征。

3.2.2 语义信息获取

很多结合深度学习的 SLAM 工作实现都依赖着许多特定版本的第三方库，这使得算法在其他设备或系统上的移植难度大大提升。同时，部分工作由于受到电池续航方面的诸多限制，很多移动机器人会选择高能耗比的 ARM 架构类型的芯片。如图 3-3，NCNN 是腾讯推出的专门为移动端进行优化的高性能神经网络前向推理计算框架^[54]。其利用纯 C++ 语言实现无任何第三方依赖，而且支持众多主流神经网络框架、支持多系统多架构平台设备部署。NCNN 的各种优秀特性使得它可以轻松的整合至 SLAM 系统，所以选用 NCNN 作为目标检测线程的基础框架。

SLAM 作为移动机器人进行状态估计的基础组件，只有具备良好的实时性能才能保证上层任务的顺利运行。然而，很多动态场景 SLAM 的相关工作由于复杂的语义分割神经网络或者不合理的系统架构，使得算法运行速度缓慢、实时性能差。为了尽可能的提升系统中目标检测线程的速度，SG-SLAM 选用了比 YOLO 更快更准

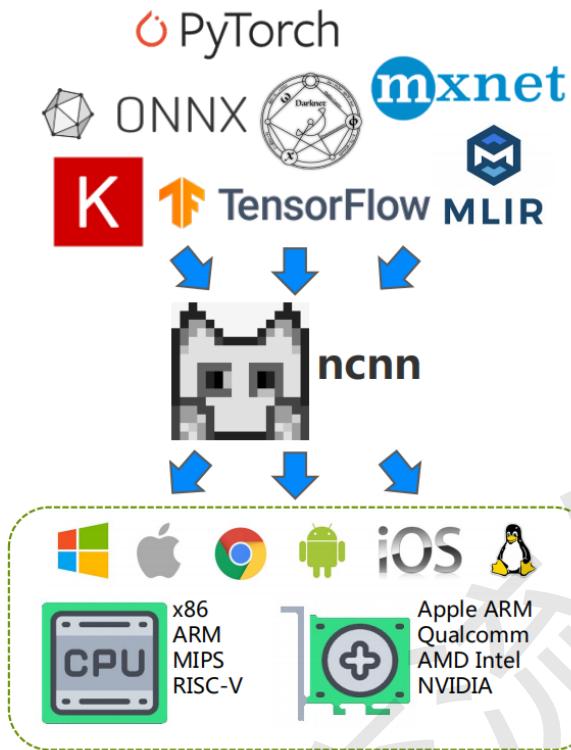


图 3-3 NCNN 神经网络前向推理计算框架生态图

确的多类别单次检测器 SSD^[45]为检测头。另外又使用为移动端设备进行优化的 MobileNetV3^[56]作为 SSD 的主干特征提取器的替代品。最后，使用 PASCAL VOC Dataset^[57]对该网络进行训练，一共可以识别 20 个常见的对象类别。

NCNN 前向推理计算框架和 MobileNetV3-SSD 目标检测神经网络的选用，再加上系统合理的架构安排，使得 SG-SLAM 克服了同类工作实时性能差的缺点。通过该目标检测网络获取的对象边界检测框便可以获得图像帧中某些常见动态对象的先验语义信息。

3.2.3 动态特征剔除融合算法

为了尽可能的解决动态场景 SLAM 以往同类工作中的其他问题，SG-SLAM 利用语义信息辅助几何信息的思路来检测和剔除动态特征点。

基于几何信息的方法根据偏移距离与经验阈值比较大小的结果来判断特征点所代表的三维地图点是否发生了移动。然而，阈值的大小相当难以界定^[40]：设定太小会使得许多静态特征点被识别为动态点造成误检，设定太大则会漏检许多真正的动态特征点。这是因为单纯的极线约束方法无法从语义层面来理解场景，只能机械的

使用一个固定阈值处理所有特征点。

为解决上述问题，首先将目标检测线程的所有对象种类根据人的先验知识分为静态对象和动态对象两类。分类标准是：凡是具有移动属性的对象就被定义为动态对象（例如人、自行车）。然后定义标准经验阈值 ε_{std} 和动态权重值 w 两个概念。标准经验阈值 ε_{std} 的选定方法十分直接：只需保证极线约束方法在使用此阈值时，仅对移动比较明显的动态特征点进行剔除。动态权重值 w 则是根据动态对象活动可能性大小设定的动态对象属性值。例如，人类平时移动的概率很大，则动态权重值为 $w=5$ ；椅子一般不会进行移动，则动态权重值为 $w=2$ 。有了这些准备之后，便可对当前帧中所有的特征点进行逐一判别，过程描述如下：

表 3-1 动态特征剔除算法伪代码

算法名称：融合几何信息和语义信息的动态特征剔除算法

输入：之前帧， F_1 ；当前帧， F_2 ；之前帧的特征点， P_1 ；当前帧的特征点， P_2 ；

标准经验阈值， ε_{std} ；之前帧的潜在动态对象区域， $rect2d$ ；

输出：当前帧特征点中所有静态特征点的集合， S ；

流程：

```

1 :  $P_1 = \text{CalcOpticalFlowPyrLK}(F_2, F_1, P_2)$ 
2 : 删除位于图像边缘且像素四周外观变化太大的特征匹配对
3 : 删除位于之前帧潜在动态对象区域  $rect2d$  内的特征
4 :  $FundmentalMatrix = \text{FindFundamentalMat}(P_2, P_1, 7\text{-point method based on RANSAC})$ 
5 : for each matched pair  $p_1, p_2$  in  $P_1, P_2$  do:
6 :   if (DynamicObjectsExist && IsInDynamicRegion( $P_2$ ))
7 :     if (CalcEpiLineDistance( $p_1, p_2, FundmentalMatrix$ )  $\times$  GetDynamicWeightValue( $P_2$ )  $< \varepsilon_{std}$ )
8 :       Append  $p_2$  to  $S$ 
9 :     end if
10:   else
11:     if (CalcEpiLineDistance( $p_1, p_2, FundmentalMatrix$ )  $< \varepsilon_{std}$ )
12:       Append  $p_2$  to  $S$ 
13:     end if
14:   end if
15: end for

```

首先，看目标检测线程是否在当前帧中检测到动态对象。如果不存在或者虽然存在但当前特征点不在动态对象检测边界框内，则计算当前特征点的偏移距离后，直接将之与标准经验阈值 ε_{std} 进行比较，根据结果决定是否将之剔除。如果存在动态对象且当前特征点位于其检测边界框内，则计算当前特征点的偏移距离后，将之与动态权重值 w 和标准经验阈值 ε_{std} 的乘积进行比较，根据结果判断是否进行剔除。算法的具体伪代码实现请见表 3-1。

目标检测线程获得的关于动态对象的先验语义信息赋予了基于几何信息的方法从更高层次理解环境的能力。对不同概率的动态区域的采取不同大小的经验阈值，克服了经验阈值难以选定的困难。由于没有过分依赖语义信息，所以该剔除算法弥补了 1.2.2 节所述的一些缺点。

3.3 实验验证及分析

3.3.1 公共数据集与评价指标介绍

3.3.1.1 慕尼黑工业大学 RGB-D SLAM 数据集

TUM RGB-D 数据集^[58]是德国慕尼黑工业大学（Technical University of Munich）计算机视觉研究组的一个室内机器人大型数据集。该数据集的目的是为视觉里程计和 SLAM 系统创建一个新的评估基准。该数据集是用 Microsoft Kinect 深度相机以 30 赫兹全帧速率和 640×480 的像素分辨率拍摄记录的，其中每个图像都包含彩色图像和相应的深度图像。相机运动的真实轨迹（ground truth）是从使用了八个高速跟踪相机的高精度动态捕捉系统中采集获得。它还提供了轨迹比较和误差度量的自动化工具，以便更好地评估算法的表现。该数据集可以免费下载使用，并且已经被广泛应用于学术研究和工业应用中。

该数据集根据使用场景的不同，将所有视频序列分成了很多个类别，包括日常办公室、仓库等场景。这些序列类别分别用于不同需求的 SLAM 算法测试，例如手持 SLAM 序列、机器人 SLAM 序列、3D 对象重建序列等。在实验后，把 SG-SLAM 系统在数据集序列下运行得到的实验轨迹与数据集给定的真实轨迹相比较，便可准确评价算法的实际性能。

3.3.1.2 波恩大学 RGB-D 动态数据集

Bonn RGB-D SLAM 数据集^[59]，是波恩大学提供的为评估动态场景 SLAM 性能

的一个数据集，其包含高度动态的视频序列。该数据集共提供了 24 个动态序列和 2 个静态序列。人们在这些序列中执行不同的动作，例如操纵盒子或玩气球。对于每个序列场景，数据集还提供了使用 Optitrack Prime 13 动作捕捉系统记录的相机真实轨迹。此外，数据集还开放了使用 Leica BLK360 地面激光扫描仪采集的处于静态环境下的地面真实 3D 点云数据。为了方便开发者使用，这些序列的格式与慕尼黑工业大学 RGB-D SLAM 数据集相同，因此可以使用相同的评估工具。

3.3.1.3 误差评价指标

为了定量评估对比算法之间的跟踪性能准确度差异，实验采用了绝对轨迹误差（Absolute Trajectory Error, ATE）和相对位姿误差（Relative Pose Error, RPE）两个误差指标。其中，绝对轨迹误差（ATE）用于评估实验估计轨迹与真实轨迹之间每帧的差值，如第 i 帧的绝对轨迹误差可以使用下式进行计算

$$\text{ATE}_i = T_{gt,i}^{-1} T_{est,i} \quad (3-7)$$

其中 $T_{gt,i}$ 代表真实轨迹在第 i 个时刻的位姿， $T_{est,i}$ 代表实验估计轨迹在第 i 个时刻的位姿。

一般来说，最常用的是通过比较估计轨迹和真实轨迹之间的欧几里得距离平均值计算所有帧下 ATE 的均方根误差（Root-Mean-Squared Error, RMSE）统计量，其具体计算过程如下式

$$\text{RMSE}(\text{ATE}_{1:N}) = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^N \left\| \log(T_{gt,i}^{-1} T_{est,i})^\vee \right\|_2^2} \quad (3-8)$$

其中 N 代表了位姿轨迹的总帧数（总时刻数），式中采用了将李群使用对数映射至李代数的位姿表示法。除了均方根误差统计量之外，还可以计算 ATE 指标的平均值、中位数、标准差等统计量来衡量位姿轨迹之间的绝对误差水平。

相对姿态误差（RPE）衡量的是 SLAM 算法中估计轨迹和真实轨迹相邻两帧之间的姿态误差。与 ATE 类似的，它将估计轨迹上相邻两帧之间的位姿变换矩阵与真实轨迹上对应两帧之间的位姿变换矩阵求逆，得到轨迹之间的相对位姿误差。如式（3-9）是第 i 帧的 RPE 计算过程

$$\text{RPE}_i = (T_{gt,i}^{-1} T_{gt,i+\Delta t})^{-1} (T_{est,i}^{-1} T_{est,i+\Delta t}) \quad (3-9)$$

其中， Δt 为固定的时间间隔。因为相对位姿误差一般包含了平移以及旋转两个误差部分，所以可单独分别表示这两部分误差。比如，只考虑平移部分的所有帧下

RPE 的均方根误差统计量计算表达式为

$$\text{RMSE}(\text{RPE}_{1:N}) = \sqrt{\frac{1}{N-\Delta t} \cdot \sum_{i=1}^{N-\Delta t} \left\| \text{trans}\left(\left(T_{gt,i}^{-1} T_{gt,i+\Delta t}\right)^{-1} \left(T_{est,i}^{-1} T_{est,i+\Delta t}\right)\right) \right\|_2^2} \quad (3-10)$$

另外，还有只考虑旋转部分的 RPE 均方根误差统计量。除了均方根误差之外，也可以计算 RPE 指标的平均值、中位数、标准差等统计量来衡量位姿轨迹之间的相对误差水平。

3.3.2 与 ORB-SLAM2 性能对比

本文所提出的 SG-SLAM 系统是基于 ORB-SLAM2 框架的基础上改进而来的。因此为了评估系统改进前与改进后在跟踪精度方面的提升情况，本节将对 SG-SLAM 与原始 ORB-SLAM2 算法在动态场景下的跟踪性能进行对比实验验证。本节实验的硬件平台为 NVIDIA Jetson AGX Xavier 计算开发套件，其软件系统运行环境为 Ubuntu 18.04 LTS。

实验数据中，通过式 (3-11) 来计算本文所提算法 SG-SLAM 相较于原始 ORB-SLAM2 算法的性能提升比率

$$\eta = \frac{o - s}{o} \times 100\% \quad (3-11)$$

上式中， o 表示 ORB-SLAM2 算法估计的轨迹位姿与真实位姿之间的误差， s 表示本文所提 SG-SLAM 算法估计的轨迹位姿与真实位姿之间的误差。

为了评估 SG-SLAM 系统在动态场景下的准确性和鲁棒性，本节实验主要使用了慕尼黑工业大学数据集中动态对象 (dynamic objects) 类别下的五个代表性视频序列。其中前四个以 fr3_walking 名称开头的为高动态场景序列，第五个 fr3_sitting_static 为低动态场景序列。下面分别简要介绍这些序列的情况：

在 fr3_walking_xyz 序列中，场景是手持的深度相机保持在 xyz 三个方向横移拍摄两个人在办公室场景里穿梭行走；在 fr3_walking_static 序列中是手持的深度相机保持在原位来拍摄办公室两个人行走的画面；在 fr3_walking_rpy 序列中，深度相机沿滚转、俯仰和偏航 (roll-pitch-yaw) 三个方向旋转拍摄两个人在办公室场景里行走移动的场景；在 fr3_walking_halfsphere 序列中，深度相机在一个直径约一米的小半球上拍摄两人在办公室行走的画面。这四个高动态序列目的是评估在可见场景中存在大面积快速移动对象时 SLAM 算法的鲁棒性。

在 fr3_sitting_static 序列中，手持的深度相机保持在一个位置拍摄坐在办公桌前说话、做手势交流的两个人。该序列场景中动态因素较小，这个低动态序列目的是评估场景中存在缓慢移动动态对象时 SLAM 算法的鲁棒性。因为该序列的跟踪难度低于高动态序列，所以作为实验的补充对照序列。

图 3-4 至图 3-8 五张图片是在 RGB-D 相机数据输入模式下，原始 ORB-SLAM2 和本文所提 SG-SLAM 两个系统算法分别运行这五个视频序列之后的估计轨迹与真实轨迹的绝对轨迹误差（ATE）结果对比图。

图中黑色画线是通过动作捕捉系统获取的真实轨迹，蓝色画线是 SLAM 算法通过视频序列的图像信息进行状态估计得到的估计轨迹，黑色和蓝色之间的红色连线则代表着真实轨迹和估计轨迹的误差值。红色连线覆盖的面积越大说明误差值越大，也就意味着算法的位姿估计精度越低。

