电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE

论文题目 室内动态场景下的 V-SLAM 算法研究

专业学位类别	电子信息
学　　号	
作者姓名	
指导教师	
学　　院	信息与软件工程学院

分类号 TN311.5 密级 公开

UDC[注 1] 004.41

学 位 论 文

室内动态场景下的 V-SLAM 算法研究

（题名和副题名）

（作者姓名）

指导教师

电子科技大学 成都

（姓名、职称、单位名称）

申请学位级别 硕士 专业学位类别 电子信息

专业学位领域 软件工程

提交论文日期 2025 年 3 月 15 日 论文答辩日期

学位授予单位和日期 电子科技大学 2025 年 6 月

答辩委员会主席

评阅人

注 1：注明《国际十进分类法 UDC》的类号。

Research on V-SLAM Algorithm in Indoor Dynamic Scenes

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Discipline **Electronic Information**

Student ID

Author

Supervisor

School **School of Information and Software Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期： 2025 年 03 月 15 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，同意学校有权保留并向国家有关部门或机构送交论文的复印件和数字文档，允许论文被查阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索及下载，可以采用影印、扫描等复制手段保存、汇编学位论文。

（涉密的学位论文须按照国家及学校相关规定管理，在解密后适用于本授权。）

作者签名：_____ 导师签名：_____

日期： 2025 年 03 月 15 日

摘 要

随着科技的持续发展进步，人们对于提升家居生活的智能化和舒适度需求日益增长，移动机器人在室内家居场景中的应用变得越来越广泛。同步定位与建图技术（Simultaneous Localization and Mapping，SLAM）是移动机器人实现自主定位和环境感知的关键。其中基于视觉的 SLAM（Visual SLAM，V-SLAM）依靠摄像头获取环境数据，通过特征提取、匹配与跟踪，实现相机运动估计与地图构建。然而，传统的 V-SLAM 算法通常假设环境是静态的。当动态区域的特征点被误跟踪时，可能对相机位姿估计精度产生不利影响，甚至导致轨迹偏差或系统崩溃。针对以上问题，本文展开了室内动态场景下的 V-SLAM 算法研究，具体研究和内容如下：

1. 针对难以判断物体真实运动状态的问题，本文设计了一个结合 YOLOv8 实例分割和几何信息（Geometric）的 GeoYOLO-SLAM 系统。GeoYOLO-SLAM 可以准确区分物体真实运动状态，从而实现复杂场景中动态物体的精确检测。为了解决单个关键帧观测不稳定的问题，本文引入了地图点移动概率，利用连续关键帧中物体的真实运动状态信息。与其他先进方法相比，本文算法在算法性能方面具有显著优势。

2. 通过联合静态物体的重投影误差、极线误差和几何段误差，非线性优化相机位姿。针对动态物体遮挡引起的图像背景缺失问题，改进传统的双线性插值算法，以修复剔除动态物体后的静态背景，并生成无重影的静态点云稠密地图。实验结果验证了本文非线性优化的有效性，并表明本文图像背景修复算法能够修复动态物体剔除后的静态背景，生成更清晰、无重影的静态点云稠密地图。

3. 结合本文室内动态场景下的 V-SLAM 算法，设计了三维可视化系统。该系统可以实时显示 V-SLAM 的位姿轨迹，并展示点云稠密地图。增加了辅助可视化功能，使用户能够通过与系统交互进行更直观的观察。

关键词：V-SLAM，动态场景，非线性优化，静态点云稠密地图

ABSTRACT

With the continuous development of technology, people's demand for enhancing the intelligence and comfort of home life is growing, and the application of mobile robots in indoor home environments is becoming increasingly widespread.Simultaneous Localization and Mapping (SLAM) is key to enabling mobile robots to achieve autonomous localization and environmental perception. Visual SLAM (V-SLAM), based on vision, uses cameras to capture environmental data and performs feature extraction, matching, and tracking to estimate camera motion and build maps. However, traditional V-SLAM algorithms typically assume that the environment is static. When dynamic region features are mis-tracked, it may negatively affect the camera pose estimation accuracy, even leading to trajectory deviation or system failure. To address this issue, this thesis conducts research on V-SLAM algorithms in indoor dynamic scenes. The specific research and contents are as follows:

1.To address the challenge of determining the true motion state of objects, this thesis designs a GeoYOLO-SLAM system that combines YOLOv8 instance segmentation with geometric information. GeoYOLO-SLAM can accurately distinguish the true motion state of objects, enabling precise detection of dynamic objects in complex scenarios. To solve the instability of using a single keyframe observation, this thesis introduces the motion probability of map points, leveraging the true motion state information of objects from consecutive keyframes. Compared to other advanced methods, the proposed algorithm demonstrates significant advantages in performance.

2.A nonlinear optimization of camera pose is performed by jointly minimizing the reprojection error of static objects, epipolar error, and geometric segment error. To address the issue of background loss caused by dynamic object occlusion, an improved bilinear interpolation algorithm is used to restore the static background after removing dynamic objects and to generate a shadow-free static point cloud dense map. Experimental results validate the effectiveness of the nonlinear optimization and demonstrate that the proposed image background restoration algorithm can recover the static background after dynamic object removal, resulting in a clearer and shadow-free static point cloud dense map.

3.A 3D visualization system is designed in conjunction with the proposed V-SLAM algorithm for indoor dynamic scenes. The system operates in real-time, displaying the V-

SLAM trajectory and dense point cloud map. It is enhanced with auxiliary visualization features, allowing users to interact with the system for more intuitive observations.

Keywords: V-SLAM, Dynamic Scenes, Nonlinear Optimization, Static Dense Point Cloud Map

目 录

第一章 绪论 1
1.1 研究背景与意义 1
1.2 国内外研究现状 2
1.2.1 V-SLAM 发展历史及研究现状 2
1.2.2 动态环境下的 V-SLAM 研究方法 4
1.3 研究内容与结构安排 8
第二章 相关理论及技术 10
2.1 相机模型 10
2.1.1 针孔相机模型 10
2.1.2 深度相机模型 11
2.2 视觉里程计 12
2.2.1 特征提取与特征匹配 12
2.2.2 2D-2D 对极几何 14
2.2.3 3D-2D：PnP 16
2.3 ORB-SLAM3 算法 17
2.4 本章小结 19
第三章 室内动态场景下的 GeoYOLO-SLAM 系统设计 20
3.1 系统框架介绍 20
3.2 基于语义和深度信息的物体掩码生成 21
3.2.1 基于 YOLOv8 的实例分割 21
3.2.2 基于改进滑动窗口的深度信息补全 23
3.3 多视图几何物体真实运动状态检测 26
3.3.1 基于深度与极线距离的联合异常检测 26
3.3.2 基于跟随式动态窗口的物体运动状态校验 31
3.4 地图点移动概率 33
3.5 实验分析 35
3.5.1 物体真实运动状态检测实验 37
3.5.2 跟随式动态窗口消融实验 39
3.5.3 视觉里程计实验 40
3.6 本章小结 45

第四章 室内动态场景下的 V-SLAM 后端优化与稠密建图 46
4.1 系统框架介绍 46
4.2 基于静态物体的非线性优化算法 47
4.2.1 BA 优化 47
4.2.2 静态物体刚性约束 48
4.2.3 因子图优化算法 49
4.3 静态点云稠密地图构建 50
4.3.1 基于改进双线性插值算法的图像背景修复 50
4.3.2 静态点云稠密地图构建 52
4.4 实验分析 54
4.4.1 因子图优化消融实验 54
4.4.2 图像背景修复实验 56
4.4.3 静态点云稠密地图构建实验 57
4.5 本章小结 58
第五章 三维可视化系统的设计与实现 60
5.1 需求分析 60
5.1.1 系统功能性需求 60
5.1.2 系统性能需求 62
5.1.3 系统用例设计 63
5.2 总体设计 64
5.2.1 系统架构设计 64
5.2.2 系统功能模块设计 65
5.2.3 数据库设计 67
5.3 详细设计 68
5.3.1 用户管理模块 68
5.3.2 数据可视化模块 69
5.3.3 辅助可视化模块 72
5.4 系统各模块实现 72
5.4.1 开发技术与环境 72
5.4.2 用户管理模块实现 73
5.4.3 数据可视化模块实现 74
5.4.4 辅助可视化模块实现 75

5.5 系统测试 ……76
5.5.1 功能测试 ……77
5.5.2 性能测试 ……78
5.6 本章小结 ……78
第六章 总结与展望 ……79
6.1 全文总结 ……79
6.2 工作展望 ……80
致 谢 ……81
参考文献 ……82

图目录

图 2-1 针孔相机模型 10
图 2-2 RGB-D 相机原理 12
图 2-3 FAST 角点 13
图 2-4 对极几何 15
图 2-5 ORB-SLAM3 系统 17

图 3-1 GeoYOLO-SLAM 系统框架 20
图 3-2 YOLO 系列性能对比图 21
图 3-3 语义延时 22
图 3-4 基于 ROS 的关键帧和实际结果的通信机制 23
图 3-5 掩码处理。(a) YOLOv8 实例分割结果；(b) 对应灰度图像 23
图 3-6 传统的深度信息补全。(a) 原始深度图像；(b) 传统滑动窗口处理后的深度图像 24
图 3-7 深度图处理方法 25
图 3-8 改进的深度信息补全。(a) 对应的 RGB 图像；(b) 原始深度图像；(c) 改进滑动窗口处理后的深度图像 25
图 3-9 掩码处理。(a) 粗略的掩码；(b) 精细化的掩码 26
图 3-10 对极几何判断特征点运动状态 27
图 3-11 图像金字塔示意图 28
图 3-12 跟随式动态窗口 32
图 3-13 地图点移动概率 34
图 3-14 TUM 数据集。(a)RGB 图像；(b) 对应深度图像 35
图 3-15 TUM 数据集中的动态场景 36
图 3-16 Bonn 数据集中的动态场景。(a) 实验者走动；(b) 实验者搬动箱子 37
图 3-17 实验者均处于动态时 GeoYOLO-SLAM 联合异常检测实验结果 37
图 3-18 实验者既有运动状态又有静止状态 GeoYOLO-SLAM 联合异常检测实验结果。(a) 检测出一位静止的实验者；(b) 检测出两位静止的实验者 38
图 3-19 ORB-SLAM3 与 GeoYOLO-SLAM 特征点对比。(a) ORB-SLAM3；(b) GeoYOLO-SLAM 38

图 3-20 ORB-SLAM3 与 GeoYOLO-SLAM 特征点对比。(a) ORB-SLAM3；(b) GeoYOLO-SLAM 39
图 3-21 实验者逐渐出现在相机视野 39
图 3-22 跟随式动态窗口消融实验 40
图 3-23 walking_xyz 轨迹..... 41
图 3-24 walking_halfsphere 轨迹。(a)ORB-SLAM3；(b) GeoYOLO-SLAM 41
图 3-25 walking_rpy 轨迹。(a)ORB-SLAM3；(b) GeoYOLO-SLAM..... 41

图 4-1 后端优化与稠密建图系统框架 46
图 4-2 刚性约束..... 48
图 4-3 因子图..... 50
图 4-4 直接剔除动态物体。(a) 场景 1；(b) 场景 2..... 50
图 4-5 walking_xyz 轨迹分解对比图。(a)XYZ 坐标轴；(b) RPY 旋转轴 55
图 4-6 sitting_xyz 轨迹分解对比图 55
图 4-7 动态物体遮挡导致静态信息缺失。(a) 场景 1 的 RGB 图像；(b) 场景 2 的 RGB 图像；(c) 场景 1 的深度图像；(d) 场景 2 的深度图像..... 56
图 4-8 图像背景修复后效果展示。(a) 场景 1 的 RGB 图像；(b) 场景 2 的 RGB 图像；(c) 场景 1 的深度图像；(d) 场景 2 的深度图像 57
图 4-9 传统的点云稠密地图。(a)walking_xyz；(b)walking_static 58
图 4-10 改进后的点云稠密地图。(a)walking_xyz；(b)walking_static..... 58

图 5-1 系统用例图..... 63
图 5-2 系统架构图..... 64
图 5-3 系统模块概要图..... 65
图 5-4 用户注册登录流程图..... 68
图 5-5 相机位姿显示功能流程图 69
图 5-6 相机位姿与点云地图显示时序 70
图 5-7 三维可视化系统用户管理模块 73
图 5-8 登录界面展示..... 73
图 5-9 文件路径选择界面..... 74
图 5-10 相机位姿显示..... 74
图 5-11 点云稠密地图显示..... 75
图 5-12 增加距离参考圈..... 76
图 5-13 选择背景配色功能..... 76

表目录

表 3-1 实验平台配置 35
表 3-2 GeoYOLO-SLAM 与 ORB-SLAM3 在 TUM 数据集上的 ATE 误差 42
表 3-3 GeoYOLO-SLAM 与 ORB-SLAM3 在 Bonn 数据集上的 ATE 误差 43
表 3-4 三种算法在 TUM 数据集上的 ATE 误差 43
表 3-5 三种算法在 Bonn 数据集上的 ATE 误差 44

表 4-1 因子图优化消融实验在 TUM 数据集上的 ATE 误差 55
表 4-2 因子图优化消融实验在 Bonn 数据集上的 ATE 误差 56

表 5-1 用户管理模块功能需求 60
表 5-2 数据显示模块功能需求 61
表 5-3 辅助显示模块功能需求 61
表 5-4 root_user 表 67
表 5-5 user 表 67
表 5-6 log 表 67
表 5-7 系统功能测试用例 77
表 5-8 系统性能测试用例 78

第一章 绪论

1.1 研究背景与意义

近年来，人工智能与移动机器人技术呈现出突飞猛进的发展态势。机器人逐步替代了人类在简单重复性任务中的工作，例如工厂流水线加工和仓库物品搬运[1]。随着社会需求的多样化和复杂化，机器人技术的应用从工业领域扩展到服务领域。移动机器人作为机器人产业的重要组成部分，其市场规模正在快速增长，尤其是在家居环境下的服务型移动机器人领域。以扫地机器人为代表的家政服务机器人因其市场潜力巨大，成为服务型机器人产业的核心组成部分[2]。然而，要在竞争激烈的市场中占据优势，家政服务机器人的核心竞争力在于其智能化程度的高低。而实现高度智能化的关键在于，机器人必须具备自主感知和理解周围环境的能力，从而能够做出精准合理的智能决策。

SLAM 是一种广泛应用于计算机视觉和机器人领域的技术，于 1986 年被首次提出[3]，旨在通过使用传感器数据在没有先验知识的情况下同时实现系统自身的定位和环境地图的构建。SLAM 通常依赖于多种类型的传感器获取环境中的感知信息[4]，如相机、激光雷达[5]、惯性测量单元（Inertial Measurement Unit，IMU）等。如果 SLAM 的传感器主要使用的是相机那么又可以称为 V-SLAM[6]。V-SLAM 利用摄像头获取环境信息，并通过前端进行特征提取、特征匹配和帧间跟踪，后端则负责进行回环检测和全局位姿优化以完成精确的相机运动估计和地图构建。目前家政服务机器人由于成本和体积的限制，其搭载的处理器算力较为有限，对算法的高效性和鲁棒性提出了更高要求。在众多感知技术中，V-SLAM 因其低成本和丰富的环境感知能力，逐渐成为家政服务机器人的核心技术之一[7]。

在家政服务机器人实际工作时，服务环境中经常会出现动态物体，例如房间内走动的人、活动的宠物以及移动的家具等等。但是传统的 V-SLAM 算法在运行过程中假设场景是静态的。在基于特征点的 V-SLAM 系统中，若这些来自动态区域的特征被跟踪，则相机姿态估计的精度会受到严重影响从而导致轨迹偏差甚至系统崩溃[8]。因此对于 V-SLAM 来说，静态特征点的提取对相机跟踪的稳定性和地图构建的质量至关重要。

如何处理动态特征点就成为了 V-SLAM 中室内动态场景的一个重点关注问题。V-SLAM 前端通常使用随机样本一致性算法（Random Sample Consensus，RANSAC）来进行离群值的判断[9]，当存在少量运动特征点的时候 RANSAC 可以将其检测为异常值，但当运动特征点较多时 RANSAC 算法往往会失效[10]。最近的一些研究

通常基于光流估计或深度学习来应对场景中的动态物体[10–14]。光流检测通过计算连续帧之间动态特征点由运动产生的光流向量判断特征点的运动方向和速度，V-SLAM 系统通过光流检测得到特征点的运动进行相机的位姿估计[11]。但是相较于室外物体而言，室内场景下物体的运动速度相对更慢。当物体的运动较慢时，像素之间的位移较小，这会导致光流估计的不准确[12]。由于光流估计的精度取决于像素的位移大小，低速物体的光流估计可能会产生较大的误差。深度学习引入目标检测或者实例分割来识别物体类别，剔除视野中识别到运动物体类别范围上的特征点[10, 13]，仅运用静态特征点实现对相机位姿的估计。现阶段，借助深度学习技术对物体类别进行识别进而剔除动态点的方法可能会误删除位于静态物体上的特征点。虽然轻量化的目标检测速度很快，但是矩形框必然会引入背景静态点造成静态特征点的错误剔除。

在 V-SLAM 中，实例分割对场景中的图像进行像素级别的语义分类，将不同的像素分配给不同的语义类别。它通过将图像分割为语义对象的集合从而提供更丰富的外界环境信息，进而筛选出对相机位姿估计有利的特征点。但是仅仅依靠实例分割对先验运动物体这一类别上的特征点进行剔除显然不合理，这可能会导致处于静止情况下的先验动态物体上的特征点也被错误剔除，使得在室内纹理信息较低的情况下特征点数目不足从而导致难以进行跟踪。因此可以结合几何信息准确区分运动物体，使得 V-SLAM 更好地剔除运动物体上的特征点并进行相机位姿估计，从而提供更精确的定位和地图构建[15]。

1.2 国内外研究现状

1.2.1 V-SLAM 发展历史及研究现状

V-SLAM 是机器人利用视觉传感器，如摄像头等，同时估算机器人或移动设备在空间中的位置并生成环境地图的一项技术。根据所用相机的类型不同，V-SLAM 可分为单目 SLAM[16]、双目 SLAM[17] 和 RGB-D SLAM。单目 SLAM 依靠单个摄像头获取的图像作为输入进行位姿估计和地图构建，而双目 SLAM 和 RGB-D SLAM 则分别利用双目摄像头和深度相机，能够捕获更加丰富的空间深度信息。近年，V-SLAM 技术经历了快速的发展与创新。

2007 年，Davison 等人提出了基于特征点的 Mono SLAM[18]。作为首个单目 SLAM 框架，Mono SLAM 证明了可以只用一个移动的单目摄像头作为 SLAM 的唯一数据来源。Mono SLAM 的核心在于通过扩展卡尔曼滤波算法来估计现实世界中的相机运动和三维点的位置[19]。尽管 Mono SLAM 缺乏全局优化和回环检测模块，但它依然在 V-SLAM 领域中具有里程碑式的意义，开创了 V-SLAM 的先河并

逐渐成为该领域的重要研究方向之一。然而，使用 Mono SLAM 方法重建的地图具有一定的局限性。这些地图仅包括地标信息，未能提供关于该区域的详细信息。例如，地图中没有包含环境的详细几何结构或其他有助于更好理解和导航的细节信息。尽管如此，Mono SLAM 作为 V-SLAM 研究的早期代表为后续的研究和发展奠定了基础，推动了 V-SLAM 技术的进步和应用。

2013 年，Salas Moreno 等人提出了 SLAM++[20]，开创了一种全新的基于物体的 3D SLAM 范式。这一框架充分利用了 RGB-D 传感器的能力，将实时物体识别和跟踪集成到 SLAM 系统中，大幅提升了环境建模的效率和精度。SLAM++ 以 RGB-D 相机扫描场景作为输入，通过检测和识别场景中的 3D 物体，将其作为显式节点纳入优化图。这些物体节点不仅代表环境中的静态特征，还提供了与相机位姿直接关联的 6 自由度约束。相较于传统基于点或特征的 SLAM 方法，这种基于物体的描述显著降低了系统的计算复杂度，同时增强了对环境变化的适应能力。

2014 年，Engel 等人[21] 提出 LSD-SLAM 被广泛认为是该领域中的一项重要突破。该方法的核心架构包括跟踪、深度估计以及地图优化三个关键模块，每个模块在提升 SLAM 性能方面都发挥了重要作用。LSD-SLAM 利用其独特的姿态图估计模块，可以高效地完成大规模场景的三维地图重建。此外，它还具备全局优化和回环检测功能，使得其在提升定位精度和地图一致性方面表现出色。然而，尽管 LSD-SLAM 在技术上具有诸多优势，其在应用过程中也暴露出一定的不足之处。该方法在初始化阶段需要平面中的所有点参与计算，这对计算资源的需求较高，导致其计算复杂度大幅增加。因此在资源受限的环境中，LSD-SLAM 可能面临性能瓶颈，限制了其实际应用的广泛性。

2015 年，Al-Mutib 等人[22] 提出了一种基于双目相机的 V-SLAM 室内自主移动机器人导航方法。该 SLAM 技术借助 2D 物体检测以及观测点分类，依据 2D 检测框和 3D 检测框之间存在的几何约束关系，对物体的位姿进行粗略计算。在完成粗略计算后，再将 3D 检测框投射到 2D 图像内，从而生成物体的轮廓。生成物体轮廓可用于引导鲁棒的特征匹配过程，进而在立体图像以及时序图像中构建稳定的关联。最后将所有语义和特征测量结果紧耦合到优化框架中，联合求解相机与物体的精确位姿。

同年，Raúl Mur-Artal 等人提出了一种高效的 V-SLAM 系统 ORB-SLAM[23]，特别适合单目相机应用场景。ORB-SLAM 的核心技术是采用 ORB 特征点[24]。其中 ORB 特征点结合了 FAST 角点检测[25] 和 BRIEF 描述子[26]，通过使用特征金字塔使得该 SLAM 系统具有旋转不变性，从而增加特征匹配的准确性和高效性。同时 ORB-SLAM 引入了单目相机初始化方法，通过单应矩阵或本质矩阵完成运动初始化，提高了初始阶段的稳定性。

2017 年，ORB-SLAM 的升级版 ORB-SLAM2[27] 被提出，以解决单目相机在纯旋转情况下易出现跟踪失败的问题。同时 ORB-SLAM2 拓展了对双目相机和 RGB-D 相机的适配情况。此外，该系统运用光束平差法（Bundle Adjustment，BA）进行位姿优化[28]。由于双目和 RGB-D 相机在远距离点测距时误差较大，系统对地图点进行了分类，并使用不同的三角化方法生成地图点。

ORB-SLAM3[29] 是 2020 年发表的 ORB-SLAM 系列目前的最新版本，延续了前两代系统的高效性与鲁棒性，并在多传感器融合和多任务场景中进一步优化。ORB-SLAM3 不仅支持单目、双目和 RGB-D 相机，还首次引入了对惯性测量单元 IMU 的支持，形成了一个紧耦合的视觉惯性 SLAM 系统。这种设计显著提升了系统在动态场景和复杂运动中的鲁棒性与精度，特别是在光线条件较差或纹理较少的环境中表现优异。

1.2.2 动态环境下的 V-SLAM 研究方法

传统的 SLAM 算法通常基于场景静态性的假设，因此在包含移动物体的动态环境中，往往会因为未能正确区分静态背景和动态目标而失效或产生不准确的定位和建图结果。这一局限性对实际应用中的机器人导航和操作造成了重大挑战，尤其是在复杂的动态场景中。近年来随着对动态环境下 V-SLAM 研究的深入，如何有效处理动态场景区分静态和动态元素，已成为该领域的一个研究热点。动态 V-SLAM 不仅需要实现对机器人自身姿态的高精度估计，还需要解决动态物体的检测与剔除问题，从而提升系统在动态环境中的适应性和鲁棒性。近年来动态 V-SLAM 成为人们的重点研究问题之一[30]。

1.2.2.1 基于物体运动信息的动态 V-SLAM

动态 V-SLAM 的一种方法是通过跟踪动态物体的运动信息来实现高精度的位姿估计与场景重建。常见的实现方式包括利用动态因子图对动态物体和相机运动进行联合建模，通过加入动态物体的运动约束和观测信息，使得优化过程能够同时求解相机和物体的轨迹。此外，还可以采用光流法跟踪动态物体，通过逐帧估计物体的像素级运动轨迹，将其投影到三维空间中，从而分离动态物体对静态背景场景的干扰。这种方法能够有效处理动态环境下的 V-SLAM 问题，提升系统在复杂场景中的鲁棒性和精确性。

2019 年，Yang 等人提出了 CubeSLAM[31]，通过最直接的方式使用立方体建模环境中的物体。CubeSLAM 在处理动态信息时，常假设物体为刚体并遵循特定的运动模型。刚体假设认为物体上的点相对位置不会随时间改变，可以通过重投影误差优化其位置。而运动模型则采用匀速运动来约束运动特性。为应对动态场景中

特征点难以匹配的问题，CubeSLAM 没有依赖三维点位置，而是通过二维 KLT 稀疏光流算法跟踪特征点并通过将各项约束加入 BA 优化来改进相机位姿估计。对于目标物体的动态跟踪，结合二维边界框预测与匹配，确保能够在帧间准确关联动态物体的位置变化。

Henein 等人[32] 延用了 CubeSLAM 中提到的动态因子图提出了一种新的基于特征、无模型、物体感知的动态 V-SLAM 算法。通过语义分割识别刚性运动物体并估计其运动轨迹，而无需预先获取物体姿态。该 V-SLAM 在前端通过实例级物体分割生成物体掩码，并利用这些掩码检测和跟踪潜在动态特征和静态特征，从而构建包含动态和静态结构的因子图。通过因子图建模将动态 V-SLAM 问题转化为非线性最小二乘优化，联合优化机器人位姿、以及动态物体的 SE(3) 姿态变换。同时，基于刚体运动模型，算法能够识别同一物体在不同时间步长上的特征点并估计物体速度。即使在没有物体质心信息的情况下，该方法也能通过检测到的特征点近似估计物体的线速度。这一创新不仅有效消除了对物体姿态估计的需求，还为动态环境中的 V-SLAM 提供了可靠的定位和场景理解能力。

2020 年，一种面向动态场景的特征化 V-SLAM 系统 VDO-SLAM[33] 被提出。在预处理模块中首先进行场景中可能移动物体的实例分割，再利用密集光流跟踪分割结果上动态物体的角点特征从而进行相机位姿估计。利用图像语义信息实现机器人定位、动态与静态结构映射，以及刚体运动的精准跟踪。系统的核心方法包括：同时估计机器人姿态、动态与静态三维点的分布及物体运动规律，从而全面建模动态场景。通过精确计算动态物体的 SE(3) 姿态变化，实现对物体运动状态的高精度描述。并进一步从场景中提取物体速度信息，为动态 V-SLAM 任务提供优化支持。

同年，Zhang 等人提出了 FlowFusion[34]。该 V-SLAM 利用 RGB-D 相机的 RGB 图像和深度图像，通过针孔相机模型生成 3D 点云数据。其次采用超体素聚类将点云分割为多个簇，并将每个簇视为刚体便于高效处理。然而传统视觉里程计在动态环境中依赖场景为静态刚体的假设，通过最小化光度和几何残差来估计相机运动，这在存在动态对象时会导致误差累积和位姿估计的不准确。为解决这一问题，FlowFusion 引入光流残差，通过分析像素的动态特性定位动态区域。从图像的光流中减去相机自运动引起的光流，从而得到投影到图像平面的场景流。对于静态像素场景流接近零，而对于动态像素场景流非零且随移动速度增加而增大。这种基于光流残差的动态处理方法有效降低了动态环境对视觉里程计精度的影响，使系统在复杂场景下依然能够保持较高的性能和鲁棒性。

2021 年，Teed 等人设计的 DROID-SLAM[35] 通过引入端到端可微分的循环优化架构。DROID-SLAM 将传统几何方法与深度学习的优势结合，显著提升了 SLAM

系统在动态场景中的性能和泛化能力。该方法借鉴 RAFT[36] 的光流估计思路，但将其扩展为对任意数量帧的全局联合优化迭代更新相机姿态和深度图，从而有效减少长路径和闭环中的漂移。核心创新在于引入了可微分的密集束调整（Dense Bundle Adjustment，DBA）层，该层通过高斯牛顿更新最大化相机姿态、深度图和光流估计的兼容性。同时利用几何约束提升了系统的稳健性与准确性，使其能够处理双目相机或 RGB-D 相机输入而无需重新训练。通过优化深度和位姿，DROID-SLAM 不仅克服了传统方法缺乏深度学习灵活性的局限，也弥补了深度学习方法缺乏几何约束的不足，从而实现了出色的效果。

2022 年，Qiu 等人设计的动态 V-SLAM 系统 AirDOS[11] 通过 Mask-RCNN[37] 执行的实例分割将动态物体行人提取出来，再进行 AlphaPose[38] 提取人体关键点后通过光流法跟踪动态目标。最终通过人体关键点的三角测量以及光流法的动态目标跟踪进行相机位姿估计。但是基于光流法的动态 V-SLAM 系统易受到光照等因素的影响，在情况复杂的动态环境下很难达到预期效果。

2024 年，Gong 等人提出了一种基于显著性区域检测的动态 RGB-D SLAM 系统 SFD-SLAM[39]。SFD-SLAM 通过显著性检测网络直接从 RGB 图像和场景流图像中提取动态区域，避免了传统阈值分割模型的复杂性，生成精确的动态物体掩码。为提高分割精度，SFD-SLAM 结合了几何约束和深度残差，采用超像素分割和模糊 C 均值聚类（Fuzzy C-Means，FCM）进一步优化动态物体的识别。光流与场景流的分离方法使得系统能够有效区分由相机自运动和动态物体引起的运动，从而提高了分割的准确性，提升了动态 SLAM 在复杂环境中的鲁棒性和精度。

1.2.2.2 基于深度学习的动态 V-SLAM

近年来，深度学习不断发展，各种网络框架不断涌现。动态 V-SLAM 的另一个方向是利用深度学习来处理动态场景。这些方法利用目标检测或实例分割算法来识别和跟踪环境中感兴趣的物体。通过将物体级信息与 V-SLAM 系统集成，这些方法可以有效处理动态物体，并保持场景的一致性。

2018 年，Zhong 等人提出了 Detect-SLAM[40]，将 V-SLAM 与目标检测算法相结合，以解决动态环境对 V-SLAM 性能的影响。Detect-SLAM 采用了 SSD 目标检测算法[41] 来识别场景中的动态物体，并去除这些物体上的特征点，从而减少动态元素对 SLAM 定位的干扰。然而由于目标检测算法在检测动态物体时可能会包含一些非动态物体范围内的背景点，这些背景点的存在可能会影响系统的准确性。因此 Detect-SLAM 进一步引入了 GrabCut 算法[42]，对检测到的背景区域进行细化和剔除，确保仅精准地去除动态特征点。通过这种方式，Detect-SLAM 显著提高了 V-SLAM 系统在动态环境下的定位性能，使其在处理动态场景时具有更高的鲁棒

性和可靠性。

同年，Yu 等人提出了 DS-SLAM[13]。该 V-SLAM 是一种面向动态环境的鲁棒语义 V-SLAM 方法，通过结合 SegNet 语义分割网络[43] 与运动一致性检查，显著减少动态目标对姿态估计的影响，提升动态场景下的鲁棒性和定位精度。该系统基于 ORB-SLAM2 增加了语义分割、局部建图、回环检测和密集语义映射等线程，并采用实时语义分割网络过滤场景中的动态部分，从而提高了定位和建图模块的性能。此外 DS-SLAM 通过构建密集语义的 3D 八叉树地图[44]，为机器人提供了基于几何和语义信息的高级感知能力，并使用机器人操作系统（Robot Operating System，ROS）[45] 验证了其在真实动态环境中的高效性和可靠性。

