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 **基于 RGB-D 相机的物体级语义 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 TN828.6 密级 公开

UDC[注 1] 621.39

学 位 论 文

基于 RGB-D 相机的物体级语义 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 1 月 1 日 论文答辩日期 2025 年 1 月 1 日

学位授予单位和日期 电子科技大学 2025 年 2 月 2 日

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 Object-level Semantic SLAM Algorithm Based on RGB-D Camera

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：作 者 签 名　　日期：　2025 年 01 月 1 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：作 者 签 名　　导师签名：导 师 签 名

日期：　2025 年 01 月 1 日

摘 要

随着移动机器人、增强现实和自动驾驶技术的快速发展，对环境感知与自主导航的精度要求不断提高。然而，传统基于几何特征的同步定位与地图构建（SLAM）方法在复杂动态环境中易受遮挡、光照变化和动态目标干扰的影响，难以提供稳定可靠的位姿估计和环境建模。为解决这一问题，融合深度学习与多模态感知的语义 SLAM 技术逐渐成为研究热点，该技术通过引入物体级语义信息提升系统的环境理解能力，从而在动态场景下实现更加鲁棒的定位和地图构建。本文提出了一种面向室内静态场景的语义 SLAM 算法，利用 RGB-D 相机获取的深度信息与语义分割结果相结合，实现物体检测与构。同时，通过语义特征约束优化位姿估计与回环检测，从而提升系统在复杂室内环境中的定位精度和稳定性。以下是本文做出的工作：

1. 本文针对 YOLOv8 分割网络生成的物体掩码进行了精细度优化，结合图像深度图信息对掩码边缘进行改进，使其更准确地贴合物体的真实轮廓。对提取的二维物体信息，构建了对应的三维物体模型，并设计了关联匹配算法，用于匹配每次观测到的物体。为了管理物体信息，建立了全局物体数据库用于动态存储物体数据。同时，引入了局部物体数据库，存储最近观测到的物体并用于物体关联。通过物体关联算法，实现了在不同观测视角下的物体匹配，从而构建更精确的物体模型，减少错误建图，提高语义 SLAM 系统的稳健性。

2. 利用物体信息优化 SLAM 系统的位姿估计，选择稳定的物体作为约束信息引入 SLAM 的捆绑优化，以利用物体质心信息作为约束项以减少位姿估计误差。同时，基于物体与其最近的 5 个物体之间的关系构建物体地图，并通过计算地图相似性和物体之间的语义相关性，实现回环检测，提高系统的全局一致性和稳健性。针对透明物体和反光物体，设计了个神经网络用于恢复物体深度，从而增强系统在复杂场景中的适应能力。

3. 针对设计的语义 SLAM 算法，构建了一个完整的 SLAM 系统。该系统采用 Qt 技术开发前端界面，并使用 MySQL 数据库存储系统信息。系统主要包括三个模块：用户管理模块、SLAM 算法模块以及文件管理模块。

关键词：室内，静态场景，语义分割，物体重建，语义 SLAM

ABSTRACT

With the rapid development of mobile robotics, augmented reality, and autonomous driving technologies, the accuracy requirements for environmental perception and autonomous navigation are continuously increasing. However, traditional geometry-based Simultaneous Localization and Mapping (SLAM) methods are prone to the effects of occlusion, lighting changes, and dynamic object interference in complex dynamic environments, making it difficult to provide stable and reliable pose estimation and environmental modeling. To address this issue, semantic SLAM technology, which integrates deep learning and multimodal perception, has gradually become a research hotspot. This technology enhances the system's environmental understanding by introducing object-level semantic information, thereby achieving more robust localization and map construction in dynamic scenes. This thesis presents a semantic SLAM algorithm for indoor static scenes, which combines depth information obtained from an RGB-D camera with semantic segmentation results to perform object detection and reconstruction. Additionally, semantic feature constraints are applied to optimize pose estimation and loop closure detection, thereby improving the system's localization accuracy and stability in complex indoor environments. The following summarizes the contributions of this work:

1.This thesis focuses on refining the object masks generated by the YOLOv8 segmentation network. By incorporating depth map information, the mask edges are improved to more accurately fit the true contours of the objects. For the extracted 2D object information, corresponding 3D object models are constructed, and an association matching algorithm is designed to match each observed object. To manage object information, a global object database is established for dynamically storing object data. Additionally, a local object database is introduced to store the most recently observed objects for object association. Through the object association algorithm, object matching is achieved from different observation viewpoints, thereby constructing more accurate object models, reducing mapping errors, and enhancing the robustness of the semantic SLAM system.