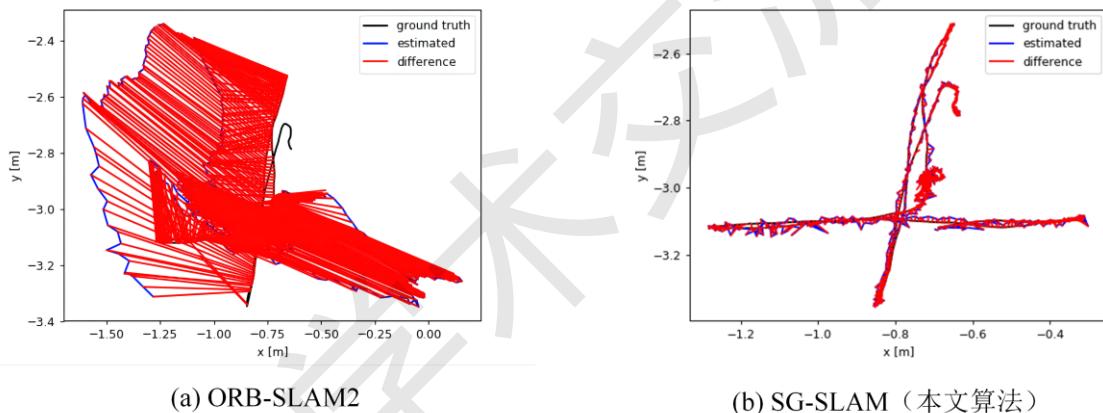


图 3-4 TUM 数据集 fr3/walking_xyz 序列的 ATE 结果图

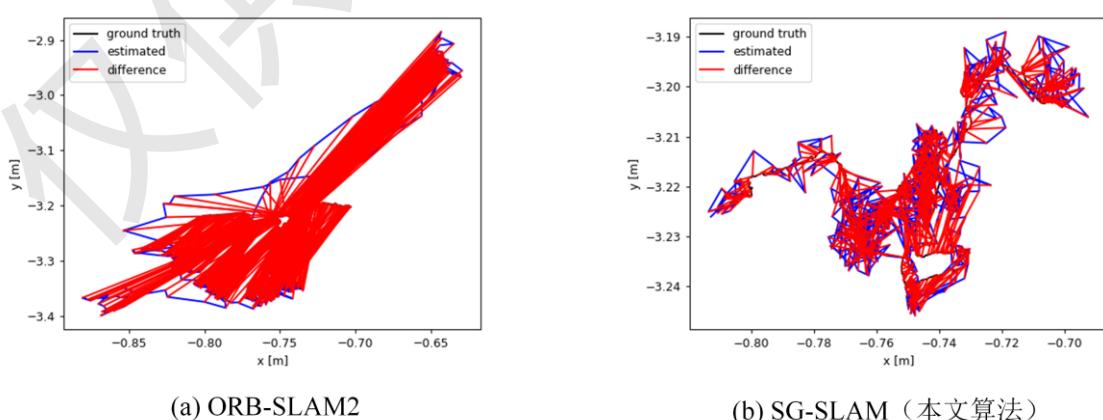


图 3-5 TUM 数据集 fr3/walking_static 序列的 ATE 结果图

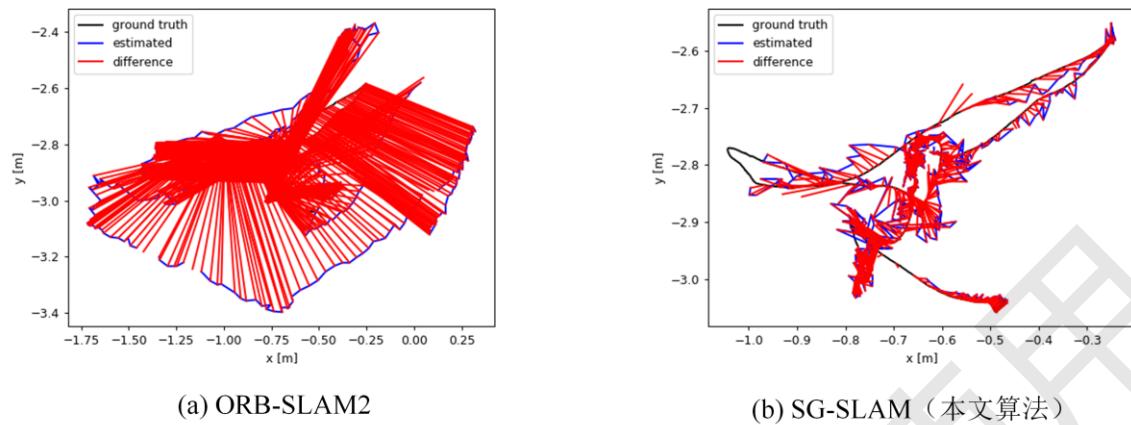


图 3-6 TUM 数据集 fr3/walking_rpy 序列的 ATE 结果图

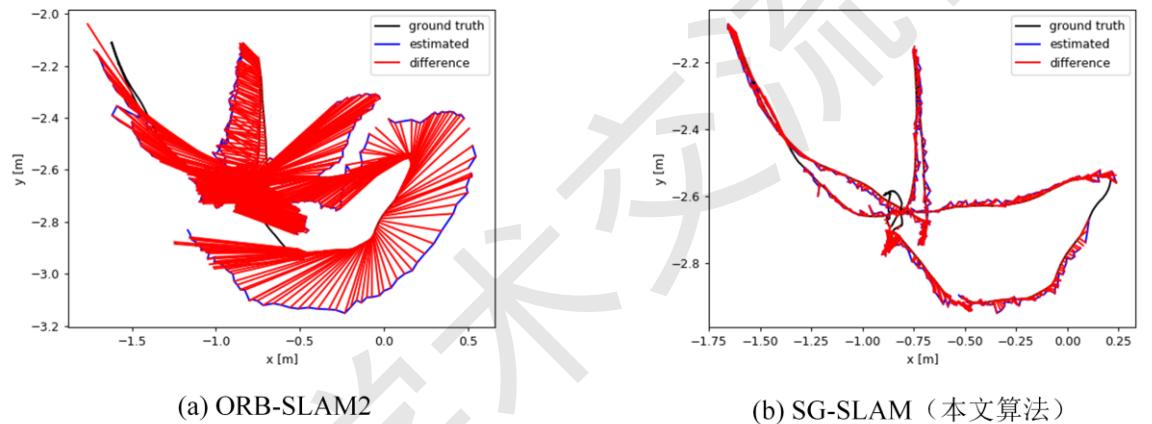


图 3-7 TUM 数据集 fr3/walking_halfsphere 序列的 ATE 结果图

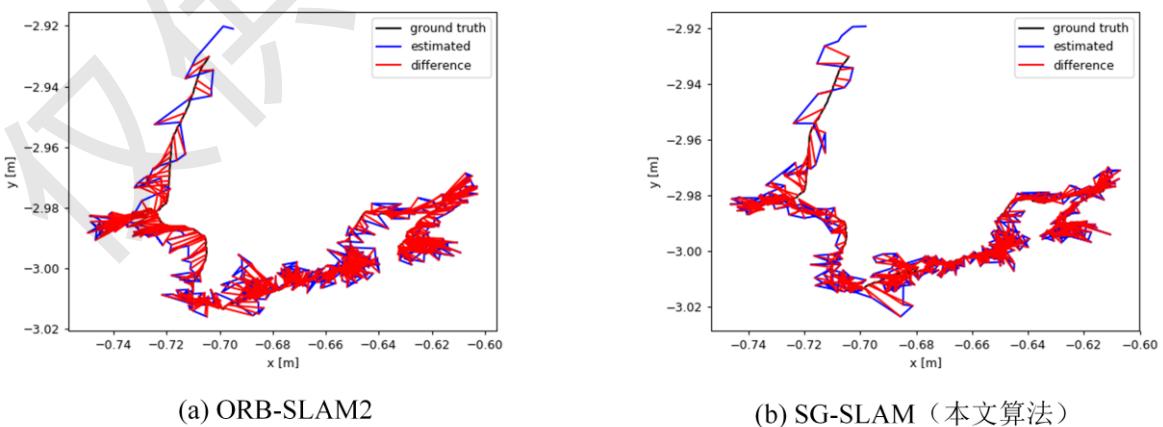


图 3-8 TUM 数据集 fr3/sitting_static 序列的 ATE 结果图

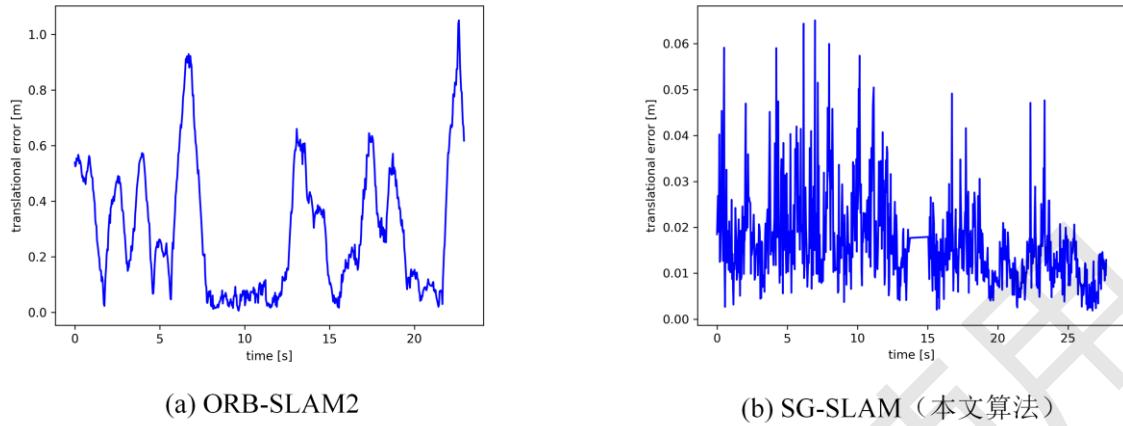


图 3-9 TUM 数据集 fr3/walking_xyz 序列的 RPE 结果图

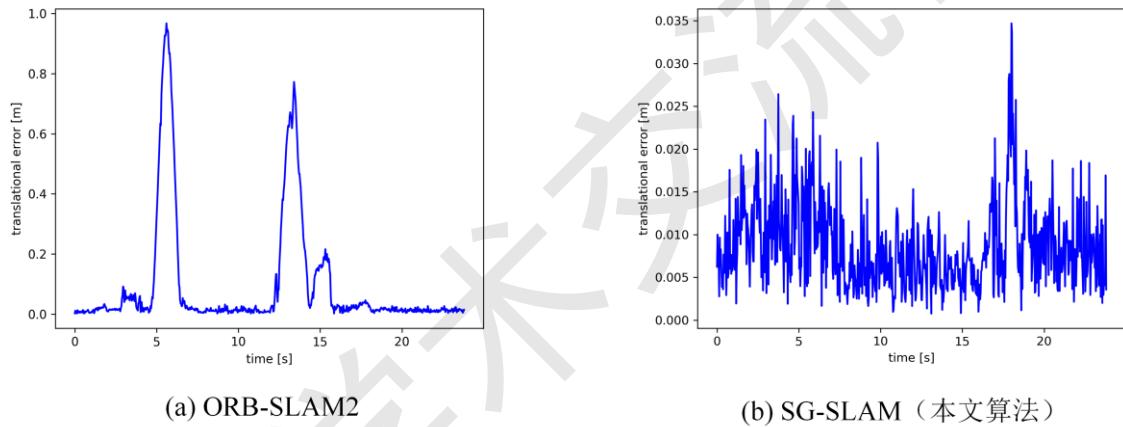


图 3-10 TUM 数据集 fr3/walking_static 序列的 RPE 结果图

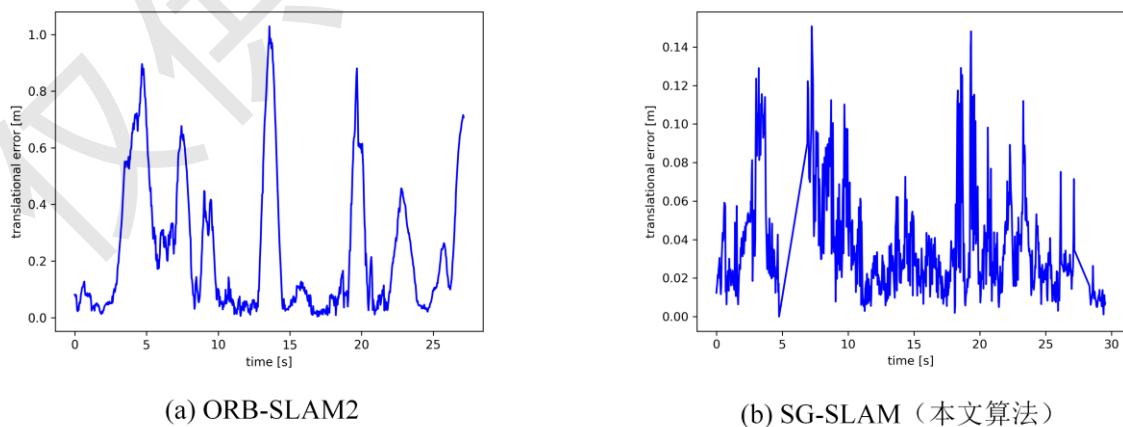


图 3-11 TUM 数据集 fr3/walking_rpy 序列的 RPE 结果图

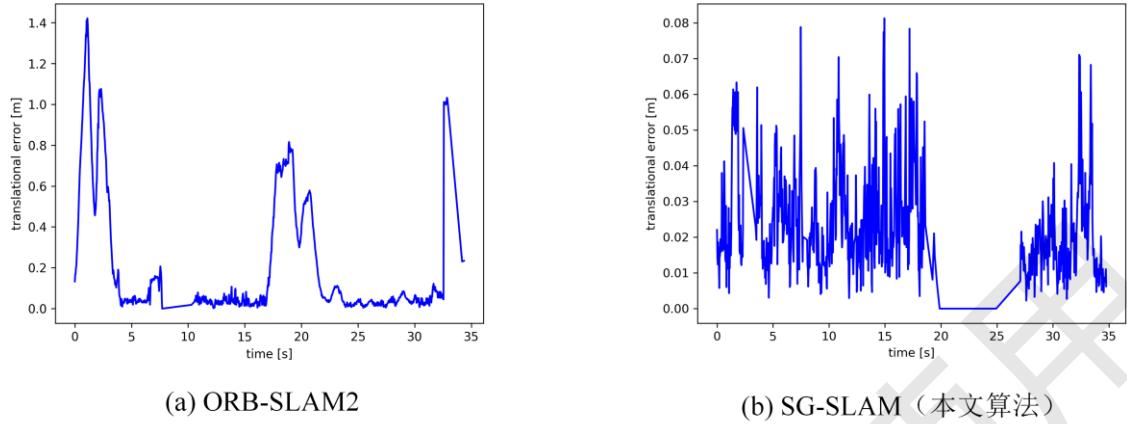


图 3-12 TUM 数据集 fr3/walking_halfsphere 序列的 RPE 结果图

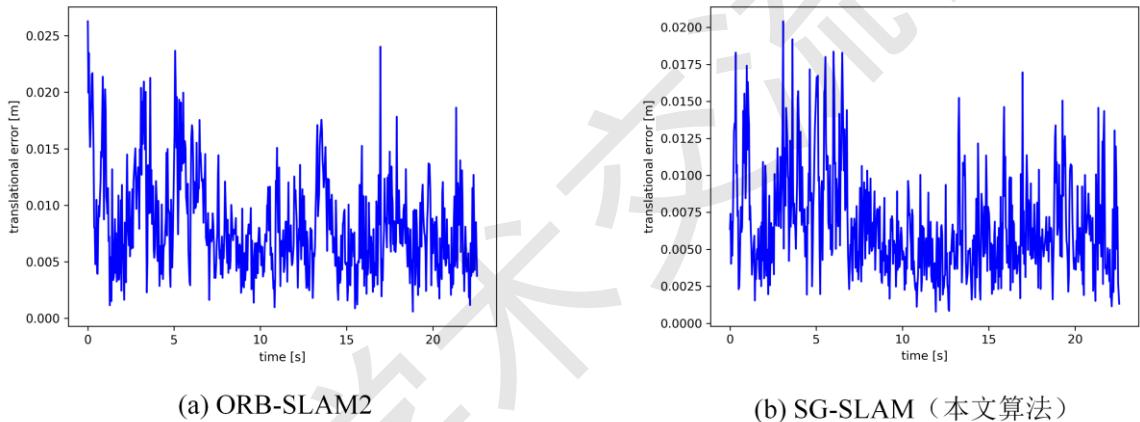


图 3-13 TUM 数据集 fr3/sitting static 序列的 RPE 结果图

由实验结果图可知，本文提出的系统 SG-SLAM 在四个高动态场景序列中的估计位姿轨迹结果相较于 ORB-SLAM2 系统而言，更加贴近拟合真实轨迹值。在低动态场景序列的实验中，由于动态对象活动的范围和幅度均较小，所以位姿轨迹估计结果的精度提升幅度较小，不如高动态序列明显。