2019 年，Xiao 等人提出一种结合深度学习与传统 V-SLAM 的语义同步定位与建图框架 Dynamic-SLAM[46]，旨在解决动态环境中的干扰问题。该系统运用卷积神经网络搭建了一个将先验知识相融合的 SSD 目标检测器，能够在独立的检测线程里对动态对象展开实时检测。考虑到 SSD 网络在动态场景中存在漏检的状况，Dynamic-SLAM 提出了基于相邻帧速度不变性的补偿算法，显著提高了检测的召回率。在此基础之上 Dynamic-SLAM 借助选择性跟踪算法，对动态目标所包含的特征点进行相应处理，有效减少了由于错误匹配引起的位姿估计误差。该系统集成了基于特征的 V-SLAM 框架，结合语义信息，实现了更高的定位精度和环境建图鲁棒性。

2020 年，Han 等人[8] 将 ORB-SLAM2 与 PSPNet 实例分割网络[47] 相结合，通过利用实例分割结果和光流检测技术针对特定类别的动态目标进行特征点的有效剔除，从而显著提高了动态场景中相机位姿估计的精度。这一方法充分体现了语义信息在处理动态场景下 V-SLAM 问题中的重要性。当前 V-SLAM 系统正逐步从传统几何信息处理向语义信息融合的方向发展，语义信息的引入不仅能够有效检测和处理动态目标，还显著提升了对场景的理解能力。

同年，Li 等人基于 ORB-SLAM2 进行了改进提出了处理动态信息的 DXS-LAM[48]。通过整合 CNN[49] 提取的关键点、局部描述子和全局描述子，实现了更高效的重定位和闭环检测。DXSLAM 提出了一种基于全局描述子的重定位方法。相较于传统的 BOW 方法[50]，不仅计算量更小，而且成功率显著提高。同时通过训练局部特征的视觉词袋，并结合局部描述子与全局描述子，开发了一种高度可靠的闭环检测机制。在工程实现上，DXSLAM 借助 OpenVINO[51] 和 FBOW[50]，使系统能够充分利用 SIMD 指令。从而在仅使用 CPU 的情况下实现实时运行，无需依赖 GPU 或其他加速器。

2022 年，Cui 等人提出了语义光流 SOF-SLAM[52]。该系统使用 SegNet 语义分割网络生成逐像素的输入掩码，通过结合基本矩阵计算过滤动态特征点，从而提

高了在动态场景下相机位姿估计的鲁棒性。然而，SOF-SLAM 在动态环境中仍存在一定的局限性：其一，使用传统的特征点匹配方法在面对剧烈场景变化时往往表现不足，容易受到遮挡和特征点漂移的影响；其二，仅依赖前后连续两帧数据进行处理的方式，未能充分利用更长时间尺度上的时序信息，导致对动态场景的建模能力和全局一致性仍有较大提升空间。

同年，Cheng 等人提出了 SG-SLAM[53]，这是一种结合语义信息与 V-SLAM 的系统。该方法在实时 RGB-D 语义线程中生成三维点云及三维语义对象，并将其发布到 ROS 进行可视化。最终，该系统以不同颜色区分物体类别，构建出包含丰富语义信息的语义八叉树地图。相比于传统 V-SLAM 系统，SG-SLAM 通过引入实时的语义分割和三维语义表达，不仅提升了环境建模的精度和直观性，还为机器人在复杂动态场景中的任务规划和决策提供了更具解释性的参考。

2023 年，Liu 等人提出了一种结合 YOLOv7 实例分割的 YES-SLAM[54]。YES-SLAM 利用 YOLOv7 的高效物体检测能力，在复杂且动态的场景中精确地识别和跟踪物体。同时，该系统通过结合语义信息、动态特征点数量以及相机运动幅度，改进了关键帧的选择策略从而提高了系统的定位精度和鲁棒性。YES-SLAM 还在关键帧中引入更丰富的语义信息，构建高质量的语义稠密地图。

总之，将语义信息结合到动态场景中已成为近年来 V-SLAM 处理动态问题的主流研究方向。这种方法通过语义分割和语义理解技术，不仅能够更准确地区分动态物体和静态物体，还能利用语义信息进行动态场景中冗余数据的剔除。从而切实提升系统在动态环境中的定位精度和场景理解能力。此外，语义信息还可以为 V-SLAM 系统提供更高级别的环境理解，例如识别场景中的目标物体类别或预测物体的动态行为，从而进一步增强系统在复杂动态场景中的鲁棒性和适应性。这种结合策略不仅提升了 V-SLAM 的场景理解深度，还为自主机器人在动态环境中的任务规划和决策提供了更可靠的支持。

1.3 研究内容与结构安排

本文围绕室内动态场景下的 V-SLAM 算法优化与系统实现展开研究，旨在解决传统 V-SLAM 算法在动态环境中因动态物体干扰导致的位姿估计偏差与地图构建不完整问题。全文共分为六章，各章节内容逻辑递进，逐步实现室内动态场景下的高精度位姿估计与静态点云稠密地图构建。

第一章为绪论，系统阐述了移动机器人在动态家居场景中的应用需求及技术挑战并分析了 V-SLAM 的发展脉络。重点总结了动态环境下 V-SLAM 的研究现状与不足。通过对比现有方法，明确了本文的研究方向。

第二章为相关理论及技术，作为全文的理论基础。详细介绍了 V-SLAM 的核心算法框架，包括相机模型、视觉里程计及 ORB-SLAM3 算法原理，为后续动态场景下的算法改进提供理论支撑。

第三章提出室内动态场景下的 GeoYOLO-SLAM 系统设计。将 YOLOv8 实例分割[55] 与改进滑动窗口策略补全的深度信息相结合，生成精细化物体掩码。结合语义信息和多视图几何校验，提出一种物体真实运动状态检测方法。并构建地图点移动概率模型，从而解决单个关键帧观测不稳定的问题。最后在 TUM 数据集[56] 以及 Bonn 数据集[57] 上进行实验，并与目前主流的动态 V-SLAM 算法进行横向比较，展现 GeoYOLO-SLAM 算法的有效性。

第四章主要是动态场景下的 V-SLAM 后端优化与稠密建图算法研究。基于第三章物体运动状态的判断，提出融合静态刚性物体运动约束。通过联合最小化重投影误差、极线误差及几何段误差，共同优化相机位姿。针对动态物体移除后的场景缺失问题，设计基于改进双线性插值的图像背景修复算法恢复被遮挡区域的静态背景，最终生成完整且无重影的静态点云稠密地图。通过在 TUM 数据集以及 Bonn 数据集上的消融实验，验证本文所提算法能够提高位姿估计的准确度。并且在 TUM 数据集上的测试结果表明，本文改进的算法能够很好的修复图像背景并创建较为清晰无重影的静态点云稠密地图。

第五章是三维可视化系统的设计与实现，作为算法与用户交互的平台。系统集成用户管理模块、数据可视化模块以及辅助可视化模块，支持实时显示 V-SLAM 位姿轨迹、显示静态点云稠密地图，并提供多视角切换、增删距离参考图形及点云坐标交互查询等功能。通过模块化开发与系统性测试，验证了三维可视化系统在动态场景分析中的实用性与高效性。

第六章是对本文所开展的研究工作进行归纳总结，并对未来相关研究方向予以展望。

第二章 相关理论及技术

2.1 相机模型

2.1.1 针孔相机模型

针孔相机模型是计算机视觉和图像处理领域中一个基本而重要的概念，它描述了把三维世界里的点映射到二维图像平面的具体方式。针孔相机模型如图2-1所示，该模型把相机类比为一个前方开有针孔的封闭盒子，而图像平面处于相机内部的特定位置。光线从场景内的物体出发，穿过针孔之后，在图像平面上形成投影点。这一过程基于几何光学原理，利用相似三角形的关系，可以精确计算出场景中任意三维点在图像平面上的位置。

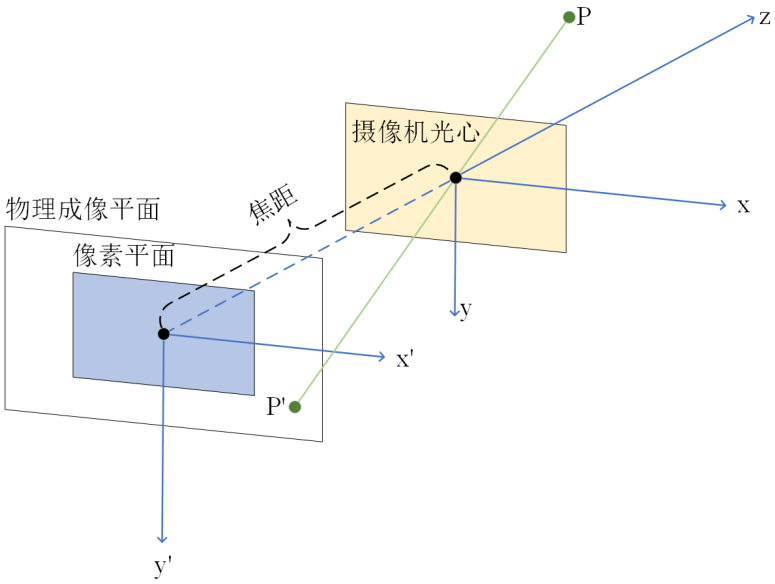


图 2-1 针孔相机模型

在针孔相机模型中，重要的参数包括焦距、主点位置等。焦距决定了相机的视野和图像的缩放程度，设定摄像机的焦距为 f。主点位置则是图像平面上光轴与平面的交点，通常对应于摄像机的光心 O。现实中的空间点 P 通过这个小孔投影，最终在成像平面上形成一个点 P'。假设点 P 的坐标为：$P = \left[X, Y, Z\right]^T$，而 P' 的坐标为:$P' = \left[X', Y', Z'\right]^T$。

根据三角形相似的原理，可以得出：

$$\frac{Z}{f} = \frac{X}{X'} = \frac{Y}{Y'} \tag{2-1}$$

通过移项，整理之后得到 P' 与成像的空间关系式：

$$\begin{aligned} X' &= f\frac{X}{Z} \\ Y' &= f\frac{Y}{Z} \end{aligned} \tag{2-2}$$

假定在物理成像平面内存在一个像素平面 $o-u-v$。那么像素平面上 P' 的像素坐标为 $P' = \left[u, v\right]^T$。通过平移和缩放，P' 坐标与像素坐标 $\left[u, v\right]^T$ 的转换关系如下所示：

$$\begin{cases} u = \alpha X' + c_x \\ y = \beta Y' + c_y \end{cases} \tag{2-3}$$

其中 α 和 β 为缩放系数，c_x 和 c_y 为平移系数。将公式2-2代入以上坐标转换公式，可以得到：

$$\begin{cases} u = f_x\frac{X}{Z} + c_x \\ y = f_y\frac{Y}{Z} + c_y \end{cases} \tag{2-4}$$

转换成矩阵的格式为：

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} \cdot \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{1}{Z} \cdot KP \tag{2-5}$$

其中由 f_x、f_y、c_x 以及 c_y 组成的矩阵被称为相机的内参矩阵 K，反映了相机的内部属性和成像特性。相机的内参通常由相机制造商提供，也可以通过相机标定法进行计算和提取。将相机坐标系和世界坐标系的转换关系带入上式得出世界坐标系与像素坐标系间的转换关系：

$$P' = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z}K(RP + t) = \frac{1}{Z}KP \tag{2-6}$$

其中 R 为相机的旋转矩阵，t 为相机的平移向量，T 为相机的外参矩阵。

2.1.2 深度相机模型

深度相机作为一种能够直接获取场景深度信息的设备，已广泛应用于计算机视觉、虚拟现实、机器人导航等领域。深度相机模型是理解和应用该类相机的理论基础，它不仅涉及深度传感器的工作原理，还包括其如何与传统的图像处理技术相结合以提供准确的空间信息。

如图2-2所示，深度相机通常基于不同的技术原理来获取深度数据，最常见的技术包括结构光和飞行时间（Time-of-Flight，ToF）[58]。每种技术都有其独特的原

理和优缺点，但它们的目标都是通过某种方式测量从相机到场景中物体表面的距离。结构光技术通过投射已知模式的光并测量其形变来计算深度。飞行时间技术是通过计算光信号在空间中传播所耗费的时间来测量距离。而双目相机则借助两台或多台相机从不同视角捕获图像，通过比较不同图像中的特征点来估计深度。

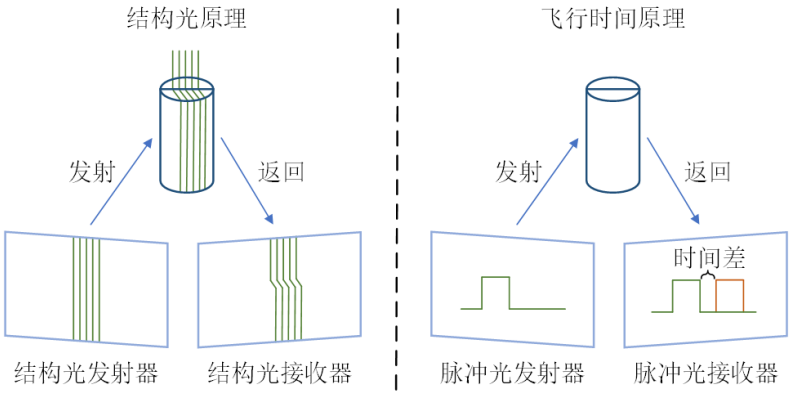


图 2-2 RGB-D 相机原理

在深度相机的建模中，除了考虑传感器本身的硬件特性外，还需要关注相机的内部参数和外部参数。相机的内部参数涵盖了焦距、主点位置以及畸变系数等，外部参数则描述了相机相对于世界坐标系的姿态。对于深度相机，除了这些传统的相机内外参外还需要考虑到传感器的特定设计特性，例如深度分辨率和测量精度。

2.2 视觉里程计

在 V-SLAM 中，特征点法通过提取和匹配图像中的特征点并结合数学模型，实现相机位姿估计。在单目 SLAM 系统中由于没有先验信息可供参考，系统需要依赖两帧图像之间的特征点匹配关系（2D-2D）来计算初始的相机位姿，进而实现系统的初始化。此外，通过在两帧图像之间进行三角化操作，确定地图点的初始深度信息。完成初始化后系统已具备地图点的初始深度信息，因此可以将新到达的图像与这些地图点进行匹配（3D-2D），从而对后续帧的相机位姿进行计算和优化。以下将详细介绍特征提取和特征匹配、三角化处理以及优化方案。

2.2.1 特征提取与特征匹配

在 V-SLAM 系统中，特征提取与特征匹配是实现精确定位与环境建图的关键技术。V-SLAM 的核心任务是通过连续传感器数据估计设备的位姿变化并构建环境地图，而这一过程高度依赖于从场景中提取的视觉特征点。通过特征提取算法，从图像或点云中识别出具有高度稳定性和区分度的特征点，为后续的帧间匹配提

供了基础。这些特征不仅用于推导设备的运动估计，还为地图表示提供了可靠的基准。在帧间匹配中，通过匹配相邻帧或回环帧中的特征点，V-SLAM 能够有效估计相对变换并进行回环检测与地图优化，从而提高全局一致性。

对于 V-SLAM，常用的特征包括尺度不变特征变换（Scale-Invariant Feature Transform，SIFT）[59]、方向 FAST 与旋转 BRIEF 特征（Oriented FAST and Rotated BRIEF，ORB）[24]、快速角点检测（Features from Accelerated Segment Test，FAST）等，这些特征能够在不同光照条件、视角变化和图像缩放中保持良好的稳定性与鲁棒性。特征提取算法为视觉系统提供了关键的特征点和描述子，广泛应用于图像匹配、位姿估计以及地图构建等任务。

SIFT 是一种具有高度鲁棒性和广泛适用性的特征提取方法，由 David Lowe 提出，并在多尺度与旋转不变性问题上开创了全新的解决思路。其核心流程包括尺度空间的构建、关键点检测、特征描述子生成和特征匹配。在尺度空间中，SIFT 利用高斯金字塔对图像进行多尺度表示，通过差分高斯（Difference of Gaussians，DoG）[60] 计算检测图像的局部极值点，从而提取具有尺度不变性的关键点。随后，通过计算梯度方向并赋予每个特征点一个主方向，SIFT 实现了对旋转的鲁棒性。在描述子生成阶段，SIFT 从关键点邻域提取梯度方向直方图并编码为 128 维的高维特征向量，这种描述子能够很好地适应光照变化和仿射变换。尽管 SIFT 在特征点匹配的精度和鲁棒性方面表现优异，但其计算复杂度较高，使得其在实时性要求较高的 SLAM 系统中受到了一定限制。

为解决计算效率问题，ORB 作为一种高效的特征提取算法应运而生，其设计目标是降低计算代价的同时，仍保持一定的特征点检测和描述能力。ORB 的设计理念源自 FAST 与 BRIEF 两种方法的结合与优化。

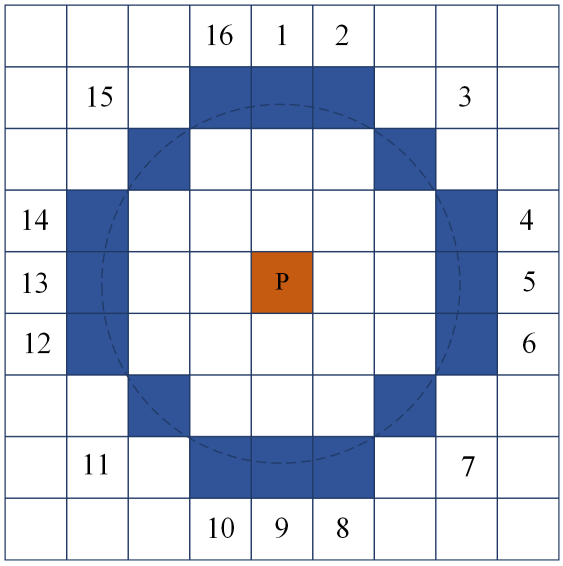


图 2-3 FAST 角点

FAST 是一种专注于快速角点检测的算法，利用圆周像素测试机制通过比较某像素点及其邻域灰度值的差异，快速判断其是否为角点。如图2-3所示其判断逻辑简单高效：若圆周上连续的 16 个像素点灰度值均亮于或暗于中心像素值设定的阈值，则该像素点被标记为角点。然而 FAST 本身不具备尺度和旋转不变性，也不提供特征描述子，需要与其他方法结合才能用于匹配任务。BRIEF 则是一种轻量化的特征描述子，通过对像素邻域进行简单的二值化比较生成二进制字符串，极大地降低了计算和存储开销。尽管 BRIEF 在匹配效率上具有显著优势，但其对图像旋转非常敏感。

ORB 结合了 FAST 和 BRIEF，并在关键环节上引入了多种优化技术，使其成为一种兼具效率与鲁棒性的特征提取算法。ORB 首先使用 FAST 进行特征点检测，但为了增强旋转鲁棒性和筛选关键点的质量，ORB 对 FAST 检测到的特征点应用了哈里斯角点响应函数进行排序仅保留响应值较高的关键点。同时为了解决 BRIEF 对旋转敏感的问题，ORB 通过在特征点邻域计算灰度质心方向，为每个特征点分配一个主方向。并对 BRIEF 描述子进行旋转调整，从而实现旋转不变性。此外，ORB 使用基于汉明距离的匹配方法，在保证匹配速度的同时提升了特征匹配的准确性。ORB 通过改进 FAST 和 BRIEF，显著降低了计算复杂度，同时实现了对尺度、旋转和一定程度的光照变化的鲁棒性，使其成为许多实时 V-SLAM 系统的核心算法。

2.2.2 2D-2D 对极几何

在两帧图像之间的运动估计问题中，对极几何提供了重要的理论基础。设两帧图像之间的相机运动由旋转矩阵 R 以及平移向量 t 描述，两个相机的光心分别记为 O_1 和 O_2。空间中的某一特征点 P 在第一帧图像中的投影为 p_1，在第二帧图像中的投影为 p_2，通过几何关系可以推导出约束条件。设在世界坐标系下，特征点 P 的空间坐标为：

$$P = \left[X, Y, Z\right]^T \tag{2-7}$$

根据针孔相机模型，第一帧图像中的投影为 p_1，在第二帧图像中的投影为 p_2 的像素其次坐标为：

$$\begin{aligned} p_1 &= KP \\ p_2 &= K(RP + t) \end{aligned} \tag{2-8}$$

其中，p_1 像素点和 p_2 像素点与归一化平面坐标 x_1 和 x_2 的关系为：

$$\begin{aligned} x_1 K &= p_1 \\ x_2 K &= p_2 \end{aligned} \tag{2-9}$$

将式2-8代入后，得到 x_1 与 x_2 的关系如下：

$$x_2 = Rx_1 + t \tag{2-10}$$

两侧同乘上 t 的外积后将最后一项消掉，可以得到：

$$t^\wedge x_2 = t^\wedge R x_1 \tag{2-11}$$

左右两边同乘 x_2^T，由于 t 和 x_2 是两个相互垂直的向量，可以得到：

$$x_2^T t^\wedge x_2 = x_2^T t^\wedge R x_1 = 0 \tag{2-12}$$

通过特征匹配确定 p_1 像素点与 p_2 像素点为同一空间点的投影，连线 O_1p_1 和连线 O_2p_2 在三维空间中相交于点 P，O_1、O_2 和 P 三点共同定义了极平面。将式2-9带入式2-12，可以得出 O_1、O_2 和 P 三点共面的对极几何关系：

$$p_2^T K^{-T} t^\wedge R K^{-1} p_1 = 0 \tag{2-13}$$

其中 $t^\wedge R$ 为本质矩阵（Essential Matrix，E）。本质矩阵 E 与基础矩阵（Fundamental Matrix，F）的转换关系如下：

$$F = K^{-T} E K^{-1} \tag{2-14}$$

如图2-4所示，极平面分别与两个成像平面相交，其交点记为极点 e_1 和 e_2。相机中心连线 O_1O_2 称为基线。此外，极平面与成像平面的交线被称为极线，极线在两幅图像中分别对应于空间点 P 的潜在投影位置。在第一帧图像中，射线 O_1p_1 上的所有空间点都会投影到同一像素点 p_1。而在第二帧图像中，该射线的投影对应于一条极线 e_2p_2，表示点 P 在第二帧中的可能投影位置。因此，若已通过特征匹配确定了点 p_2，即可进一步推算出点 P 的三维空间位置以及相机之间的运动关系。若缺乏匹配点，则需沿极线搜索潜在的对应关系。

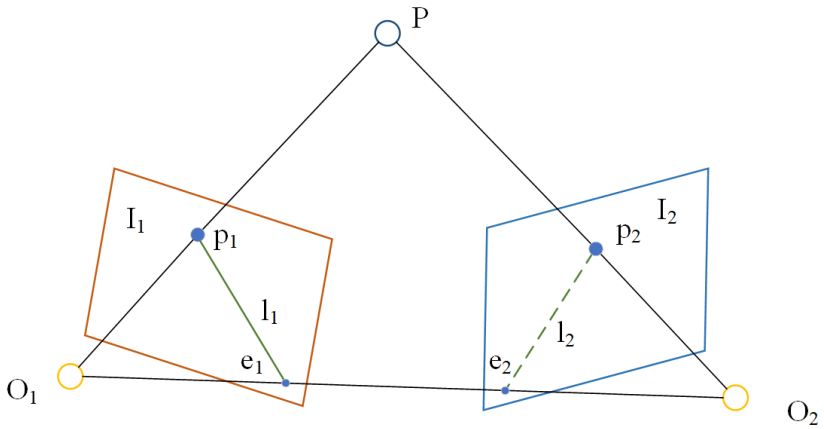


图 2-4 对极几何

2.2.3 3D-2D：PnP

透视 n 点（Perspective-n-Po int，PnP）是 V-SLAM 与计算机视觉中的核心技术，用于通过已知的 3D 空间点及其在图像上的 2D 投影点求解相机的位姿（旋转矩阵 R 与平移向量 t）。其数学模型基于针孔相机投影原理，将 3D 点通过相机坐标系转换至像素坐标，并通过最小化重投影误差构建优化目标。经典的求解方法主要为直接线性变换（Direct Linear Transformation，DLT）[61]、EPnP[62] 以及非线性优化（如 Bundle Adjustment）。

DLT 方法是一种较为经典的线性求解 PnP 问题的方式，通过建立一个包含相机外参的线性方程组来求解 PnP 问题。在已知 3D 点和对应的 2D 投影的情况下，DLT 假设相机的投影过程可以用一个线性模型来描述：

$$x_i = K[R|t]X_i \tag{2-15}$$

其中，$x_i = [u_i, v_i]^T$ 是 2D 图像中的点，$X_i = [X_i, Y_i, Z_i]^T$ 是 3D 点，K 是相机内参矩阵，$[R|t]$ 是相机的外参矩阵。DLT 方法通过将上述方程展开，将多个 2D-3D 点对的关系转化成一个线性方程组，并通过最小二乘法来求解该方程组，从而得到相机的外参。尽管 DLT 计算高效，但其对噪声敏感且需至少 6 组点，限制了其在复杂场景的应用。

EPnP 方法是在 PnP 问题中进行优化的经典方法之一，它通过引入辅助变量，将问题简化为一个相对高效的求解过程。EPnP 算法通过假设 3D 点集与投影之间的关系，可以通过一个中介的虚拟平面来描述，从而将外参估计转化为一个优化问题。EPnP 通过选择一组特定的基准点和通过对这组点进行适当变换，提出了一种基于预定义基准点的方案，使得 PnP 问题的求解更加高效。

然而无论是 DLT 还是 EPnP，这些方法通常依赖于线性优化。虽然能提供初步的外参估计，但结果的精度和鲁棒性仍然受到限制。为了解决这些问题，通常使用非线性优化方法（如 BA 优化）进一步精细化估计。BA 优化通过最小化所有 3D 点与其对应图像点之间的重投影误差，从而优化相机的内外参数和 3D 点的空间位置。BA 优化通过非线性最小二乘法优化目标函数：

$$\mathcal{E} = \sum_{i=1}^{n} \|x_i - \hat{x}_i(X_i, K)\|^2 \tag{2-16}$$

其中，x_i 是图像中已知的点，$\hat{x}_i(X_i, K)$ 是通过相机内外参数和 3D 点估算得到的投影点，X_i 是 3D 点的位置。根据以上公式，系统能够同时优化相机的内参、外参以及三维点的位置，从而得到全局最优解。BA 优化被广泛应用于计算机视觉中的相机标定、三维重建和多视角几何问题中。虽然该方法计算量较大，尤其是在点

数和视角较多时，但它能够提供非常精确的结果，是高精度 PnP 问题求解的理想选择。

2.3 ORB-SLAM3 算法

如图2-5所示，ORB-SLAM3 是当前 V-SLAM 领域中功能最为全面且性能优异的系统之一。作为 ORB-SLAM 系列的最新版本。ORB-SLAM3 在前代算法的基础上进行了重要扩展，支持单目、双目、RGB-D 相机，以及与 IMU 数据的融合使得系统具备了更高的鲁棒性和适应性。ORB-SLAM3 通过多线程架构实现了高效的视觉里程计、建图与全局优化，具备实时性、精确性和灵活性。ORB-SLAM3 继承了 ORB-SLAM2 的核心思想，通过多线程架构将 SLAM 系统分为三个主要线程：跟踪线程（Tracking）、局部建图线程（Local Mapping）和回环检测线程（Loop Closing），并在此基础上融合了 IMU 数据，从而显著提高了定位与建图的精度与稳定性。这三个线程协同工作，实现了从输入传感器数据到输出全局地图的完整功能。

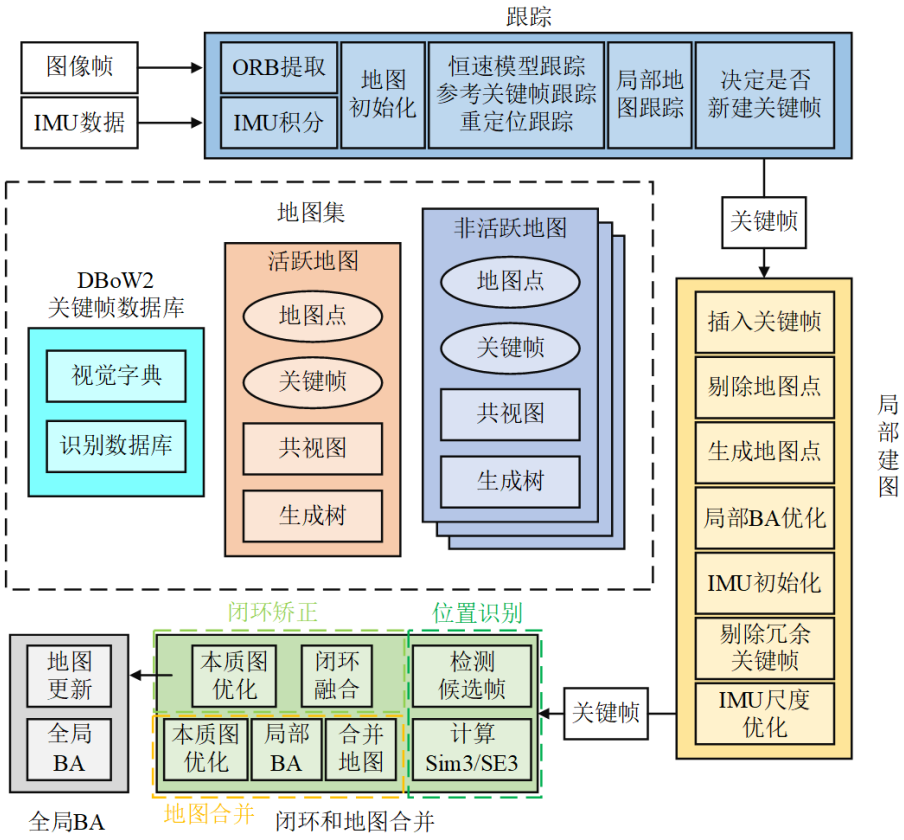


图 2-5 ORB-SLAM3 系统

首先跟踪线程负责处理传感器输入数据，通过特征提取与匹配估计相机的位姿，实现实时跟踪。跟踪线程将关键帧发送给局部建图线程，同时保持当前帧与地

图之间的关联关系。其次，局部建图线程接收关键帧并进行局部地图构建，维护地图点的稠密性与准确性，通过局部优化不断提高关键帧与地图点的精度。最后，回环检测线程通过检测历史帧与当前帧的相似性闭合回环，减少累积误差，并利用全局 BA 优化整个地图的全局一致性。ORB-SLAM3 同时引入了 IMU 数据融合机制，为纯视觉 SLAM 提供运动约束，使系统能够在快速运动或低纹理场景下依然保持稳定工作。此外，ORB-SLAM3 支持单目、双目和 RGB-D 相机输入，并针对不同传感器配置进行适配，确保系统具备通用性和灵活性。

跟踪线程是 ORB-SLAM3 的前端模块，主要功能是通过输入的传感器数据实时估计相机的位姿，并完成特征点的匹配与跟踪。在跟踪过程中，系统首先使用 ORB 特征提取算法对图像中的关键点进行提取，并计算其描述符。随后，通过与局部地图或历史关键帧进行特征匹配，估计当前帧相对于地图的位姿。跟踪过程中采用了 PnP 算法来解决位姿估计问题，同时结合 RANSAC 算法剔除误匹配点，确保位姿估计的准确性。对于单目相机，跟踪线程还需进行初始化，通过两帧图像间的视差进行三角化恢复初始的 3D 点云。双目相机通过视差计算间接估计深度，RGB-D 相机则可以直接获取深度信息。因此对于双目和 RGB-D 相机，跟踪过程更加高效且准确。此外，ORB-SLAM3 引入了 IMU 数据融合，通过 IMU 预积分技术提供先验位姿约束，进一步提高跟踪过程的精度与鲁棒性。当跟踪线程检测到帧间特征匹配质量下降时会触发重定位模块，通过与全局地图进行匹配重新恢复位姿，确保系统能够在跟踪丢失时快速恢复。