2.The object information is utilized to optimize the pose estimation of the SLAM system by selecting stable objects as constraint information and incorporating them into the SLAM bundle adjustment. This allows the use of the object's centroid information as a constraint term to reduce pose estimation errors. Meanwhile, an object map is constructed

based on the relationships between an object and its five nearest objects. By calculating the map similarity and the semantic correlation between objects, loop closure detection is achieved, improving the global consistency and robustness of the system. For transparent and reflective objects, a neural network is designed to recover the object's depth, thereby enhancing the system's adaptability in complex scenarios.

3.A complete SLAM system has been built for the designed semantic SLAM algorithm. The system uses Qt technology to develop the front-end interface and employs a MySQL database to store system information. The system mainly consists of three modules: the user management module, the SLAM algorithm module, and the file management module.

Keywords: Indoor, Static Scene, Semantic Segmentation, Object Reconstruction, Semantic SLAM

目 录

图目录

表目录

第一章 绪论

1.1 研究背景与意义

近年来，同时定位与建图（Simultaneous Localization and Mapping，SLAM）已成为基础研究的一个重要领域，因为它有望解决与自主探索型人工智能移动机器人相关的众多问题[1]。SLAM 技术的主要应用是使机器人能够在未知环境中实现自我定位和地图构建。其独特优势在于无需依赖先验地图或外部干预，能够自主探索并感知动态环境。因此，SLAM 在城市搜索与救援、地下采矿、水下监控、行星探索等多个领域展现出广泛的应用前景。SLAM 的核心目标是实现机器人在未知环境中的实时定位与建图，确保其能够在复杂环境中准确估计自身的位置，并同时构建环境的高精度地图。在此过程中，机器人通过感知设备采集周围环境的信息，并利用先进的算法对信息进行处理与融合，从而获得位置与地图的估计。根据不同的感知方式，SLAM 技术可以细分为多个类型，主要包括激光 SLAM（基于激光雷达）、视觉 SLAM（基于相机）以及惯性 SLAM（基于惯性测量单元）等。每种类型的 SLAM 都有其独特的优势和应用场景，激光 SLAM 在高精度定位方面表现突出，视觉 SLAM（Visual SLAM，VSLAM）则更适用于低成本和高度灵活的应用，而惯性 SLAM 则能够在复杂的动态环境下提供较强的鲁棒性。这些不同类型的 SLAM 系统在多种实际应用中，发挥着重要作用，并推动着机器人技术的不断进步与发展。

VSLAM 是一种基于相机图像数据的定位与建图技术。它通过相机获取环境的图像信息，并利用计算机视觉算法提取图像中的特征点，进而估算相机的位姿，同时构建环境地图。与激光 SLAM 相比，VSLAM 能够利用丰富的图像数据来捕捉更多的环境信息，尤其在纹理丰富、细节复杂的场景中，VSLAM 表现出较强的鲁棒性。由于相机设备的低成本、易于集成和高精度，VSLAM 已成为当前 SLAM 技术研究和实际应用中的重要方向，广泛应用于机器人导航、增强现实、自动驾驶等领域。在 VSLAM 系统中，前端的数据处理环节至关重要，它负责从相机图像中提取有用的信息并计算相机的运动轨迹。VSLAM 的前端主要分为两种方法：直接法和特征点法。直接法假设光度不变性，即假设在相邻两帧图像中，相同像素点的光强度保持不变。通过在相邻图像中匹配相同光强的像素点，直接法能够估算相机的运动。例如，LSD-SLAM（Large-Scale Direct Monocular SLAM）[2] 便采用了直接法，通过优化光度误差来推算相机位姿。这种方法的优点是能够利用每个像素的信息，因此适用于低纹理场景，但对光照变化较为敏感。特征点法通过提取

图像中的角点或其他特征点进行匹配，从而估算两帧之间的运动。ORB-SLAM[3]便是一个典型的采用特征点法的 VSLAM 系统，它通过提取并匹配图像中的 ORB 特征点，估算相机位姿并构建环境地图。特征点法通常具有较高的鲁棒性，特别在纹理丰富的场景中，能够较好地处理光照变化和局部模糊问题。此外，Tong Qin 等人提出的 VINS[4, 5] 系列是不同于 ORB-SLAM 的又一经典框架。其支持多种惯性传感器类型，包括 IMU、GPS 等。在此算法基础上也提出了很多新算法，比如 DynaVINS 算法[6] 设计了一个鲁棒捆绑优化来丢弃一些异常的特征。PL-VINS[7] 对回环检测误差、惯性误差及点线视觉重投影误差三项误差进行整合, 由此构建目标优化函数, 并利用滑动窗口双向边缘化策略, 实现回环检测与重定位。