图 3-9 至图 3-13 五张图片是在两个系统算法分别运行五个视频序列之后的相对轨迹误差 (RPE) 在平移部分的实验结果图。从图中平移误差的坐标轴刻度单位可以明显看出，本文所提出的 SG-SLAM 算法相比原始 ORB-SLAM2 系统的误差要小一到两个数量级。未经处理的 ORB-SLAM2 算法误差随着场景中人的走动幅度而跳动，而 SG-SLAM 算法虽然在波动，但是全程的误差波动保持了相对的平稳。这说明场景中人物的走动给本文所提出的系统算法的负面影响较小，系统在位姿估计方

表 3-2 SLAM 算法在 TUM 数据集的 ATE 结果（对比统计量为 RMSE、Mean）

数据集序列	ORB-SLAM2（框架）		SG-SLAM（本文）		性能提升	
	RMSE	Mean	RMSE	Mean	RMSE(%)	Mean(%)
w_xyz	0.6826	0.6086	0.0152	0.0132	97.77	97.83
w_static	0.4032	0.3690	0.0073	0.0065	98.19	98.24
w_rpy	0.5396	0.5012	0.0324	0.0264	94.00	94.73
w_half	0.4462	0.4096	0.0268	0.0232	93.99	94.34
s_static	0.0087	0.0078	0.0060	0.0053	31.03	32.05

表 3-3 SLAM 算法在 TUM 数据集的 ATE 结果（对比统计量为 Median、S.D.）

数据集序列	ORB-SLAM2（框架）		SG-SLAM（本文）		性能提升	
	Median	S.D.	Median	S.D.	Median(%)	S.D.(%)
w_xyz	0.6661	0.3091	0.0118	0.0075	98.23	97.57
w_static	0.3164	0.1627	0.0059	0.0034	98.14	97.91
w_rpy	0.4974	0.1999	0.0215	0.0187	95.68	90.65
w_half	0.3860	0.1770	0.0203	0.0134	94.74	92.43
s_static	0.0072	0.0039	0.0047	0.0029	34.72	25.64

表 3-4 SLAM 算法在 TUM 数据集的 RPE 平移部分结果（对比统计量为 RMSE、Mean）

数据集序列	ORB-SLAM2（框架）		SG-SLAM（本文）		性能提升	
	RMSE	Mean	RMSE	Mean	RMSE(%)	Mean(%)
w_xyz	0.3752	0.2944	0.0194	0.0166	94.83	94.36
w_static	0.2182	0.0950	0.0100	0.0087	95.42	90.84
w_rpy	0.3374	0.2344	0.0450	0.0366	86.66	84.39
w_half	0.3685	0.2072	0.0279	0.0238	92.43	88.51
s_static	0.0093	0.0082	0.0075	0.0066	19.35	19.51

表 3-5 SLAM 算法在 TUM 数据集的 RPE 平移部分结果（对比统计量为 Median、S.D.）

数据集序列	ORB-SLAM2（框架）		SG-SLAM（本文）		性能提升	
	Median	S.D.	Median	S.D.	Median(%)	S.D.(%)
w_xyz	0.2394	0.2326	0.0147	0.0100	93.86	95.70
w_static	0.0169	0.1965	0.0076	0.0051	55.03	97.40
w_rpy	0.1137	0.2426	0.0296	0.0262	73.97	89.20
w_half	0.0491	0.3047	0.0198	0.0146	59.67	95.21
s_static	0.0074	0.0044	0.0059	0.0035	20.27	20.45

面保持了良好的准确性和鲁棒性。绝对轨迹误差（ATE）的实验数据结果如表 3-2 和表 3-3 所示。与实验图片结果相符合的，在四个高动态场景序列中，本文所提出的 SG-SLAM 算法相比原始 ORB-SLAM2 框架在均方根误差（RMSE）、均值误差（Mean）、中值误差（Median）和标准差（S.D.）所有统计量的性能提升幅度都达到了 90%以上的数量级提升。这就从定量上证明了本文动态特征检测与剔除算法在视角中含有高度动态对象时的有效性。由于低动态序列 fr3/stting_static 场景本身动态因素较少，因此对系统位姿跟踪算法的不利影响不如高动态序列大。所以在实验数据上本文算法相比于基础框架 ORB-SLAM2 而言，绝对轨迹误差参数的均方根误差统计量上仅提升 30%左右。

表 3-6 SLAM 算法在 TUM 数据集的 RPE 旋转部分结果（对比统计量为 RMSE、Mean）

数据集序列	ORB-SLAM2（框架）		SG-SLAM（本文）		性能提升	
	RMSE	Mean	RMSE	Mean	RMSE(%)	Mean(%)
w_xyz	7.1415	5.6403	0.5040	0.4393	92.94	92.21
w_static	3.8068	1.6993	0.2676	0.2419	92.97	85.76
w_rpy	6.4220	4.5134	0.9565	0.7834	85.11	82.64
w_half	7.9219	4.4695	0.8119	0.7133	89.75	84.04
s_static	0.2899	0.2606	0.2657	0.2389	8.35	8.33

表 3-7 SLAM 算法在 TUM 数据集的 RPE 旋转部分结果（对比统计量为 Median、S.D.）

数据集序列	ORB-SLAM2（框架）		SG-SLAM（本文）		性能提升	
	Median	S.D.	Median	S.D.	Median(%)	S.D.(%)
w_xyz	4.6159	4.3804	0.3848	0.2469	91.66	94.36
w_static	0.3888	3.4065	0.2269	0.1144	41.64	96.64
w_rpy	2.2990	4.5685	0.6289	0.5487	72.64	87.99
w_half	1.2568	6.5406	0.6452	0.3878	48.66	94.07
s_static	0.2484	0.1271	0.2287	0.1163	7.93	8.50

表 3-4 至表 3-7 是 ORB-SLAM2 和 SG-SLAM 算法在相对位姿误差在平移和旋转部分的实验数据结果。与前文绝对轨迹误差的原因分析相似的，在 RPE 实验数据上本文所提出的 SG-SLAM 算法同样在四个高动态序列中性能提升明显，而在低动态序列中性能提升相对较低。而且，在实际实验中 ORB-SLAM2 甚至在跟踪过程中因为动态对象出现丢失现象，而 SG-SLAM 则在动态场景中表现出了良好的鲁棒性。

3.3.3 消融实验

本文所提出的 SG-SLAM 动态特征点检测与剔除算法是在原始 ORB-SLAM2 系统上添加改进的，本章 3.3.2 节的实验结果已经论证了该算法在动态环境下跟踪性能的整体提升效果。由 3.2 节算法原理介绍可知，SG-SLAM 动态特征剔除算法是基于几何信息和基于语义信息的算法相结合而成的融合算法。因此有必要通过实验验证这两种信息融合的必要性。

由 1.2.2 节的研究现状分析可知，如果单纯依靠极线约束的几何信息方法来剔除动态点，那么算法的经验阈值选取不当容易使得检测动态特征点出现错检或者漏检。而单纯依靠基于语义信息的方法同样存在无法正确处理先验对象范围以外的动态特征点等其他问题。因此，SG-SLAM 利用语义信息辅助几何信息的方法来进行动态特征点的检测和剔除工作，充分吸取两种方法的优点并避免缺点。

图 3-14 展示了消融实验的实际动态特征剔除效果。首先，SG-SLAM (S) 名称代表仅采用基于语义信息的方法来检测和剔除动态特征点。其次，SG-SLAM (G) 则表示其完全基于极线约束的几何方法来处理动态环境的影响。最后，SG-SLAM

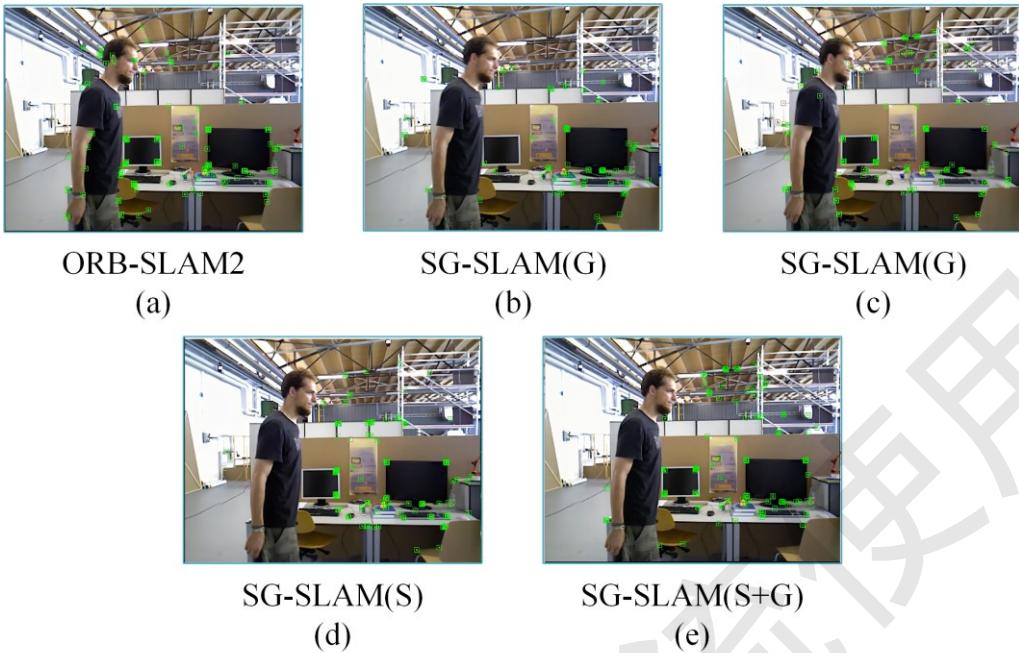


图 3-14 消融实验的动态特征剔除效果对比图。其中，图 (b) 的经验阈值为 0.2，图 (c) 的经验阈值为 1.0

表 3-8 在 TUM 数据集的消融实验 ATE 结果对比 (对比统计量为 RMSE、Mean)

数据集序列	SG-SLAM (S)		SG-SLAM (G)		SG-SLAM(S+G)	
	RMSE	Mean	RMSE	Mean	RMSE	Mean
w_xyz	0.0171	0.0144	0.1497	0.1323	0.0152	0.0132
w_static	0.0139	0.0088	0.0153	0.0099	0.0073	0.0065
w_rpy	0.0381	0.0255	0.2757	0.1830	0.0324	0.0264
w_half	0.0811	0.0766	0.0448	0.0342	0.0268	0.0232
s_static	0.0084	0.0071	0.0094	0.0084	0.0060	0.0053

表 3-9 在 TUM 数据集的消融实验 ATE 结果对比 (对比统计量为 Median、S.D.)

数据集序列	SG-SLAM (S)		SG-SLAM (G)		SG-SLAM(S+G)	
	Median	S.D.	Median	S.D.	Median	S.D.
w_xyz	0.0124	0.0092	0.1152	0.0702	0.0118	0.0075
w_static	0.0065	0.0108	0.0077	0.0118	0.0059	0.0034
w_rpy	0.0190	0.0283	0.1021	0.2061	0.0215	0.0187
w_half	0.0762	0.0268	0.0281	0.0290	0.0203	0.0134
s_static	0.0059	0.0045	0.0076	0.0043	0.0047	0.0029

(S+G) 则表示采用本文所提出的融合几何信息与语义信息两种方法的动态特征点剔除策略。这些方法的所有实验数据结果都记录在表 3-8 至表 3-9 中。

首先，图 3-14 (a) 显示了未加改动的原始 ORB-SLAM2 算法提取特征点的直观结果：其基本没有对动态区域做任何处理。其次，图 3-14 (b) 和图 3-14 (c) 显示了仅使用极线约束的几何信息方法在不同大小经验阈值下提取特征点的结果。在较低的经验阈值（如图 3-14 (b)）下，许多原本的静态特征点被错误识别为动态点遭到剔除（如图中显示器四角的特征点）；而在较高的经验阈值（如图 3-14 (c)）

下，尽管已经剔除了行走的人身体范围内的部分动态特征点，但仍然有一部分动态特征点发生了被漏检的情况。然后，图 3-14 (d) 则显示了仅使用基于语义信息的目标检测方法对特征点的直观提取结果：人体边界检测框范围内，包括显示器的两个角和椅子周围的一些特征均遭到了粗暴的剔除。最后，融合了语义信息、几何信息的 SG-SLAM 系统的动态特征点剔除结果在图 3-14 (e) 中显示。从 (e) 图中可以看出，SG-SLAM 所采取的融合算法基本剔除了行走人体范围内的所有特征点并尽可能保留了人体轮廓以外（例如显示器边角和桌椅周边）的静态特征点，剔除效果明显优于前两种仅依靠单一信息的算法。

表 3-8 和表 3-9 中的数据记录了 SG-SLAM 系统在单独基于几何信息方法、单独基于语义信息方法和两种信息相结合三种方法的消融实验结果。其中，SG-SLAM (G) 中采用的经验阈值是图 3-14 (b) 中较为严苛的低经验阈值。从表 3-8 中绝对轨迹误差 (ATE) 的各种统计量数据结果来看，使用了融合算法的 SG-SLAM (S+G) 显示出与图 3-14 直观动态点剔除效果相符合的数据结果。在几乎所有实验序列的所有误差统计量数值上，融合算法均低于其他仅依靠单一信息的算法，这就证明了本文提出的 SG-SLAM 动态特征点剔除算法在融合信息上的有效性。

3.3.4 与近期同类先进工作对比

为了进一步分析 SG-SLAM 系统在动态场景下的实际性能表现，本节使用最具代表性的绝对轨迹误差 ATE 的均方根误差 (RMSE) 和标准差 (S.D.) 两个统计量进行实验比较，其中 RMSE 可以衡量系统准确性，S.D. 则衡量鲁棒性。所有算法均是以 RGB-D 相机作为数据采集传感器。我们共选择了近年来八个同类先进工作来与本文所提 SG-SLAM 算法进行结果对比，实验结果见表 3-10 至表 3-12 中的数据，所有表格中的“-”字符表示该工作对应的论文内容中缺失该部分数据。

通过与近年来八个先进动态环境 SLAM 工作对比，本文提出的 SG-SLAM 动态特征剔除算法达到的精确度和鲁棒性在大多数实验序列下都取得了领先水平。其中，DynaSLAM 通过采用像素级别的语义分割神经网络以及新增低成本跟踪算法在个别序列的实验结果上取得了微弱的领先优势。然而代价却是其算法的实时性能存在缺陷（实验数据见表 4-2）。ORB-SLAM3 是 ORB-SLAM2 算法的下一代改进版本，但是其主要的改进创新在于视觉里程计部分加入了惯性测量单元 (IMU) 以及多地图管理。对于动态场景并没有做出针对性改进，因此在追踪性能上与 ORB-SLAM2 相

第3章 动态场景下的鲁棒视觉SLAM

表3-10 与近期先进算法在TUM数据集的ATE结果对比（对比统计量为RMSE、S.D.）

数据集序列	YOLO-SLAM ^[39]		DS-SLAM ^[37]		DynaSLAM (N+G) ^[36]	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
w_xyz	0.0146	0.0070	0.0247	0.0161	0.015	-
w_static	0.0073	0.0035	0.0081	0.0036	0.006	-
w_rpy	0.2164	0.1001	0.4442	0.2350	0.035	-
w_half	0.0283	0.0138	0.0303	0.0159	0.025	-
s_static	0.0066	0.0033	0.0065	0.0033	0.006	-