局部建图线程作为 ORB-SLAM3 后端的重要组成部分，其核心功能是对跟踪线程传递的关键帧进行处理，同时构建和优化局部地图。局部建图的主要任务包括：关键帧的插入与管理、地图点的三角化以及局部 BA。在接收到新的关键帧后，局部建图线程首先对关键帧进行筛选，剔除冗余的关键帧，保证地图的稀疏性与实时性。随后，系统通过三角化方法恢复新的三维地图点，并将其与现有地图点进行融合与优化。为进一步提高地图的精度，系统对局部关键帧和地图点进行局部 BA 优化。使用非线性优化算法最小化观测误差，确保局部地图的一致性。局部建图线程与跟踪线程并行运行，有效分担了计算负担，并通过不断优化局部地图，为跟踪线程提供高质量的地图参考。

回环检测线程是 ORB-SLAM3 实现全局地图优化的关键模块，主要用于检测当前帧与历史帧之间的回环，并通过回环校正消除累计误差。回环检测的核心是通过 ORB 特征匹配找到与当前帧相似的历史关键帧，判断是否存在回环。当检测到回环后，系统会计算当前帧与回环帧之间的 Sim3 相似变换[63]，并通过全局 BA 对整个地图进行优化确保地图的全局一致性。回环检测的引入有效消除了长期累计的漂移误差，使得 ORB-SLAM3 能够在大规模场景下构建精确的地图。

ORB-SLAM3 相比前代算法及其他主流 SLAM 系统具有显著优势。其支持单目、双目、RGB-D 相机与 IMU 数据的融合，拓展了系统的应用范围。通过 IMU 数据融合、回环检测与全局优化，系统在弱纹理、快速运动的环境下依然具备高精度和鲁棒性。多线程架构的引入将跟踪、建图与优化任务高效分离，确保实时性能。此外，ORB-SLAM3 针对单目、双目和 RGB-D 传感器提供灵活的初始化机制，适应不同应用需求。

2.4 本章小结

本章围绕 V-SLAM 系统的理论基础与核心技术展开论述。首先从相机成像模型出发，系统介绍了针孔相机模型与深度相机模型的数学表达及其物理意义，为后续算法奠定理论框架。在视觉里程计部分，重点剖析了特征提取与匹配的关键技术，详细推导了基于 2D-2D 特征对应的对极几何约束方程，以及通过 3D-2D 点对应关系建立 PnP 位姿估计模型的求解方法。最后介绍了 ORB-SLAM3 经典算法的核心思想及其框架结构，为后续算法优化与系统实现奠定了必要的理论基础。

第三章 室内动态场景下的 GeoYOLO-SLAM 系统设计

传统的 V-SLAM 算法通常假设周围环境是静态的，因此在固定场景下能够实现较为准确的定位与建图。然而在动态场景中，摄像头可能会捕捉到移动的物体，干扰姿态估计的准确性。从而导致定位误差，甚至引发定位失败。因此，动态环境中识别运动的物体，减少这些物体对机器人定位的干扰至关重要。此外，先验动态物体在实际情况中不一定处于运动状态，例如坐下静止不动的人。在纹理信息丰富的情况下可以提取较多的特征点，如果仅依靠类别将先验动态物体上的特征点全部去除，虽然可能会误删静止物体上的特征点但仍然有大量特征点可用于匹配。然而在纹理信息较少的背景下，例如静止的人靠在室内白墙上，简单地按照类别删除特征点会导致丢失大量静态特征点从而导致特征点数量不足等问题。

针对以上问题，本文设计了一个结合语义模块和几何模块的 GeoYOLO-SLAM 系统。本章的结构如下：首先，概述整个 GeoYOLO-SLAM 系统的架构。其次，详细讲解如何通过语义模块生成物体精细化掩码。接着，详细讲解如何通过几何模块确定物体的真实运动状态以及如何更新地图点的移动概率。最后在开源数据集上测试本文提出的 GeoYOLO-SLAM 系统，并将其结果与当前其他主流算法比较，验证本算法提高姿态估计精度的有效性。

3.1 系统框架介绍

本章的系统框架如图3-1所示，在传统的 ORB-SLAM3 算法的基础上增加语义模块、几何模块以及地图点移动概率更新算法。以 RGB 图像和深度图像作为输入，以相机的位姿与建图作为输出。

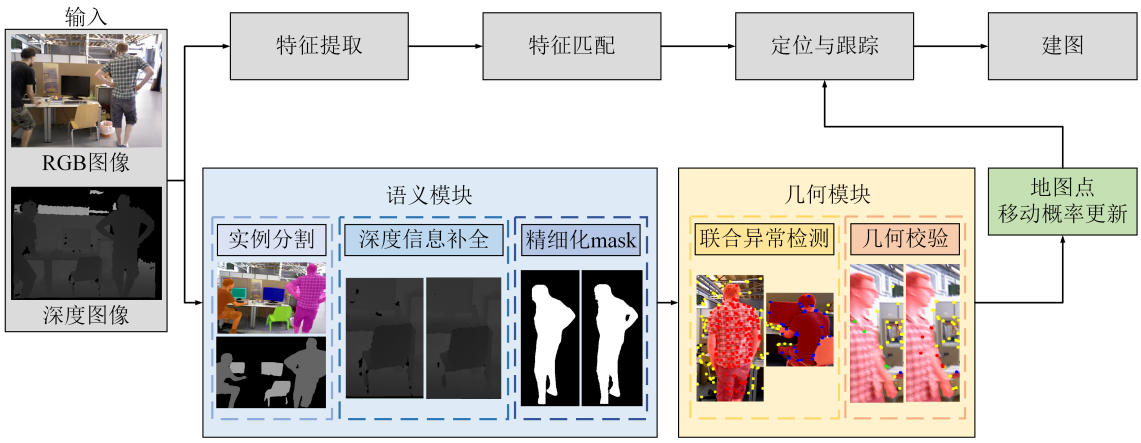


图 3-1 GeoYOLO-SLAM 系统框架

本文以室内动态场景为研究对象，以 RGB 图像以及深度图像作为输入。在语义模块中 RGB 图像通过 YOLOv8 实例分割得到粗略的物体掩码信息，再通过深度图精细化物体掩码信息。将精细化物体掩码信息输入到几何模块进行联合异常检测，并结合多视图几何校验以实现在复杂场景中准确得到物体真实运动状态。为了解决仅使用一个关键帧观测不稳定的问题，本文引用移动概率利用连续关键帧的物体真实运动状态信息。将地图点增加移动概率属性，根据物体的运动状态更新地图点的动态概率。系统跟踪阶段只使用静态地图点匹配的静态特征点从而解决动态特征点对相机位姿估计的干扰问题。

3.2 基于语义和深度信息的物体掩码生成

3.2.1 基于 YOLOv8 的实例分割

YOLOv8 作为 YOLO 系列的最新版本，相较于之前的模型在多个方面进行了显著优化。首先，YOLOv8 采用了更先进的网络架构设计，包括改进的特征提取模块和轻量化的模型结构，从而在保持高精度的同时显著提高了推理速度和计算效率。其次，YOLOv8 通过引入自适应锚点机制和更高效的标签分配策略，进一步提升了对多尺度目标的检测能力，尤其是在小目标检测任务中表现出色。此外，YOLOv8 还优化了训练策略，结合了最新的数据增强方法与优化算法，使得模型在复杂场景中的泛化能力更强。如图3-2所示是 Ultralytics 官方提供的 YOLO 系列性能对比图。综合来看，YOLOv8 在检测精度、速度和模型轻量化方面均优于此前的 YOLO 系列。因此本文采用了最新的 YOLOv8 实例分割框架。

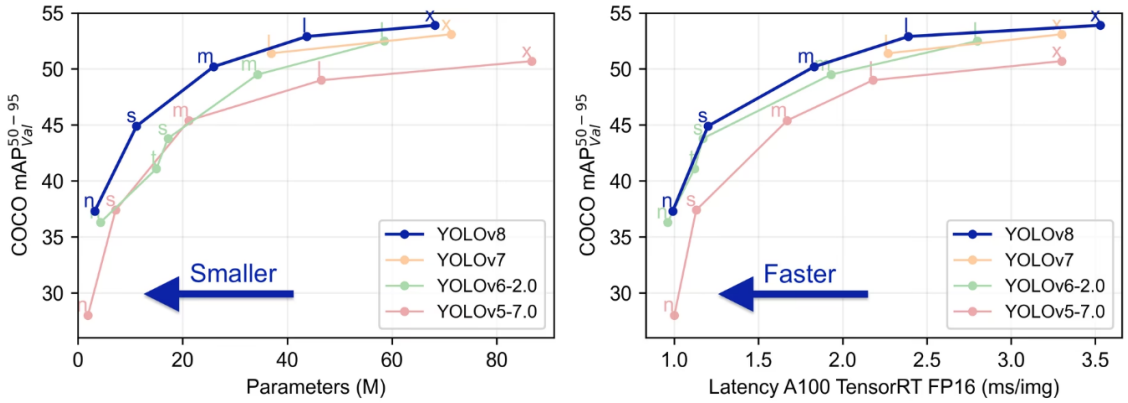


图 3-2 YOLO 系列性能对比图

由于现阶段，实例分割处理一帧的速度要慢于 V-SLAM 系统处理一帧的速度。因此如果对每个关键帧顺序的进行实例分割会导致 V-SLAM 系统被动等待实例分割处理结果，从而造成 V-SLAM 系统不能实时。为了解决 V-SLAM 系统不能实时

的问题，本文借鉴了 RDS-SLAM[64] 中的关键帧队列策略，每次处理都只将关键帧队列中的第一个关键帧和最后一个关键帧进行实例分割。

在 RDS-SLAM 中定义了语义延时 D_s 为当前处理帧 F_t 的 id 与具有最新实例分割信息的帧 S_t 的 id 的差值：

$$D_s = FrameID(F_t) - FrameID(S_t) \tag{3-1}$$

由于现阶段实例分割的速度要慢于 V-SLAM 系统的速度，所以对每一帧或者每一个关键帧都执行实例分割会导致语义延时语义延时 D_s 的线性增长，从而造成误差的累积。采用关键帧队列选择队首和队尾的关键帧进行分割，不仅可以尽可能利用最新的语义信息还能使语义延时 D_s 保持在一个常数。如图3-3所示，这时语义延时 D_s 成为一个常数而非线性增加。

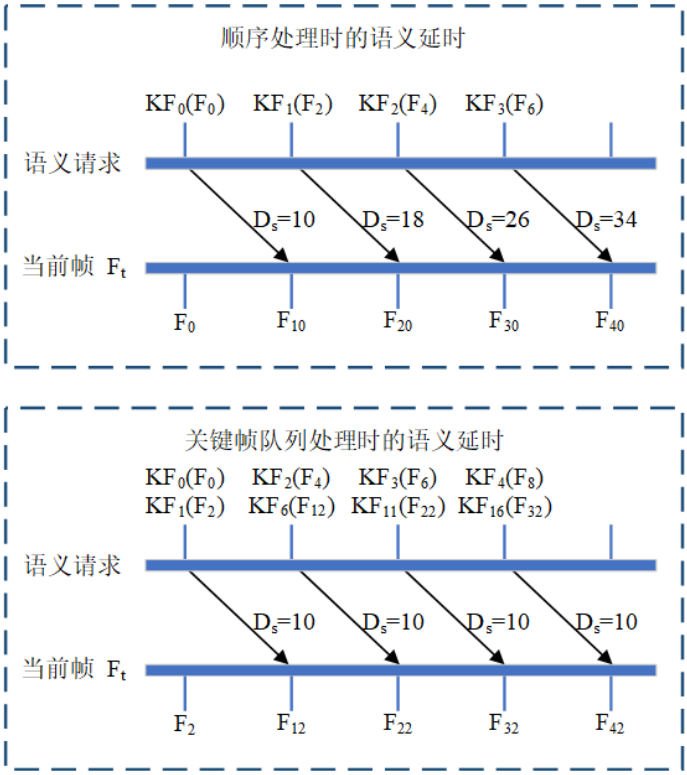


图 3-3 语义延时

为了解决 V-SLAM 系统不能实时的问题本文引入了关键帧队列，将关键帧队列中头部关键帧和尾部关键帧进行实例分割，进而得到关键帧中的物体分割结果。将做完实例分割的关键帧进行标记，确保在之后申请实例分割时避开已经具有语义信息的关键帧。关键帧队列申请实例分割请求与结果返回过程如下图3-4所示，客户端每收到两个关键帧组合就将其发送至服务端进行实例分割，服务端进行 YOLOv8 实例分割后返回这两帧的处理结果组合。

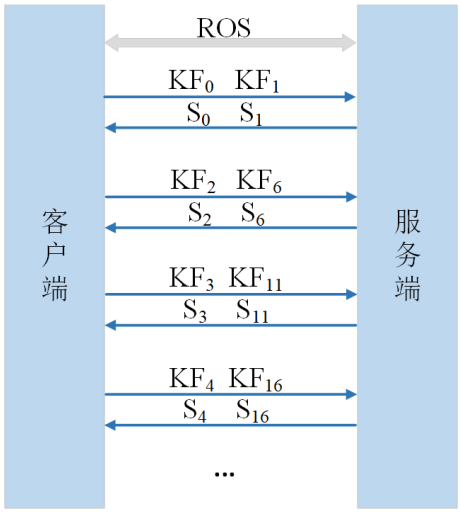


图 3-4 基于 ROS 的关键帧和实际结果的通信机制

语义线程在接收 YOLOv8 实例分割输出的物体检测结果后，基于灰度值差异对多目标掩码进行分离处理。根据灰度值的区别，提取出每个物体单独的掩码。物体掩码如图3-5所示，每个物体的掩码能够清晰地与其他目标区分开来。

(a) (b)

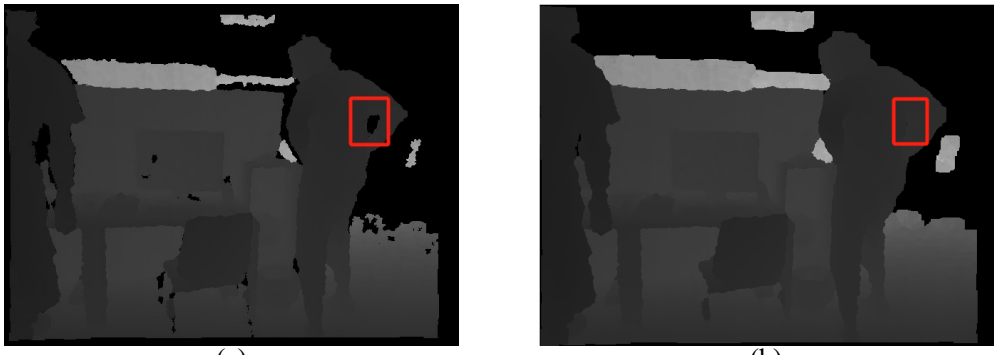
图 3-5 掩码处理。(a) YOLOv8 实例分割结果；(b) 对应灰度图像

3.2.2 基于改进滑动窗口的深度信息补全

实例分割算法的结果通常在边界区域存在一定的不准确性。这种不准确性主要体现在前景物体和背景的分离上，特别是在复杂场景中，算法可能难以精确区分物体的轮廓区域。这种边界误差不仅影响了分割结果的精确性，还可能对后续的动态特征点的判断带来负面影响。因此本文通过结合深度图信息进一步提升掩码的精度。

现阶段滑动窗口技术被用于补全深度图中的异常值，然而这种方法存在显著的缺点。滑动窗口算法的主要思想是通过一个固定尺度的窗口通过固定的步长遍历图像中每个区域，以滑动的方式截取图像从而达到补全异常值的目的。然而，采

用这种方法遍历全图时，固定窗口尺寸难以适应不同大小的异常区域。如果设定窗口小于异常区域，则无法捕捉足够有效像素，导致深度图噪点补全失败。如果设定窗口远大于异常区域，则会导致除了深度图噪点被补全之外，背景中的深度也会被错误补全。如图3-6红色框所示，使用传统滑动窗口补全深度图像中的异常值会错误地将背景区域补全，造成深度图像细节缺失。



(a) (b)

图 3-6 传统的深度信息补全。(a) 原始深度图像；(b) 传统滑动窗口处理后的深度图像

本文通过改进传统的滑动窗口技术，修复深度图中由于噪声或传感器限制导致的零值像素问题，并且避免深度图背景细节被赋予错误深度值的问题。针对深度值为 0 的候选像素，提出了一种基于两层滑动窗口的填充方法，以提高深度图的完整性和鲁棒性。该方法通过构建以候选像素为中心的两个邻域窗口，将其划分为内层和外层两个区域分别统计邻域内非零像素的频度。只有当该像素内外层计数器都超过对应阈值的时候才进行深度补全，如果外层计数器没有超过对应阈值则不进行深度补全，从而避免深度图背景细节被错误赋予深度值的问题。最终利用高斯权重矩阵对候选像素进行填充。

如图3-7所示，算法逐像素遍历深度图。对于深度值非零的像素直接保留其原始深度值。对于深度值为零的候选像素，基于其位置定义 5×5 的滑动窗口邻域。该邻域分为两个子区域：内层窗口和外层窗口。内层窗口由候选像素周围的 3×3 区域组成，包含 8 个紧邻像素。外层窗口则由整个 5×5 区域的边界像素组成，包含候选像素外围的 16 个像素。通过遍历窗口内的每个像素，记录所有非零像素的深度值并统计其出现频度。同时，根据像素所在的区域分别累积内层和外层窗口的非零像素数量。当内层窗口或外层窗口的非零像素数量超过预设的阈值时，认为候选像素具备足够的上下文信息进行填充。否则，若内层或外层的非零像素数量未达到阈值，则保持候选像素深度值为零以避免误填充带来的噪声引入。找到要填充的候选像素后，使用窗口大小 5×5 创建一个二维高斯分布矩阵。

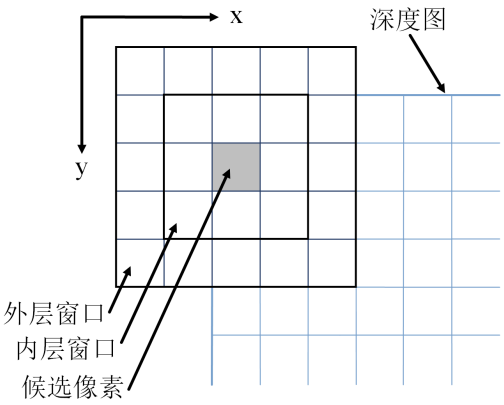


图 3-7 深度图处理方法

窗口中心点权重最大，越靠近边缘权重越小。高斯权重矩阵公式：

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3-2}$$

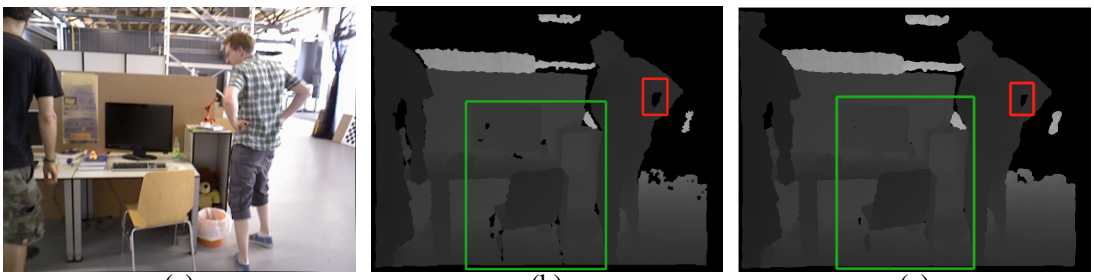
其中，(x,y) 是窗口中每个像素的相对坐标，σ 控制权重分布的范围。在高斯平滑过程中，标准差 σ 决定了高斯分布的扩散范围，从而影响邻域像素的权重分配。根据高斯分布的数学性质，99.73% 的权重集中在 [−3σ，3σ] 范围内。为了确保高斯分布的主要权重能够被窗口覆盖，窗口的大小为 6σ。因此 σ 与窗口大小 $Msize$ 的关系为：

$$\sigma=\frac{Msize}{6} \tag{3-3}$$

因此对于 5×5 大小的外层窗口，σ 的值取为 0.83。遍历外层窗口与内层窗口内的所有非零像素，使用高斯权重乘以其深度值，通过加权和并归一化得出候选像素的填充值：

$$d_{pixel}=\frac{\sum_{i,j\in N}G(i,j)\cdot D(i,j)}{\sum_{i,j\in N}G(i,j)} \tag{3-4}$$

其中 N 为邻域内的非零像素，$D(i,j)$ 是邻域像素的深度值。

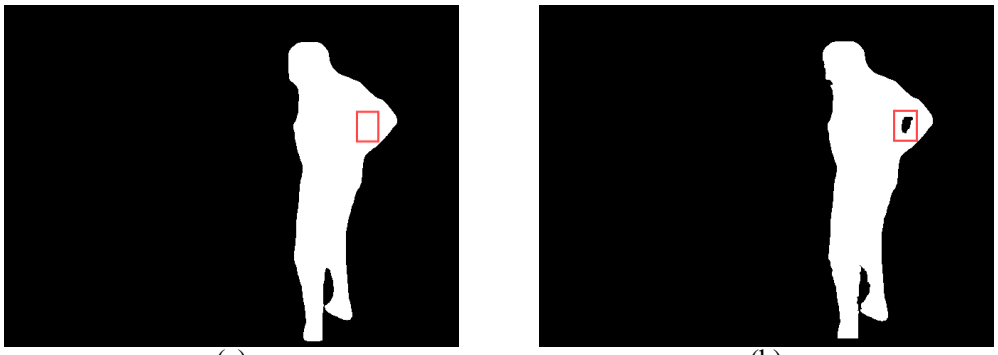


(a) (b) (c)

图 3-8 改进的深度信息补全。(a) 对应的 RGB 图像；(b) 原始深度图像；(c) 改进滑动窗口处理后的深度图像

如图3-8所示，改进后的算法在补全深度图时表现出明显优势。算法不仅有效地去除了深度图中的噪点（如绿框所示），而且能够较好地保留背景细节信息（如红框所示）。这一改进使得深度图在精度和细节保留方面均得到了显著提升。

在实例分割过程中，算法有时会错误地将背景区域划分为物体的一部分，这可能导致物体掩码的准确性下降。这种误分割会导致在判断物体运动状态时，背景区域的特征点被错误地识别为物体上的特征点，从而影响后续处理的准确性和效果。为了解决这一问题，本文针对每个实例分割物体的掩码进行精细化处理。首先结合该物体区域内深度值的均值与方差，动态确定前景的深度范围。将深度值限制在均值的上下两倍标准差范围内，以此过滤掉异常的深度值。在掩码区域内通过对每个像素的深度值进行判断，将符合深度范围的像素标记为前景从而生成更加精细化的掩码。处理后的结果如图3-9所示，红框中成功地将物体与背景区域从掩码中有效区分开来。



(a) (b)

图 3-9 掩码处理。(a)粗略的掩码；(b)精细化的掩码

3.3 多视图几何物体真实运动状态检测

3.3.1 基于深度与极线距离的联合异常检测

先验动态物体在实际情况中不一定处于运动状态，例如室内静止不动的人。在纹理信息丰富的情况下可以提取较多的特征点，如果仅依靠类别将先验动态物体上的特征点全部去除，虽然可能会误删静止物体上的特征点但仍然有大量特征点可用于匹配。然而在纹理信息较少的背景下，例如静止的人靠在室内白墙上，简单地按照类别删除特征点会导致丢失大量静态特征点从而导致特征点数量不足等问题。由于 RDS-SLAM 中的关键帧队列策略是直接把关键帧中的物体认为是动态物体进行处理，实际上错误地将很多静态的物体标注为动态，进而误删除了静态物体上特征点对应的地图点。针对静态物体误标注为动态的问题，本文增加几何模块与语义模块共同判断物体真实运动状态。如图3-10所示显示了两帧之间的对

极几何约束。

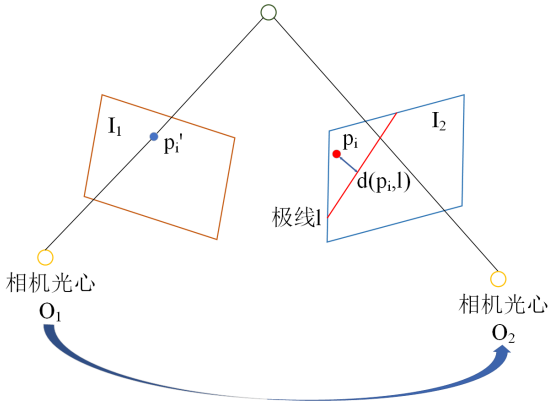


图 3-10 对极几何判断特征点运动状态

设 p_i' 和 p_i 分别表示前一帧和当前帧中匹配的特征点。

$$\begin{aligned} p_i' &= \left[x_{p_i'}, y_{p_i'}, 1\right] \\ p_i &= \left[x_{p_i}, y_{p_i}, 1\right] \end{aligned} \tag{3-5}$$

其中，x_{p_i} 和 y_{p_i} 为当前帧匹配特征点的像素坐标，极线 l 的公式如下：

$$l = \left[X_{p_i'}, Y_{p_i'}, Z_{p_i'}\right]^T = F p_i' \tag{3-6}$$

因此，极线 l 为：

$$l = F\left[x_{p_i'}, y_{p_i'}, 1\right] \tag{3-7}$$

极线 l 的单位法向量为：

$$\hat{n}_l = \frac{n_l}{\|n_l\|} = \frac{[-Y_{p_i}, X_{p_i}]^T}{\sqrt{X_{p_i}^2 + Y_{p_i}^2}} \tag{3-8}$$

其中 $X_{p_i'}$，$Y_{p_i'}$，$Z_{p_i'}$ 表示极线向量的坐标，F 表示基础矩阵。则匹配点 p_i 到极线 $l_{p_i'}$ 的矢量距离表示为：

$$d_{p_i,l} = \frac{|p_i^T F p_i'|}{\sqrt{||X_{p_i'}||^2 + ||Y_{p_i'}||^2}} \cdot \hat{n}_l \tag{3-9}$$

代入可得：

$$d_{p_i,l} = \frac{|p_i F (p_i')^T|}{\sqrt{||X_{p_i'}||^2 + ||Y_{p_i'}||^2}} \cdot \frac{[-Y_{p_i'}, X_{p_i'}]^T}{\sqrt{X_{p_i'}^2 + Y_{p_i'}^2}} \tag{3-10}$$

其中 $d_{p_i,l}$ 服从高斯分布：

$$d_{p_i,l} \sim \mathcal{N}(0, \Omega) \tag{3-11}$$

其中，Ω 表示协方差矩阵，其值根据特征点提取过程中所在的金字塔层级进行确定。如图3-11所示为多层金字塔结构，多层金字塔结构中不同层的特征点往往具有

不同的尺度和分布特性。

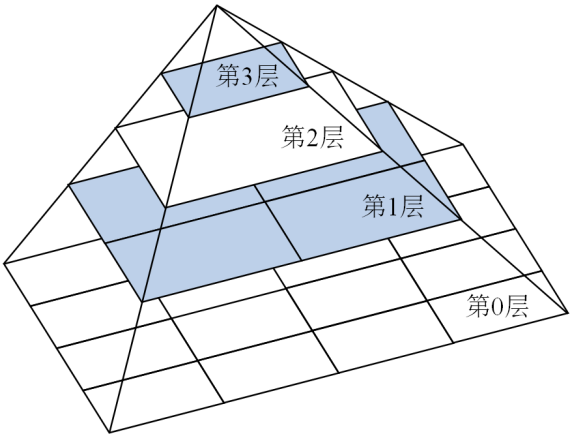


图 3-11 图像金字塔示意图

不同层的特征点对内积的贡献程度并不一致，高层的特征点通常对应更大的特征范围，而低层特征点则更加细致。因此在通过匹配特征点计算极线距离时，如果对所有层的特征点都使用相同的阈值进行判断并不准确。为了解决这一问题，在计算内积时引入了协方差矩阵对特征点进行加权。通过协方差矩阵的加权，可以根据不同特征点的分布特性对其匹配贡献进行调整。在图像金字塔第 n 层的特征点到极线距离的协方差矩阵 Ω 为：

$$\Omega = (s^n \times p)^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3-12}$$

其中，s 表示图像金字塔中每一层的尺度缩小因子，在 ORB-SLAM3 中设定为 1.2。p 表示图像金字塔第 0 层的标准差，在 ORB-SLAM3 中设定为 1 个像素。

加权后的极线距离误差 $\mathcal{E}_d$ 为：

$$\begin{aligned} \mathcal{E}_d &= d_{p_i,l}^T \Omega^{-1} d_{p_i,l} \\ &= (\Omega^{-\frac{1}{2}} d_{p_i,l})^T (\Omega^{-\frac{1}{2}} d_{p_i,l}) \end{aligned} \tag{3-13}$$

其中，$(\Omega^{-\frac{1}{2}} d_{p_i,l}) \sim \mathcal{N}(0, I)$ 为二维标准正态分布，因此整体物体的极线距离误差 $\mathcal{E}_d$ 符合自由度为 2 的卡方分布。此时不同层金字塔结构中的极线距离被归一化，当特征点 p_i' 到极线的加权内积 $\mathcal{E}_d$ 大于一个阈值 γ 时，则认为该特征点 p_i' 为动态特征点。在这里认为内点被错误地判定为外点的概率为 5%，因此阈值 γ 被设定为 5.99。

尽管极线距离是动态特征点检测中的重要指标，其反映了特征点在多视图几何约束下的偏离程度，但仅依赖极线距离并不足以准确判断动态特征点。这是因为极线距离在某些场景中可能受到其他因素的干扰，例如图像噪声、匹配误差或摄像头运动模式的复杂性，导致静态特征点也可能出现一定的极线距离。虽然静

态物体和动态物体中匹配特征点对产生的极线距离会有较大差异，但是这种差异很难确定阈值将两种情况分开。在这些情况下，仅依赖极线距离可能导致漏检或误检。与此同时，深度信息提供了一个更直观的物理属性，通过分析特征点在多帧图像中的深度变化可以进一步区分静态物体与动态物体。将深度差与极线距离结合起来，通过联合异常检测能够更全面地捕捉特征点的动态性，利用两者的互补性有效提升检测的鲁棒性和准确性。

基础矩阵 F 描述了两幅图像中对应点对之间的几何约束。基本矩阵 F 可以进一步分解为相机之间的相对姿态信息，即旋转矩阵 R 和平移向量 t。以下具体描述了从基本矩阵 F 到相对运动参数 R 和 t 的分解过程。根据公式2-14可将基础矩阵 F 转化为本质矩阵 E。本质矩阵 E 是一个 3×3 的矩阵，具有以下奇异值分解形式：

$$E = U\Sigma V^{\top} \tag{3-14}$$

其中 U、V 均为正交矩阵，$\Sigma = \text{diag}(\sigma, \sigma, 0)$ 为本质矩阵的奇异值矩阵。本质矩阵的前两个奇异值相等，第三个奇异值为零。根据奇异值分解，可以从 E 中提取出旋转矩阵 R 和平移向量 t。引入一个辅助矩阵 W，其定义为：

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3-15}$$

结合奇异值分解，本质矩阵可以分解为：

$$\begin{aligned} R_1 &= UWV^{\top} \\ R_2 &= UW^{\top}V^{\top} \\ t &= u_3 \end{aligned} \tag{3-16}$$

其中 u_3 是 U 的第三列向量，表示平移方向的单位向量。分解结果存在四种可能性，即旋转矩阵 R 与平移向量 t 的组合为 (R_1, t)、$(R_1, -t)$、(R_2, t) 和 $(R_2, -t)$。然而只有一种解中，使得点 P_2 在两个相机下均具有正深度。因此，只需将任意点代入以上四种解中检查该点在两个相机下的深度值，即可判断哪种解是正确的。通过上述过程，基础矩阵 F 被分解为旋转矩阵 R 和平移向量 t，从而实现了相机之间的相对姿态估计。