随着技术的不断发展，移动机器人在多种复杂应用场景中面临更高的环境感知要求。深度学习的快速进步，尤其是 SegNet[8] 和 YOLO[9] 等神经网络的出现，为 SLAM 系统在应对复杂环境和提升定位精度方面提供了强大的支持。通过结合语义分析算法，SLAM 能够利用语义信息将数据关联从传统的像素级别提升到物体级别，极大地增强了机器人对环境的理解能力，从而形成语义 SLAM（Semantic SLAM）。语义 SLAM 系统分成两个部分：语义提取器（Semantic Extractors）和现代视觉 SLAM 框架[10]。这种方法不仅能够先验性地判断并处理动态目标，减少动态物体对系统的干扰，还能显著提高 SLAM 在回环检测和位姿优化等核心任务中的精度。此外，语义分析有助于机器人实现更高层次的自主理解以及改善人机交互。然而，目前的语义 SLAM 在语义地图构建上仍存在诸多不足，尤其是在静态物体数据关联、物体精准重建和前景分割的准确性等方面。为提升系统的整体性能，仍需对静态物体进行准确关联以此优化物体的三维重建，并进一步提高前景分割和物体检测的准确度，确保语义 SLAM 能够在复杂环境中提供更加精确、可靠的定位与建图能力。

1.2 国内外研究现状

SLAM 技术是机器人领域中一个重要的研究方向，其目标是使移动机器人在未知环境中能够实时进行自身位姿估计和地图构建。Cesar Cadena 等人将 SLAM 发展历程分为了三个阶段，经典时代（1986-2004）、算法时代（2004-2015）以及鲁棒感知时代（2015-现在）[11]。在经典时代，引入了 SLAM 的主要概率公式，包括基于扩展卡尔曼滤波器[12]、Rao-Blackwellized 粒子滤波器和最大似然估计等方法；此外，这个时期还涵盖了与效率和强大的数据关联相关的基本挑战。Durrant-Whyte 和 TimBailey 的两项工作[13, 14] 对经典时代的早期发展和主要公式结论进行了详细回顾，内容基本全面覆盖了整个经典时代的发展。接着是算法时代，Gamini Dissanayake 等人的

工作[15] 回顾内容涵盖这个时期的一些发展，并提出了一些 SLAM 面临的挑战。目前正处于鲁棒感知时代，其中涉及到一些新的挑战如，鲁棒性能、高层次理解、资源感知和任务感知、驱动推理。

VSLAM 前端传感器可以分为单目、双目以及深度相机。单目相机器尺寸小、价格低廉、易于校准且功耗较低[16]，仅利用单个镜头通过小孔成像原理捕捉周围环境的二维图像。根据几何光学原理，光线通过镜头的光圈（即小孔）投射到成像平面上，从而形成图像。单目相机计算上通常较为轻便，但深度估计较为困难，且对环境纹理及光照变化较为敏感。双目相机由两个相机镜头组成，通常以一定的基线（即两个相机之间的物理距离）并行布置。通过左右两个镜头从不同的视角拍摄同一场景，双目相机可以利用图像匹配算法计算出图像之间的视差，从而推导出场景中各个点的深度信息。其深度估计精度依赖于视差计算的质量，且计算复杂度较高。深度相机是以主动探测的方式直接捕捉场景深度信息的传感器，能够直接提供场景中每个像素的深度值，极大简化了深度估计过程。在低纹理环境或动态场景中仍能提供稳定的深度数据，但其缺点是较高的硬件成本和有限的测量精度，尤其是在较远距离或大范围场景中的深度精度较低。

Davison 等人于 2007 年提出的 MonoSLAM[17] 是较早使用单目相机的实时 SLAM 系统之一。该系统采用基于扩展卡尔曼滤波器的方法，通过跟踪图像中的特征点来构建环境地图并估计机器人位姿。MonoSLAM 的创新之处在于能够仅依赖单目相机提供的视觉信息来实现即时定位与地图构建，而无需借助额外的传感器。这种方法通过使用扩展卡尔曼滤波器有效地融合了视觉信息与运动模型，从而在动态环境中实现了准确的定位和建图。MonoSLAM 的成功为 VSLAM 技术的广泛研究和应用奠定了基础，也为后续的研究者提供了宝贵的思路和方法。此后，许多基于视觉的 SLAM 系统都借鉴并扩展了 MonoSLAM 的基本思想，推动了 VSLAM 在机器人导航等领域的迅速发展。