表3-11 与近期先进算法在TUM数据集的ATE结果对比（对比统计量为RMSE、S.D.）

数据集序列	YOLACT based SLAM ^[40]		RDS-SLAM ^[34]		M-removal DVO ^[30]	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
w_xyz	0.017	-	0.0571	0.0229	0.0657	0.0354
w_static	0.009	-	0.0206	0.0120	0.0334	0.0207
w_rpy	0.038	-	0.1604	0.0873	0.0729	0.0335
w_half	0.026	-	0.0807	0.0454	0.0668	0.0266
s_static	-	-	0.0084	0.0043	-	-

表3-12 与近期先进算法在TUM数据集的ATE结果对比（对比统计量为RMSE、S.D.）

数据集序列	ORB-SLAM3 ^[22]		DP-SLAM ^[60]		SG-SLAM（本文）	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
w_xyz	0.9178	0.4859	0.0141	0.0702	0.0152	0.0075
w_static	0.3614	0.1522	0.0079	0.0118	0.0073	0.0034
w_rpy	1.0197	0.5122	0.0356	0.2061	0.0324	0.0187
w_half	0.6572	0.3124	0.0254	0.0290	0.0268	0.0134
s_static	0.0090	0.0043	0.0059	0.0043	0.0060	0.0029

表3-13 与近期先进算法在Bonn数据集的ATE结果对比（对比统计量为RMSE、Mean）

数据集序列	ORB-SLAM2（框架）		YOLO-SLAM		SG-SLAM（本文）	
	RMSE	Mean	RMSE	Mean	RMSE	Mean
crowd1	0.8632	0.6284	0.033	-	0.0234	0.0185
crowd2	1.3573	1.2071	0.423	-	0.0584	0.0420
crowd3	1.0772	1.0070	0.069	-	0.0319	0.0231
moving_no_box1	0.1174	0.0935	0.027	-	0.0192	0.0174
moving_no_box2	0.1142	0.0973	0.035	-	0.0299	0.0275
person_tracking1	0.7959	0.7090	0.157	-	0.0400	0.0375
person_tracking2	1.0679	0.9590	0.037	-	0.0376	0.0343
synchronous1	1.1411	0.9884	0.014	-	0.3229	0.2665
synchronous2	1.4069	1.3201	0.007	-	0.0164	0.0105

表3-14 与近期先进算法在Bonn数据集的ATE结果对比（对比统计量为Median、S.D.）

数据集序列	ORB-SLAM2（框架）		YOLO-SLAM		SG-SLAM（本文）	
	Median	S.D.	Median	S.D.	Median	S.D.
crowd1	0.3592	0.5918	-	-	0.0161	0.0143
crowd2	1.1163	0.6207	-	-	0.0301	0.0406
crowd3	0.9733	0.3823	-	-	0.0187	0.0219
moving_no_box1	0.0785	0.0710	-	-	0.0156	0.0081
moving_no_box2	0.0955	0.0598	-	-	0.0261	0.0119
person_tracking1	0.7410	0.3617	-	-	0.0380	0.0139
person_tracking2	0.8732	0.4699	-	-	0.0312	0.0154
synchronous1	0.9179	0.5703	-	-	0.1722	0.1824
synchronous2	1.3259	0.4864	-	-	0.0073	0.0126

差不大，所以在实验数据上没有明显提升。这是本文算法采用纯视觉 ORB-SLAM2 系统做为基础框架的原因之一。Ao Li 等人提出的 DP-SLAM^[60]采用了与 DynaSLAM 类似的语义分割网络做为先验信息的获取来源，然而语义分割本身的准确度和鲁棒性也限制了其算法的稳定发挥。因此，在某些实验序列（如 fr3/walking_rpy）场景下，该算法的稳定性有所下降（S.D.统计量误差过大）。

本文所提出的 SG-SLAM 虽然在慕尼黑工业大学数据集下的多数序列存在了微弱的领先优势，但是依然存在数据集针对性过拟合的风险。所以还需要在其它数据集和场景下继续进行更多的实验，以验证算法的泛化性能。表 3-13 和表 3-14 是原始 ORB-SLAM2、近期先进算法 YOLO-SLAM 和本文所提算法 SG-SLAM 在波恩大学数据集各类场景下的绝对轨迹误差实验结果。

波恩大学数据集实验主要选择了九个具有代表性的视频序列。其中，名称为“crowd”的三个序列是三个人在房间里进行随机行走的场景；名称是“moving_no_box”的两个序列是一个人将盒子从地板移动到桌子上；名称为“person_tracking”的两个序列是指摄像机一直跟踪一个行走的人的场景。最后，名称为“synchronous”序列呈现了几个人一下又一下地向同一个方向连续跳跃的场景。为了评估我们的系统的精度性能，它主要与原来的 ORB-SLAM2 系统和目前最先进的 YOLO-SLAM 系统进行了比较。

从表 3-13 和表 3-14 的实验数据结果可以看出，本文所提 SG-SLAM 算法在绝大多数序列场景下均取得了领先的优势。只有在两个“synchronous”序列中，SG-SLAM 的性能不如 YOLO-SLAM。主要原因可能是在“synchronous”场景中人群的跳跃方向与极线方向相似，导致算法出现不同程度的退化^[24]。另外，场景中人体间断性静止、运动的特殊状态场景也使算法稳定性有所下降。

波恩大学数据集上的优异性能表现不仅再次证明了本文所提 SG-SLAM 是目前在动态场景中精确度和鲁棒性方面最先进的系统之一，而且也证明了该算法具有良好的泛化性能。

3.3.5 现实场景动态特征剔除效果

前文三组实验已经在公开数据集上验证了本文所提动态特征剔除融合算法的准确性、融合有效性、先进性以及泛化性。为了继续检验所提算法的实用价值，本节使用移动机器人系统在现实场景中测试算法效果，实验结果如图 3-15。

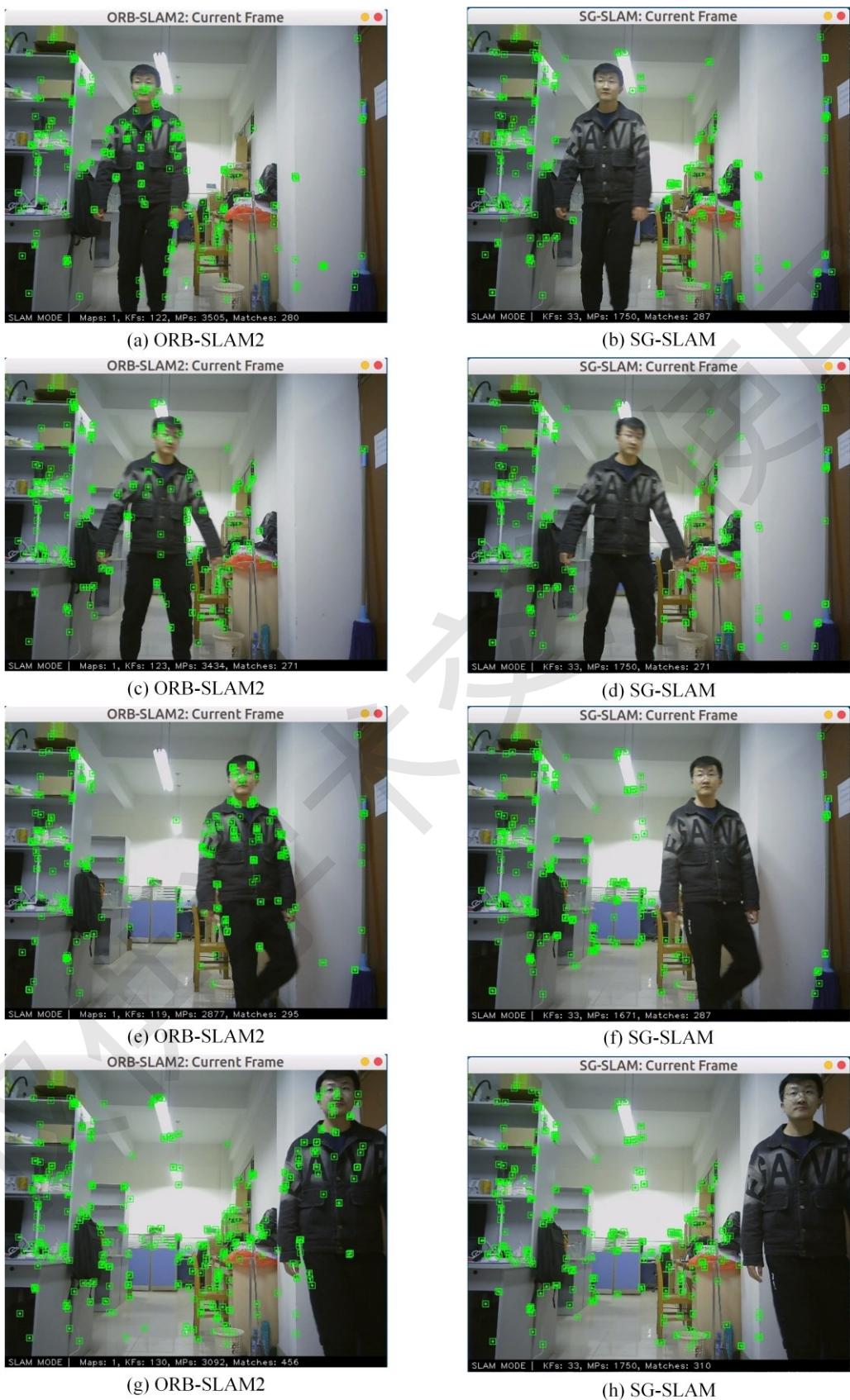


图 3-15 现实场景动态特征剔除算法实验效果对比图

如图 3-15 所示, (a) 到 (h) 八张图片分别是移动机器人运行 ORB-SLAM2 和 SG-SLAM 时随机截取的四组实验效果对比画面。从 ORB-SLAM2 实验图 3-15 (a)、3-15 (c)、3-15 (e)、3-15 (g) 中可以看出, 在场景中出现移动的行人时, 其在人体轮廓内依然提取了大量的特征点, 为后续的状态估计带来了很多错误的数据关联。而右侧 SG-SLAM 系统的实验结果图 3-15 (b)、3-15 (d)、3-15 (f)、3-15 (h) 则展示出本文所提动态特征剔除融合算法的优越效果: 精准剔除了人体范围内的所有动态特征, 尽可能保留了人体范围外的一切静态特征。

使用移动机器人在现实场景下的优异实验效果表明本文所提动态特征剔除算法具备实用价值, 确实可以使移动机器人在动态场景下提高状态估计的准确性。

3.4 本章小结

针对 SLAM 系统在动态场景下因为静态场景假设而导致的精确性和鲁棒性下降的问题, 本章在基于对极约束的几何信息算法和基于深度学习的语义信息算法基础上, 提出了融合几何信息与语义信息的动态特征点剔除算法。为了验证算法的性能, 分别进行了四组实验。一是在慕尼黑工业大学数据集上将本文所提算法和其原始框架 ORB-SLAM2 进行对比试验, 验证了该算法相比其原始框架在动态场景中的准确性和具体性能提升幅度; 二是在慕尼黑工业大学数据集上将信息融合算法 SG-SLAM (S+G) 与两个仅依靠单一信息的算法 SG-SLAM (S) 和 SG-SLAM (G) 进行消融实验对比, 证明了本文所提算法在融合信息上的有效性; 三是在慕尼黑工业大学和波恩大学两个数据集上再次将本文所提算法与近期的八个同类动态场景 SLAM 先进算法进行实验对比, 证明了本文所提 SG-SLAM 算法是目前在动态场景中跟踪性能最先进的系统之一, 同时还具有良好的泛化性; 四是在现实场景中使用移动机器人系统验证所提算法的实用性。

第 4 章 多类型直观感知语义地图构建

4.1 引言

在第 3 章中已经介绍了 SLAM 系统在动态场景下提升准确性的动态特征剔除算法，准确的定位和动态对象信息可以更好服务于语义建图。绝大多数视觉 SLAM 系统构建的地图均是用于状态估计的稀疏点云度量地图，机器人很难利用这类地图进行后续的高级任务规划。本章提出了一种语义地图构建方法，可以使系统具备较为直观的语义交互能力。首先，简要介绍 2D 语义目标检测的基本原理。其次，介绍 3D 点云滤波和分割的相关情况。然后，论述本文所提的语义对象度量地图构建算法以及语义对象的数据关联原理。再次，解释八叉树地图和三维点云重建原理以及如何消除动态因素。接下来，介绍实验所用公开数据集。由于直观语义地图目的是为后续交互任务做粗略定位，且目前尚未有统一实验基准，因此本章只进行语义建图定性实验，展示直观效果并分析。最后，对系统做实时性对比实验并进行分析。

4.2 语义地图构建算法原理

4.2.1 2D 语义目标检测

在深度学习中，神经网络通常包含两个主要功能部件：特征提取框架（backbone feature extractor）和检测头（detection head）。

特征提取框架（也称为骨干网络）是用于从输入数据中提取高级特征表示的深度卷积神经网络，通常由多个卷积层和池化层组成，其可以自主学习图像数据中的高级特征。在特征提取框架提取完高级特征后，便可以将输出结果传递给检测头。检测头是用于从特征提取框架的输出中预测目标边界框的神经网络模块，通常由卷积层和全连接层组合而成，并使用各种技术来预测目标对象的类别和位置。为了尽可能提高系统运行效率，本文所提算法选用了速度快、准确率高的多类别单次检测器 SSD^[45]为检测头。SSD 采用了先验框（prior boxes）的方式来预测边界框的位置和大小，这些先验框是在训练阶段根据样本统计信息生成的。与传统的回归方式相比，SSD 使用了多个先验框，可以更好地适应不同尺度和长宽比的物体。另外，由于本文算法运行于英伟达 Jetson AGX Xavier 计算平台，因此又使用为移动端设备进

行优化的 MobileNetV3^[56]作为 SSD 的主干特征提取器的替代品。MobileNetV3 是一种轻量级的卷积神经网络，主要用于图像分类和目标检测任务。与之前的 MobileNet 系列相比，MobileNetV3 在准确性和计算效率之间实现了更好的平衡，同时引入了一些新的轻量级特性。它的轻量级特性使得它在嵌入式设备上的应用有很大的潜力。

由图 2-7 所示的系统整体框架可知，本节采用的语义目标检测实际上就是第三章动态场景下检测动态对象的数据复用，数据复用机制使得系统的效率大大提高。当然，实际使用中可以根据具体的情况通过适当的改动程序，灵活选用其他检测器以取得准确率和速度之间的平衡。

4.2.2 3D 点云滤波和分割

在通过 RGB-D 相机采集的深度图像获取 3D 点云数据时，由于相机设备误差、工作环境（如环境中包含水、玻璃等反射材料）等因素带来的影响，3D 点云数据中可能包含噪声、杂散点等无效或错误的信息。在实际算法应用过程中，除了这些因为环境和测量误差导致的噪声信息以外，3D 点云数据中往往还存在着很多与主体点云团相距很远的离群点。这些无效信息会干扰 3D 点云数据的可靠性和准确性，使得点云处理任务（例如点云配准、目标检测、语义分割等）难以进行。因此，需要对 3D 点云进行滤波处理，目的是去除点云中的噪声和无效信息，同时尽可能保留原始点云的形状和特征。

在点云处理流程中，滤波作为预处理的第一步，往往对后续处理影响很大，只有在滤波预处理中将噪声点、杂散点、离群点等按照后续处理定制，才能够更好地进行点云分割、可视化等后续应用处理。本文采用 PCL 点云库^[61]进行点云的各项处理操作。PCL 点云库中的滤波模块提供了很多灵活实用的滤波处理算法，例如双边滤波、高斯滤波、体素（voxel grid）滤波、基于随机采样一致性滤波和统计离群点去除（statistical outlier removal）等。