在相机运动的计算过程中假设在第一帧中，某特征点的像素坐标为 $p_1 = [u_1, v_1, 1]^{\top}$，其对应的深度为 d_1，则该特征点在相机坐标系中的三维坐标可以通过内参矩阵 K 的逆矩阵计算为：

$$P_1 = d_1 \cdot K^{-1}[u_1, v_1, 1]^{\top} \tag{3-17}$$

其中，$P_1 = [x_1, y_1, z_1]^\top$ 表示该特征点的三维坐标。在第二帧中，相机的相对运动可以表示为一个旋转矩阵 R 和一个平移向量 t，通过相机运动模型可以将第一帧中的三维坐标 P_1 转换到第二帧中的三维坐标 P_2，即：

$$P_2 = R \cdot P_1 + t \tag{3-18}$$

其中 $P_2 = [x_2, y_2, z_2]^\top$ 是特征点在第二帧中的三维坐标。在第二帧中，特征点的像素坐标可以通过其三维坐标投影到图像平面上计算得到。令 $p_2 = [u_2, v_2, 1]^\top$ 表示该特征点的像素坐标，结合相机内参矩 K 和三维坐标 P_2，有：

$$[u_2, v_2, 1]^\top = \frac{1}{z_2} \cdot K \cdot P_2 \tag{3-19}$$

其中 z_2 为特征点 p_2 的深度值。在已知内参矩阵 K 和像素坐标 p_2 的情况下，可通过反推计算得到该特征点在第二帧中的深度：

$$\hat{z} = z_2 \tag{3-20}$$

通过上述计算过程，可以得到第二帧中该特征点的三维坐标 P_2 和对应的深度信息 $\hat{z}$，从而实现特征点在不同帧中的三维位置和深度映射。深度差 Δz 是由第二帧对应深度图中该特征点实际深度值 z 与根据相机位姿信息计算所得的估计深度值 $\hat{z}$ 计算得到：

$$\Delta z = |z - \hat{z}| \tag{3-21}$$

对于静态物体，其投影后的估计深度值跟真实深度值较为接近。因此静态物体的 Δz 通常较小，与动态物体存在显著差异。在这里动态物体的阈值 θ 设定为 1.12。

仅依赖极线距离或深度信息进行判断存在局限性。极线距离反映了点与极线之间的几何关系，但无法有效捕捉三维空间中深度的不一致性。特别是在动态场景中，运动物体可能会导致几何关系依然满足但深度存在显著偏差。同样，仅依赖深度信息虽然可以检测深度上的异常，但缺乏几何约束的辅助，在某些情况下可能误将正常的深度变化识别为异常。

为了同时捕捉极线距离误差 $\mathcal{E}_d$ 和深度差 Δz 的异常情况，本文提出了物体第 i 个特征点的联合异常检测得分 S_i：

$$S_i = \begin{cases} 0, & \text{其他} \\ 1, & \mathcal{E}_d > \gamma \text{且} \Delta z > \theta \end{cases} \tag{3-22}$$

当极线距离超过阈值 γ 则认为该特征点几何异常得分，当深度差超过阈值 θ 则认

为该特征点深度异常得分。则整个物体的联合异常检测得分为：

$$S_{\text{object}} = \frac{1}{N}\sum_{i=1}^{N} S_i \tag{3-23}$$

其中，N 为该物体上匹配特征点的总数。整个物体的联合异常检测模型得分 S_{object} 越接近 1 越可能处于运动状态；反之，越接近 0 越可能处于静止状态。

3.3.2 基于跟随式动态窗口的物体运动状态校验

深度与极线距离的联合异常检测依赖于两帧之间的相对运动。然而现有的 ORB-SLAM3 系统虽然能够估计每个关键帧的粗略位姿，但在包含动态物体的场景中，其估计的位姿往往存在动态误差。为解决上述问题，本文创新性地使用图片中较少的先验静态信息计算出较为可靠的基础矩阵 F，以提升对极几何判断的准确性。首先利用除了实例分割检测出的物体之外的背景中的先验静态特征点的匹配结果得到匹配的特征点对。其次使用这些匹配点对，通过最小化重投影误差估计基础矩阵 F 以描述其几何关系。

在 ORB-SLAM3 中关键帧的生成依赖于多个条件，旨在确保系统在环境中能够持续获得有效的位置信息，同时避免冗余计算。通常当相机的位移或旋转角度超过一定阈值，或当相机的视角发生显著变化时，系统会生成新的关键帧。如果当前帧在局部地图中增加了新的特征点，或当前帧与最近的关键帧之间的视差较大时，也会触发关键帧的生成。因此出现在本关键帧中物体，可能在上一关键帧中还没有出现。本关键帧中物体突然出现可能是该物体静止但相机视角转动使得物体出现在该关键帧中，也可能是该物体处于运动状态进入相机视角。因此，无法仅与上一关键帧进行联合异常检测就判断出该物体的真实运动状态。

关键帧中物体突然出现导致运动状态无法判断引起误差的问题，仅仅在两个关键帧中进行物体的几何校验是不够的，还要加入对普通帧的处理以避免出现以上影响。同时，ORB-SLAM3 中并没有保存普通帧，在跟踪阶段处理完普通帧后就会进行丢弃。因此关键帧队列头部关键帧周围的普通帧已经被丢弃，并不能进行关键帧队列头部关键帧与普通帧的几何校验。并且实例分割过程较为耗时，在实例分割的同时，SLAM 系统中仍会有新的普通帧不断进入并被处理。为了解决以上问题，本文针对关键帧队列设计了一种扩展机制。即在关键帧队列中最后一个关键帧的基础上引入一个固定大小为 5 的跟随式动态窗口，用于存储该关键帧之后依次进入的 5 个普通帧。该设计旨在保留这些普通帧的信息，使得关键帧队列最后一个关键帧中有突然出现的物体时，可以与跟随式动态窗口中的普通帧进行联合异常检测从而判断该物体的真实运动状态。需要注意的是，跟随式动态窗口的大小设置需兼顾多种因素。一方面，其大小不宜过大以免超出后续关键帧的范

围；另一方面，需要保持与执行对极几何校验的关键帧之间的适当距离，以便准确判断物体的运动状态。因此，本研究将跟随式动态窗口的大小设定为 5，以在计算效率与信息完整性之间取得平衡。如果关键帧中包含的物体较少实例分割的处理速度会显著加快，从而导致实例分割完成时进入的普通帧数量少于 5 个。在这种情况下，跟随式动态窗口将动态调整，仅存储已进入的普通帧直至该窗口被新的关键帧刷新。

假设处于某时刻的跟随式动态窗口如图3-12所示。绿色表示已经实例分割并完成物体运动状态检测的关键帧。该时刻选取关键帧队列的第一个未进行实例分割的关键帧 KF_3 和最后一个关键帧 KF_{11}（由黄色表示）进行实例分割并精细化掩码后，KF_3 直接与前一个相邻的关键帧进行联合异常检测。而最后一个关键帧 KF_{11} 与前一个相邻的关键帧进行联合异常检测后，如果 KF_{11} 关键帧上所有物体中都存在与前一个相邻关键帧中特征点匹配的特征点，则认为 KF_{11} 关键帧上所有物体在前一个相邻关键帧中都出现过。此时仅通过与前一个相邻关键帧进行联合异常检测就可以判断 KF_{11} 关键帧上所有物体的真实运动状态，不需要 KF_{11} 关键帧与跟随式动态窗口中的普通帧（由灰色表示）进行联合异常检测。然而，如果 KF_{11} 关键帧上某个物体中所有特征点在 KF_{110} 关键帧中都没有与之匹配的特征点，则认为该物体在前一个相邻关键帧中没有出现过。此时 KF_{11} 关键帧会从跟随式动态窗口中选择相距最远的 CF_{27} 普通帧开始检测该普通帧中是否存在与物体中特征点匹配的特征点。若存在，则 KF_{11} 关键帧上该物体将与 CF_{27} 普通帧进行联合异常检测，判断该物体真实运动状态。若不存在，则 KF_{11} 关键帧继续与跟随式动态窗口中前一个普通帧进行校验。

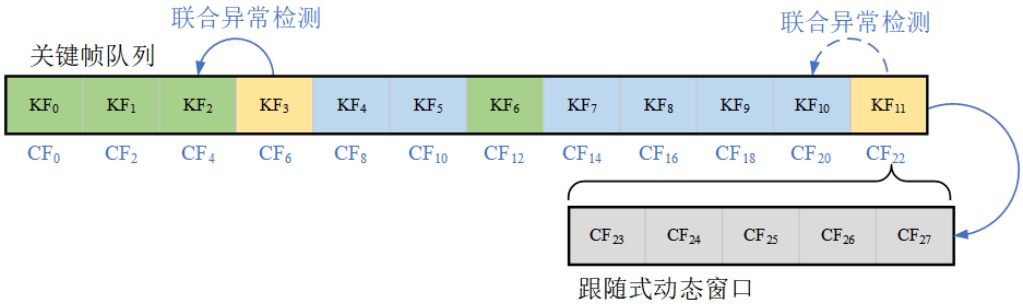


图 3-12 跟随式动态窗口

关键帧队列尾部关键帧多视图几何校验的伪代码流程如算法3-1所示。首先，根据语义模块已经得到了关键帧队列中头部关键帧和尾部关键帧实例分割的语义信息。根据关键帧的 ID 取得前一个相邻关键帧，对于头部关键帧中的物体直接依次对每个物体上的特征点求前一个相邻关键帧上的匹配特征点。通过每个物体上的匹配特征点对进行联合异常检测。如果整个物体的联合异常检测超过阈值 0.6，

则认为该物体真实处于运动状态。如果某个物体上的特征点在前一个相邻关键帧上没有任何匹配特征点对，由于不能分清是因为相机自身运动还是物体运动导致，因此该物体在此次判断中不做处理。其次，对于关键帧队列尾部的关键帧，跟关键帧队列中头部关键帧进行同样的判断。与之不同的是，如果某个物体上所有特征点在前一个相邻关键帧上都没有匹配的特征点，则认为当前关键帧队列的最后一个关键帧中观测到的物体未能在相邻的前一个关键帧中被观测到。这时从跟随式动态窗口的最后一个普通帧开始逐帧查询是否存在该物体的匹配特征点。如果在最后一个普通帧中存在匹配特征点，则在关键帧队列尾部关键帧和跟随式动态窗口的最后一个普通帧之间进行该物体的联合异常检测。如果在最后一个普通帧中没有匹配特征点，则继续在跟随式动态窗口中向前查找，直到出现匹配特征点对。

算法 3-1: 关键帧队列尾部关键帧几何校验模块伪码示例

```
输入: 1) backKF; //关键帧队列的尾部关键帧
      2) slidingWindow; //尾部关键帧的跟随式动态窗口
      3) threshold; //联合异常检测阈值
输出: 物体真实运动状态
prevKeyframe = getPreviousKeyframe(backKF.mImID);          // 获取前一个相邻关键帧
for dynamicObject in backKF.priorDynamicObjects do
    featurePoints = dynamicObject.featurePoints;
    matchingFeaturePoints = findMatchingFeaturePoints(prevKeyframe, featurePoints);
    if matchingFeaturePoints.isEmpty() then
        for frame : slidingWindow.reverse() do
            matchingFeaturePoints = findMatchingFeaturePoints(frame, featurePoints);
            if !matchingFeaturePoints.isEmpty() then
                break;                                     // 找到匹配点后停止查找
    if !matchingFeaturePoints.isEmpty() then
        outlierScore = computeJointOutlierScore(matchingFeaturePoints);
        if outlierScore > threshold then       // 如果联合异常检测超过阈值，标记为动态
            dynamicObject.markAsMoving();
return
```

3.4 地图点移动概率

由于实例分割有时候无法完全检测，可能会出现错检和漏检的情况。同时，基于深度与极线距离的联合异常检测方法在绝大多数情况下能够有效执行，但在特定情况下仍可能发生误判。因此仅依赖单个关键帧的语义信息进行判断是不够的。为了解决这个问题，本文引入了地图点移动概率模型[64]来连续关键帧的语义信息，

从而增强系统的鲁棒性。每个地图点在当前时刻的移动概率如图3-13所示：

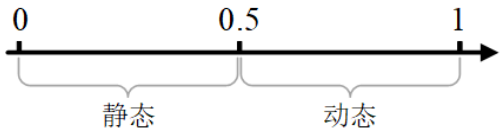


图 3-13 地图点移动概率

若地图点移动概率趋近于 1，那么该地图点就更可能处于动态。若地图点移动概率趋近于 0，那么该地图点就更可能处于静态。每个地图点初始概率设为 0.5。考虑到实例分割不是完全准确，因此假设地图点的真实运动状态与实例分割结果认为的地图点运动状态的关系如下所示：

$$\begin{aligned} p(z_t = d \mid m_t = d) &= \alpha \\ p(z_t = s \mid m_t = d) &= 1 - \alpha \\ p(z_t = s \mid m_t = s) &= \beta \\ p(z_t = d \mid m_t = s) &= 1 - \beta \end{aligned} \tag{3-24}$$

其中 d 代表动态，s 代表静态。$m_t = d$ 表示 t 时刻地图点的真实运动状态为动态，$z_t = d$ 表示 t 时刻该地图点对应的特征点位于实例分割中的物体掩码上。本文将 α 和 β 值设为 0.9。那么当前时刻地图点的移动概率依赖于之前时刻的实例分割结果以及地图点的初始状态。那么移动概率的更新问题可以表示为：

$$\begin{aligned} bel(m_t) &= p(m_t \mid z_{1:t}, m_0) \\ &= \mu p(z_t \mid m_t, z_{1:t}, m_0) p(m_t \mid z_{1:t-1}, m_0) \\ &= \mu p(z_t \mid m_t) p(m_t \mid z_{1:t-1}, m_0) \\ &= \mu p(z_t \mid m_t) \overline{bel(m_t)} \end{aligned} \tag{3-25}$$

当前时刻地图点移动概率 $bel(m_t)$ 转换成依赖于预测的地图点移动概率 $\overline{bel(m_t)}$。由于现有技术手段在实时监测物体运动状态方面存在局限性，难以有效捕捉目标对象的瞬间运动状态突变现象。因此假设状态转移概率 $p(m_t = d \mid m_{t-1} = s) = 0$，$p(m_t = d \mid m_{t-1} = d) = 1$，$\mu = bel((m_t = d) + bel(m_t = s))/2$。地图点处于动态的移动概率为：

$$\overline{bel}(m_t = d) = p(m_t = d \mid m_{t-1} = d)\, bel(m_{t-1} = d) \tag{10}$$

在更新地图点移动概率之后，将概率值小于 0.5 的地图点认为是静态物体对应的地图点，概率值大于 0.5 的地图点认为是动态物体对应的地图点。在跟踪阶段只处理静态地图点对应的特征点，忽略动态特征点匹配，从而减少动态物体对跟踪精度的影响。

3.5 实验分析

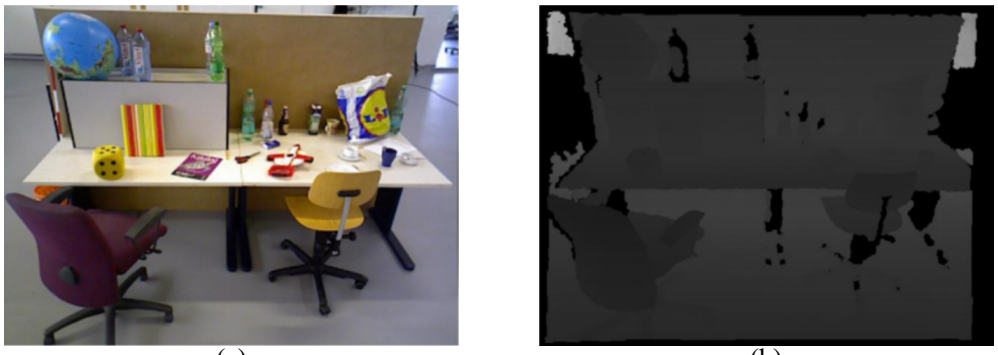
为了验证第三章算法的有效性，本实验以 ORB-SLAM3 为基础框架实现 GeoYOLO-SLAM 算法。本文实验使用笔记本电脑作为实验平台，在 Ubuntu18.04 操作系统下进行。使用了 C++11、python2.7、python3.8、ROS、g2o、Eigen、ultralytics 等作为软件开发环境，并使用 COCO 数据集[65] 上训练好的 YOLOv8 实例分割模型。本文实验分析采用 TUM 数据集和 Bonn 数据集。实验平台配置如表3-1所示:

表 3-1 实验平台配置

环境配置	型号和参数
CPU	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
GPU	NVIDlA GeForce GTX 1050 Ti
内存	16G
操作系统	Ubuntu 18.04
CUDA	10.2

本文采用在 COCO 数据集上预训练的 YOLOv8 实例分割模型，该数据集包含 80 类常见室内外物体的 33 万张标注图像，提供像素级实例分割掩码、边界框及语义标签。COCO 数据集的场景多样性和多尺度遮挡标注特性，使模型能够精准识别复杂光照条件下的物体边界，特别适用于室内环境中的动态目标检测。其丰富的类别覆盖与密集标注为实例分割模型提供了跨场景泛化能力，保障了物体掩码提取的几何完整性与语义一致性。

TUM 数据集由慕尼黑工业大学（Technical University of Munich，TUM）开发，是评估 V-SLAM 和视觉里程计算法中常用的基准数据集之一。TUM 数据集通过 Microsoft Kinect 相机传感器采集，涵盖了 39 个各具特色的室内场景序列。并通过高精度系统采集相机设备的实际位姿，提供相机的真实移动轨迹。如图3-14所示，TUM 数据集中每个系列都包括了 RGB 图像和与之对应的深度图。

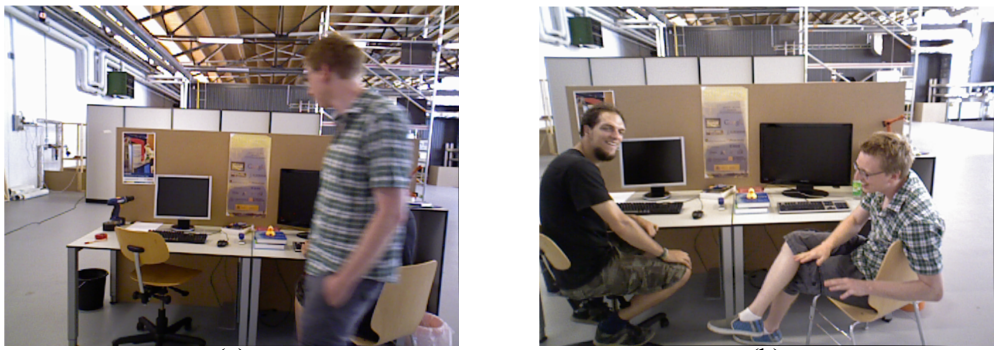


(a) (b)

图 3-14 TUM 数据集。(a)RGB 图像；(b)对应深度图像

TUM 数据集是一个综合性测试数据集，广泛应用于衡量算法在各种环境条件下的性能。它通过细致划分场景类型，旨在检验算法在多种环境中的鲁棒性。数据集依据场景中目标的种类及其运动模式，被归类为两大类别：动态场景以及静态场景。TUM 数据集中的 freiburg3 场景为动态环境的测试提供了丰富的数据。为了在这些场景中测试 V-SLAM 算法的多样性，TUM 数据集使用了四种互不相同的相机运动模式，其中包括半球面运动模式的 halfsphere 序列、大范围旋转的 rpy 序列、静止模式的 static 序列以及沿坐标轴平移的 xyz 序列。

freiburg3 场景记录了两位实验者在室内环境中活动的场景，按照动态目标的运动特点，可以将数据进一步划分为高动态场景和低动态场景两类。在高动态场景中，walking 系列数据集尤为突出。该系列记录了两个实验者在一个固定的室内环境中执行多种活动，如行走、交谈、搬动家具等，同时相机进行不同类型的运动。由于这些动态目标的快速变化及其显著存在，walking 系列成为评估 V-SLAM 算法在复杂、高动态环境中的重要数据集。如图3-15(a)所示，以 walking 系列为例的高动态场景记录了两个实验者在一个固定的室内环境中执行多种活动。在低动态场景中，sitting 系列数据集尤为突出。该系列记录了两个实验者在一个固定的室内环境中进行各种坐姿活动。如图3-15(b)所示，以 sitting 系列为例的低动态场景展示了实验者坐着时的微小运动。这些设计使得 TUM 数据集成为测试 V-SLAM 算法的强大工具。



(a) (b)

图 3-15 TUM 数据集中的动态场景。(a)高动态场景；(b)低动态场景

Bonn 数据集（Bonn RGB-D Dynamic Dataset）是由波恩大学（University of Bonn，Bonn）计算机科学与人工智能研究所制作。该数据集使用 ASUS Xtion Pro LIVE RGB-D 传感器进行录制，格式与 TUM 数据集相同。Bonn 数据集包含了 24 个序列，其中包含具有高度动态特征的场景。高度动态特征的场景中可能包含了多个动态对象，如图3-16所示多个实验者在相机视角前走动、实验者搬动箱子运动等等。为了提供精确的地面真实数据，Bonn 数据集结合了来自 Optitrack Prime 13

动作捕捉系统的标定数据。

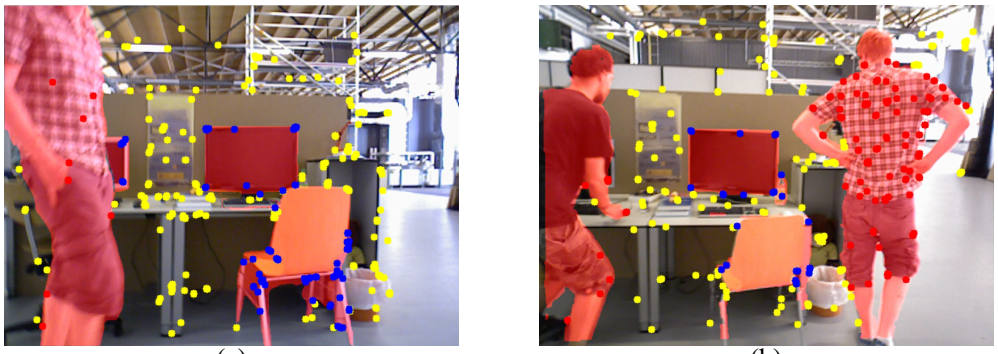
(a) (b)

图 3-16 Bonn 数据集中的动态场景。(a)实验者走动；(b)实验者搬动箱子

3.5.1 物体真实运动状态检测实验

为了验证动态场景下本文提出的基于深度与极线距离的联合异常检测的有效性，本节物体真实运动状态检测实验是基于 TUM 数据集中的 walking_xyz 序列和 sitting_xyz 序列作为实验对象。分别在该序列上运行 ORB-SLAM3 算法与 GeoYOLO-SLAM 算法将两者提取特征点进行对比。

本文 GeoYOLO-SLAM 算法的实验结果图如下，可以看出联合异常检测能够很好的将物体的真实运动状态识别出来。其中，动态特征点使用红色圆点表示，静态特征点蓝色圆点表示。物体之外的背景中的静态特征点用黄色圆点表示。可以看出，GeoYOLO-SLAM 算法不是将所有先验动态物体都认定为动态，而是经过联合异常检测有效识别真正处于运动状态的物体上的特征点。如图3-17所示，绕着桌子运动的实验者身上的特征点均被标注为动态，屏幕、椅子、瓶子等物体经过联合异常检测被标注为静态。

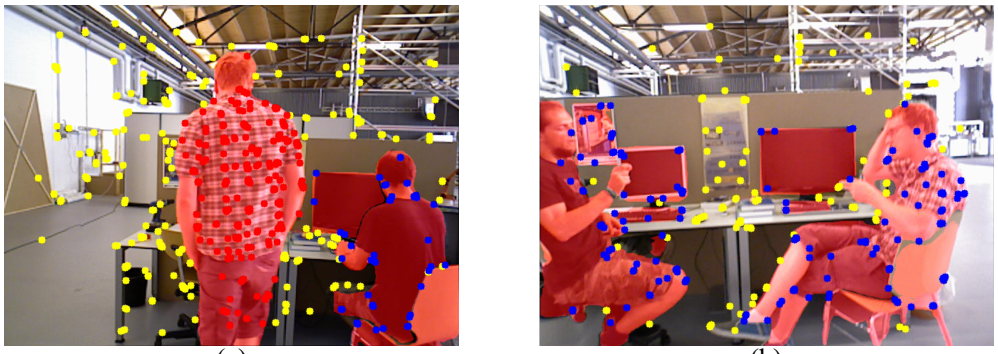


(a) (b)

图 3-17 实验者均处于动态时 GeoYOLO-SLAM 联合异常检测实验结果。

(a) 检测出一位运动的实验者；(b) 检测出两位运动的实验者

当相机视角中存在多位实验者时，GeoYOLO-SLAM 通过联合异常检测判断实验者的真实运动状态。如图3-18所示，当实验者运走动时会被检测出动态，当实验者坐下静止的时候将被检测为静态：



(a) (b)

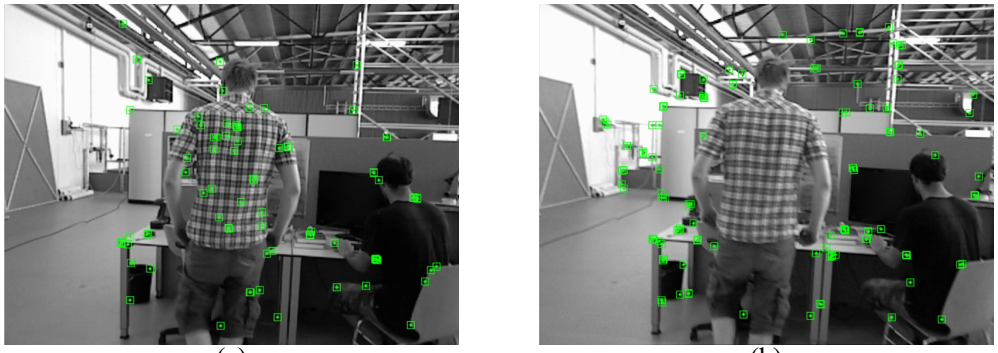
图 3-18 实验者既有运动状态又有静止状态 GeoYOLO-SLAM 联合异常检测实验结果。(a) 检测出一位静止的实验者；(b) 检测出两位静止的实验者

如图3-19所示，展示了将 GeoYOLO-SLAM 剔除动态特征点后的结果与 ORB-SLAM3 中提取的特征点结果进行对比。图中的特征点提取结果以绿色矩形框包围的圆点形式呈现。由于 ORB-SLAM3 缺乏物体运动状态检测功能，图中行走的人物身上也被标记了大量特征点。这些特征点被纳入匹配过程中，可能会导致在动态物体与静态场景共存的情况下发生匹配错误，从而影响系统精度。而 GeoYOLO-SLAM 通过滤除处于运动状态的物体上的特征点，验证了其能够有效判断物体的真实运动状态。

(a) (b)

图 3-19 ORB-SLAM3 与 GeoYOLO-SLAM 特征点对比。(a) ORB-SLAM3；(b) GeoYOLO-SLAM

当一位实验者静止坐着、另一位实验者处于运动状态的特征点对比如图3-20所示。ORB-SLAM3 会提取两位实验者身上的特征点，而 GeoYOLO-SLAM 只滤除处于运动状态的实验者身上的特征点，保留了静止的实验者身上的特征点。



(a) (b)

图 3-20 ORB-SLAM3 与 GeoYOLO-SLAM 特征点对比。(a) ORB-SLAM3；(b) GeoYOLO-SLAM

3.5.2 跟随式动态窗口消融实验

为了验证动态场景下本文提出的多视图几何校验中跟随式动态窗口的有效性，本节实验是基于 TUM 数据集中的 walking_rpy 序列作为实验对象。本序列中存在动态物体突然出现在相机视角的场景，如图3-21所示。穿着栅格衬衫的实验者本来没有出现在相机视野，随着时间的推移逐渐从左侧进入相机视野中。

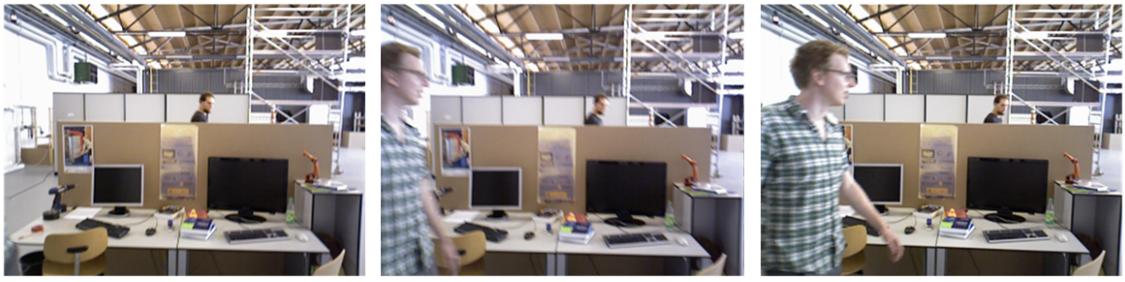
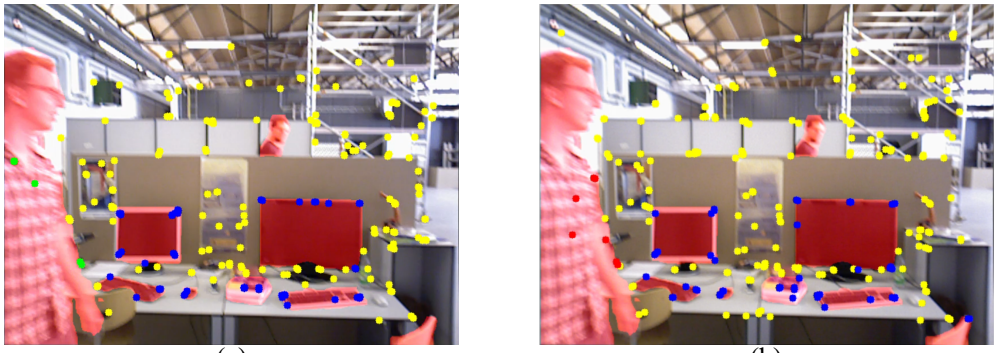


图 3-21 实验者逐渐出现在相机视野

为了验证跟随式动态窗口的有效性，在 walking_rpy 序列上进行了消融实验。实验方法为保持 SLAM 整体框架不变，不使用跟随式动态窗口进行试验，增加跟随式动态窗口进行试验。如图3-22(a)所示为不使用跟随式动态窗口的实验结果，如图3-22(b)所示为使用跟随式动态窗口的实验结果。由于该关键帧的前一个关键帧中还未出现穿着栅格衬衫的实验者，因此图3-22(a)所示本关键帧中穿着栅格衬衫的实验者新产生的地图点对应的特征点无法与上一关键帧进行匹配，使得该实验者的真实运动状态无法判断。因此图3-22(a)中该实验者掩码上的特征点显示为初始状态，从而标记为绿色。图3-22(b)中虽然本关键帧新产生的地图点对应的特征点无法与上一关键帧进行匹配，但由于跟随式动态窗口，本关键帧与跟随式动态窗口中的普通帧进行匹配。因此图3-22(b)中该实验者的掩码上的特征点被判定为动态，从而标记为红色。



(a) (b)