在 2007 年 Georg Klein 等人提出了 PTAM（Parallel Tracking and Mapping for Small AR Workspaces）算法[18]。PTAM 的追踪模块通过提取图像中的特征点并对其进行优化，精确估算相机的运动轨迹；而建图模块则基于这些特征点构建了一个包含环境信息的地图。PTAM 通过并行化处理追踪与建图任务，实现了实时的相机轨迹更新和环境地图构建，极大提高了系统的效率和实时性，尤其适用于室内环境。PTAM 的创新设计使得其成为双目 SLAM 的经典算法之一，为后续的双目 SLAM 技术的发展奠定了重要基础。其有效的追踪与建图分离策略为更大规模的 VSLAM 系统提供了宝贵的经验，并推动了机器人自主导航的进一步发展。

在 2011 年，Richard 等人提出了 DTAM（Dense Tracking and Mapping in Real-Time）[19] 算法。与传统的基于特征点的 SLAM 系统（如 PTAM）不同，DTAM 采

用了稠密建图的策略，不依靠稀疏的特征点来进行定位，而是通过计算每一帧图像的深度来进行三维场景的建模。该算法利用图像的每个像素来计算场景的三维结构，从而获得更加精细和准确的场景信息。这种方法通过对图像的深度信息进行追踪和优化，能够构建出密集的地图，极大提高了地图的细节和准确性。DTAM 采用了一种基于图像亮度一致性的优化方法，通过最小化当前视图与已建立的三维模型之间的亮度误差来实现图像对齐。这种方法避免了传统 SLAM 系统中依赖于特征点匹配的限制，改为直接使用图像的亮度值进行密集匹配，因此能够更高效地处理图像中的每个像素。然而，这也使得 DTAM 对光照变化非常敏感，可能受到环境光照条件的影响较大。DTAM 的提出为密集 SLAM 系统的发展提供了新的方向，尤其适用于需要高精度地图和细节的应用场景，如虚拟现实、增强现实和机器人导航等领域。其方法论突破了稀疏特征点的限制，在实时性和准确性上都取得了显著的进展。

在 2015 年 Mur-Artal 等人提出的 ORB-SLAM[3] 是一种基于特征的单目 SLAM，可以在小型和大型、室内和室外环境中实时运行，成为模块化 SLAM 领域的一项重要工作，很多后续出现的基于特征匹配的 SLAM 系统都是由 ORB-SLAM 发展而来。如 Mur-Artal 等人接下来提出的 ORB-SLAM2，在保持框架整体性的基础上对一些细节进行了改进，使其能够适用于更多种类的相机，包括深度相机和双目相机。此外，跟踪线程中引入了预处理模块，最后有一个全局捆绑优化（Bundle Adjustment, BA）提高系统的鲁棒性；ORB-SLAM3[20] 在此基础之上耦合了惯性传感器 IMU、加入了融合估计以及子地图功能。Wang 等人提出的算法[21] 利用基于开放窗口的关键帧聚类算法对关键帧进行聚类，减少位姿图优化的规模，以加速位姿图优化的速度。

SLAM 框架早期是建立在静态假设成立的基础上，即认为环境中的所有物体都是静态不动的，唯一移动的物体是传感器本身。这种假设导致在存在动态物体的环境中，位姿估计容易变得不准确，甚至在高度动态的环境中可能完全失效。为了解决这一问题，提出了动态场景下的 SLAM，即动态 SLAM。动态 SLAM 将环境中的物体分为了动态和静态两类来进行区分。在一些动态 SLAM[22–29] 中，动态物体被剔除，不纳入位姿估计计算当中。另一方面，一些 SLAM 框架采用不同的策略，将动态物体位姿进行估计并纳入到优化中[30]。

2012 年 Guo 等人提出 Coslam 算法[25]，该算法通过多个机器人协同工作，使得每个机器人都能共享其所感知的环境信息。并利用不同的视角或位置，提供更多的静态环境信息来帮助剔除动态物体的影响。最终在地图构建过程中忽略这些动态物体，确保仅对静态环境进行建图。