由于处理大量点云数据需要消耗比较大的计算资源，为了尽可能提高算法的实时性能，所以本文所提系统 SG-SLAM 首先将深度图像转化的 3D 点云数据进行体素滤波处理。体素滤波是一种基于体素（voxel）的下采样滤波器。体素滤波的原理是将 3D 点云创建划分为规则的三维体素网格，然后使用每个体素网格内所有点云计算出每个网格的重心（即所有点云的平均值）。这样就可以利用每个体素网格的

重心来近似代表网格内的其他点云。通过这种方法，可以降低 3D 点云的密度，使点云数据更加均匀，也减少了一些点云的噪声和不必要的细节。

在对原始点云数据进行体素下采样滤波之后，点云数量已经大幅度减少，因此已经降低了数据处理难度。但此时还需要对这些点云进行统计离群点去除，以便过滤掉点云中的离群点，从而继续提高点云数据的质量。如果这些离群点不被滤除，它们可能会对后续的处理和分析造成影响。统计离群点去除的原理是基于统计学方法。它首先计算每个点与其最近邻点之间的距离，并根据距离的分布情况确定一个局部范围。然后遍历每个点，计算它局部范围内所有点的平均距离和标准差。用标准差作为阈值，从而剔除那些距离大于阈值的离群点。

为了更方便的获取到某特定对象的点云团，预处理的第二步是对每个位于目标检测框内的点云进行分割处理。点云分割是指将一个点云数据集中的点云根据不同特征进行划分，使得每个划分后的部分对应相似的特征。点云分割的过程会根据所选的模型来拟合数据集中的点，通过比较拟合结果和原始数据之间的差异来判断哪些点属于该模型，哪些点不属于该模型。

显然，现实环境中每个对象都是由距离很近的点云团聚合在一起。因此，本文使用欧式聚类（Euclidean Cluster）分割算法作为点云分割方法。其主要原理是将点云中距离较近的点聚合在一起，形成一个个点云团。其具体步骤如图 4-1 所示：

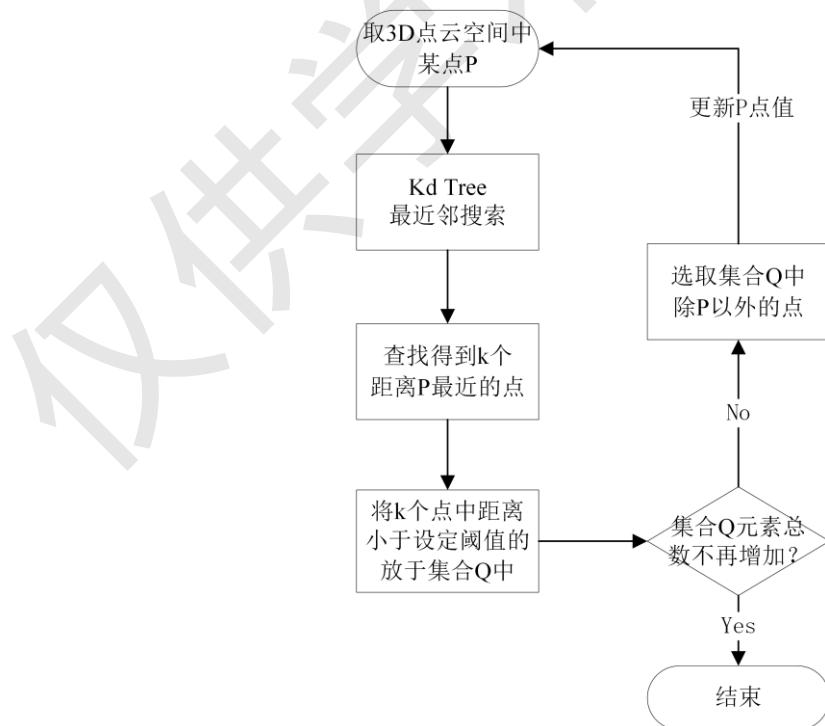


图 4-1 欧式聚类分割算法流程图

4.2.3 语义对象度量地图构建算法

机器人操作系统（ROS）提供了一套软件工具库，可以帮助开发者快速建立机器人应用。Rviz 是机器人操作系统框架中的一个可视化工具。除了跟踪线程向 ROS 系统发布机器人位姿之外，SG-SLAM 系统的语义建图线程也会发布两种数据：3D 点云和 3D 语义对象。这些数据经由 Rviz 处理后，便可显示出直观的地图界面。

为了加快系统建图效率，语义建图线程只利用关键帧中的信息进行地图构建。每当跟踪线程生成新的关键帧时，语义建图线程就立即使用该关键帧中的深度图像与该帧相机位姿生成一个局部 3D 有序点云。随后该局部 3D 点云被发布至 ROS 系统，通过 octomap_server 功能包增量式的创建一个全局八叉树地图（OctoMap）。靠发布 3D 点云构建的全局八叉树地图是一个不需要事先假设就可以对任意环境进行建模的完全 3D 模型地图，其具有可更新、灵活性强、结构紧凑等优点，所以可以很容易的服务于移动机器人的导航与避障任务。然而，这种地图不包含语义层次的信息，无法赋予机器人与语义对象之间进行高级任务规划的能力。所以，还需要一个具有带有坐标的语义对象地图，这里称之为语义对象度量地图。

关键帧的彩色图像中含有之前目标检测线程获得的 2D 语义信息（边界检测框），3D 有序点云则是获取自深度图像的与之对应的 3D 场景解释。语义建图线程通过结合 2D 语义信息与处理过后的 3D 点云信息来获取 3D 语义对象。如图 4-2 所示，语义对象度量信息获取算法主要思路如下所述：

利用目标检测算法可以从彩色图像中获得目标对象的 2D 语义信息边界检测框。但由于 2D 语义信息边界检测框包含了很多非目标对象的噪声区域，所以无法准确的分割语义对象轮廓。因此需要结合深度图像信息做更深一步的处理。

根据相机小孔成像原理，将深度图像的数据转化为对应的 3D 点云。然后获取位于 2D 语义信息边界检测框内的 3D 点云。原始的 3D 点云一般数据量大，而且含有异常的离群点。因此需要通过体素滤波算法来降低数据量，通过统计离群点去除算法来去除离群点。此时，处理后的 3D 点云来源为目标对象和一部分背景中的其他对象。由于目标对象的点云团和背景中其他对象的点云团一般是分离的，所以使用欧式聚类分割算法将 3D 点云分割为几个点云团。接下来要做的就是找到目标对象的对应点云团，即找出与目标对象相似度最大的点云团。目标对象对应点云团求解过程伪代码如下表 4-1 所示。相似度概念可以看作衡量该点云团是否是目标对象

表 4-1 目标对象点云团求解算法伪代码

算法名称: 目标对象点云团求解算法

输入: 所有候选目标点云团, *object_cloud_clusters*; 目标对象 2D 语义信息边界检测框, *rect2d*;

输出: 与目标对象相似度最高的点云团, *best_cloud_cluster*;

流程:

```

1: for each object_cloud_cluster in object_cloud_clusters do
2:   GetBackProjectedBox(object_cloud_cluster,rect3d)
3:   area1 = rect2d.area(), area2 = rect3d.area(), area0 = (rect2d & rect3d).area()
4:   overlap =  $\frac{\text{area0}}{\text{area1} + \text{area2} - \text{area0}}$ 
5:   deviate =  $\sqrt{(\text{rect2d}.x - \text{rect3d}.x)^2 + (\text{rect2d}.y - \text{rect3d}.y)^2}$ 
6:   point_cloud_nums = object_cloud_cluster.size()
7:   similarity =  $\frac{\text{overlap} \times \text{point\_cloud\_nums}}{\text{deviate}}$ 
8:   if (similarity > best_similar)
9:     best_cloud_cluster = object_cloud_cluster, best_similar = similarity
10:   else if (similarity > 2nd_best_similar)
11:     2nd_best_similar = similarity
12:   end if
13: end for
14: if (best_similar × 0.1 > 2nd_best_similar) return true
15: else return false
16: end if

```

所真正对应点云团的可能性大小量。

在以上算法中, 对于相似度的计算主要用到了以下几个假设:

- 一、点云团反投影至相机图像帧的反投影框与目标对象的 2D 语义信息边界检测框重合度越大越好;
- 二、点云团反投影至相机图像帧的反投影框的中心与目标对象的 2D 语义信息边界检测框中心距离越近越好;
- 三、因为目标对象在检测框内占比最高, 所以点云团的点云数越多越好;

最后，如果某点云团相似度远大于其他点云团，那么便认为该点云团是目标对象点云团。通过计算该点云团的包围框尺寸（长宽高）和其空间中心点坐标，便可以获得目标对象的空间信息。3D语义目标对象度量信息的整个获取流程如下图4-2所示。

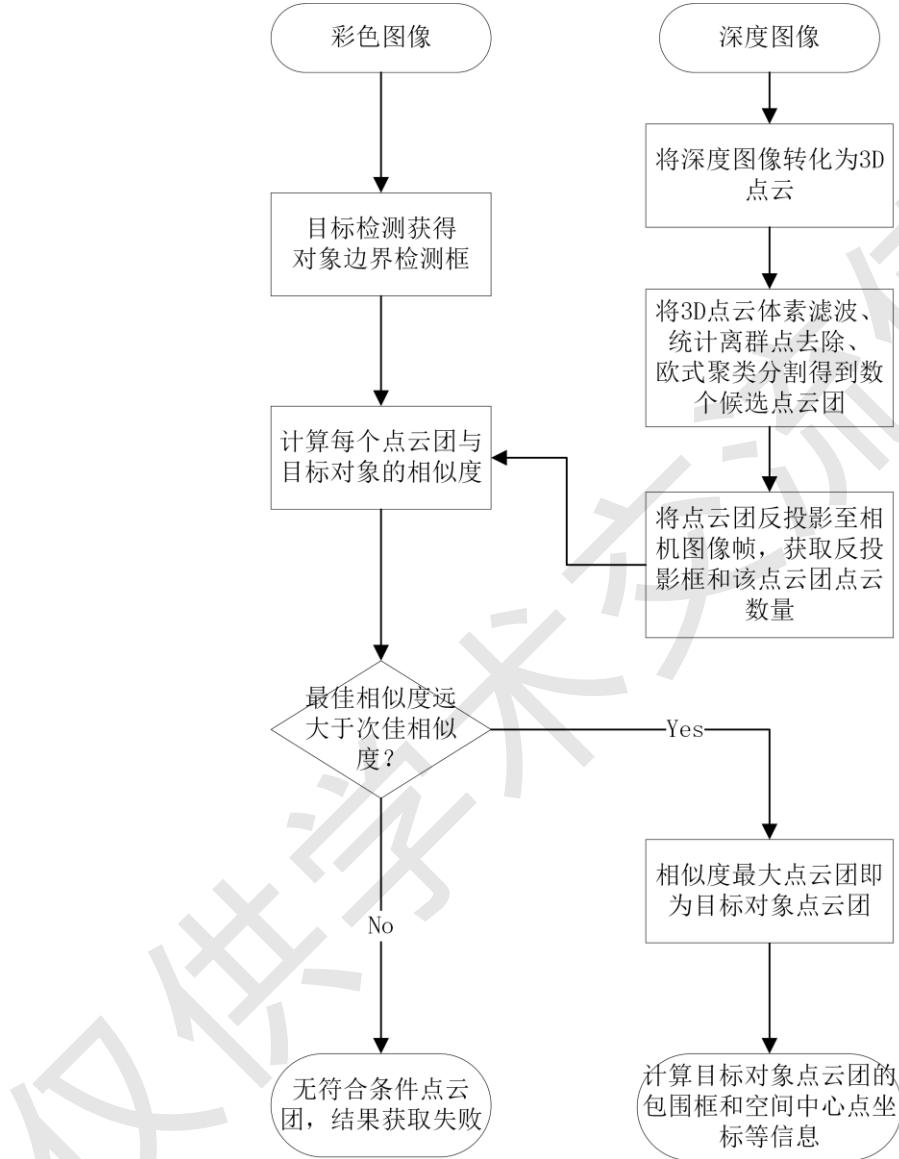


图 4-2 3D 语义对象度量信息获取流程图

语义建图线程对当前关键帧中的每个 2D 语义信息（除可以自主移动的动态对象以外，例如人、狗等）执行上述操作后，便可获得当前关键帧视角中的 3D 语义对象数据信息。在系统运行过程中，根据对象类别、中心点坐标以及对象体积尺寸等信息便可以持续的关联或更新 3D 语义对象数据库，并通过相应的接口将数据库发布至 ROS 系统做语义对象度量地图的可视化。

4.2.4 算法数据关联原理

4.2.3 节获取语义对象度量信息的一个问题是：如何判断当前语义对象是之前就已经获取过的地图原有对象，还是最近才检测到的未加入地图的新对象。本文所提 SG-SLAM 系统在这里使用 Kuhn-Munkres^[62]算法来进行语义对象之间的数据关联。

在这里，数据关联要解决的是获取的 3D 语义对象和 3D 语义数据库中的对象之间进行关联匹配的问题。到目前为止，数据关联方法涌现出了许多理论工具，其中包括但不限于：

- 一、卡尔曼滤波器和贝叶斯决策理论应用于目标跟踪发展出的概率数据关联（Probabilistic Data Association, PDA）方法，联合概率数据关联（Joint Probability Data Association, JPDA）方法等。这些方法通过在概率框架下进行观测和目标之间的关联来实现目标跟踪中的数据关联；
- 二、基于图模型理论的各种方法，如最小生成树（Minimum Spanning Tree, MST）、基于图的多假设跟踪（Graph-based MHT, GMT）、匈牙利算法（Hungarian Algorithm）等；
- 三、基于学习的方法，如基于卷积神经网络的目标检测和跟踪（Object Detection and Tracking, ODT）。

数据关联是创建语义对象数据库的难点。因为任何单一的匹配算法都有可能发生匹配错误。一旦发生错误，将会对后续的任务规划造成严重的影响。因此，必须尽可能使对象之间的数据关联准确、可靠。本文所采用的 Kuhn-Munkres 算法是基于图模型理论提出的改进版匈牙利算法，其具有诸多优点，比如：

- 一、算法能够保证最优匹配结果，即每个目标只与一个观测值匹配，并且匹配结果具有最小总成本。这种保证最优匹配的特点在实际应用中非常重要；
- 二、适用于一般的二分图匹配问题，不需要特殊的先验假设。在实际应用中，目标数量和观测值数量往往是不确定的，Kuhn-Munkres 算法的灵活性使其可以适应不同场景的数据关联任务。

Kuhn-Munkres 算法的原理可以简要理解为：给定两组节点，通过计算每组节点之间的权重，将它们连接成一个二分图，并通过算法计算出一组最优的匹配。最优的匹配意味着每个节点都可以与它的匹配节点建立最小权重的连接关系，从而实现最优的匹配效果。在具体实现中，可以将该有权二分图的最小权匹配问题转化为最

小代价问题。即，可以将每组节点之间的边权重值看作通过某种方法计算得到的代价（cost），然后将所有组节点之间的边代价构造为代价矩阵。算法的目的是找到使所有组节点之间的边代价和最小的匹配方式。实现过程如下：

设 $costf$ 为语义对象之间匹配的总代价函数，本文算法利用语义对象类别、对象中心点坐标位置、对象体积尺寸等误差信息进行对象之间的匹配。式（4-1）为总代价函数的计算表达式

$$costf = w_a f_a + w_b f_b + w_c f_c + \dots \quad (4-1)$$

其中， f_a 代表新的语义对象类别和语义对象数据库的对象类别是否相同，也称为对象类别代价； f_b 代表新的语义对象中心点坐标和语义对象数据库的对象中心点坐标距离是否相近，也称为位置距离代价； f_c 代表新的语义对象和语义对象数据库的对象体积尺寸大小是否相近，也称为尺寸误差代价。 w_a 、 w_b 和 w_c 是各自对应项的权重系数。式（4-1）的省略号表示算法的代价项可以继续无限添加。即，如果系统感知能力升级，此总代价函数子项可以继续扩充（比如新增对象颜色代价等），以提高匹配算法的精确度和鲁棒性。