图 3-22 跟随式动态窗口消融实验。(a) 不使用跟随式动态窗口；(b) 使用跟随式动态窗口

3.5.3 视觉里程计实验

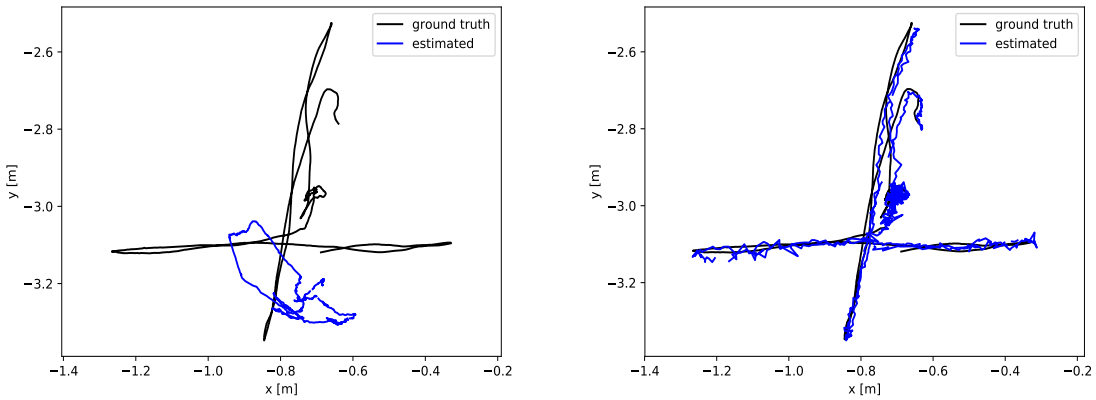
为了评估 GeoYOLO-SLAM 算法性能，本文使用可扩展视觉里程计工具（Extensible Visual Odometry，EVO）进行位姿对比分析，并采用常用的误差度量方法：绝对轨迹误差（Absolute Trajectory Error，ATE）。ATE 用于衡量估计轨迹与真实轨迹之间的偏差，通过计算每个时刻估计轨迹的位姿与真实轨迹对应时刻的位姿之间的欧氏距离。该指标能够反映整个轨迹的整体精度。在本文实验中，将 ORB-SLAM3 作为基准 V-SLAM 系统与 GeoYOLO-SLAM 进行 ATE 的计算与比较。同时，还将 GeoYOLO-SLAM 与先进的动态 V-SLAM 进行横向对比，以全面分析并量化 GeoYOLO-SLAM 在实际环境中的表现。

以下是 TUM 数据集中 walking_xyz、walking_halfsphere 和 walking_rpy 序列中，ORB-SLAM3 与 GeoYOLO-SLAM 的轨迹对比图。该实验使用了 TUM 官方提供的 evaluate_ate.py 文件绘制算法轨迹与相机真实轨迹的对比图。其中黑色线条表示相机真实轨迹，蓝色线条表示算法轨迹。

由 ORB-SLAM3 与 GeoYOLO-SLAM 的轨迹对比图可以看出，GeoYOLO-SLAM 相比 ORB-SLAM3 均有相应的提升。由图3-23、图3-24以及图3-25所示，GeoYOLO-SLAM 在 walking 高动态环境序列中性能的提升较为明显。使用 EVO 工具评估得到的 ATE 评价指标的数据，为了更好的量化实验结果，本文通过计算得到 ATE 的均方根误差（RMSE）与标准差（S.D.）进行对比。并计算出 GeoYOLO-SLAM 相比 ORB-SLAM3 性能的提升幅度 η：

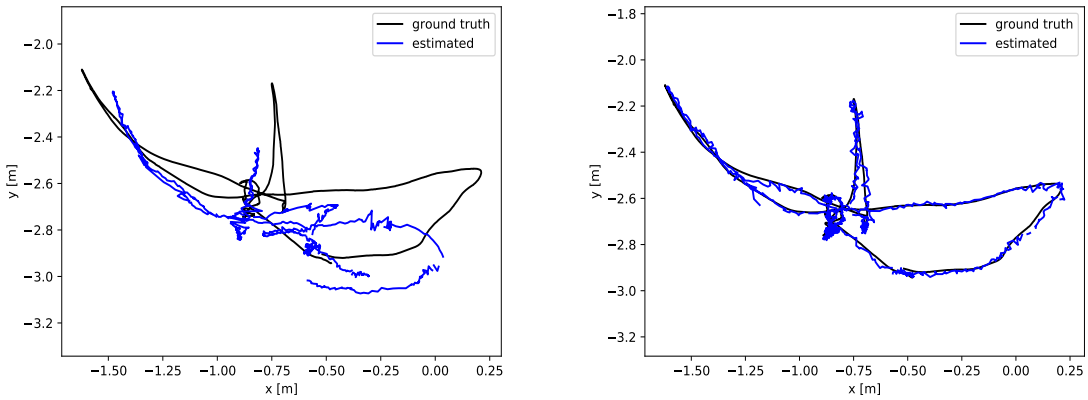
$$\eta = \frac{V_{\text{ORB-SLAM3}} - V_{\text{GeoYOLO-SLAM}}}{V_{\text{ORB-SLAM3}}} \times 100\% \tag{3-26}$$

其中 $V_{\text{ORB-SLAM3}}$ 表示 ORB-SLAM3 的 ATE 结果值，$V_{\text{GeoYOLO-SLAM}}$ 表示 GeoYOLO-SLAM 的 ATE 结果值。



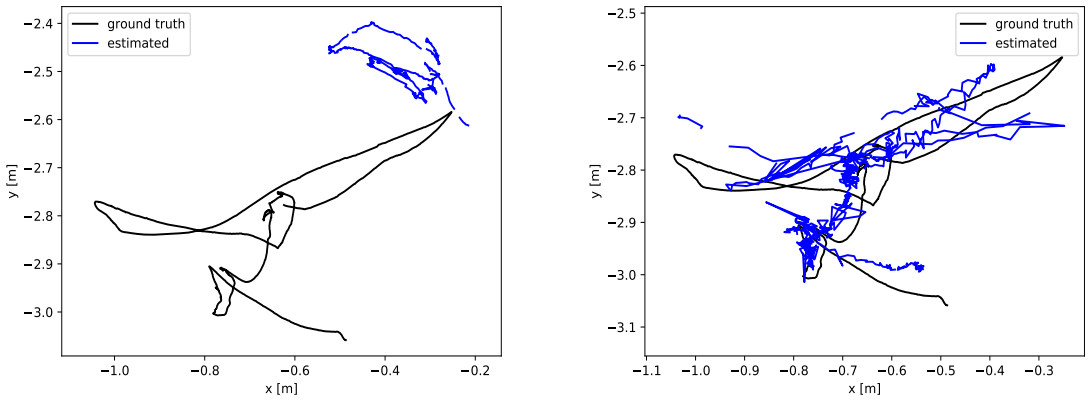
(a) (b)

图 3-23 walking_xyz 轨迹。(a)ORB-SLAM3；(b) GeoYOLO-SLAM



(a) (b)

图 3-24 walking_halfsphere 轨迹。(a)ORB-SLAM3；(b) GeoYOLO-SLAM



(a) (b)

图 3-25 walking_rpy 轨迹。(a)ORB-SLAM3；(b) GeoYOLO-SLAM

ORB-SLAM3 与 GeoYOLO-SLAM 在 TUM 数据集上的绝对轨迹误差 ATE 对比结果如表3-2所示。表内以黑体加粗呈现的数据即为本文中算法精度较优的成果，并且表内数据单位均为 m。

表 3-2 GeoYOLO-SLAM 与 ORB-SLAM3 在 TUM 数据集上的 ATE 误差

Sequences	ORB-SLAM3		GeoYOLO-SLAM		提升幅度	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
walking_xyz	0.695344	0.366273	**0.032020**	**0.026180**	95.4%	92.9%
walking_static	0.024349	0.013870	**0.009585**	**0.004373**	60.7%	68.5%
walking_halfsphere	0.339097	0.107399	**0.023703**	**0.011878**	93.0%	89.0%
walking_rpy	0.522199	0.224315	**0.108142**	**0.073974**	79.3%	67.0%
sitting_xyz	0.016571	0.007282	**0.010104**	**0.005124**	39.0%	29.6%
sitting_static	0.011176	0.005581	**0.007000**	**0.003650**	37.3%	34.6%
sitting_halfsphere	0.041811	0.016126	**0.018366**	**0.009578**	56.0%	40.6%

根据表3-2中的实验结果，GeoYOLO-SLAM 在所有 TUM 数据集的测试序列中相较于 ORB-SLAM3 存在显著的提升，尤其在以 walking 系列为例的高动态序列。在 walking_xyz 序列中，GeoYOLO-SLAM 的 RMSE 相比于 ORB-SLAM3 减少了约 95.4%，而标准差减少了 92.9%。在 walking_halfsphere 和 walking_rpy 序列中，GeoYOLO-SLAM 也分别获得了 93.0% 和 79.3% 的 RMSE 性能提升，同时标准差的减少幅度为 89.0% 和 67.0%。GeoYOLO-SLAM 在低动态场景 sitting 序列也有一定的提升。由于 ORB-SLAM3 会将提取到的全部特征点都进行特征匹配。尽管在静态场景中，这些特征点提取和匹配能够提供可靠的定位信息。但在动态场景中，由于存在大量的动态物体，ORB-SLAM3 会提取动态物体中的特征点并进行特征匹配。这些特征点的匹配往往是错误的，从而导致了定位误差的增加。在 walking 高动态环境的序列中，动态物体（如行人等）会严重影响特征点的匹配，进而引发较大的相机位姿估计误差。在 sitting 低动态环境的序列中由于大部分时间实验者处于静止的状态，因此 GeoYOLO-SLAM 算法的提升幅度没有在 walking 高动态环境的序列中明显但仍有较大提升。相对于 ORB-SLAM3，GeoYOLO-SLAM 会通过语义模块和几何模块判断物体真实运动状态，从而避免动态物体上特征点的匹配。因此 GeoYOLO-SLAM 有效解决了动态物体干扰，提升了系统的整体性能。

ORB-SLAM3 与 GeoYOLO-SLAM 在 Bonn 数据集上的 ATE 对比结果如表3-3所示。可以看出 GeoYOLO-SLAM 在所有序列上均有提升，其中在 crowd、person_tracking、moving_nonobstructing_box 以及 placing_nonobstructing_box 序列上提升较为明显，在 kidnapping_box 序列上的提升幅度略低于其他序列的提升幅度。这

是由于 kidnapping_box 序列在相机视角中没有出现明显物体运动的画面，而是由于相机视角转换之后箱子消失在相机视野范围内并当箱子再次出现的时候更换了位置。因此相对于其他序列，kidnapping_box 序列的动态信息较少，因此 GeoYOLO-SLAM 提升幅度相对较少。

表 3-3 GeoYOLO-SLAM 与 ORB-SLAM3 在 Bonn 数据集上的 ATE 误差

Sequences	ORB-SLAM3		GeoYOLO-SLAM		提升幅度	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
crowd	1.292393	0.795335	**0.022116**	**0.009746**	98.29%	98.77%
person_tracking	0.485994	0.254110	**0.049148**	**0.012202**	89.89%	95.21%
kidnapping_box	0.028967	0.015345	**0.025653**	**0.013627**	11.47%	11.86%
moving_nonobstructing_box	0.218968	0.075571	**0.020915**	**0.008298**	90.46%	89.02%
placing_nonobstructing_box	0.827057	0.203091	**0.018090**	**0.009085**	97.81%	95.53%

本文对 GeoYOLO-SLAM 与两种近年来在动态环境中表现优异的 SLAM 算法 Dyna-SLAM 和 RDS-SLAM 进行性能对比。Dyna-SLAM 基于 ORB-SLAM2 框架，结合 Mask-RCNN 进行动态物体的检测与去除，提升了动态环境下的定位精度。RDS-SLAM 则在 ORB-SLAM3 框架上集成了实例分割算法和关键帧队列策略，进一步增强系统在动态场景的鲁棒性与实时性。在 TUM 数据集中对以上两种出色 SLAM 算法与本文 GeoYOLO-SLAM 算法展开对比。

GeoYOLO-SLAM 与以上两种 SLAM 在 TUM 数据集上的 ATE 对比结果如表3-4所示:

表 3-4 三种算法在 TUM 数据集上的 ATE 误差

Sequences	DynaSLAM		RDS-SLAM		GeoYOLO-SLAM	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
walking_xyz	0.032223	**0.017317**	0.046455	0.036053	**0.032020**	0.026180
walking_static	0.009836	0.004614	0.009614	**0.004159**	**0.009585**	0.004373
walking_halfsphere	0.029317	0.013233	0.031211	0.016125	**0.023703**	**0.011878**
walking_rpy	**0.049567**	**0.024387**	0.120477	0.071754	0.108142	0.073974
sitting_xyz	0.018170	0.009079	0.012159	0.008030	**0.010104**	**0.005124**
sitting_static	0.008248	0.004115	0.007582	**0.003425**	**0.007000**	0.003650
sitting_halfsphere	0.025868	0.011046	0.037862	0.026093	**0.018366**	**0.009578**

GeoYOLO-SLAM 在大部分序列中表现出了较低的 RMSE 和标准差，尤其是在 walking_xyz、walking_static、walking_halfsphere、sitting_xyz、sitting_static 和 sitting_halfsphere 这些序列中，相较于 DynaSLAM 和 RDS-SLAM，GeoYOLO-SLAM

的 RMSE 值普遍较低，并且标准差也显示出更好的稳定性。而在 walking_rpy 序列中，GeoYOLO-SLAM 没有 DynaSLAM 处理效果好。这是由于 walking_rpy 序列的相机视角包含旋转（roll）、俯仰（pitch）和偏航（yaw）等变化，又存在实验者在场景中走动，具有较强的动态性。DynaSLAM 是对每一个关键帧都进行实例分割去除先验动态物体，并且使用多视图几何信息去除其他动态物体，直接在像素层面剔除了动态物体。而 GeoYOLO-SLAM 为了增强算法实时性，是使用关键帧队列对前后关键帧进行处理。判断出物体真实运动状态后，通过更新地图点移动概率从而剔除移动概率较高的地图点对应的特征点，从而达到剔除动态物体的目的。然而在具有较强的动态性的场景中地图点的匹配可能会出现错误，从而导致位姿估计时出现偏差。

GeoYOLO-SLAM 与 DynaSLAM 和 RDS-SLAM 在 Bonn 数据集上的 ATE 对比结果如表3-5所示:

表 3-5 三种算法在 Bonn 数据集上的 ATE 误差

Sequences	DynaSLAM		RDS-SLAM		GeoYOLO-SLAM	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
crowd	**0.020909**	**0.009635**	0.022905	0.010145	0.022116	0.009746
person_tracking	**0.048407**	0.013742	0.054734	0.016307	0.049148	**0.012202**
kidnapping_box	0.028403	0.015855	0.027060	0.014424	**0.025653**	**0.013627**
moving_nonobstructing_box	0.027578	0.014787	0.023788	0.008327	**0.020915**	**0.008298**
placing_nonobstructing_box	0.020230	0.065081	0.018717	0.009695	**0.018090**	**0.009085**

可以看出 GeoYOLO-SLAM 在 kidnapping_box、moving_nonobstructing_box 序列以及 placing_nonobstructing_box 序列上的效果明显优于其他动态 SLAM。因为这些序列中出现了先验静态物体（如箱子等）处于运动状态，而在 GeoYOLO-SLAM 中可以判断物体真实运动状态从而避免使用这些处于运动状态的先验静态物体上的特征点。因此在 GeoYOLO-SLAM 中如果序列中出现先验动态物体处于静态或者先验静态物体处于动态的情况，结果会优于其他动态 SLAM 算法。DynaSLAM 在 crowd 序列的 RMSE 及标准差表现优于 GeoYOLO-SLAM，且在 person_tracking 序列的 RMSE 上也略微优于 GeoYOLO-SLAM。这是由于 crowd 序列和 person_tracking 序列中的先验动态物体全部处于运动状态。DynaSLAM 中会对每个关键帧顺序进行实例分割，这虽然会使得语义信息更加完善但是会导致 SLAM 系统等待每个关键帧实例分割的结果，因此 DynaSLAM 很难实时进行。GeoYOLO-SLAM 使用关键帧队列弥补了 DynaSLAM 不能实时的问题，并且在 crowd 序列和 person_tracking 序列上的误差结果也与 DynaSLAM 相差无几。从整体数据分析来看，本文 GeoYOLO-SLAM 算法展现出较为良好的性能。

3.6 本章小结

本章主要针对室内动态场景下 GeoYOLO-SLAM 系统的设计与实现展开了详细讨论。通过语义模块结合 YOLOv8 的实例分割与改进的滑动窗口深度补全算法共同生成物体的精细化掩码。进一步提出基于深度与极线距离的联合异常检测方法，结合多视图几何校验策略共同判断物体真实运动状态。并且引入地图点移动概率解决单个关键帧观测不稳定的问题，增强系统的鲁棒性。最后在 TUM 数据集与 Bonn 数据集上的实验结果充分验证了 GeoYOLO-SLAM 系统在室内动态环境中的有效性与可靠性。

第四章 室内动态场景下的 V-SLAM 后端优化与稠密建图

在 ORB-SLAM3 中，通过提取关键帧和特征点构建稀疏点云地图。但是稀疏点云地图的点云密度较低，只包含少数具有显著视觉特征的点，这使得其在表示环境时缺乏足够的细节。为了克服稀疏点云地图中存在的细节不足问题，可以通过生成点云稠密地图来提高地图的精度和完整性。点云稠密地图通过结合多个视角的图像信息，采用深度估计和优化技术，将稀疏的特征点填补为更为密集的点云。然而如果场景中存在运动物体，图像中的运动物体会在稠密点云地图中形成多个位置的重影。此外，相机位姿的误差会直接影响图像与地图点的对齐程度，进而影响到点云的密集程度和位置准确性。

第三章中提出的物体运动状态检测方法有效地区分了场景中的动态物体和静态物体。静态物体通常在空间中保持固定位置，为了增加更多已知的约束项共同优化相机位姿，本文引入了静态物体的非线性优化。同时，补全动态物体造成的图片静态背景缺失，从而实现静态点云稠密地图的构建。

4.1 系统框架介绍

本文以 ORB-SLAM3 算法作为基础，本章的系统框架如图4-1所示。

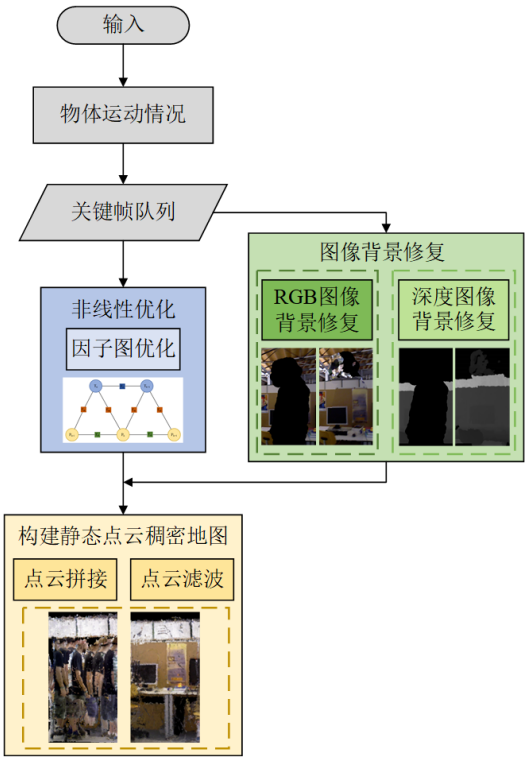


图 4-1 后端优化与稠密建图系统框架

本文在第三章确认物体真实运动状态的基础上，提出了一种基于静态物体的非线性优化算法。通过利用静态物体的重投影误差、极线误差以及几何段误差作为因子节点共同优化相机位姿。其次，通过改进双线性插值算法进行 RGB 图像和深度图像的修复。最后，利用优化后的相机位姿，结合静态背景修复后的图像信息进行静态点云稠密地图的构建。确保能够有效去除地图中动态物体重影的影响，获得高质量的三维地图重建结果。

4.2 基于静态物体的非线性优化算法

4.2.1 BA 优化

投影模型和 BA 优化是 V-SLAM 后端优化中的核心技术。投影模型用于准确描述三维世界坐标系和二维图像坐标系之间的映射关系。BA 优化则通过最小化重投影误差，以达到调整优化相机位姿和地图点位置的目的。

首先，将世界坐标系中的三维点 p 通过公式2-8转换到相机坐标系中，如下所示：

$$p' = Rp + t \tag{4-1}$$

其中，$p' = [X', Y', Z']^T$ 是在相机坐标系中的三维点。三维点 p' 被投影到相机的图像平面。将三维点 p' 的 Z 坐标归一化，得到二维像素坐标 p_c：

$$\begin{aligned} p_c &= [u_c, v_c, 1]^T \\ &= \left[\frac{X'}{Z'}, \frac{Y'}{Z'}, 1\right]^T \end{aligned} \tag{4-2}$$

在现实相机中，由于镜头畸变，图像中的像素位置与理想的二维投影位置之间存在偏差。通常，通过径向畸变和切向畸变来建模这种影响。畸变修正公式如下：

$$\begin{aligned} u' &= u_c \left(1 + k_1 r^2 + k_2 r^4\right) \\ v' &= v_c \left(1 + k_1 r^2 + k_2 r^4\right) \end{aligned} \tag{4-3}$$

其中，$r = \sqrt{{u_c}^2 + {v_c}^2}$ 是像素坐标的径向距离，k_1 和 k_2 是畸变系数，u_c 和 v_c 是无畸变的像素坐标。最后，畸变修正后的像素坐标为：

$$\begin{aligned} u &= f_x u' + c_x \\ v &= f_y v' + c_y \end{aligned} \tag{4-4}$$

其中，f_x 和 f_y 是相机的焦距，c_x 和 c_y 是主点偏移量。根据上述公式，可以得到相机图像中地图点投影后的坐标 u、v。那么通过当前的相机位姿和三维点位置计算

得到的投影 u^{proj} 与对应像素点的实际观测的误差 e_m 为：

$$e_m = \begin{bmatrix} u - u^{proj} \\ v - v^{proj} \end{bmatrix} \tag{4-5}$$

BA 优化旨在通过最小化重投影误差来优化相机的位姿和三维点的坐标。重投影误差指的是实际图像中的像素位置与通过三维点投影到图像平面后计算出的像素位置之间的差异。其通过协方差矩阵加权后的代价函数 $\mathcal{E}_m$ 可以表示为：

$$\mathcal{E}_m = e_m^T \Omega_m^{-1} e_m \tag{4-6}$$

4.2.2 静态物体刚性约束

本文增加针对静态物体的刚性约束，以此来优化相机位姿估计。在大多数情况下，室内静态物体被视为刚性物体。刚性物体是指在运动过程中，其形状和体积始终保持不变。根据刚性物体的特性，无论物体如何移动或旋转，其内部各点之间的相对距离始终保持不变。如图4-2所示：

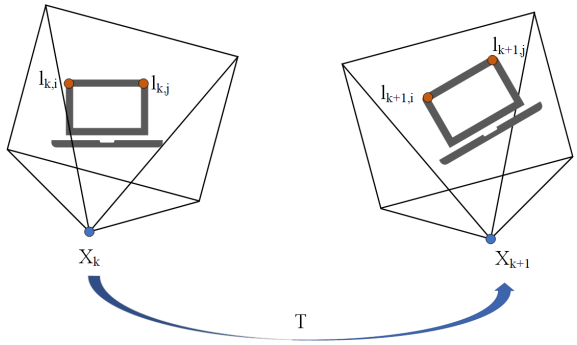


图 4-2 刚性约束

设定 k 时刻同一个静态刚性物体上两个特征点 $l_{k,i}$ 和 $l_{k,j}$，这两个特征点对应地图点之间的距离为 $s_{k,ij}$。那么在 $k+1$ 时刻这两个特征点对应地图点之间的距离 $s_{k+1,ij}$ 在整个过程中保持不变。因此，刚性约束可以表示为：

$$s_{k,ij} = s_{k+1,ij} \tag{4-7}$$

即保证静态刚性物体上的特征点在运动过程中不发生距离变化。为了量化这种约束，本文定义了刚性误差 e_s，用来衡量物体是否偏离了刚性约束的要求。刚性误差的计算公式为：

$$e_s = \| s_{k,ij} - s_{k+1,ij} \| \tag{4-8}$$

通过协方差矩阵 Ω_s 加权后的代价函数 $\mathcal{E}_s$ 可以表示为：

$$\mathcal{E}_s = e_s^T \Omega_s^{-1} e_s \tag{4-9}$$

4.2.3 因子图优化算法

因子图是一种图形模型，用于高效表示和推理高维概率分布。它通过分解全局联合概率分布，将复杂的全局模型表示为一组较小的局部因子，每个因子仅依赖于少量的变量。形式上，因子图表达了一个联合概率分布 $P(X)$ 的分解形式：

$$P(X) \propto \prod_i \phi_i(X_i) \tag{4-10}$$

其中，$X = \{X_1, X_2, \ldots, X_n\}$ 是变量，$\phi_i(X_i)$ 是定义在变量子集 $X_i \subseteq X$ 上的因子函数。因子图由两种节点组成：变量节点和因子节点。变量节点表示需要优化的未知量，因子节点代表与这些变量相关的约束或观测模型，变量节点和因子节点由边连接。边表示因子节点如何约束变量节点，并在优化过程中传递误差或约束的信息。在贝叶斯概率框架中，因子图通常用于表示后验概率分布：

$$P(X|Z) \propto P(Z|X)P(X) \tag{4-11}$$

其中 $P(Z|X)$ 是似然函数，表示观测数据 Z 条件于变量 X 的分布，$P(X)$ 是先验分布。可以转换为最大化乘积的形式：

$$X^* = \arg\max_X \phi(X) = \arg\max_X \prod_i \phi_i(X_i) \tag{4-12}$$

其中，$\prod_i \phi_i(X_i)$ 代表了所有因子 $\phi_i(X_i)$ 的乘积。每一个独立项子 $\phi_i(X_i)$ 对应一个因子。将这些因子乘积结合在一起，并用图的形式进行表示，就称为因子图。通过对数变换，可以将后验概率的优化转化为最小化负对数似然问题，从而转换为最小二乘问题：

$$X^* = \arg\min_X \sum_i -\log \phi_i(X_i) \tag{4-13}$$

在 V-SLAM 中，因子图被广泛用于联合优化相机位姿、地图点 3D 坐标以及运动参数等变量节点。本文以所有静态物体重投影误差 $\mathcal{E}_m$、所有静态物体极线误差 $\mathcal{E}_d$ 以及静态刚性物体几何段误差 $\mathcal{E}_s$ 作为因子节点，共同优化变量节点相机位姿 T。

重投影误差 $\mathcal{E}_m$ 以及刚性物体几何段误差 $\mathcal{E}_s$ 如上两节所示。极线误差因子用于约束所有静态物体上匹配特征点到极线的距离，根据式3-13极线误差 $\mathcal{E}_d$ 表示为：

$$\mathcal{E}_d = e_d^T \Omega_d^{-1} e_d \tag{4-14}$$

通过因子图联合优化这些变量节点，最小化上述误差项的加权总和：

$$X^* = \arg\min_T (\mathcal{E}_s + \mathcal{E}_m + \mathcal{E}_d) \tag{4-15}$$

即：

$$X^* = \arg\min_T \left(e_s^T \Omega_s e_s + e_m^T \Omega_m e_m + e_d^T \Omega_d e_d\right) \tag{4-16}$$

其中 Ω_r、Ω_m、Ω_p 是对应误差的协方差矩阵。通过最小二乘法求得最优符合观测值和先验信息的相机位姿 T。因子图如图4-3所示，P 为地图点的 3D 坐标。重投影误差 $\mathcal{E}_m$、几何段误差 $\mathcal{E}_s$ 和极线误差 $\mathcal{E}_d$ 作为因子节点对变量节点相机位姿 T 进行优化。

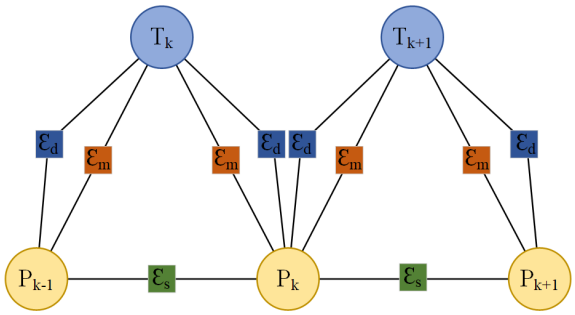
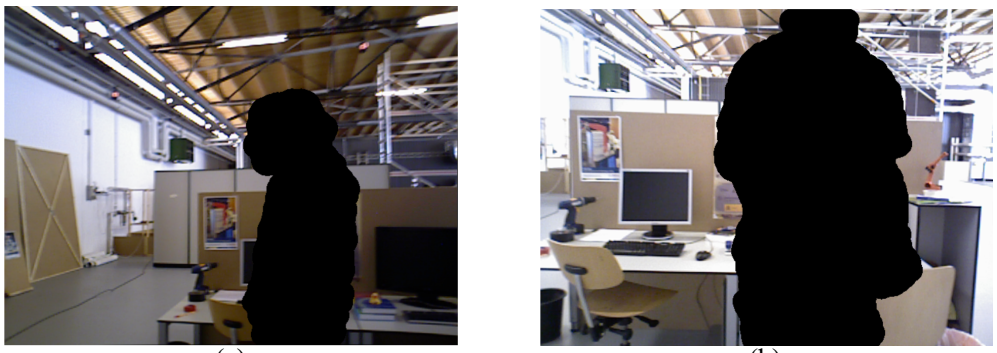


图 4-3 因子图

4.3 静态点云稠密地图构建

4.3.1 基于改进双线性插值算法的图像背景修复

在动态 V-SLAM 系统中，动态物体是影响地图准确性和稳定性的一项重要挑战。通常情况下，动态物体会被误认为是场景的一部分从而会影响对静态环境的准确建图。如图4-4所示，如果直接剔除图像中动态物体会造成图像中场景完整性缺失的问题。为了解决这一问题本文提出了一种图像背景修复算法，通过将先前关键帧的 RGB 图像与深度图像共同投影到当前帧，来修复被动态物体遮挡的背景区域。该方法利用前后关键帧之间的相对位置关系，填补当前关键帧中缺失的背景数据，从而实现了无动态物体干扰的背景修复。



(a) (b)

图 4-4 直接剔除动态物体。(a)场景 1；(b)场景 2

在第三章中，最终会将关健帧队列中的所有关键帧的 RGB 图像进行实例分割并筛选出处于真实运动状态的物体掩码，并且还会对关键帧中的所有关键帧深度图进行深度信息补全。因此每个关键帧都包含补全后的深度图 D、动态物体掩码信息、RGB 图像 I 以及变换矩阵 T。

对于每一关键帧，首先需要将前一关键帧的深度图 D 转换为 3D 点。通过使用相机的内参矩阵 K，可以将每个像素 (u, v) 对应的深度值转换为相机坐标系中的 3D 点 P_{3D}。利用变换矩阵 T 将 3D 点从前一关键帧的坐标系转换至当前关键帧所处的坐标系中：

$$P_{3D}^{curr} = T \cdot P_{3D} \tag{4-17}$$

当 3D 点转换到当前帧的坐标系之后，它们将被重新投影到 2D 图像平面，得到当前帧中的像素坐标 (u', v')：

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = K \cdot P_{3D}^{curr} \tag{4-18}$$

当前关键帧的 RGB 图像就可以通过将前一关键帧的颜色值转移到对应的像素位置来实现修复。基于历史关键帧中未被掩码覆盖的静态区域，将其三维信息投影至当前关键帧的图像坐标系，并筛选出投影位置恰好位于当前帧需补全的掩码区域内的像素点。通过这一策略，补全操作仅针对当前帧尚未修复的掩码区域进行填充，且严格限定数据源为历史帧中已验证的静态背景信息。既避免了因重复处理已补全区域导致的冗余计算，又确保了修复内容仅来源于环境中的静态部分，从而提升修复效率与时空一致性。

由于前后关键帧之间的相对运动不能完全通过变换矩阵 T 精确计算，通常会存在一定的误差，进而使得直接转移像素时无法实现精确对齐。此外，当像素位置不完全重合时，常会出现像素位置的偏差。为了实现颜色值的平滑过渡，本文采用双线性插值来解决这一问题。