Daniela 等人在 2022 年提出的 STDyn-SLAM[27] 采用对极几何的方法，通过建

立当前帧和上一帧光流，根据对极几何约束判断是否为动态点，从而将动态点进行标记并剔除，避免这些物体影响定位和地图构建。

Wu 等人于 2022 年提出的 YOLO-SLAM[28] 将物体检测与 RANSAC 算法相结合，以去除动态物体。并且通过几何约束，确保动态物体在定位和地图更新时不会导致误差。

传统 SLAM 系统主要依赖于几何信息，在某些场景下可能限制了对环境的深度理解。随着人工智能技术的不断推进与发展[31]，机器人定位与导航技术越发成熟。深度学习方法被广泛用于图像特征提取、深度图的生成、对抗性训练，可以提高 SLAM 系统的鲁棒性等性能。深度学习为 VSLAM 系统获取更多的环境语义信息，增强对环境的高层次理解能力，从而更好的感知环境。在 SLAM 系统中加入语义信息，可以形成语义 SLAM。语义 SLAM 利用深度学习网络对物体进行分割，能更好的识别可能的动态物体，同时构建出包含语义信息的地图，在导航和环境交互等方面有更好的效果。

2013 年 Salas-Moreno 等人提出了 SLAM++ 算法[32]。该算法是一种基于对象级地图表示的增强型 SLAM 方法，能提高大规模和长期运行任务中的鲁棒性与精度。算法采用了对象级别的地图表示，能够捕捉更高层次的空间结构和语义信息，从而使得系统能更好地适应环境的变化。该方法通过图优化框架同时进行定位与建图，能够处理复杂的环境，并提升系统在长期运行中的稳定性。

2017 年 Martin 等人提出的 Co-Fusion[33] 利用 SharpMask[34] 通过使用运动和语义信息将场景分割成不同的对象，同时跟踪和重建真实的 3D 形状，并随时间推移改进物体在地图上的模型。

2018 年 Martin 等人进行改进，提出了 Mask-Fusion[35]，使用 MASK-RCNN[36] 网络对场景中的不同对象进行识别，并在 SLAM 线程之外添加了一个用于分割的语义线程，以提高系统的实时性。

2018 年 Nicholson 等人提出了 QuadricSLAM[37]。该算法将物体检测的二次曲面模型（如椭球体等）作为三维地标，结合传统的 VSLAM 技术进行优化，来提高相机姿态估计的准确性。算法利用因子图进行联合优化，其中包括了通过物体检测获得的曲面模型以及通过传统视觉里程计得到的运动估计。通过这种方法，QuadricSLAM 能够在复杂和动态的环境中提供更稳定的 SLAM 性能，尤其是在物体变化和视角变化较大的情况下，显著提高了相机定位和地图构建的鲁棒性。

2019 年 Yang 等人提出了 CubeSLAM[38]。该算法创新性的引入了具有明确几何形状的物体建模，特别是立方体模型。通过从图像中提取深度信息，并将物体识别与其几何模型结合，来提高稠密地图构建的效率。在此基础上，算法使用基于立方体几何的优化方法来精确估计物体在三维空间中的位置和姿态，同时通过全

局优化技术将物体信息融入到 SLAM 系统的地图中，从而提高了动态环境中的鲁棒性和精确性。这种方法特别适用于需要精准场景重建的应用，如增强现实、室内导航等。

2022 年 M Zins 等人提出了 OA-SLAM 算法[39]。该算法通过结合物体和点特征，增强了传统 SLAM 的重定位能力。算法使用 3D 椭球体模型表示物体，并通过 YOLO 检测网络在视频帧中实时检测物体。在相机跟踪丢失的情况下，OA-SLAM 能够利用这些物体进行相机姿态恢复，并恢复 SLAM 跟踪。该算法的创新点在于将物体信息与传统的点特征结合，使用物体检测提供更多的重定位锚点，使得在复杂场景中能够从不同视角进行相机重新定位。即使在一些经典方法失败的情况下，也能有效地继续 SLAM 工作。

2022 年 Cheng 等人提出的 SG-SLAM[40] 在 ORB-SLAM2 的基础框架上添加了两个新的并行线程：一个用于获取 2D 语义信息的对象检测线程和一个语义地图线程，然后利用语义信息和几何信息快速剔除动态点，使用静态特征点进行位姿估计，将静态物体构建到语义地图中。