对象类别代价，位置距离代价以及尺寸误差代价计算方式可分别如式（4-2）至式（4-4）所示

$$f_a = \begin{cases} 1 & (\text{type}_a = \text{type}_b) \\ 0 & (\text{type}_a \neq \text{type}_b) \end{cases} \quad (4-2)$$

$$f_b = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \quad (4-3)$$

$$f_c = length_a \times width_a \times height_a - length_b \times width_b \times height_b \quad (4-4)$$

其中， type_a 和 type_b 分别是新检测语义对象 a 和对象数据库中某对象 b 的对象类别信息； x_a, y_a, z_a 和 x_b, y_b, z_b 分别是新检测语义对象 a 和对象数据库中某对象 b 的中心点坐标值； $length_a, width_a, height_a$ 和 $length_b, width_b, height_b$ 分别是新检测语义对象 a 和对象数据库中某对象 b 的长宽高值，式（4-4）意味着它们的体积值误差。

最后，每次检测到新的语义对象组时，求解计算它们与对象数据库的代价矩阵后，然后使用 Kuhn-Munkres 算法求解其二分图的最小权重匹配。如果某对象与数据库成功匹配，则使用均值滤波算法将新旧语义对象的各项信息进行融合优化，合并为一个语义对象。如果未能匹配成功，则认为该对象为检测到的全新语义对象，将之直接放入 3D 语义对象数据库中。

4.2.5 八叉树地图构建原理

八叉树地图（OctoMap）是一种基于八叉树结构的三维环境地图表示方法^[55]，主要用于机器人导航、场景重建和物体识别等领域。

八叉树地图的基本原理是将三维空间划分为一系列小立方体（voxel），然后使用八叉树来递归地对其进行细分，直到达到预设的分辨率或满足一定的停止条件。在八叉树的数据结构中，每个节点有八个子节点，代表将其分成八个等分的小立方体区域（如图 4-3 所示）。每个节点代表一个立方体区域，其包含有该区域是否空闲以及该区域的颜色属性等信息。当某个节点的所有子节点具有相同的状态（都被占据或者都为空）时，可以将该节点及其子节点合并为一个单一节点，这就达到了节约空间和提高查询效率的目的。

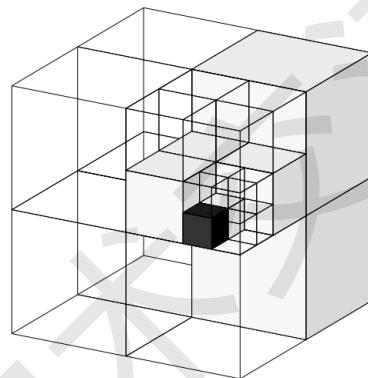


图 4-3 八叉树存储示意图，白色相当于该节点空闲，而黑色相当于该节点被占据

实际上，八叉树地图选择了使用概率形式来描述区域是否被占据的信息。具体的，在给定传感器观测值 $z_{1:t}$ 的基础上，一个节点 n 被占据的概率 $P(n | z_{1:t})$ 可以由式 (4-5) 进行估计

$$P(n | z_{1:t}) = \left[1 + \frac{1 - P(n | z_t)}{P(n | z_t)} \cdot \frac{1 - P(n | z_{1:t-1})}{P(n | z_{1:t-1})} \cdot \frac{P(n)}{1 - P(n)} \right]^{-1} \quad (4-5)$$

假设节点先验占据概率为均匀分布（即 $P(n) = 0.5$ ），然后使用概率对数值进行表示，则有某节点的概率对数值（Log-odds）为

$$L(n | z_{1:t}) = L(n | z_{1:t-1}) + L(n | z_t) \quad (4-6)$$

即每次新的观测值到来时，只需要直接加在原来的节点概率对数值上，便可完成概率的更新。利用这个原理，可以很好的对地图进行增量化更新，从而具备处理含运动物体地图的能力。而且通过调节八叉树节点数据结构的深度，可以控制地图

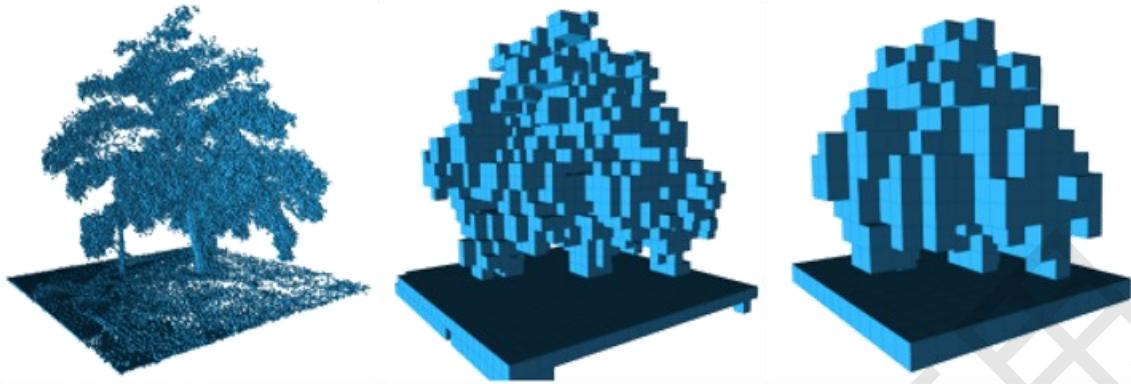


图 4-4 分辨率分别为 0.08 米、0.64 米和 1.28 米的八叉树地图
的分辨率大小，从而根据不同设备的算力实现不同的地图效果。不同分辨率的八叉
树地图构建效果如图 4-4 所示。

本文所提 SG-SLAM 系统中的全局八叉树地图（OctoMap）构建由 ROS 社区中
的第三方软件 `octomap_server` 建图服务功能包实现。将局部 3D 点云及对应关键帧
的位姿信息通过消息机制发布后，设置好相关参数的建图服务功能包将会对其进行
处理，然后便可增量生成对应的全局静态八叉树地图。使用 Rviz 工具订阅对应的消
息话题即可查看该地图。

4.2.6 全局三维点云重建原理

全局三维点云重建是指通过多个视角的图像扫描环境数据，然后将它们融合在
一起以重建场景的三维点云模型，用于机器人导航、虚拟现实、增强现实等应用。
本文设计的 SG-SLAM 系统实现了一个简单的全局 3D 点云重建算法。实现步骤简
要表述如下：

一、将每个关键帧的深度图像转换为局部 3D 点云后，根据跟踪线程估计的关
键帧位姿变换，将局部 3D 点云的所有点都变换至世界坐标系下，从而获
得该关键帧视角下的全局 3D 点云。数学表述如式（4-7）所示

$$P_{\text{world}}^i = T_{\text{wc}}^i \times P_{\text{camera}}^i \quad (4-7)$$

其中， P_{camera}^i 表示第 i 个关键帧所在的相机坐标系下的局部 3D 点云位置坐
标， P_{world}^i 表示由 P_{camera}^i 转换完成的世界坐标系下全局 3D 点云位置坐标。
 T_{wc}^i 是第 i 个关键帧对应的从相机坐标系到世界坐标系的位姿变换矩阵，维
度为 4×4 ，属于李群 $SE(3)$ 。

二、对变换后的世界系点云团进行统计离群点去除和体素滤波，提高点云质量。

三、将第二步从单个关键帧获得的世界系点云团添加到全局 3D 点云变量。

四、循环前三步，增量式构建全局 3D 点云。设置一个关键帧处理数量阈值，

一旦前三步循环次数超过该阈值，便对此时的全局 3D 点云进行一次统计离群点去除和体素滤波处理，提高整体点云质量。处理完成后重置循环次数变量。

进行第四步操作是因为对全局 3D 点云进行噪声去除和滤波十分消耗算力，很难对数据量庞大的全局 3D 点云进行实时滤波处理。每当第四步处理完成后，便可将全局 3D 点云通过消息机制发布，使用 Rviz 工具订阅对应的消息话题即可查看该地图。

4.2.7 消除动态因素对建图的影响

环境中的动态对象不仅严重影响机器人状态估计的精度，还会影响语义地图构建的效果。如果不对环境中的动态对象加以处理，那么构建的地图中可能会出现大量的噪声。这就要求在地图创建的过程中忽略动态对象部分，尽可能生成一个全局一致的静态地图。

得益于第 3 章对动态场景处理时获取到的 2D 动态对象语义信息，在语义地图构建的过程中可以方便的消除动态因素。消除动态对象影响的原理十分简单：如果已知环境中会存在动态对象，那么只需要在通过深度图像获取局部 3D 点云数据时，直接把位于 2D 动态对象边界检测框内的点云滤除即可。由于 2D 动态对象边界检测框早在动态特征剔除阶段就已经获取，属于数据复用。因此该操作并不需要复杂计算，不会对系统的实时性产生消极影响。在消除动态对象对建图的消极影响后，动态场景下构建的地图相比之前有了较大的提升，第 4.3.6 节将进行对比实验以展示消除动态对象后的建图效果。

4.3 实验验证及分析

4.3.1 公共数据集介绍

OpenLORIS-Scene 是通过机器人在真实场景中记录数据，然后使用动作捕捉系统获取真实轨迹的一个数据集^[63]。该数据集旨在帮助评估 SLAM 和场景理解算法在

现实部署中的成熟度。为了捕获到由人类活动、昼夜交替和其他各种情况引起的场景变化，数据集中的每个场景都提供了多条不同的运行轨迹。这些因素在解决机器人长期自主性等问题上至关重要。

该数据集数据由轮式移动机器人不高于人类步行的速度收集。机器人上的主要传感器包括一个 RealSense D435i 相机和一个 RealSense T265 相机，它们都安装在距离地面大约 1 米的固定高度。RealSense D435i 配备了全局快门感应器以及更大的相机镜头，其传感器可以 30 帧每秒的速度输出分辨率达 1280×720 的图像。两个摄像头都提供 IMU 测量以及与对应图像的硬件同步。另外，本章地图构建实验还使用了慕尼黑工业大学 RGB-D 数据集，相关内容可见本文 3.3.1 节。

4.3.2 语义对象度量地图构建

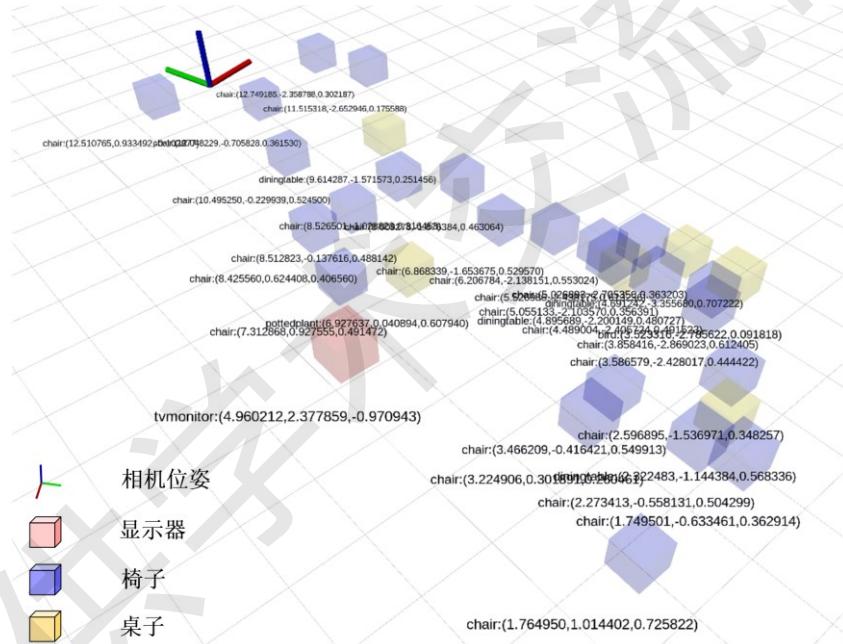


图 4-5 OpenLORIS-Scene 数据集 cafe1-2 序列语义对象度量地图

图 4-5 是 OpenLORIS-Scene 数据集的 cafe1-2 序列中构建的语义对象度量地图。图 4-6 是 SG-SLAM 在 TUM RGB-D 数据集的 fr3/walking_xyz 序列中的语义对象度量地图。从图中可以看出，语义对象度量地图给出了相机位姿和所有场景中采集到的语义对象及其位置坐标。地图中显示的对象的坐标是从运行 SLAM 系统的原点（将首张图像帧的坐标系视为原点）转换而来，坐标轴的顺序是 x, y, z 。

全局八叉树地图赋予了机器人避障和规划可行路线的能力，语义对象度量地图使得机器人拥有可以导航至语义对象目标附近的方位信息。这两种地图使移动机器

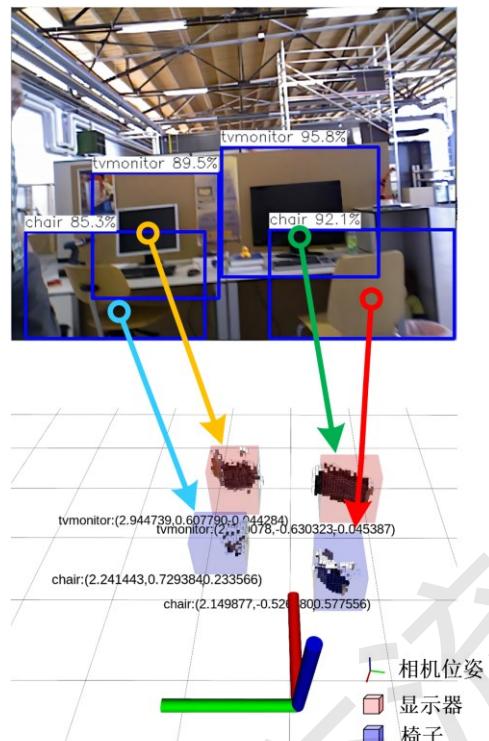


图 4-6 TUM RGB-D 数据集 fr3/walking_xyz 序列语义对象度量地图

人可以在更高层次上理解周围场景，并执行较为高级的人机交互任务。该地图场景对应的全局八叉树地图见下节实验结果（如图 4-7）。

4.3.3 全局八叉树地图构建

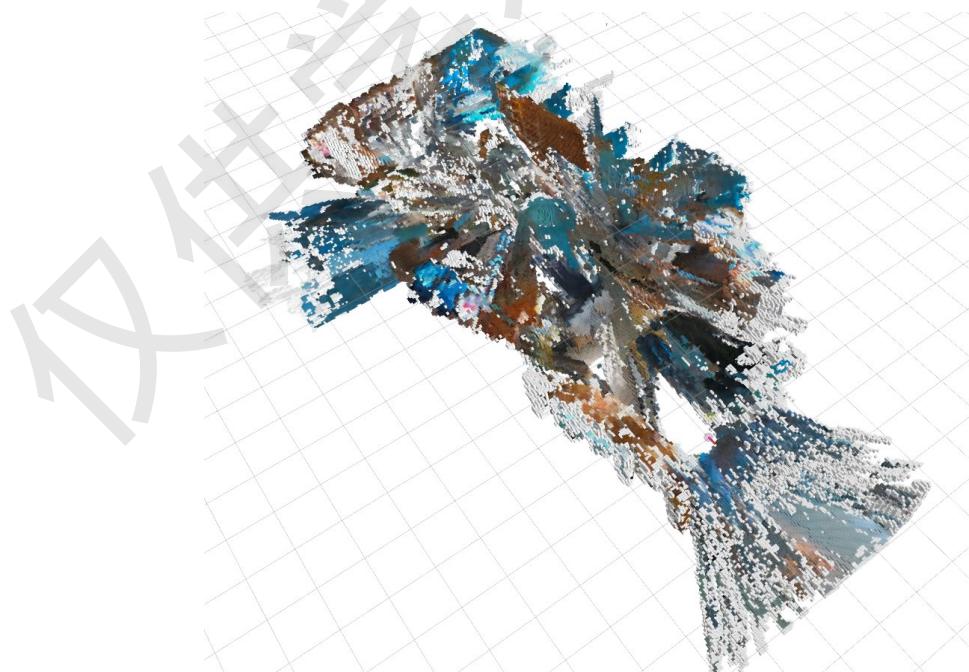
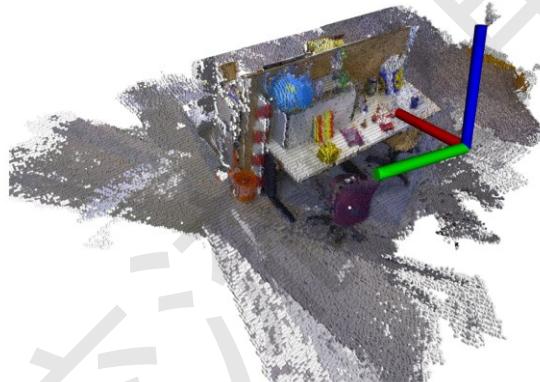