双线性插值是一种常用的插值方法，主要用于像素位置不完全重合时估算新像素的颜色值。不同于仅在一个方向上进行插值，双线性插值同时考虑水平方向和垂直方向上的相邻像素，从而能更准确地估算二维空间中的任意位置。这种方法虽然简单且常用，但在处理复杂图像特征时存在一定局限性。由于传统的双线性插值方法对所有像素区域的权重分配相同，无法根据图像的局部特性进行调整，导致在边缘区域容易引入伪影从而影响图像的质量。

为了提升插值精度和视觉效果，本文引入了非均匀加权机制。在传统双线性插值的基础上，为每个像素点的插值增加与梯度信息相关的权重。通过图像的局部梯度信息来动态调整每个像素的权重。边缘区域权重值减少，避免插值过程中

引入不准确的边缘信息。平坦区域的权重值增加，以确保插值过程中平坦区域的像素保持更为平滑和一致。

首先通过 Sobel 算子来获取深度图像 D 在水平方向以及垂直方向的梯度。对于每个像素点 (x, y) 计算其水平方向梯度 G_x 以及垂直方向梯度 G_y：

$$\begin{aligned} G_x &= \frac{\partial D}{\partial x} \\ G_y &= \frac{\partial D}{\partial y} \end{aligned} \tag{4-19}$$

然后，通过计算梯度的幅值 $G(x, y)$ 来衡量像素的局部变化强度：

$$G(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)} \tag{4-20}$$

该幅值反映了图像中每个像素点在深度图中的变化强度，较高的梯度值通常意味着该区域存在边缘。基于梯度幅值 $G(x, y)$，确定区域权重 $W(x, y)$：

$$W(x, y) = \frac{1}{1 + \lambda \cdot G(x, y)} \tag{4-21}$$

梯度较小的平坦区域权重值更高，而梯度较大的边缘区域权重值更低。其中，λ 是一个超参数，用于控制幅值的敏感度。

使用传统的双线性插值方法选择目标像素 (x, y) 的邻近像素 (x_1, y_1)、(x_1, y_2)、(x_2, y_1) 和 (x_2, y_2)，首先计算这些邻近像素的权重 W，然后使用加权平均来得到最终的插值结果：

$$I(x, y) = \sum_{x'} \sum_{y'} W(x', y') \cdot I(x', y') \tag{4-22}$$

其中 $x' \in \{x_1, x_2\}$，$y' \in \{y_1, y_2\}$。

本文通过引入非均匀加权机制改进传统的双线性插值方法，有效地提高了插值结果的精度和质量。该方法根据图像区域的不同特性，动态调整每个像素点的权重。使用同样的改进双线性插值方法补全深度图 D 的完整性。

4.3.2 静态点云稠密地图构建

在 ORB-SLAM3 中提供了稀疏点云地图的构建。稀疏点云地图通过少量的关键点和描述符来表示环境的几何结构，这种方式具有较低的计算成本。然而在室内动态场景中，仅依靠稀疏地图可能无法提供足够的细节描述，构建精确的三维地图是确保系统能够理解和导航其环境的关键。点云稠密地图作为其中一种重要的表示方式，包含了场景中大量的三维点，每个点都包含了空间位置以及颜色等属性。可以提供丰富的环境信息，帮助机器人感知周围环境。

如果使用包含动态物体的 RGB 图像和深度图像来进行点云地图的构建，可能

会导致点云地图中出现动态物体的重影。这种重影不仅会影响点云地图的精度，还会影响地图拼接和优化过程。本文使用剔除动态物体背景修复后的 RGB 图像以及深度图像生成点云地图，从而准确地反映环境的真实结构。

对于单个像素，通过内参矩阵与深度信息的结合，可以将每个像素的二维坐标 (x, y) 与对应的深度值 $D(x, y)$ 转换为三维空间中的坐标 (X, Y, Z)。其转换公式如下：

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = K^{-1} \cdot D(x, y) \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4-23}$$

可以将图像中的每个像素映射到三维空间，从而生成单帧的点云图。由于相机在不同时间拍摄的图像对应不同视角的点云,这些点云需要经过配准才能合并为一个完整的三维点云稠密地图。因此得到单帧点云图后，要通过相机位姿信息对齐不同帧的点云，实现不同帧点云图的拼接。设当前帧的点云为 $P_i = \{p_{i1}, p_{i2}, \ldots, p_{in}\}$，其中 p_{ij} 表示第 i 帧中的第 j 个点的三维坐标，且 $j \in \{1, 2, \ldots, n\}$。通过相机位姿矩阵 T_i，可以将这些点从相机坐标系转换到全局坐标系中：

$$p_{ij}^{w} = T_i \cdot p_{ij} \tag{4-24}$$

其中，p_{ij}^{w} 表示第 i 帧中第 j 个点在全局坐标系中的位置，T_i 是相机位姿矩阵。

然而由于估计的位姿存在一定误差，点云拼接通常不能达到精确对齐，点云之间可能仍存在偏差。此外，在点云稠密地图生成中，难免会出现噪声和冗余点。因此进行滤波和降采样可以有效地提升点云稠密地图的效率和质量。

本文使用体素网格滤波减少点云拼接时的重复点云信息。体素网格滤波是将点云所在的空间分成许多被称为体素的立方体单元，在每个体素内计算点的位置的平均值，以此来减少点云的密度。体素网格滤波通过将相近的点合并为一个平均点，保留了点云的空间结构，但又有效地减小了点的数量。对于一组点云数据 $P = \{p_1, p_2, ..., p_n\}$，其中每个点 $p_i = (x_i, y_i, z_i)$ 均代表三维空间中的点。在体素网格滤波中，首先将整个点云空间划分成固定大小的网格体素，然后将每个体素内的所有点的坐标取平均值，得到一个新的代表点 p_i^{avg}，计算公式如下：

$$p_i^{avg} = \frac{1}{m} \sum_{j=1}^{m} p_j \tag{4-25}$$

其中，m 是体素内的点的数量，p_j 是体素中的第 j 个点。

在体素网格滤波减少点云稠密地图中的重复点云后，通过统计离群点移除（Statistical Outlier Removal，SOR）对点云中的离群点进行过滤。在点云数据中，离群点指的是那些与周围点的距离显著不同的点，通常是由传感器的噪声或者外

界环境的干扰造成的。SOR 首先用 K 近邻算法计算每个点的邻域，并通过统计学的方法判断该点是否是离群点。接着判断该点是否远离其他点，超过一定的阈值就认为它是离群点。点云中的某个点 p_i 及其 k 个最近邻点，点 p_i 到邻域内点的平均距离 d_i 如下所示：

$$d_i = \frac{1}{k}\sum_{j=1}^{k}\|p_i - p_j\| \tag{4-26}$$

如果距离 d_i 超过了该点邻域内的平均距离 μ_i，那么这个点就被认为是离群点，会被从点云中移除。

4.4 实验分析

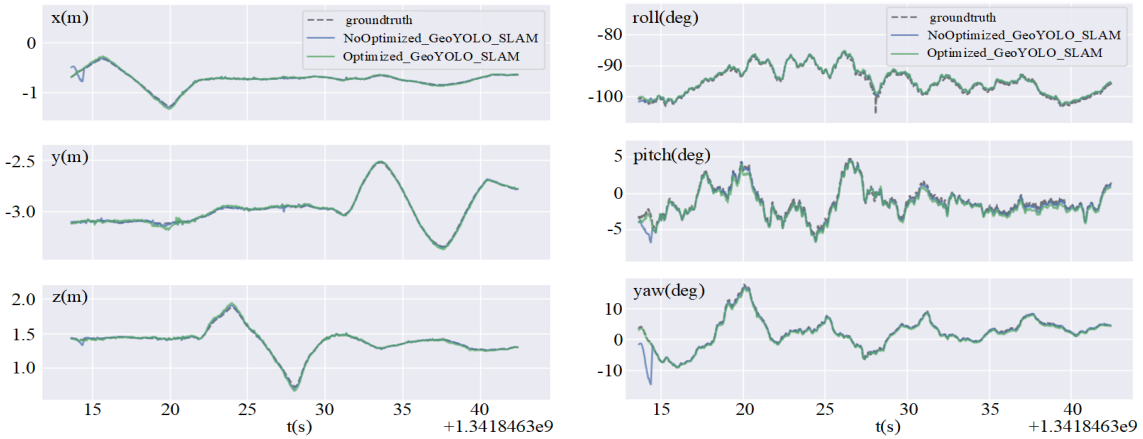
为了验证第四章算法的有效性，本实验在 Ubuntu18.04 操作系统下进行，使用了 C++11、python2.7、python3.8、ROS、g2o、Eigen、pcl 等作为软件开发环境。实验使用的数据集为 TUM 数据集和 Bonn 数据集。

4.4.1 因子图优化消融实验

本文第四章中动态场景下的 V-SLAM 后端优化算法是在第三章判断出物体真实运动状态之后通过因子图优化，将所有静态物体重投影误差、所有静态物体极线误差以及静态刚性物体几何段误差作为因子，共同优化相机位姿。本文通过消融实验对比增加因子图优化之后的 GeoYOLO-SLAM 算法性能与没有增加因子图优化的 GeoYOLO-SLAM 算法性能，通过实验结果分析动态场景下的 V-SLAM 后端优化算法的效果。

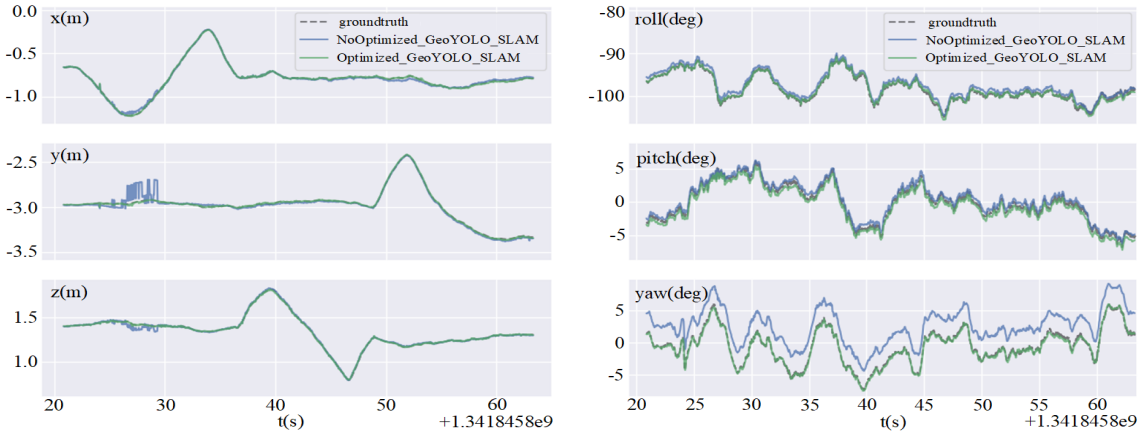
通过 EVO 工具在 TUM 数据集中的 walking_xyz 序列和 sitting_xyz 序列上进行实验。EVO 工具将相机位姿轨迹分别在 X、Y、Z 三个坐标轴以及 R、P、Y 三个旋转轴上进行了分解，便于观察对比每个轴向的精度表现。图中黑色虚线代表相机真实轨迹，蓝色实线代表没有增加因子图优化的 GeoYOLO-SLAM 算法轨迹，绿色实现代表增加因子图优化之后的 GeoYOLO-SLAM 算法轨迹。

如图4-5和图4-6所示，增加因子图优化之后的 GeoYOLO-SLAM 算法轨迹和没有增加因子图优化的 GeoYOLO-SLAM 算法轨迹都比较贴合相机真实轨迹，得到的相机位姿轨迹都较为准确。而与未增加因子图优化的 GeoYOLO-SLAM 算法轨迹相比，增加因子图优化后的轨迹在这些序列中展现出更加平稳的趋势。没有增加因子图优化的 GeoYOLO-SLAM 算法因累积误差或局部观测约束不足，轨迹易出现小幅振荡。增加因子图优化的 GeoYOLO-SLAM 算法能够有效地纠正这些未处理好的部分，进一步提高了轨迹的稳定性和准确性。



(a) (b)

图 4-5 walking_xyz 轨迹分解对比图。(a)XYZ 坐标轴；(b) RPY 旋转轴



(a) (b)

图 4-6 sitting_xyz 轨迹分解对比图。(a)XYZ 坐标轴；(b) RPY 旋转轴

GeoYOLO-SLAM 算法因子图优化消融实验的绝对轨迹误差 ATE 在 TUM 数据集上的结果如表4-1所示。

表 4-1 因子图优化消融实验在 TUM 数据集上的 ATE 误差

Sequences	不使用因子图优化		增加因子图优化		提升幅度	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
walking_xyz	0.032020	0.026180	**0.031645**	**0.016792**	1.17%	35.6%
walking_static	0.009585	0.004373	**0.009326**	**0.004114**	2.7%	5.93%
walking_halfsphere	0.023703	0.011878	**0.022858**	**0.011043**	3.56%	7.05%
walking_rpy	0.108142	0.073974	**0.092871**	**0.060812**	14.08%	17.75%
sitting_xyz	0.010104	0.005124	**0.010010**	**0.004998**	0.93%	2.46%
sitting_static	0.007000	0.003650	**0.006219**	**0.003151**	11.16%	3.77%
sitting_halfsphere	0.018366	0.009578	**0.016141**	**0.007210**	12.07%	24.69%

消融实验的绝对轨迹误差 ATE 在 Bonn 数据集上的对比结果如表4-2所示。

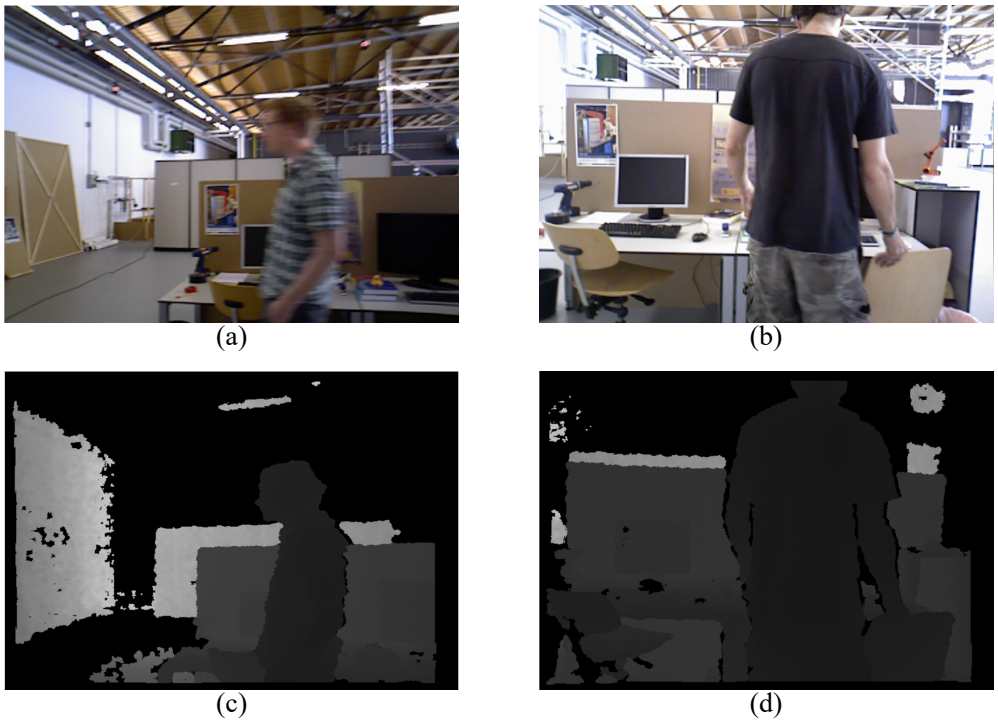
表 4-2 因子图优化消融实验在 Bonn 数据集上的 ATE 误差

Sequences	不使用因子图优化		增加因子图优化		提升幅度	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
crowd	0.022116	0.009746	**0.020677**	**0.009627**	6.49%	1.22%
person_tracking	0.049148	0.012202	**0.047424**	**0.013764**	3.51%	12.85%
kidnapping_box	0.025653	0.013627	**0.024096**	**0.011920**	6.07%	12.53%
moving_nonobstructing_box	0.020915	0.008298	**0.020652**	**0.007350**	1.26%	11.44%
placing_nonobstructing_box	0.018090	0.009085	**0.016774**	**0.007647**	7.28%	15.8%

消融实验结果表明，增加因子图优化后 GeoYOLO-SLAM 系统利用静态物体信息对相机位姿进行了有效更正，整体表现相比优化前有所提升。在所有测试的序列中，增加因子图优化后 GeoYOLO-SLAM 系统的 ATE 结果相比不使用因子图优化的结果均有不同程度的提升。这表明基于静态物体的非线性优化算法能够有效地减少 GeoYOLO-SLAM 系统位姿估计误差。

4.4.2 图像背景修复实验

本文选用 TUM 数据集的 walking_xyz 序列进行图像背景修复实验。

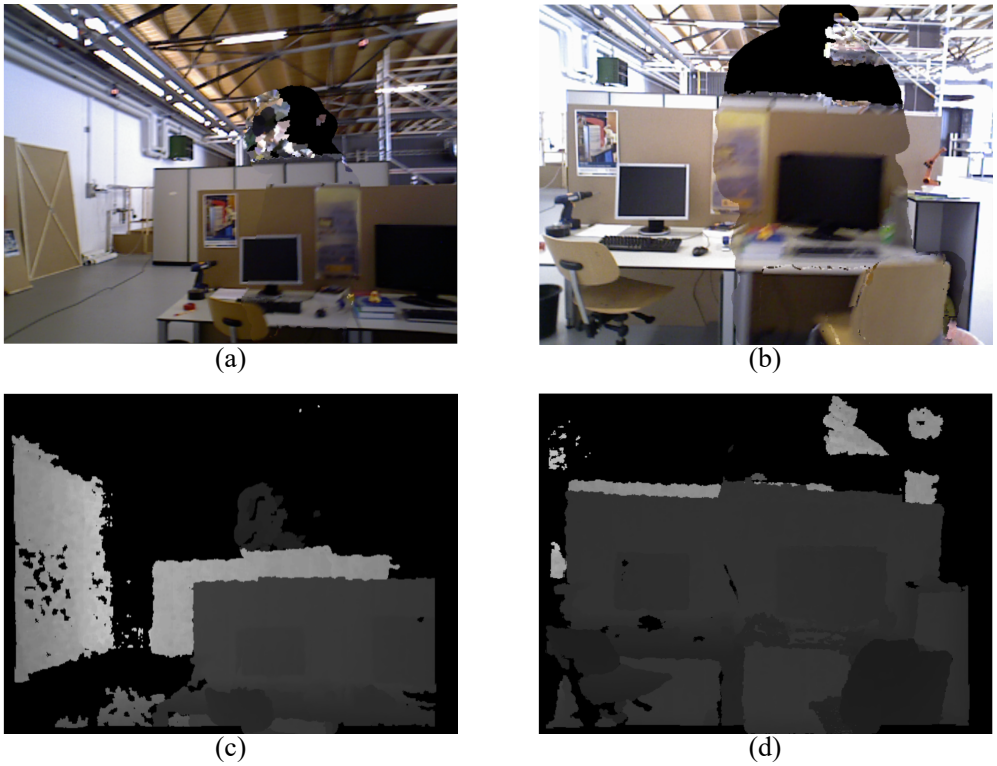


(a) (b) (c) (d)

图 4-7 动态物体遮挡导致静态信息缺失。(a)场景 1 的 RGB 图像；(b)场景 2 的 RGB 图像；(c)场景 1 的深度图像；(d)场景 2 的深度图像

RGB 图像和深度图像如图4-7所示，行走的实验者将桌面背景信息挡住，导致被实验者遮挡的静态背景缺失，如果直接剔除动态物体会造成画面完整度缺失。基于改进双线性插值算法的图像背景修复通过推断缺失区域的像素值，使图像恢复相对自然的视觉效果。

基于改进双线性插值算法恢复后的 RGB 图像和深度图像如图4-8所示。

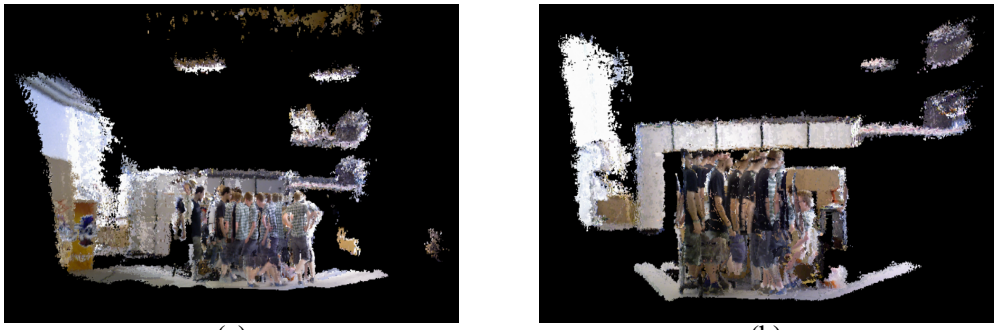


(a) (b) (c) (d)

图 4-8 图像背景修复后效果展示。(a)场景 1 的 RGB 图像；(b)场景 2 的 RGB 图像；(c)场景 1 的深度图像；(d)场景 2 的深度图像

4.4.3 静态点云稠密地图构建实验

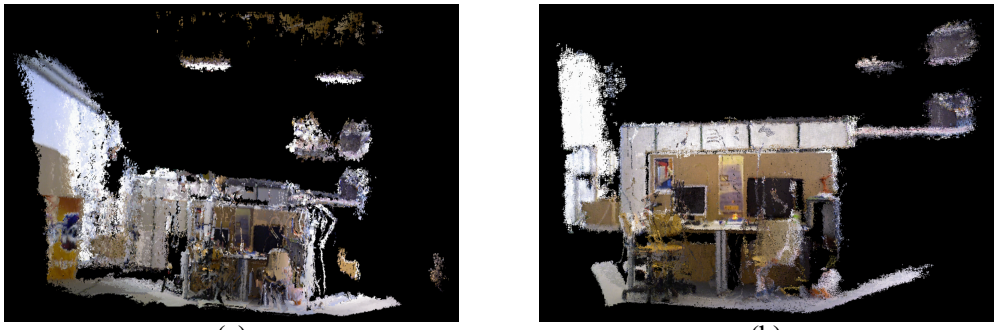
在传统的 ORB-SLAM3 算法中通常创建的是稀疏点云地图。由于稀疏点云地图所含信息有限，既难以应用于导航，也无法提供直观的地图信息。本节选用 TUM 数据集 walking 高动态场景中的 walking_xyz 序列以及 walking_static 序列实现对现实场景的稠密点云建图。在传统的构建点云稠密地图过程中若场景内存在运动物体，图像中的运动物体因其在场景中的不同位置和姿态会在稠密点云地图中形成多个位置的重叠点云，导致地图信息的不准确和不一致。如图4-9所示，运动物体在不同时间出现在地图的不同位置从而导致重影现象的产生，最终形成包含重影的点云稠密地图。此时点云稠密地图的静态背景被动态物体的重影遮挡，无法对环境进行有效观测。



(a) (b)

图 4-9 传统的点云稠密地图。(a)walking_xyz；(b)walking_static

使用图片背景修复后的 RGB 图像与深度图像构建点云稠密地图结果如图4-10所示。



(a) (b)

图 4-10 改进后的点云稠密地图。(a)walking_xyz；(b)walking_static

针对动态物体在传统点云稠密地图构建中易产生的运动模糊与重影问题，通过对原始图像进行背景修复，解决图像因运动物体遮挡导致的静态背景缺失的问题。随后基于修复后的 RGB 图像和深度图像生成点云信息，通过点云拼接以及点云滤波和降采样等操作生成静态点云稠密地图。实验结果表明，在包含动态物体的复杂室内场景中，改进后的静态点云稠密地图显著降低了动态物体由于运动残留的重叠点云。并且静态背景在改进后的点云稠密地图中得以完整创建，验证了本文静态点云稠密地图构建算法在动态干扰下的鲁棒性。

4.5 本章小结

本章基于第三章判断出的物体真实运动状态，提出了基于静态物体的非线性优化算法。通过结合静态物体的重投影误差、极线误差以及几何段误差进行联合优化，显著提升了相机位姿的精度。针对动态物体移除后的场景完整性缺失问题，提出基于改进双线性插值的图像背景修复算法实现图像中动态遮挡区域的静态背景修复，从而有效提高了静态点云稠密地图的质量。通过在公开数据集 TUM 数据

集和 Bonn 数据集上进行实验，验证了本文提出的非线性优化算法的有效性。并且通过实验结果表明，基于改进双线性插值的图像背景修复算法能一定程度上修复图像剔除动态物体后缺失的静态背景，并得到了可视化效果更好的、无动态物体重影的静态点云稠密地图。

第五章 三维可视化系统的设计与实现

5.1 需求分析

现代社会对家政服务机器人的依赖逐渐加深，尤其是在扫地机器人和智能机器人领域。为了实现高效、准确的导航，机器人系统必须具备感知和理解周围环境的能力。V-SLAM 技术作为一种高效的定位与地图构建方法，能够通过传感器实时获取周围环境的数据，进而推算出机器人的位置与环境地图。然而，仅仅通过 V-SLAM 算法获取的位姿和地图数据，往往无法直观展现，难以为开发者、用户或者研究人员提供有价值的反馈。这些数据通常需要进一步的处理和可视化，以便更好地理解机器人的环境及其运动轨迹。

本文设计了一个基于 Qt 的三维可视化系统，不仅能够将 V-SLAM 系统的相机位姿信息和点云数据直观展示，还能使得用户与系统的交互更加简便。本文从三维可视化系统的需求分析出发，证明了三维可视化系统开发的必要性。本文明确了三维可视化系统的核心功能，由总体设计逐步深入至详细设计，对系统架构以及关键算法的运行流程等方面进行了阐述。

5.1.1 系统功能性需求

本系统的功能性需求主要包括用户登录、V-SLAM 位姿生成、点云地图展示、点云格式转换以及辅助显示。首先，用户需要通过注册并登录到三维可视化平台，然后选择 V-SLAM 数据所在的文件夹。后台通过运行 V-SLAM 算法生成相机位姿并在界面上显示。其次，V-SLAM 算法结束后会生成点云稠密地图。用户可以选择点云地图进行显示，也可以将点云地图转换成其他格式方便后续需求。最后，三维可视化平台的辅助显示功能可以使用户更好进行观测。

用户管理模块功能需求如表5-1所示：

表 5-1 用户管理模块功能需求

需求编号	需求名称	需求描述
UserManage01	新增用户	用户能够通过新增用户功能提供个人用户名和密码进行注册，新增用户之后进行登录使用平台。
UserManage02	用户登录和退出	用户能够通过提供个人用户名和密码登录平台。用户使用完成后点击退出能够退出账号登陆。
UserManage03	修改密码	用户能够通过点击修改密码按钮并提供用户名、旧密码和新密码的方式重置登录密码。

数据显示模块功能需求如表5-2所示：

表 5-2 数据显示模块功能需求

需求编号	需求名称	需求描述
VSLAM01	选择文件夹并运行	用户能够选择数据集所在文件夹并运行 V-SLAM 算法生成位姿文件。在 V-SLAM 算法结束时生成点云地图文件。
VSLAM02	相机位姿显示	用户在运行 V-SLAM 算法能够展示相机位姿轨迹。
VSLAM03	点云地图显示	用户能够选择点云文件并展示。
VSLAM04	显示点云坐标	用户能够选择 3D 点并展示点云坐标位置信息。
VSLAM05	点云格式转换	用户能够选择点云文件并将点云文件转换为不同的 PLY、PCD 格式。

辅助显示模块功能需求如表5-3所示：

表 5-3 辅助显示模块功能需求

需求编号	需求名称	需求描述
SupportDisplay01	移动视角	用户能够拖动视角进行全方位位姿轨迹与点云地图的观察。
SupportDisplay02	增加和删除距离参考图形	用户能够选择增加或删除辅助显示的距离参考圈和坐标系。
SupportDisplay03	选择参考坐标系	用户能够选择从正视图、俯视图、左视图以及右视图进行观察。
SupportDisplay04	改变栅格背景配色	用户能够选择栅格背景的配色。

为达成上述功能性要求，对系统的主要功能做出如下设计：

1. 用户注册与登录功能。用户注册与登录功能是系统的核心入口，负责用户的身份验证与账户管理。该功能允许新用户通过输入个人信息（如用户名、密码）进行注册。注册信息将被保存到系统数据库中，确保每个用户拥有唯一的身份标识。注册成功后，用户可以通过提供已注册的用户名和密码进行登录。若信息无误用户将成功登录，系统主界面上的功能将对用户开放。用户退出后，系统会清除当前会话的相关数据确保其他用户无法访问当前用户的个人信息，提升系统安全性。如果用户需要重置，可以通过提供个人用户名和旧密码并提供新密码进行密码修改。

2.V-SLAM 位姿生成功能。平台验证完用户身份之后，用户可以提供给平台 V-SLAM 数据集位置，系统会自动识别并在后台通过运行 SLAM 算法估计相机位姿并显示在系统平台上。V-SLAM 生成位姿数据结束后，根据位姿信息生成点云数据并将其保存。

3. 点云地图显示与格式转换功能。点云地图展示与格式转换功能是系统的可视化模块，主要用于选择 V-SLAM 生成的点云数据并将其展示为可交互的点云稠密地图。用户可以选中点云中的 3D 点，平台会显示该点云的坐标信息使得用户直观观察点云所在世界坐标系下的位置。该功能还支持点云 PLY 数据与 PCD 格式之间的相互转换，以方便用户进行后续处理。用户可以选择将生成的点云数据并选择转换格式，转换后的点云文件可以保存在本地。

4. 视角控制功能。用户可以使用鼠标拖动界面，在三维空间中旋转地图视角，查看不同角度的地图信息。该功能使得用户在分析复杂环境时更全面地理解点云地图的空间结构。系统允许用户缩放地图，查看更细节的区域或从远距离整体查看地图。通过平移功能，用户可以将地图在可视化窗口中平移，聚焦于某一特定区域。用户可以快速切换到正视图、俯视图、左视图以及右视图这些特定视角，以便于查看不同视角的地图信息。

5. 空间辅助功能。空间辅助功能是系统的补充模块，旨在提供额外的可视化工具帮助用户更好地理解和分析环境数据。用户可以在地图中显示坐标系，以便用户清楚地了解当前地图的空间位置和方向。坐标系可以帮助用户快速判断环境中的相对位置。用户还可以在地图中添加一个或多个参考圈，用以标记不同位置之间的相对距离。用户可以通过这些参考圈快速了解目标之间的空间距离。

5.1.2 系统性能需求

三维可视化系统的性能需求是确保系统能够高效运行并提供稳定可靠服务的基础。该系统的性能需求主要涵盖响应速度快、稳定性强以及易使用性三个方面。

1. 响应速度快。三维可视化系统必须能够在极短的时间内响应用户的各种请求，避免长时间的等待，确保用户在操作过程中获得流畅的体验。

2. 稳定性强。为了确保系统的高可用性，系统设计应具备故障检测和自动恢复机制，能够在发生故障时迅速恢复并减少对用户操作的影响。此外，系统还应定期进行用户数据备份并具备快速数据恢复的能力，确保在系统出现问题时能够最大限度地避免数据丢失。

3. 易使用性。易使用性直接影响到用户的使用体验和系统的普及程度。系统设计必须考虑到用户的便捷性，使得各类用户能够快速上手，轻松完成所需操作。系统界面设计应直观、简洁，功能模块的布局应符合用户习惯，并尽可能减少不必要的操作步骤。界面中的元素应合理分组避免信息过载，并提供清晰的指引和反馈。

5.1.3 系统用例设计

三维可视化系统主要由三大模块组成：用户管理模块、数据可视化模块、辅助可视化模块。如图5-1所示展示了三维可视化系统的用例图。在图中用户被视为参与者，与系统中的用例进行交互。

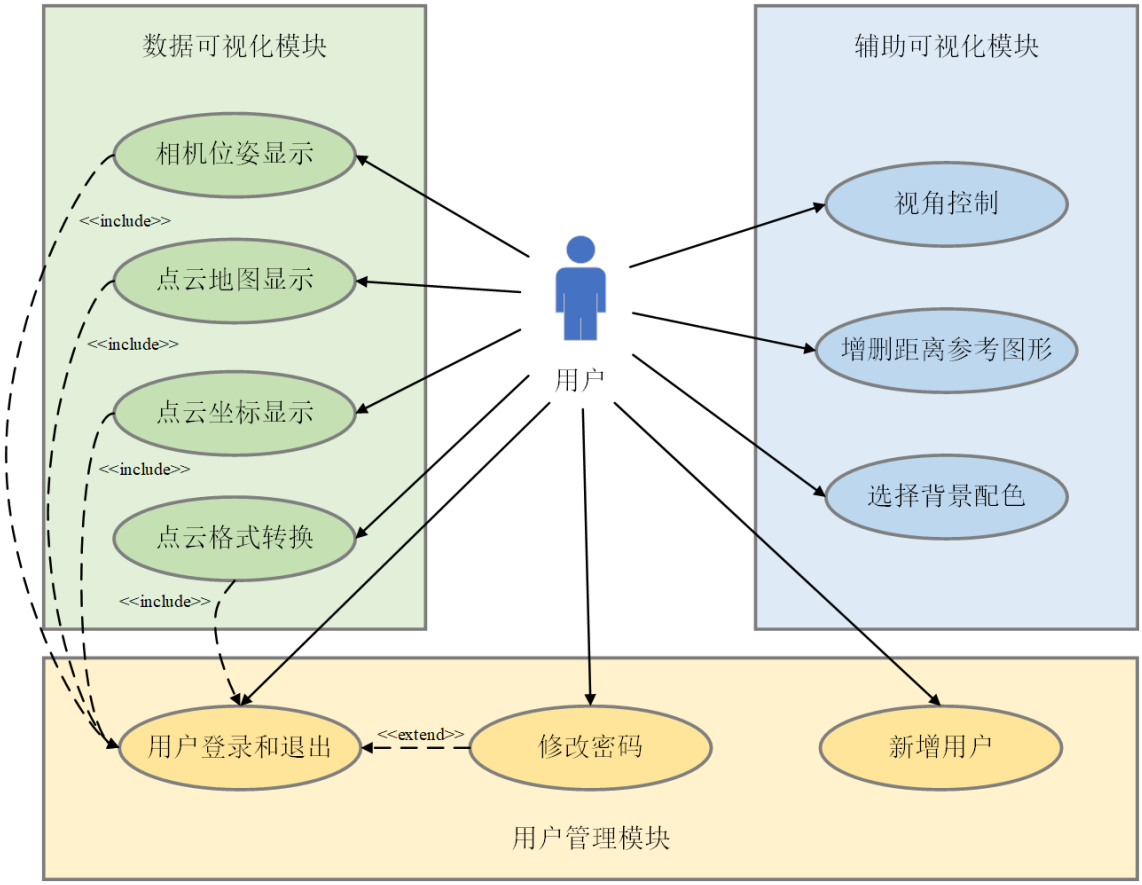


图 5-1 系统用例图

用户管理模块承担了系统对用户身份的验证职责。它为每个用户创建独立的账户，确保系统的安全性与个性化体验。用户通过此模块登录系统后，可以访问各类功能。用户管理模块的设计使得系统能够灵活地应对多用户的管理需求，同时确保数据的安全与隐私。数据可视化模块是系统的核心部分，它实现了对 V-SLAM 算法生成的地图和位姿数据的可视化显示。该模块根据 V-SLAM 算法实时获取的位姿信息，生成与之对应的点云稠密地图，并且在界面上进行动态显示。用户可以在地图中查看不同时间点的位姿变换。此模块通过图形渲染技术，确保用户能够清晰直观地看到 V-SLAM 系统所构建的点云稠密地图，支持用户对数据进行实时监控与分析。辅助可视化模块提供了多个功能，包括移动视角、增加和删除距离参考图形、选择参考坐标系以及改变栅格背景配色等。该模块通过多种灵活的交互方式提升了数据展示的可视化效果，使用户能够更高效、准确地理解和分析信息。

5.2 总体设计

在本节内容中，将针对系统的总体设计展开论述，其涵盖系统架构设计、系统功能模块设计以及数据库设计。

5.2.1 系统架构设计

本文三维可视化系统架构如图5-2所示，其中包括客户端、服务器端、算法端以及数据库这四个层次。

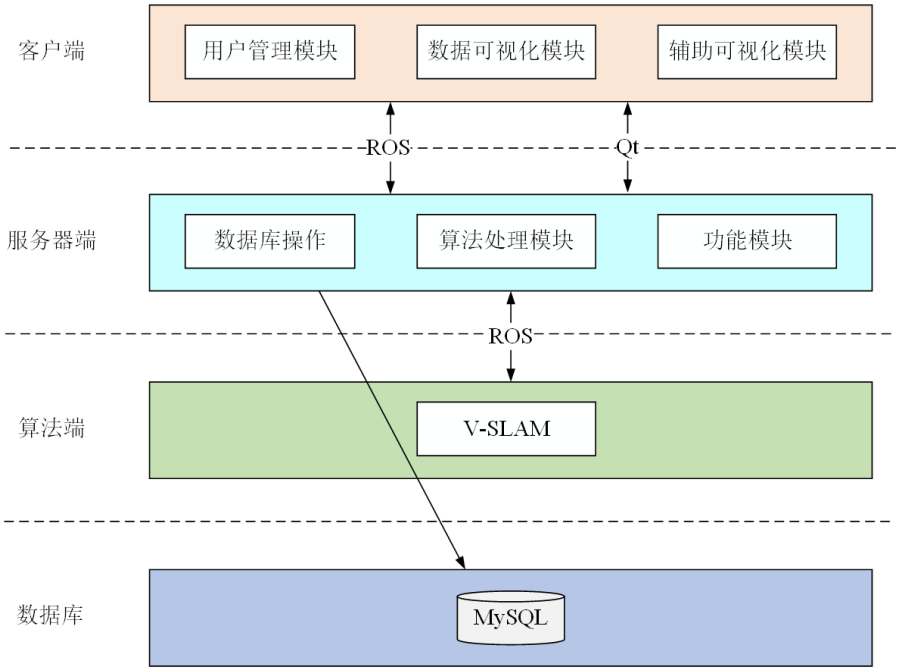


图 5-2 系统架构图

客户端负责与用户的直接交互，使用 Qt 框架实现。客户端具体可以分为用户管理模块、数据可视化模块和辅助可视化模块这三个部分。在用户登录前，显示界面只显示背景栅格地图，客户端功能界面中的功能按钮都是处于不能使用的状态。用户在用户管理模块上进行注册并登录，在身份验证成功后可以随意使用功能界面中的功能按钮。用户登录后可以在数据可视化模块中选择文件夹并通知服务端进行 V-SLAM 算法，在算法进行途中显示界面会实时显示相机位姿。在算法结束后用户可以选择生成的点云地图并展示。用户通过鼠标点击按钮或者键盘输入信息发出信号，Qt 框架的信号与槽机制可以实现客户端通过信号与槽机制通知服务端进行相应的操作。辅助可视化模块包括视角控制、增加或删除参考图形、修改栅格背景配色等。显示界面上根据用户在功能界面上的操作，显示服务端的不同响应结果。

服务器端的核心职责是处理来自客户端的请求，通过 ROS 与算法端进行通信

并通过 TCP/IP 协议与数据库保持联系。服务器端具体可以分为数据库操作模块、算法处理模块和功能模块这三个部分。数据库操作模块负责管理用户的身份信息，包括用户注册、登录与退出。在用户注册账号的时候将账号和密码信息保存在数据库中，确保用户信息的安全性。算法处理模块用来接收 V-SLAM 数据集文件并触发算法。用户在客户端选择文件夹并请求 V-SLAM 生成位姿轨迹服务器端通过 ROS 的服务调用机制向算法端发起请求，指示算法端执行 V-SLAM 算法生成位姿轨迹并返回到客户端显示界面上。功能模块用来处理客户端发来的不同功能请求。

算法端通过与服务器端的 ROS 通信，接收来自服务器端的请求数据并出发 V-SLAM 算法并生成相机的位姿，同时返回给服务器进行相机位姿的处理和显示。

数据库承担着系统内所有数据的存储与管理职责。其中，用户信息表用于存放用户的账户相关信息，像用户名、密码以及系统日志信息之类，这些数据均保存在 MySQL 数据库当中。

5.2.2 系统功能模块设计

三维可视化系统功能模块设计如图5-3所示，主要由用户管理模块、数据可视化模块和辅助可视化模块这三个模块组成。

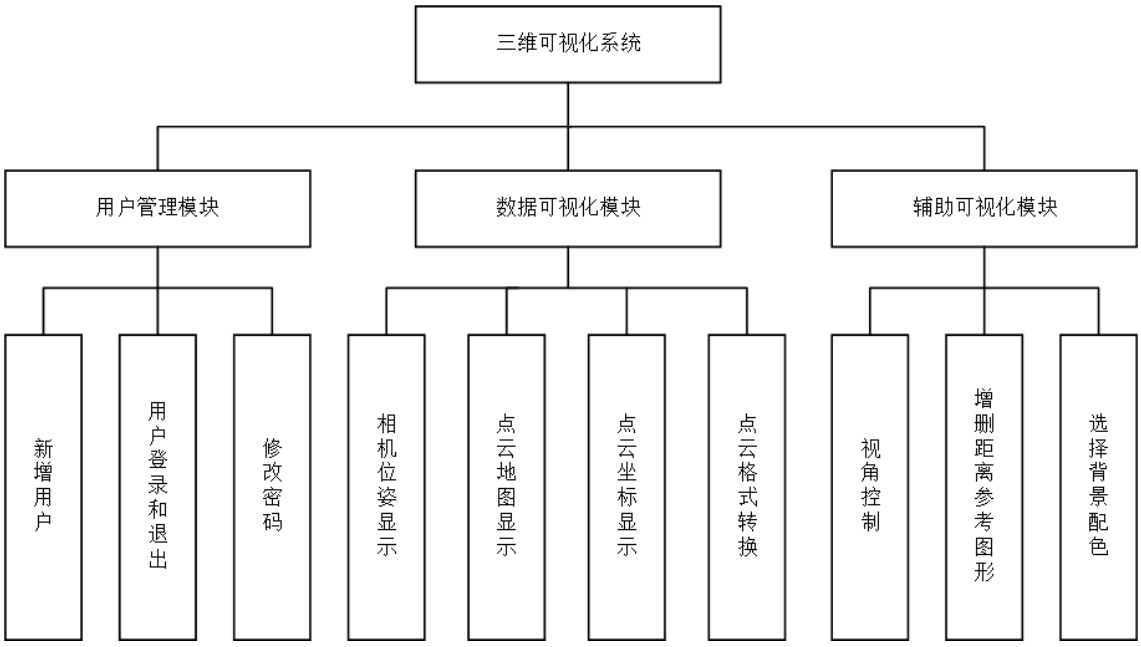


图 5-3 系统模块概要图

用户管理模块作为系统与用户交互的核心部分，主要负责用户的注册、登录和退出以及权限管理功能。通过该模块，用户能够创建账户、修改密码并退出登录。用户可通过注册功能在系统中创建新账户，注册时需要填写关键的个人信息，如用户名、密码等。系统提供用户登录接口，允许已注册用户通过输入正确的用

户名和密码进行身份验证。同时用户也可以随时退出登录，结束使用系统。为确保用户账户的安全性，系统提供密码修改功能允许用户在登录后更改其账户密码。如果是 root 用户则可以查看系统日志信息，方便系统的管理和维护。

数据可视化模块负责展示由 V-SLAM 算法生成的数据，特别是三维地图和位姿信息。该模块是系统的核心功能之一，需要实现以下功能：

1. 相机位姿显示。该功能显示实时的相机位姿信息，使用户能够了解三维空间中相机所在位置。要实现该功能，首先要选定输入数据集并传输到算法端进行 V-SLAM 算法。同时，要实时地将位姿信息从算法端传送到服务端。通过服务端的操作将位姿轨迹在客户端的可视化界面进行实时展示。

2. 点云地图显示。V-SLAM 算法在运行结束后生成三维点云稠密地图并保存在文件夹下。用户选择点云文件后，点云稠密地图显示在客户端的可视化界面。用户可以直观地查看环境的空间结构。用户也可以加载并查看之前生成的点云地图文件，该功能支持 PCD 点云格式和 PLY 点云格式的显示。

3. 点云格式转换。支持将生成的点云数据从一种格式转换为另一种格式，可以支持 PCD 点云格式与 PLY 点云格式之间的相互转化，便于与其他工具或系统进行集成。

4. 点云坐标显示。该功能可以通过用户拖动鼠标选中一个或多个点云点，返回被选中点云点的 x、y、z 世界坐标信息供用户参考点云稠密地图中点云的位置。

辅助可视化模块提供额外的可视化功能，增强系统的用户体验和操作便利性。此模块包括：

1. 视角控制。允许用户调整视角，以便从不同角度查看三维点云地图。用户可以通过鼠标设备进行交互式控制，放大、缩小或旋转视图。并且系统提供正视图、俯视图、左视图和右视图视角控制按钮，能够让点击相应按钮后可视化界面发生相应的视角转换。

2. 增加距离参考图形。为帮助用户更好地理解地图上的空间关系，系统提供了增加距离参考图形的功能。该功能包含显示坐标系和增删距离参考圈。用户可以在地图上添加 xyz 坐标系，帮助量化各个点之间的实际距离。用户通过键盘输入参考圈半径，通过点击按钮增加一个或多个距离参考圈辅助观察。在不需要参考圈的时候可以点击按钮进行删除。

3. 选择背景配色。用户可以根据需要调整背景的颜色，以提高点云地图的可视性和对比度。不同的背景颜色有助于增强用户在不同环境条件下的可视效果。

5.2.3 数据库设计

本系统选用 MySQL 数据库管理系统，以此保障系统数据能够实现快速的读取与写入。为方便用户管理以及数据记录存储，设计了 root_user 表格、user 表格以及 log 表格。

表 5-4 root_user 表

字段	类型	是否为空	键类型	默认值
account	VARCHAR(255)	否	PRIMARY KEY	NULL
password	VARCHAR(255)	否		NULL

如表5-4所示，root_user 表存储系统超级管理员的账户信息，主要包含账户名和密码。account 为超级用户的账户，不允许为空并且采用可变长字符串类型，作为主键确保账户作为用户的唯一标识符。password 为超级用户的密码，同样采用可变长字符串类型，不允许为空，用于存储管理员的登录密码。表中的数据通过 account 字段确保唯一性，系统中只允许有一个超级管理员账户。

表 5-5 user 表

字段	类型	是否为空	键类型	默认值
account	VARCHAR(255)	否	PRIMARY KEY	NULL
password	VARCHAR(255)	否		NULL

如表5-5所示，user 表用于存储普通用户的账户信息。

表 5-6 log 表

字段	类型	是否为空	键类型	默认值
id_pk	INT	否	PRIMARY KEY	NULL
time	TIMESTAMP	否		NULL
level	VARCHAR(255)	否		NULL
functionName	VARCHAR(255)	否		NULL
threadId	VARCHAR(255)	否		NULL
action	VARCHAR(255)	否		NULL
account	VARCHAR(255)	否	FOREIGN KEY	NULL

如表5-6所示，log 表用于存储系统运行时的日志信息，只允许 root 用户访问以便进行系统调试。id_pk 字段作为日志的主键，采用整数类型并设置为自动递增，以保证每条日志记录的唯一性。time 字段记录每次日志事件发生的精确时间，确保能够追溯事件的具体时刻。为了分类和区分不同的日志事件，level 字段采用可

变长字符串类型，表示日志的严重性级别。常见的级别包括信息、警告和错误等，能够帮助用户根据日志的级别进行事件的筛选和处理。functionName 字段记录生成该日志的具体函数名，便于 root 用户快速定位代码中的问题来源。日志记录还包括当前操作的线程信息。action 字段则用于详细描述日志所记录的具体操作，帮助用户了解系统中发生了哪些行为，例如用户请求、系统错误等。account 字段作为外键与 user 表中的 account 字段关联，确保每条日志能够明确记录操作用户的信息。通过与用户表的外键关系，系统能够追溯到具体的操作用户以便问题排查。

5.3 详细设计

5.3.1 用户管理模块

用户注册登录流程图如5-4所示：

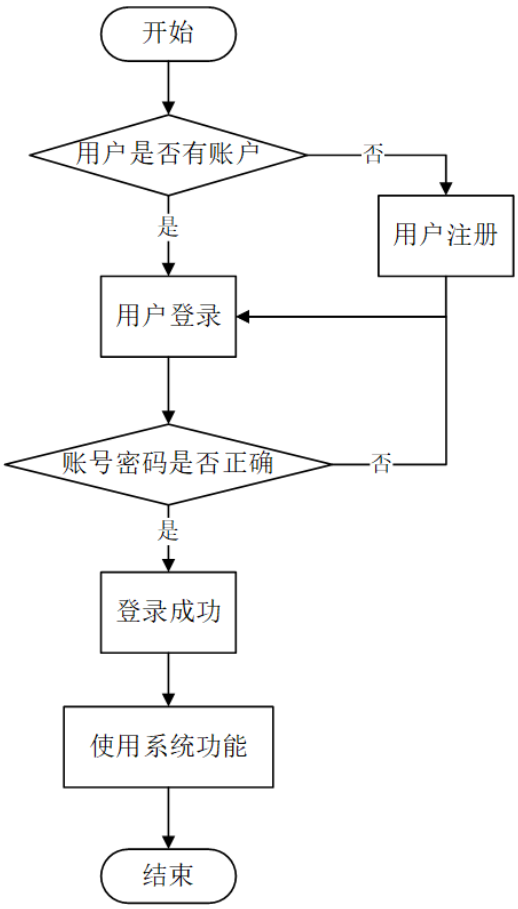


图 5-4 用户注册登录流程图

用户能够通过注册功能在系统中创建新账户。在注册环节，用户需要提供关键的个人信息，如用户名、密码。系统会对这些信息进行验证，确保用户提供的用户名是唯一的，并通过加密算法保存用户的密码，确保用户的账户信息得到安全

保护。一旦用户完成注册，系统会为其创建一个新账户，并为其分配初步的权限，允许其登录和使用系统的基本功能。用户可以随时通过登录界面输入正确的用户名和密码进行身份验证，验证通过后即可进入系统。为了增强账户安全性，系统还提供密码修改功能，允许用户在登录后根据需要更改密码。系统还提供了用户退出功能，用户在完成使用后可以通过点击退出登录按钮退出系统，系统会清除用户的登录信息，确保账户不被他人滥用。

5.3.2 数据可视化模块

5.3.2.1 相机位姿与点云地图显示

相机位姿显示算法的流程图如图5-5所示，详细描述了用户在显示相机位姿轨迹的操作步骤以及系统的流程。

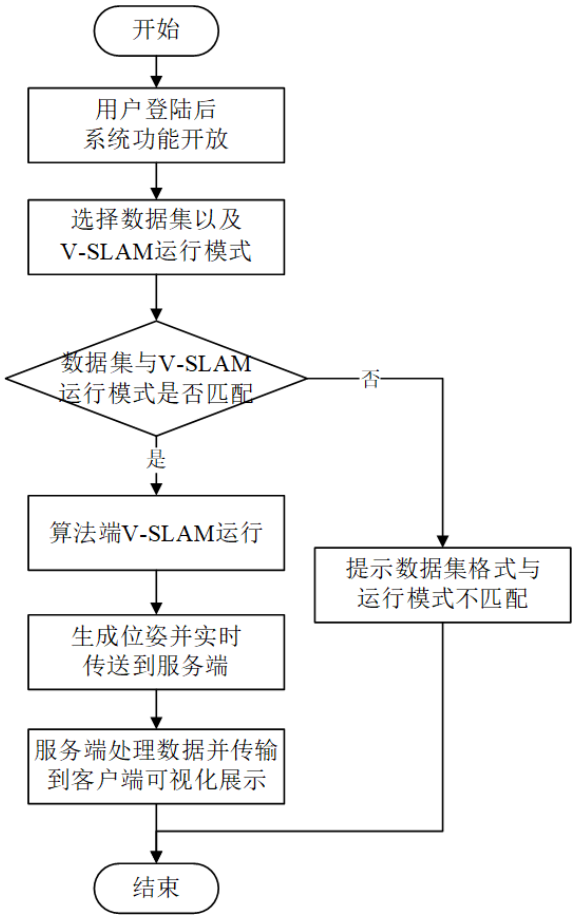


图 5-5 相机位姿显示功能流程图

相机位姿与点云地图显示时序图如图5-6所示。以此阐释相机位姿与点云地图显示模块内各组件之间的动态交互情况以及时间依赖关系。该时序图呈现出用户、客户端、服务器端以及算法端之间的交互顺序。

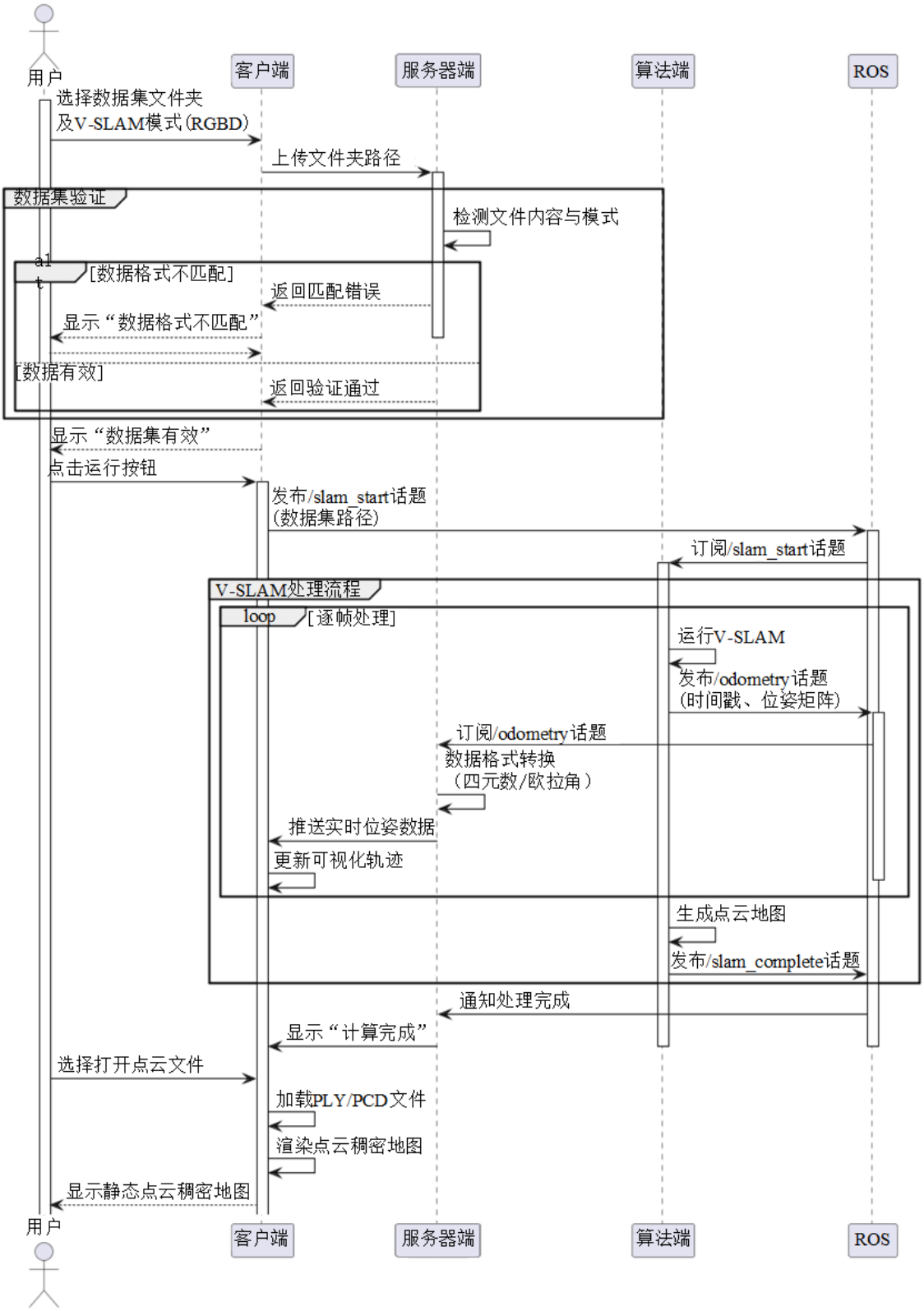


图 5-6 相机位姿与点云地图显示时序

当用户成功登录后，系统功能对用户开放。用户需要选择一个本地文件夹作为 V-SLAM 算法的输入数据集并且选择 V-SLAM 运行模式为 RGBD 模式。用户

选择好文件夹后，服务端检测文件夹中的数据集是否符合 V-SLAM 运行模式。如果不符合则提示选择文件夹中数据集格式与 V-SLAM 运行模式不匹配。如果上传的文件不包含有效的 V-SLAM 数据或者文件夹中数据格式不符合 V-SLAM 运行模式，系统提示后会跳过算法计算的步骤直接结束流程。如果文件中包含所需的数据，用户点击运行按钮就会通过 Qt 的信号与槽函数机制跳入槽函数发送 ROS 话题，算法端接收到启动信号运行 V-SLAM 算法。在这个过程中，算法端会将计算得到的位姿数据并通过 ROS 发布话题传输到服务器端。服务器端接收到数据后进行数据处理并将处理结果返回给客户端，在客户端的 GUI 可视化界面上进行位姿轨迹显示。在 V-SLAM 算法位姿估计结束后，会通过相机位姿文件，以及 V-SLAM 算法生成的补全后的 RGB 图像和深度图像生成点云地图并保存在用户选择的本地文件夹中。此时 V-SLAM 位姿已经全部生成，客户端可视化界面上位姿显示结束。用户通过选择打开点云文件在可视化界面上进行点云稠密地图显示。

5.3.2.2 点云坐标显示

为了显示选择的点云地图中 3D 点的坐标信息，通过创建一个自定义面板来实现这一功能。首先，创建标签 QLabel 来显示静态文本，以提示用户该面板展示的是与当前选定 3D 点相关的数据。当用户通过鼠标在可视化界面中选择特定点云时，交互事件经由 SelectionManager 捕获并生成结构化属性数据，该数据通过 getPropertyModel 接口转化为树状数据模型注入 PropertyTreeWidget 组件。该组件采用 MVC 设计模式（Model-View-Controller），下方部署动态扩展的树状视图控件，通过网格布局管理器进行精确位置排布。系统将点云的空间坐标（XYZ）、色彩属性（RGB）、法向量等几何特征以层级化节点形式呈现，每个属性节点支持数值的实时刷新与交互式展开与折叠操作，同时通过 Qt 信号槽机制绑定数据变更事件以确保可视化界面的同步更新。

5.3.2.3 点云格式转换

点云格式转换功能允许用户选择 PLY 或 PCD 这两种点云数据格式，然后选择目标格式进行转换。用户通过点击客户端可视化界面上的格式转换按钮，系统会弹出本地文件选择界面。用户选择完输入的点云数据以及转换点云格式后通过点击确认按钮，服务器端会根据选择的格式进行转换并将转换后的点云文件存储在原文件同一目录下。此功能适用于需要进行点云数据格式转换的场景，特别是在点云数据处理、分析和可视化的工作流程中，用户可能需要不同格式之间的转换以适配不同的工具和系统。

5.3.3 辅助可视化模块

三维可视化系统不仅具备 V-SLAM 位姿和点云稠密地图等数据可视化功能，还具备辅助可视化功能模块。该模块的设计旨在提供多种功能增强用户的交互体验，并帮助用户更直观地理解和操作点云数据。辅助可视化模块主要包含视角控制、增删距离参考圈以及选择栅格背景配色。视角控制功能允许用户从不同的角度查看三维点云地图。用户可以通过鼠标设备进行交互式控制，灵活地放大、缩小或旋转视图，以便全方位地探索点云数据。此外，系统还提供了标准的视角控制按钮，包括正视图、俯视图、左视图和右视图，这些按钮能够快速切换到预设的视角，方便用户根据需求快速定位和观察目标区域。增加距离参考图形的功能可以增删坐标系和距离参考圈，帮助用户更好地理解地图上的空间关系。用户可以通过键盘输入参考圈的半径值，并通过点击按钮来添加一个或多个参考圈，以便辅助观察不同区域的空间关系。当不再需要参考圈时，用户也可以通过简单的操作删除不必要的参考圈，进一步简化界面显示。背景配色的选择功能使得用户可以根据不同的需求调整栅格背景的颜色，以提高地图的可视性和对比度。辅助可视化模块通过用户在客户端进行操作将信号传给 Qt 槽函数进行相应函数的处理，不仅提升了三维点云数据的可视化效果，还增强了用户的交互性和可操作性，满足了不同用户在点云数据分析中的多样化需求。

5.4 系统各模块实现

5.4.1 开发技术与环境

为了确保三维可视化系统的高效性、可扩展性和稳定性，采用了多种技术和开发环境。以下将详细介绍三维可视化系统的开发技术架构和不同开发环境的配置，涵盖了算法端、客户端、服务器端以及数据库的开发环境。

三维可视化系统的算法端主要涉及点云数据处理、三维重建、数据分析等核心算法的实现。为了实现高效的点云处理和实时数据分析，算法端采用了 Ubuntu 18.04 操作系统，并结合了 Python、PyTorch、ROS 和 C++ 进行开发。客户端是三维可视化系统中的用户交互界面，负责展示处理后的点云数据并与用户进行交互。客户端的开发环境基于 Ubuntu 18.04，采用 C++ 和 Qt 框架进行开发。服务器端主要负责管理系统的核心功能，同样在 Ubuntu 18.04 操作系统下进行开发，使用了 C++ 和 ROS 技术栈来提供高效的服务。为了存储和管理大量的点云数据和其他相关数据，本文使用 MySQL 数据库进行数据管理。

5.4.2 用户管理模块实现

三维可视化系统如图5-7所示，其中用户管理模块被红框标注出。

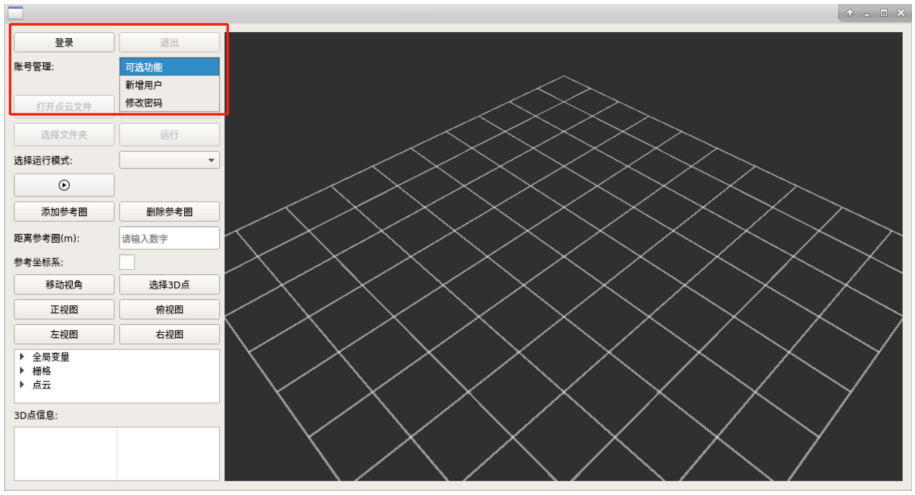


图 5-7 三维可视化系统用户管理模块

用户在没有登录的时候数据可视化的功能模块呈现灰色不可用状态。新用户通过点击账号管理中的新增用户功能，通过输入用户名和密码进行注册。注册后点击登录按钮，通过输入之前注册的用户名和密码登录系统。在客户端点击登录按钮后服务器端中用户管理模块接收到信号，此时用户管理模块会将用户输入的用户名和密码与数据库中的信息进行比对。如图5-8所示，当用户名和密码全部正确时服务器端会将登录成功的信息发送给客户端。此时会有窗口提示成功登录，这时数据可视化的功能模块解锁呈现可用的状态。如果用户名或密码错误，则将有对应窗口提示用户名错误或者密码错误。