图 4-7 OpenLORIS-Scene 数据集 cafe1-2 序列 0.05 米分辨率全局八叉树地图

图 4-7 显示了本文所提 SG-SLAM 系统在 OpenLORIS-Scene 数据集 cafe1-2 序列中构建的分辨率为 0.05 米的全局八叉树地图。该序列是一个室内咖啡厅，里面有吧台、桌椅、人群、广告牌和窗户等物品。地图左上角和右上角蓝色部分和右下角灰色部分都是大块的玻璃门窗场景，虽然系统已经对 3D 点云做了体素滤波和统计离群点去除的处理，但是依然不可避免的出现了噪声。对于其他位置，八叉树地图对原始场景做了很好的还原。



(a) TUM 数据集 fr3/long office household 图像帧



(b) TUM 数据集 fr3/long office household 八叉树地图

图 4-8 TUM 数据集 fr3/long_office_household 序列 0.01 米分辨率八叉树地图构建效果

图 4-8 (a) 是慕尼黑工业大学 RGB-D 数据集中 fr3/long_office_household 序列的某一图像帧，(b) 是由该序列构建的全局八叉树地图，相机位姿由红绿蓝（分别代表 x, y, z 方向）三色坐标轴表示。在全局八叉树建图时，分辨率更高的 fr3/long office household 序列相比于 cafe1-2 序列保留了更多细节。但代价是算力消耗过大，建图速度较慢。后续进行导航与路径规划时，可以根据高度、点云分割等信息滤除地面体素格。

4.3.5 三维点云重建地图构建

与上节相同，图 4-9 是由慕尼黑工业大学 RGB-D 数据集中 fr3/long_office_household 序列构建的全局三维点云重建地图，相机位姿由红绿蓝（分别代表 x, y, z 方向）三色坐标轴表示。其中，图 4-9 (a) 为该全局三维点云重建模型正面画面，图 4-9 (b) 为模型背面画面，该三维重建模型可以用于机器人导航、虚拟现实、增强现实等应用。由于全局三维重建时处理的点云数量巨大，因此该应用十分消耗算力，所以不建议在规模庞大的场景中使用该功能。

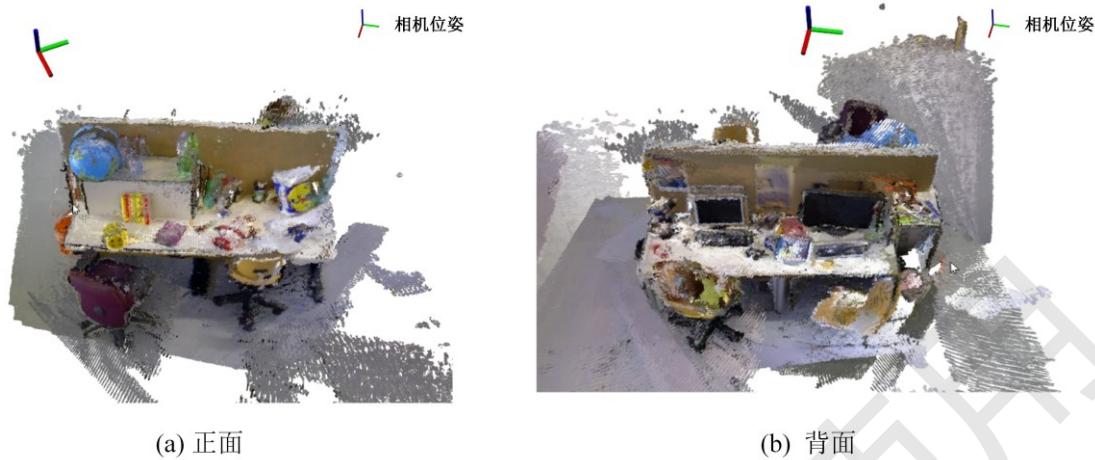


图 4-9 TUM 数据集 fr3/long_office_household 序列全局三维点云重建地图效果

4.3.6 消除动态因素对比实验

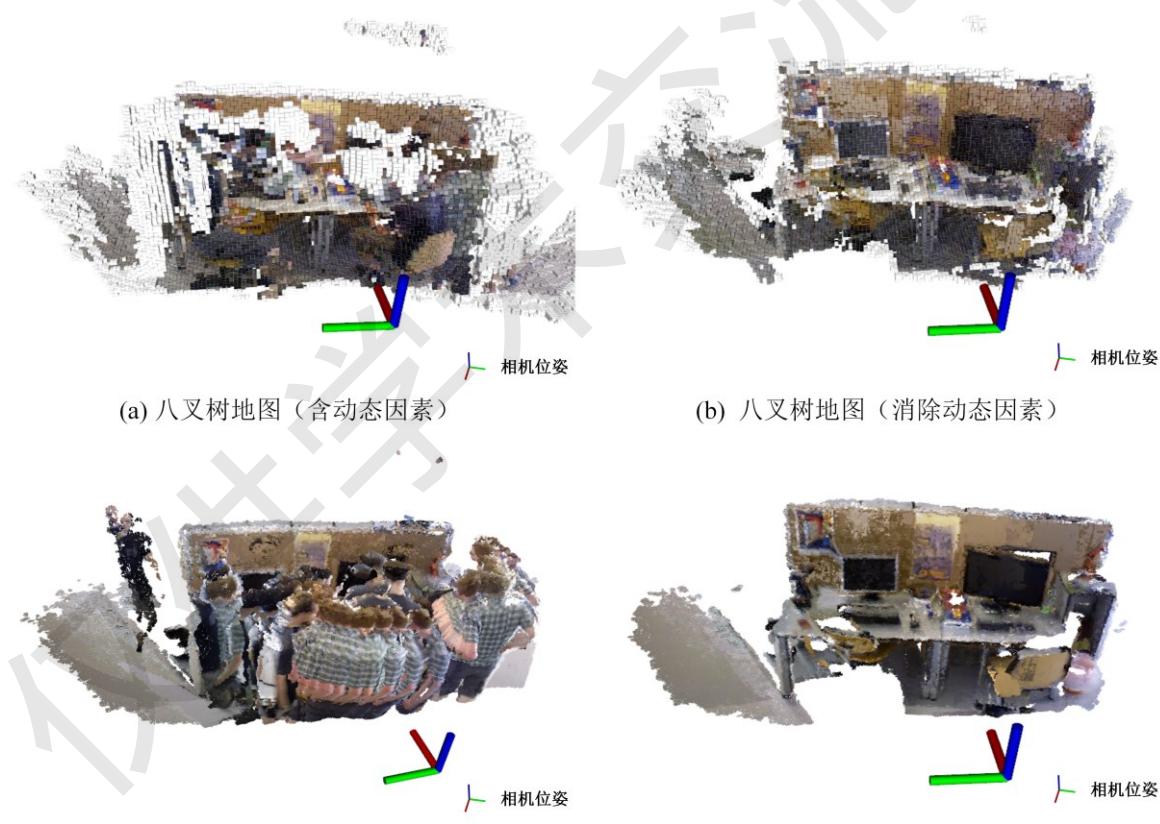


图 4-10 TUM RGB-D 数据集 fr3/walking_xyz 序列地图消除动态因素对比实验结果

图 4-10 中的所有地图均是在慕尼黑工业大学 RGB-D 数据集 fr3/walking_xyz 动态序列下构建。一方面，从图 4-10 (a) 和 (c) 中可以看出，环境中动态对象的存

在对八叉树地图和三维点云重建地图的构建产生了极其不利的影响。八叉树地图由于其概率更新原理的缘故，噪声相对较少。全局三维点云重建地图夹杂了大量的动态因素（此处为移动的人群），严重破坏了模型的重建效果。另一方面，如图 4-10 (b) 和 (d) 所示，使用 2D 动态对象边界检测框消除动态因素的地图基本没有噪声存在，效果良好。这说明，在动态场景中消除动态因素对提升建图效果很有帮助。

4.3.7 现实场景语义地图构建

多个公开数据集序列下的语义建图效果如上文实验所见，为进一步验证本文所提语义对象建图算法的泛化性、有效性，本节实验在现实场景中进行语义建图效果测试。实验用硬件设备、软件系统已在第 2 章中进行介绍。实验结果见下图 4-11。

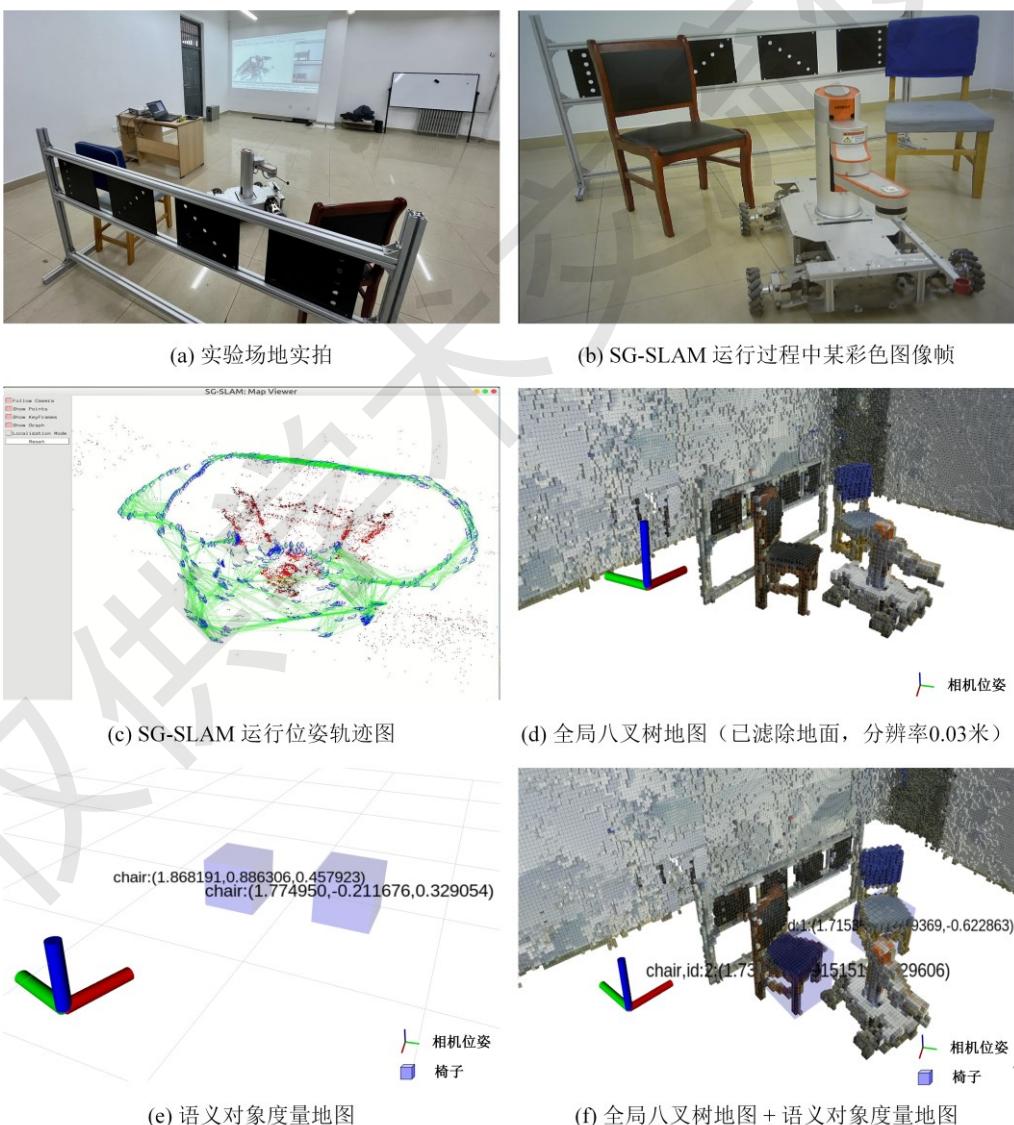


图 4-11 现实场景中 SG-SLAM 语义建图实验结果

如图 4-11 (a) 和 (b) 所示, 现实场景实验场地中包含挡板栅栏一个、棕黑色椅子一把、青蓝色椅子一把、带机械臂移动机器车一辆。辅助实验设备有笔记本电脑、投影仪(实时查看建图效果)等。在 SG-SLAM 系统启动后, 实验平台持续围绕该场地移动两圈以进行充足的数据采集, 图 4-11 (c) 展示了 RGB-D 相机的实验位姿轨迹。为方便后续导航、路径规划等任务的进行, 语义建图算法已将全局八叉树地图的地面占据体素格进行了滤除, 因此图 4-11 (d) 中仅剩地面上方的物体占据体素格。

图 4-11 (e) 展示了本文所提的语义对象度量地图构建效果。图中的蓝色包围框旁的两组数字便是两个椅子相对于世界坐标系(也即系统初始化时的首帧坐标系)的位置坐标。根据相机位姿坐标和语义对象坐标, 便可求出移动机器人和某语义对象的相对位置向量。然后, 再根据已经滤除地面的全局八叉树地图(或者其投影的 2D 栅格地图)便可求解机器人前往目标的全局规划路径。图 4-11 (f) 便是本文所提语义地图构建算法的最终效果图。可以看到语义对象度量地图的对象包围框与全局八叉树地图的物体基本重合, 精度可以满足导航至目标对象附近的任务需求。

4.3.8 系统实时性分析

SLAM 作为移动机器人状态估计的基础功能之一, 只有尽可能的保持高实时性, 才能满足其他高级应用任务的需求。SG-SLAM 系统的算法时间分析实验如表 4-2 所示。由于各算法硬件平台的不同, 所以实时性分析实验数据结果难以直接横向比较。但是, 该实验结果依然具备相当的参考价值, 原因陈述如下:

首先是本文所提 SG-SLAM 系统的实验主要运行在 Nvidia Jetson AGX Xavier 计算平台。其作为嵌入式移动计算平台, 硬件设备算力性能低于其他搭载对比算法的桌面级计算平台(例如 Intel i7 CPU、Nvidia RTX2080)。在此情况下, SG-SLAM 系统的每帧平均处理耗时仍然低于其他同类工作, 这足以说明本文所提算法的速度具有优势。而且, 为了测试本文所提 SG-SLAM 系统的泛化性能, 从而排除特定平台的针对性优化因素, 又将 SG-SLAM 移植到笔记本电脑平台(硬件配置为 AMD Ryzen 7 4800H, Nvidia GTX 1650)继续进行测试。在新平台的实验结果再次表明: 本文所提算法的每帧平均处理时间, 相比原始框架 ORB-SLAM2 增加耗时低于 10 毫秒, 可以满足移动机器人的实时性要求。

表 4-2 SG-SLAM 算法实时性能分析

系统名称	平均每帧处理时间 (毫秒)	硬件平台
ORB-SLAM2 ^[15] (框架)	59.26	Nvidia Jetson AGX Xavier
SG-SLAM (本文)	65.71	Nvidia Jetson AGX Xavier
ORB-SLAM2 ^[15] (框架)	30.04	AMD Ryzen 7 4800H, Nvidia GTX 1650
SG-SLAM (本文)	39.51	AMD Ryzen 7 4800H, Nvidia GTX 1650
YOLO-SLAM ^[39]	696.09	Intel Core i5-4228U CPU
DS-SLAM ^[37]	59.40	Intel i7 CPU, P4000 GPU
DynaSLAM ^[36]	192.00 (至少)	Nvidia Tesla M40 GPU
YOLACT based SLAM ^[40]	58.80	i7-9700K CPU, Nvidia RTX 2080
RDS-SLAM ^[34]	57.50	Nvidia RTX 2080Ti GPU

本文设计的 SG-SLAM 系统相比同类工作之所以实时性能优秀，主要有以下几个要点：

- 一、基于多线程的编程机制提升了系统的整体运行效率；
- 二、基于目标检测方法（而非耗时的语义分割），降低了多线程中阻塞模型的等待时间；
- 三、语义建图线程中，基于关键帧（而非普通帧）的机制在不明显影响效果的前提下提高了语义建图效率；
- 四、目标检测线程中的 2D 语义信息和语义建图线程中的 3D 点云信息的多次数据复用机制提升了系统效率。
- 五、使用恰当合理的数据结构，使得数据处理的流程更加迅速。

另外，专门为嵌入式平台的优化 NCNN 神经网络前向推理框架和支持 CUDA 加速的英伟达 Jetson AGX Xavier 移动计算平台等因素也是系统实时性能良好的重要原因。

4.4 本章小结

本章针对移动机器人直观的语义交互问题，提出了以 2D 语义目标检测和 3D 点云处理相结合获取 3D 语义对象的语义地图构建算法。首先，在介绍目标检测、点云滤波和分割基本原理的基础上，阐述语义对象度量地图的数据关联和构建的算法

原理和流程。然后，介绍八叉树地图和全局三维点云重建的构建原理和实现方式，并介绍了建图时消除动态对象影响的原理。接下来，在两个公开数据集和现实具体场景中对上述语义地图构建算法进行测试。实验结果表明，本文所提建图方法可以对室内环境进行语义对象度量地图、八叉树地图和全局三维点云地图的基本构建，赋予了移动机器人从语义层级理解周围环境的基本能力。最后，对本文所提 SG-SLAM 系统和其他同类工作进行实时性能对比分析实验。实验结果表明，SG-SLAM 系统与原始框架 ORB-SLAM2 相比单帧新增耗时较少，领先于其他相对比的动态场景系统算法，可以满足移动机器人的实时性要求。

本章所提地图创建完成后，可通过对象序列化技术以及 octomap 等软件库接口存储于硬盘。其他用户使用本系统时，可加载已有的地图文件，然后以纯定位模式来运行使用。

结 论

同时定位与地图构建（SLAM）是移动机器人在未知环境下进行状态估计的重要研究方向。为使机器人在复杂的非结构化环境下具备更加准确的状态估计、直观的语义交互能力，本文对动态场景下的视觉 SLAM 和语义地图构建等算法进行了深入的研究和探索。本论文主要包含三个研究内容，分别是：移动机器人系统设计研究、针对动态场景的视觉 SLAM 算法研究以及多种类型的室内直观感知语义地图构建方法研究。主要取得以下几项研究成果：

(1) 建立了一个软硬件相结合的移动机器人系统平台。在硬件设备上，针对机器人移动上自由度、计算上能耗比等问题，给出了麦克纳姆轮式移动平台、英伟达 Jetson AGX Xavier 计算平台和深度相机传感器的组合设计方案；在软硬件协作上，针对系统灵活性、扩展性等问题，使用了 ROS 系统；在软件算法上，详细分析了视觉 SLAM 系统的基本原理和模块作用。针对软件运行上实时性、稳定性等问题，设计了具有多项工程技术创新的实时 SG-SLAM 系统。相比原始框架 ORB-SLAM2，SG-SLAM 系统平均每帧新增处理耗时低于 10 毫秒，实时性领先于多数同类系统。

(2) 提出了一种融合几何信息和语义信息的动态特征剔除算法。算法思想是通过目标检测线程获取先验动态对象的语义信息，然后根据语义信息自适应调整基于极线约束的几何信息算法的经验阈值。根据 TUM、Bonn RGB-D 动态场景数据集以及现实场景中的多项实验数据结果表明：相比原始框架，所提算法在高动态场景序列中绝对轨迹误差的 RMSE 统计量和 S.D. 统计量上分别至少提升 93% 和 90%。在绝大多数场景中，所提算法的精确度和稳定性都领先于其他同类算法。

(3) 提出了一种直观语义地图构建方法。其流程是先通过 RGB-D 相机获取关键帧的 3D 点云，然后对该 3D 点云进行体素滤波、统计离群点去除和欧式聚类分割处理。接下来，一方面是将处理过后的点云转化为全局八叉树地图。另一方面是通过点云信息与目标检测获取的语义信息联合获取语义对象及其位姿，在 Kuhn-Munkres 算法的数据关联基础上构建 3D 语义对象度量地图。根据 TUM RGB-D、OpenLORIS-Scene 数据集以及现实场景下的多项实验测试表明：所提方法可以实时有效的生成具有对象坐标、尺寸信息的语义对象度量地图、全局八叉树地图和三维点云重建地图。

结 论

在对本文系统的多次测试中发现，其仍然存在一些问题需要在未来进行改进：

- (1) 动态对象沿极线方向移动产生的退化现象会造成动态特征剔除算法在原理上短暂失效。可以考虑增加其他约束项来纠正该退化现象。
- (2) 对于动态特征剔除算法中极线约束部分而言，借助光流法进行的基础矩阵求解在动态区域过大时会失效，导致基础矩阵结果不可信。未来可以考虑添加惯性测量单元（IMU）来辅助求解基础矩阵，使系统升级为视觉惯性 SLAM 系统。
- (3) RGB-D 相机因测量范围、反射性材料等各种因素采集错误的深度信息，导致八叉树地图出现噪声的问题。

参考文献

- [1] 习近平. 不断做强做优做大我国数字经济[J]. 求是,2022(2): 4-8.
- [2] 中国电子学会 . 中国机器人产业发展报告 [R/OL]. (2022-8-18)[2023-3-17].
<http://lib.ia.ac.cn/news/newsdetail/68443>
- [3] Barfoot T D. State estimation for robotics[M]. Cambridge University Press, 2017: 1-1
- [4] Kostavelis I, Gasteratos A. Semantic mapping for mobile robotics tasks: A survey[J]. Robotics and Autonomous Systems, 2015, 66: 86-103.
- [5] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I[J]. IEEE robotics & automation magazine, 2006, 13(2): 99-110.
- [6] Durrant-Whyte H F. Uncertain geometry in robotics[J]. IEEE Journal on Robotics and Automation, 1988, 4(1): 23-31.
- [7] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. The international journal of Robotics Research, 1986, 5(4): 56-68.
- [8] Smith R, Self M, Cheeseman P. Estimating uncertain spatial relationships in robotics[J]. Autonomous robot vehicles, 1990: 167-193.
- [9] Durrant-Whyte H, Rye D, Nebot E. Localization of autonomous guided vehicles[C]//Robotics Research: The Seventh International Symposium. Springer London, 1996: 613-625.
- [10] Thrun S, Burgard W, Fox D. A probabilistic approach to concurrent mapping and localization for mobile robots[J]. Autonomous Robots, 1998, 5: 253-271.
- [11] Cadena C, Carlone L, Carrillo H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age[J]. IEEE Transactions on robotics, 2016, 32(6): 1309-1332.
- [12] Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (SLAM): Part II[J]. IEEE robotics & automation magazine, 2006, 13(3): 108-117.
- [13] Dissanayake G, Huang S, Wang Z, et al. A review of recent developments in simultaneous localization and mapping[C]//2011 6th International Conference on Industrial and Information Systems. IEEE, 2011: 477-482.
- [14] Forster C, Zhang Z, Gassner M, et al. SVO: Semidirect visual odometry for monocular and

参考文献

- multicamera systems[J]. IEEE Transactions on Robotics, 2016, 33(2): 249-265.
- [15] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgbd cameras[J]. IEEE transactions on robotics, 2017, 33(5): 1255-1262.
- [16] Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]//Proceedings 2007 IEEE international conference on robotics and automation. IEEE, 2007: 3565-3572.
- [17] Chaney K. Monocular MSCKF[DB/OL]. (2018-4-16)[2023-3-17]. https://github.com/daniilidis-group/msckf_mono
- [18] 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.
- [19] Von Stumberg L, Usenko V, Cremers D. Direct sparse visual-inertial odometry using dynamic marginalization[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 2510-2517.
- [20] Usenko V, Demmel N, Schubert D, et al. Visual-inertial mapping with non-linear factor recovery[J]. IEEE Robotics and Automation Letters, 2019, 5(2): 422-429.
- [21] Zubizarreta J, Aguinaga I, Montiel J M M. Direct sparse mapping[J]. IEEE Transactions on Robotics, 2020, 36(4): 1363-1370.
- [22] Campos C, Elvira R, Rodríguez J J G, et al. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.
- [23] Tian Y, Chang Y, Arias F H, et al. Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems[J]. IEEE Transactions on Robotics, 2022, 38(4).
- [24] Kundu A, Krishna K M, Sivaswamy J. Moving object detection by multi-view geometric techniques from a single camera mounted robot[C]//2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009: 4306-4312.
- [25] Hartley R, Zisserman A. Multiple view geometry in computer vision[M]. Cambridge university press, 2003.
- [26] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381-395.
- [27] Piaggio M, Fornaro R, Piombo A, et al. An optical-flow person following

- behaviour[C]//Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell. IEEE, 1998: 301-306.
- [28] Nguyen D, Hughes C, Horgan J. Optical flow-based moving-static separation in driving assistance systems[C]//2015 IEEE 18th International Conference on Intelligent Transportation Systems. IEEE, 2015: 1644-1651.
- [29] Zhang T, Zhang H, Li Y, et al. Flowfusion: Dynamic dense rgb-d slam based on optical flow[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 7322-7328.
- [30] Sun Y, Liu M, Meng M Q H. Motion removal for reliable RGB-D SLAM in dynamic environments[J]. Robotics and Autonomous Systems, 2018, 108: 115-128.
- [31] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.
- [32] Zhang L, Wei L, Shen P, et al. Semantic SLAM based on object detection and improved octomap[J]. IEEE Access, 2018, 6: 75545-75559.
- [33] Xiao L, Wang J, Qiu X, et al. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment[J]. Robotics and Autonomous Systems, 2019, 117: 1-16.
- [34] Liu Y, Miura J. RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods[J]. Ieee Access, 2021, 9: 23772-23785.
- [35] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [36] Bescos B, Fácil J M, Civera J, et al. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.
- [37] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 1168-1174.
- [38] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481-2495.

参考文献

- [39] Wu W, Guo L, Gao H, et al. YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint[J]. Neural Computing and Applications, 2022: 1-16.
- [40] Chang J, Dong N, Li D. A real-time dynamic object segmentation framework for SLAM system in dynamic scenes[J]. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1-9.
- [41] Mozos O M, Triebel R, Jensfelt P, et al. Supervised semantic labeling of places using information extracted from sensor data[J]. Robotics and Autonomous Systems, 2007, 55(5): 391-402.
- [42] Quigley M, Conley K, Gerkey B, et al. ROS: an open-source Robot Operating System[C]//ICRA workshop on open source software. 2009, 3(3.2): 5.
- [43] Nieto-Granda C, Rogers J G, Trevor A J B, et al. Semantic map partitioning in indoor environments using regional analysis[C]//2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010: 1451-1456.
- [44] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.
- [45] Sünderhauf N, Pham T T, Latif Y, et al. Meaningful maps with object-oriented semantic mapping[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 5079-5085.
- [46] Thrun S. Probabilistic robotics[J]. Communications of the ACM, 2002, 45(3): 52-57.
- [47] Särkkä S. Bayesian filtering and smoothing[M]. Cambridge university press, 2013.
- [48] Bailey T, Nieto J, Guivant J, et al. Consistency of the EKF-SLAM algorithm[C]//2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2006: 3562-3568.
- [49] Martinez-Cantin R, Castellanos J A. Unscented SLAM for large-scale outdoor environments[C]//2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2005: 3427-3432.
- [50] Kim C, Sakthivel R, Chung W K. Unscented FastSLAM: a robust and efficient solution to the SLAM problem[J]. IEEE Transactions on robotics, 2008, 24(4): 808-820.
- [51] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91-110.
- [52] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]//2011 International conference on computer vision. IEEE, 2011: 2564-2571.

- [53] DeTone D, Malisiewicz T, Rabinovich A. Superpoint: Self-supervised interest point detection and description[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018: 224-236.
- [54] Tencent. NCNN[DB/OL]. (2018-4-16)[2023-3-17]. <https://github.com/Tencent/ncnn>
- [55] Hornung A, Wurm K M, Bennewitz M, et al. OctoMap: An efficient probabilistic 3D mapping framework based on octrees[J]. Autonomous robots, 2013, 34: 189-206.
- [56] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1314-1324.
- [57] Everingham M, Eslami S M A, Van Gool L, et al. The pascal visual object classes challenge: A retrospective[J]. International journal of computer vision, 2015, 111: 98-136.
- [58] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]//2012 IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2012: 573-580.
- [59] Palazzolo E, Behley J, Lottes P, et al. ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 7855-7862.
- [60] Li A, Wang J, Xu M, et al. DP-SLAM: A visual SLAM with moving probability towards dynamic environments[J]. Information Sciences, 2021, 556: 128-142.
- [61] Rusu R B, Cousins S. 3d is here: Point cloud library (pcl)[C]//2011 IEEE international conference on robotics and automation. IEEE, 2011: 1-4.
- [62] Munkres J. Algorithms for the assignment and transportation problems[J]. Journal of the society for industrial and applied mathematics, 1957, 5(1): 32-38.
- [63] Shi X, Li D, Zhao P, et al. Are we ready for service robots? the openloris-scene datasets for lifelong slam[C]//2020 IEEE international conference on robotics and automation (ICRA). IEEE, 2020: 3139-3145.

攻读硕士学位期间承担的科研任务与主要成果

(一) 参与的科研项目

[1] * , 科学技术部高技术研究发展中心, 国家重点研发计划. 项目编号: *

[2] * . *, 河北省教育厅 2022 年省级研究生创新资助项目. 项目编号: *

(二) 发表与完成的学术论文

[1] Cheng Shuhong, Sun Changhe, Zhang Shijun, Zhang Dianfan. SG-SLAM: A Real-Time RGB-D Visual SLAM toward Dynamic Scenes with Semantic and Geometric Information[J]. IEEE Transactions on Instrumentation and Measurement, 2023, 72(1): 1-12 (中科院 SCI 二区)

(三) 申请及已获得的软件著作权

[1] * . 三摄像头手势识别软件 v1.0, 完成日期: 2021.4.10 软件登记号: *

[2] * . 进行动态特征点剔除的 ORB-SLAM2 系统软件, 完成日期: 2021.12.12 软件登记号: *

致 谢

略

求学漫漫，二十余载，岁近而立，略有心得，聊表于此，与君共勉。
书读百遍，量变引起质变；纸上得来终觉浅，实践、认识、再实践。

生也有涯，知也无涯。以有涯随无涯，路漫漫其修远，吾将上下而求索。