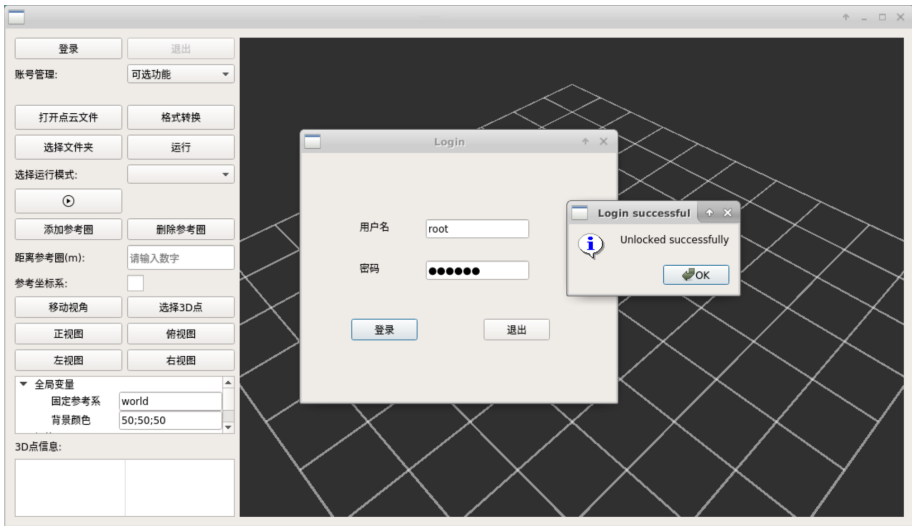


图 5-8 登录界面展示

5.4.3 数据可视化模块实现

用户登录之后数据可视化模块功能解锁，数据可视化功能模块的按钮从灰色不可用状态变成可用状态。如图5-9所示，用户通过点击选择文件夹按钮弹出文件路径选择界面进行 V-SLAM 输入数据集路径选择。通过选择文件夹以及从选择运行模式的下拉框中选择 RGB 格式后点击运行按钮开始生成位姿轨迹。如果运行模式与数据集文件夹不匹配则会弹出提示框提醒用户检查所选项。

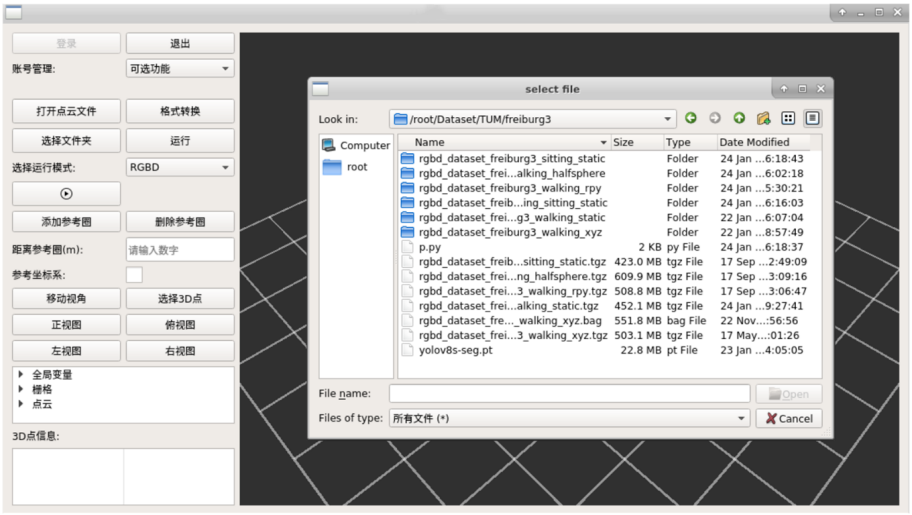


图 5-9 文件路径选择界面

如果运行模式与数据集文件夹匹配则会弹出提示框提醒用户文件匹配。如图5-10所示，此时点击运行按钮就会在右部可视化界面上显示相机位姿轨迹。

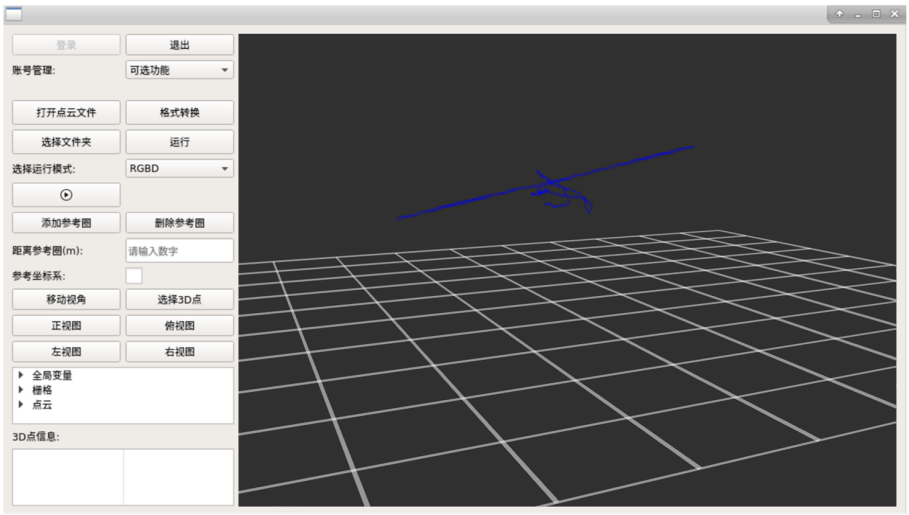


图 5-10 相机位姿显示

在 V-SLAM 运行结束后会生成对应的点云稠密地图保存在用户选择的文件夹目录下。用户通过点击打开文件夹按钮会弹出本地文件夹选择窗口。如图5-11所

示，选择点云文件之后用户输入对应点云格式的话题点击播放按钮，右部可视化界面上就会进行点云稠密地图显示。

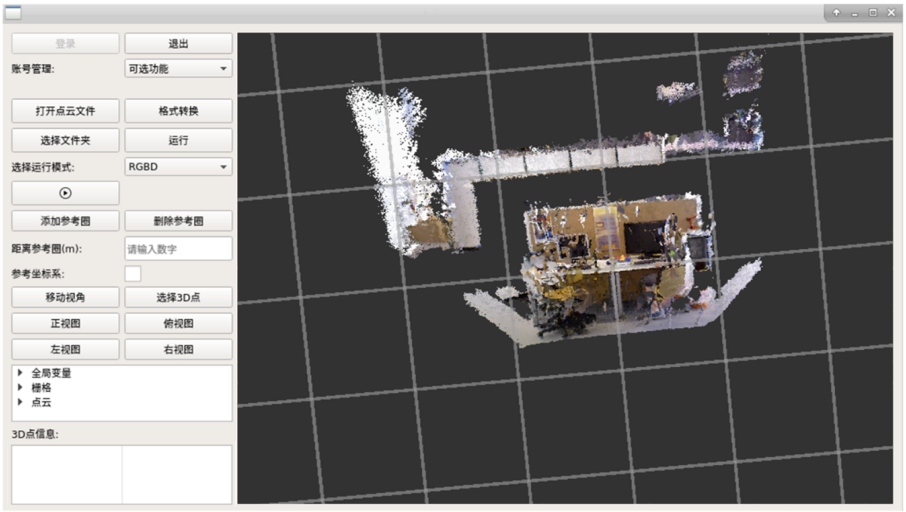


图 5-11 点云稠密地图显示

用户如果需要观察点云稠密地图中 3D 点的坐标信息，可以通过点击选择 3D 点按钮。此时就可以拖动鼠标选择需要查看的点云。此时 3D 点信息就会出现在系统界面左下方的信息框中。用户如果拖动鼠标选中多个 3D 点，则左下方信息栏中每个点云的信息都处于折叠状态。用户可以通过鼠标点击将信息进行展开。其中 3D 点信息包括 3D 点的坐标信息、点云颜色信息以及点云透明度信息。该 3D 信息栏支持实时刷新与交互式展开与折叠，确保界面与数据同步更新。

其次，系统支持点云格式转换功能，允许用户在 PLY 和 PCD 格式之间进行转换。用户通过点击格式转换按钮选择输入文件以及住处格式，点击确认后系统会进行格式转换并保存转换后的文件。

5.4.4 辅助可视化模块实现

用户在查看 V-SLAM 位姿轨迹或者点云稠密地图的时候，可以使用辅助可视化模块进行辅助观察。辅助可视化模块主要包含视角控制、增删距离参考圈以及选择背景配色。

用户点击移动视角按钮就可以改变固定视角，此时用户可以通过鼠标灵活地放大、缩小或旋转视图以便全方位地观察数据。用户如果需要图形辅助确定方位，可以点击参考系坐标进行 xyz 坐标轴显示。用户如果需要要图形辅助观察点云之间的距离可以输入距离参考圈的半径，通过点击添加参考圈按钮即可在 V-SLAM 数据可视化期间增加圆圈辅助观察。如图5-12所示，增加了半径为 1 米的参考圈辅助观察。当不再需要参考圈时，用户通过输入想要删除的距离参考圈的半径再点

击删除参考圈操作删除不必要的参考圈。

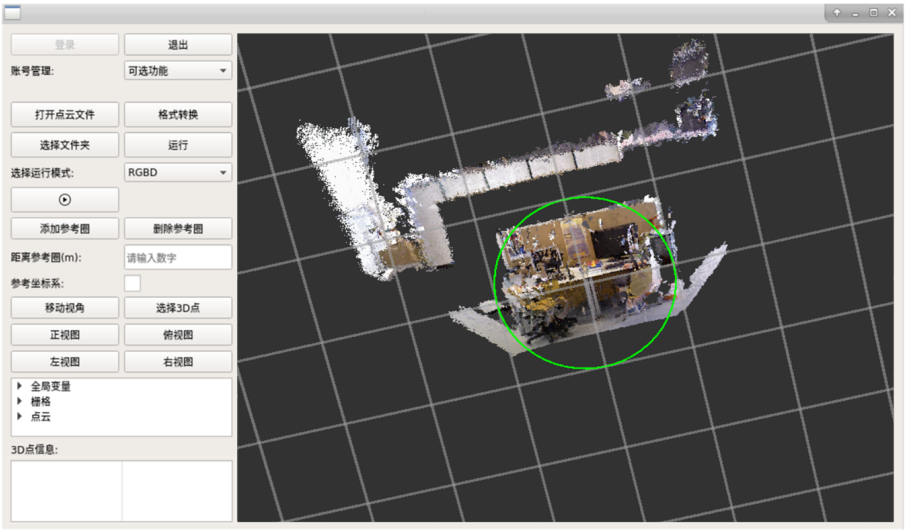


图 5-12 增加距离参考圈

背景配色的选择功能使得用户可以根据不同的需求调整可视化界面背景颜色以及栅格的颜色，以提高地图的可视性和对比度。用户可以通过点击辅助可视化栏中的栅格，对栅格的颜色、网格大小以及网格线条粗细进行调整，以符合用户自身使用习惯。也可以通过点击全局变量，通过键盘输入背景颜色对可视化界面的背景颜色进行更改。如图5-13所示，用户可以随意修改配色。

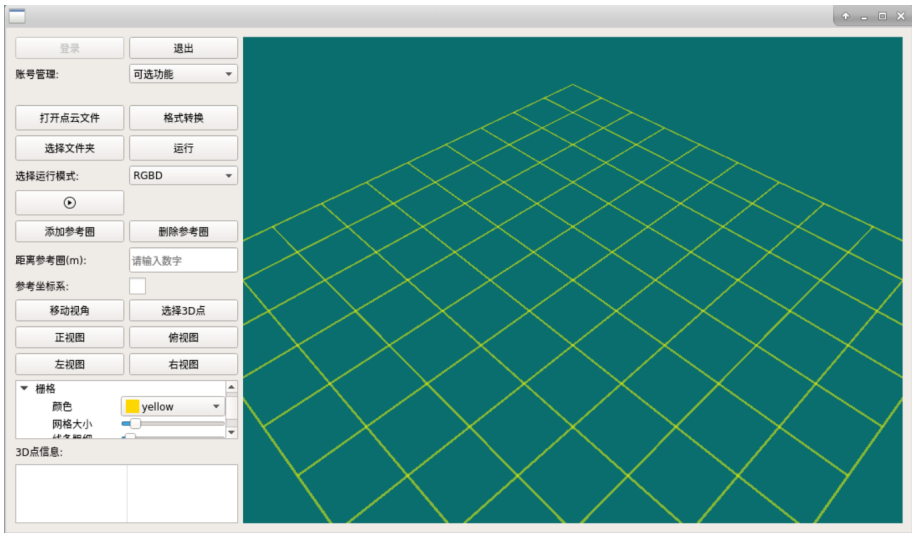


图 5-13 选择背景配色功能

5.5 系统测试

为了确保三维可视化系统的功能完备性及其性能稳定性，本章将从功能测试、性能测试展开系统验证，并基于测试结果对系统的有效性进行全面分析。功能测

试主要用于验证系统各项功能是否能够按预期运行，确保系统能够实现所设计的各项任务。性能测试则关注系统在复杂场景下的响应能力以及稳定性，旨在评估系统在高负载环境下的处理能力与流畅度。

测试环境配置涵盖了硬件方面和软件方面。硬件方面，采用 Intel Core i7-8750H 处理器、16GB 内存和 NVIDIA GeForce GTX 1050 Ti 显卡。软件方面，数据库采用 MySQL，系统运行 Ubuntu 18.04 LTS 操作系统，搭载 ROS Noetic 框架并使用 4.0 版本 Qt。

5.5.1 功能测试

在系统开发的过程中，为了确保最终交付的系统能够满足用户需求且具备高质量的性能，进行全面的功能测试是至关重要的。通过把整个系统视为黑盒实施测试，模拟用户对系统中各个功能模块的使用情况。对系统的功能及模块展开全方位的测试，以此来保证各个功能模块运行得到的结果与预期结果相符。三维可视化系统功能测试用例如表5-7所示。

表 5-7 系统功能测试用例

测试功能	预期结果	测试结果
用户注册与登录	输入合法用户名和密码，验证注册成功并生成账户文件。系统正确创建账户，拒绝重复用户名。	通过
功能权限	在没有正确登录的时候不能使用数据可视化模块功能按钮，只有在登录成功后才能使用。	通过
数据集与运行模式匹配	在选择 V-SLAM 运行模式与 V-SLAM 输入数据集时，如果格式不匹配则会弹窗提示用户重新选择，如果格式匹配则显示文件载入成功。	通过
点云格式转换	成功加载 PCD 文件并转换为 PLY 格式，成功加载 PLY 文件并转换为 PCD 格式。如果选择加载格式与转换格式相同则弹窗提示。	通过
点云坐标显示	选择单个点云只显示单个点云信息，拖动鼠标选择多个点云显示选中的多个点云信息。	通过
距离参考圈功能	添加多个不同半径的参考圈并删除其中一个或多个，距离参考圈按照添加和删除顺序显示。	通过
视角控制	点击移动视角按钮可以随意拖动可视化界面调整视角。点击不同视图按钮可以切换成相应的视角。	通过
选择背景配色	点击不同栅格颜色可以根据选择调整相应的栅格颜色。通过键盘键入背景颜色可以根据输入值调整相应的背景颜色。	通过

5.5.2 性能测试

为验证系统核心功能的运行效率，本节针对登录速度（数据库交互）、位姿显示速度（ROS 通信）和大型点云加载速度三个关键性能指标展开定量分析。系统性能测试用例如表5-8所示。

表 5-8 系统性能测试用例

测试功能	平均相应时间	最大相应时间
登录速度	43ms	59ms
位姿显示速度	172ms	235ms
5G 点云加载速度	6s	8s
10G 点云加载速度	10s	14s

5.6 本章小结

本章基于前两章提出的 GeoYOLO-SLAM 算法设计了一个三维可视化系统。在需求分析阶段，通过功能性需求分析与性能指标量化，结合用例图明确了系统的功能模块。三维可视化系统的三大功能模块分别是用户管理模块、数据可视化模块以及辅助可视化模块。通过总体设计，明确了三维可视化系统结构由客户端、服务器端、算法端以及数据库这四个层次。随后介绍了各个模块的详细设计与实现，展示了各个模块的客户端界面。最后进行测试，以验证系统的实际应用效果及其稳定性。

第六章 总结与展望

6.1 全文总结

随着智能家居的不断发展和移动机器人技术的不断进步，室内环境中的移动机器人应用正逐步走向成熟。在此背景下，SLAM 技术成为了机器人自主定位与环境感知的核心技术。尤其是基于视觉的 V-SLAM 凭借其低成本、高精度和较强的适应性，在实际应用中得到了广泛的关注和应用。然而，传统的 V-SLAM 算法通常假设环境为静态。动态物体对算法的影响尚未得到充分考虑，导致动态区域的特征点被识别并跟踪从而对相机位姿的估计精度产生了不利影响，甚至可能导致轨迹偏差或系统崩溃。本文通过对室内动态场景中的 V-SLAM 算法进行研究，旨在提升 V-SLAM 在复杂室内动态环境中的定位与建图性能。

首先，针对先验动态物体在实际环境中可能并不处于运动状态的问题，本文设计了 GeoYOLO-SLAM。该算法通过融合 YOLOv8 实例分割与深度图处理生成精细化物体掩码，并结合物体实例分割结果与多视图几何信息，准确区分并检测物体的真实运动状态。系统构建关键帧队列对前后关键帧进行处理，保证了实时性。通过为地图点赋予移动概率属性，解决单个关键帧观测不稳定的问题，并且有效避免了跟踪阶段动态特征点对相机位姿估计造成干扰。与现有开源算法相比，本文方法在动态物体处理上的优势使得系统能够更好地应对室内动态环境中的挑战。尤其是在先验动态物体处于静态和先验静态物体处于动态的状况，能够有效保持系统的稳定性和准确性。在公开数据集 TUM 数据集和 Bonn 数据集上进行了测试，证实了 GeoYOLO-SLAM 与先进算法相比取得了相当甚至更好的效果。

其次，本文提出了基于静态物体的非线性优化方法。结合所有静态物体的重投影误差、极线误差以及静态刚性物体的几何段误差进行联合优化，显著提升了相机位姿的精度。通过在公开数据集 TUM 数据集和 Bonn 数据集上进行了消融实验，证实了加上非线性优化后 GeoYOLO-SLAM 在原算法基础上有着较大的性能提升。在动态物体移除后，由于遮挡引起的背景缺失问题，本文提出了基于改进双线性插值算法的图像背景修复算法。图像背景修复算法恢复了被遮挡区域的几何和纹理信息，从而能够生成一个无动态物体重影的静态稠密点云地图。不仅提升了静态场景下地图的精度，也为机器人在动态环境中生成高质量地图提供了有效支持。在 TUM 数据集上进行测试，实验结果表明，本文构建的静态点云稠密地图与传统建图相比去除了动态物体的重影，得到了可视化效果更好的点云稠密地图。

此外，本文还设计并实现了一个三维可视化系统，该系统能够实时显示 V-

SLAM 的位姿轨迹，并展示动态更新的点云稠密地图。该可视化系统不仅具备实时性，还通过增加用户交互功能，使得用户能够更加直观地理解和操作 SLAM 系统。这种交互式的可视化方式为机器人应用提供了更好的可操作性和用户体验，也为研究人员提供了强有力的工具，便于调试和优化 SLAM 系统。

6.2 工作展望

尽管本文提出的算法在复杂室内动态场景中展现了显著的优势，但仍存在一些挑战和改进空间。具体需要优化的方向如下所示：

1. 动态物体的长期状态管理与地图一致性维护仍需深入优化。当前系统在处理间歇性运动的物体（如走走停停的人）时，可能因物体多次处于静态状态而在物体静止时保留关键帧图像上的物体信息，导致物体重复建图和场景冗余。这一问题的核心在于缺乏物体的概念，因此可以考虑增加对物体的管理。当物体多次在相机视角中静止的时候选择对比当前关键帧静止物体与之前关键帧中的静止物体进行匹配，如果确定是同一个物体可以消除之前关键帧中物体在点云地图中的点云，只保留最新关键帧中的该静止物体。这样在物体发生阶段性移动时自动触发局部地图修正，可以避免重复建图问题。

2. 实例分割的精度与鲁棒性仍有提升空间。现有 YOLOv8 实例分割模型在复杂遮挡或非刚性物体场景中可能出现过度分割，如将单物体分割为多个实例。针对此问题，会将同一个物体划分为多个物体影响系统管理物体信息。可进一步设计轻量化边缘细化模块，联合优化分割掩码的完整性与边缘精度，确保物体实例的准确提取。

3. 背景修复技术需突破对未知区域的生成能力限制。现有基于区域匹配的纹理补全算法依赖历史观测数据，难以修复动态物体移除后暴露的完全未知背景，如从未被相机观测过的墙面结构。之后可加入生成式深度学习与物理约束融合的修复框架，构建时空连贯的生成对抗网络（ST-GAN），利用历史关键帧序列的上下文信息指导缺失区域生成。在对抗训练中引入平面约束、光照一致性等物理规则作为损失函数，可增强生成内容的物理可信度，减少视觉伪影。

致　谢

行文至此，落笔为终。值此论文付梓之际，回顾三载硕士求学历程，恍如白驹过隙。从初入科研领域的懵懂探索到今日能独立开展系统性研究，这段旅程不仅是对学术能力的锤炼，更是对心智与品格的深刻磨砺。在此，谨以诚挚之心，向所有曾经给予我关心、支持与帮助的人表达我最真挚的感谢。

首谢恩师，春风化雨。承蒙 XXX 和 XXX 的悉心指导，老师们的无私帮助让我在学术研究和人生道路上受益匪浅。XXX 不仅在学术上给予我深刻的启发，还在生活中关心我的成长和发展。XXX 引导我思考更广阔的职业道路，帮助我理清了自己未来发展的思路。无论是走政界、学界还是商界，XXX 都教导我要脚踏实地、不断提升自己的能力与视野。XXX 则教给我许多做学术的基本原则，尤其是教会我做学术前先学会做人。XXX 经常提醒我们，做人诚恳、踏实，做学术要有严谨的态度和宽广的胸怀。XXX 不仅在专业知识上给予我无私的传授，更在为人处事的原则上为我树立了榜样。

感激师兄，指点迷津。我要特别感谢团队中的师兄，在学术研究的道路上给予了我大量的帮助和支持。无论是在实验设计的细节上，还是在数据分析的技巧上，师兄总是耐心地指导我，分享他丰富的经验。在我遇到困惑时，师兄不仅给予我技术性的帮助，更通过讨论和建议启发了我思考的方向。此外，师兄不仅是学术上的引路人，更是我生活中的良师益友。无论是在忙碌的实验室时光，还是在轻松的团队聚会中，都让我感受到了团队的温暖与支持。

同窗之谊，金兰之契。我还要感谢我的朋友们。感谢你们在这三年里的陪伴与鼓励，与你们的相识和相伴让我度过了一个充实且愉快的研究生时光。感谢我的同学们，你们在我学术与生活中提供了无数帮助和温暖的关怀，陪伴我一起度过了研究生阶段的每一个难关。尽管我们即将各自踏上不同的道路，但我坚信我们的友谊会一直延续，愿你们未来无论身处何地，都能拥有光明的前途，追逐梦想，勇敢前行。

再谢亲恩，椿萱并茂。感谢父母在这三年里给予我无条件的支持与鼓励，成为我最坚实的后盾。在我最需要支持与理解的时刻，是父母给了我最温暖的陪伴与鼓励，尤其是在忙碌的研究工作和生活压力下，家人始终是我最坚强的支撑。无论未来我走到哪里，你们的爱与支持都将是我前行的不竭动力。

参考文献

[1] Gonzalez-Aguirre J A, Osorio-Oliveros R, Rodríguez-Hernández K L, et al. Service robots: Trends and technology[J]. Applied Sciences, 2021, 11(22): 10702.

[2] Bogue R. Domestic robots: Has their time finally come?[J]. Industrial Robot: An International Journal, 2017, 44(2): 129–136.

[3] 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.

[4] Zhong X, Chai H, Pu Q, et al. A semantic visual slam for dynamic environments[J]. EasyChair, 2022, (9027).

[5] 危双丰, 庞帆, 刘振彬, et al. 基于激光雷达的同时定位与地图构建方法综述 [J]. Application Research of Computers/Jisuanji Yingyong Yanjiu, 2020, 37(2).

[6] Tourani A, Bavle H, Sanchez-Lopez J L, et al. Visual slam: What are the current trends and what to expect?[J]. Sensors, 2022, 22(23): 9297.

[7] Lee T j, Kim C h, Cho D i D. A monocular vision sensor-based efficient slam method for indoor service robots[J]. IEEE Transactions on Industrial Electronics, 2018, 66(1): 318–328.

[8] Han S, Xi Z. Dynamic scene semantics slam based on semantic segmentation[J]. IEEe Access, 2020, 8: 43563–43570.

[9] FISCHLER AND M. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Commun. ACM, 1981, 24(6): 381–395.

[10] 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.

[11] Qiu Y, Wang C, Wang W, et al. Airdos: Dynamic slam benefits from articulated objects[C]. 2022 International Conference on Robotics and Automation (ICRA), IEEE, 2022: 8047–8053.

[12] Liao X, Cai Z, Chen J, et al. Physics-based optical flow estimation under varying illumination conditions[J]. Signal processing: Image communication, 2023, 117: 117007.

[13] 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.

[14] Furrer F, Novkovic T, Fehr M, et al. Incremental object database: Building 3d models from multiple partial observations[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018: 6835–6842.

[15] 邹斌, 林思阳, 尹智帅, et al. 基于 yolov3 和视觉 slam 的语义地图构建 [J]. Laser & Optoelectronics Progress, 2020, 57(20): 201012.

[16] Liu H, Zhang G, Bao H. A survey of monocular simultaneous localization and mapping[J]. Journal of Computer-Aided Design & Computer Graphics, 2016, 28(6): 855–868.

[17] 曹之乐, 严中红, 王洪. 双目立体视觉匹配技术综述 [J]. 重庆理工大学学报 (自然科学), 2015, 29(2): 70–75.

[18] Davison A J, Reid I D, Molton N D, et al. Monoslam: Real-time single camera slam[J]. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(6): 1052–1067.

[19] Hu B, Luo J. A robust semi-direct 3d slam for mobile robot based on dense optical flow in dynamic scenes[J]. Biomimetics, 2023, 8(4): 371.

[20] Salas-Moreno R F, Newcombe R A, Strasdat H, et al. Slam++: Simultaneous localisation and mapping at the level of objects[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2013: 1352–1359.

[21] Engel J, Schöps T, Cremers D. Lsd-slam: Large-scale direct monocular slam[C]. European conference on computer vision, Springer, 2014: 834–849.

[22] Al-Mutib K, Mattar E, Alsulaiman M. Implementation of fuzzy decision based mobile robot navigation using stereo vision[J]. Procedia Computer Science, 2015, 62: 143–150.

[23] 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.

[24] 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.

[25] Rosten E, Drummond T. Machine learning for high-speed corner detection[C]. Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9, Springer, 2006: 430–443.

[26] Calonder M, Lepetit V, Strecha C, et al. Brief: Binary robust independent elementary features[C]. Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11, Springer, 2010: 778–792.

[27] 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.

[28] Triggs B, McLauchlan P F, Hartley R I, et al. Bundle adjustment—a modern synthesis[C]. Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings, Springer, 2000: 298–372.

[29] 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.

[30] 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.

[31] Yang S, Scherer S. Cubeslam: Monocular 3-d object slam[J]. IEEE Transactions on Robotics, 2019, 35(4): 925–938.

[32] Henein M, Zhang J, Mahony R, et al. Dynamic slam: The need for speed[C]. 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020: 2123–2129.

[33] Zhang J, Henein M, Mahony R, et al. Vdo-slam: A visual dynamic object-aware slam system[J]. arXiv preprint arXiv:2005.11052, 2020.

[34] 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.

[35] Teed Z, Deng J. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras[J]. Advances in neural information processing systems, 2021, 34: 16558–16569.

[36] Teed Z, Deng J. Raft: Recurrent all-pairs field transforms for optical flow[C]. Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, Springer, 2020: 402–419.

[37] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]. Proceedings of the IEEE international conference on computer vision, 2017: 2961–2969.

[38] Fang H S, Li J, Tang H, et al. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time[J]. IEEE transactions on pattern analysis and machine intelligence, 2022, 45(6): 7157–7173.

[39] Gong C, Sun Y, Zou C, et al. Sfd-slam: a novel dynamic rgb-d slam based on saliency region detection[J]. Measurement Science and Technology, 2024, 35(10): 106304.

[40] Zhong F, Wang S, Zhang Z, et al. Detect-slam: Making object detection and slam mutually beneficial[C]. 2018 IEEE winter conference on applications of computer vision (WACV), IEEE, 2018: 1001–1010.

[41] 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, 2016: 21–37.

[42] Rother C, Kolmogorov V, Blake A. " grabcut" interactive foreground extraction using iterated graph cuts[J]. ACM transactions on graphics (TOG), 2004, 23(3): 309–314.

[43] 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.

[44] 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.

[45] Quigley M, Conley K, Gerkey B, et al. Ros: an open-source robot operating system[C]. ICRA workshop on open source software, Kobe, vol. 3, 2009: 5.

[46] 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.

[47] Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017: 2881–2890.

[48] Li D, Shi X, Long Q, et al. Dxslam: A robust and efficient visual slam system with deep features[C]. 2020 IEEE/RSJ International conference on intelligent robots and systems (IROS), IEEE, 2020: 4958–4965.

[49] Ajit A, Acharya K, Samanta A. A review of convolutional neural networks[C]. 2020 international conference on emerging trends in information technology and engineering (ic-ETITE), IEEE, 2020: 1–5.

[50] Gálvez-López D, Tardos J D. Bags of binary words for fast place recognition in image sequences[J]. IEEE Transactions on robotics, 2012, 28(5): 1188–1197.

[51] Gorbachev Y, Fedorov M, Slavutin I, et al. Openvino deep learning workbench: Comprehensive analysis and tuning of neural networks inference[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019: 0–0.

[52] Cui L, Ma C. Sof-slam: A semantic visual slam for dynamic environments[J]. IEEE access, 2019, 7: 166528–166539.

[53] Cheng S, Sun C, Zhang S, et al. Sg-slam: A real-time rgb-d visual slam toward dynamic scenes with semantic and geometric information[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 72: 1–12.

[54] Liu H, Luo J. Yes-slam: Yolov7-enhanced-semantic visual slam for mobile robots in dynamic scenes[J]. Measurement Science and Technology, 2023, 35(3): 035117.

[55] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 779–788.

[56] 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.

[57] 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.

[58] Hansard M, Lee S, Choi O, et al. Time-of-flight cameras: principles, methods and applications[M]. Springer Science & Business Media, 2012.

[59] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60: 91–110.

[60] Lowe D G. Object recognition from local scale-invariant features[C]. Proceedings of the seventh IEEE international conference on computer vision, Ieee, vol. 2, 1999: 1150–1157.

[61] Chen L, Armstrong C W, Raftopoulos D D. An investigation on the accuracy of three-dimensional space reconstruction using the direct linear transformation technique[J]. Journal of biomechanics, 1994, 27(4): 493–500.

[62] Li S, Xu C, Xie M. A robust o (n) solution to the perspective-n-point problem[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 34(7): 1444–1450.

[63] Montemerlo M, Thrun S. Simultaneous localization and mapping with unknown data association using fastslam[C]. 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), IEEE, vol. 2, 2003: 1985–1991.

[64] Liu Y, Miura J. Rds-slam: Real-time dynamic slam using semantic segmentation methods[J]. Ieee Access, 2021, 9: 23772–23785.

[65] Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]. Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13, Springer, 2014: 740–755.