2025 年 Bai 等人提出一种使用椭球模型表示物体的语义 SLAM[41]。这个算法会跟踪动态物体情况，并实时修改地图中物体位置，最终使用静态物体在算法跟踪丢失之后进行重定位。

语义 SLAM 通过有效结合语义信息与 SLAM 系统，使得机器人能够实现更高级的感知和理解。这不仅提高了定位与地图构建的精度和鲁棒性，还拓展了其在复杂应用场景中的适用性，是 SLAM 技术在理解和应用语义信息方面的一次重要演进，为实现智能机器人在现实世界中的自主行动和决策提供了更强的支持。

1.3 研究内容与创新点

本文针对室内静态场景下的语义 SLAM 算法开展研究。语义 SLAM 不仅仅侧重于几何结构的感知，还注重对环境语义信息的理解。在实际场景中，算法识别到的 2D 检测框中包含的背景信息可能对物体信息的准确使用造成一定的干扰，影响不同观测角度下的同一物体的数据关联，导致物体建模不准确。通过优化掩码增加物体准确性并使用物体信息加入捆绑优化以及使用物体结构信息检测回环情况。在 TUM 数据集上进行测试，相比于近期的算法有一定的优势。

本文的主要创新点如下：

1. 结合深度信息优化物体掩码，并利用三维信息进行物体关联。本文首先针对 YOLOv8[42] 过度分割与深度检测问题，提出了优化策略；接着，结合处理后的物体掩码与相应的深度信息，剔除前景干扰并对物体边缘进行精细化处理。同时，

构建了全局物体数据库，用于动态更新物体信息，并基于局部数据库进行物体关联操作。算法主要通过物体质心和类别信息，并利用颜色特征，进行精准的物体关联。

2. 利用物体信息加入捆绑优化和回环检测。挑选数据库中物体大小稳定的物体加入到捆绑优化中，构建新的优化项，加强对关键帧的位姿估计。在原本的回环检测流程之前，通过计算物体地图的相似性判断是否存在回环，加强对回环情况的检测，增强算法的鲁棒性。

3. 针对深度相机在反光与透明物体深度检测中存在的误差问题，本文提出了一个基于多尺度特征对比学习方法。并且通过在网络的不同层次提取特征，并通过特征融合模块进行整合，增强了模型对复杂物体表面细节的感知能力。

1.4 论文结构安排

第一章为绪论部分，首先介绍了本文的研究背景与意义，叙述了国内外 SLAM 的发展与研究现状，阐述了本文的研究内容与创新点，最后给出了本文的结构安排。

第二章为相关理论与技术部分，首先介绍了 VSLAM 前端所使用的相机类型及其对应的模型，接着阐述了 SLAM 的结构流程，并深入解析了前端优化中的特征点提取与匹配技术，以及后端优化中涉及的图优化方法。随后，介绍了本文所参考的 ORB-SLAM2 框架。最后介绍了使用到的八叉树地图。

第三章提出了一种基于 YOLOv8 识别网络的物体提取与关联算法，以及物体优化位姿估计算法。通过结合 IoU 与深度信息的方式，融合 YOLOv8 可能存在的过度分割问题，并在剔除深度图中的坏点后进行物体掩码优化。根据物体的深度信息，动态计算深度阈值和梯度阈值，以优化掩码，最终提取出三维物体。物体提取后，物体关联通过物体类别、质心位置以及颜色信息进行判断。并且通过挑选出稳定物体并将其加入到算法的捆绑优化中，进一步提升位姿估计的准确性。最后在 TUM 数据集上验证了提出的算法的可行性。

第四章提出了利用物体信息辅助回环检测的流程，并针对反光、透明物体等特殊情况，改进了神经网络算法，以恢复深度信息。回环检测算法通过物体与最邻近物体构建物体地图，并通过计算物体地图的相似性，辅助判断回环的存在。此外，设计了基于多尺度特征对比学习方法用于增强 RGB 图像和深度图像特征向量相似性，以提升深度恢复效果。

第五章基于本文提出的 SLAM 算法，构建了一个语义 SLAM 系统。首先，进行了系统需求分析，明确了各个模块的组成结构；接着，对每个模块进行了详细设

计，并逐一实现了相关内容；最后，展示了实现效果与系统的测试结果，充分验证了系统的可行性与性能。

第六章对本文的工作进行了总结，分析了当前研究中存在的不足之处，并针对未来可深入探讨的方向进行了展望。

第二章 相关理论及技术

2.1 相机模型

2.1.1 针孔相机成像模型

VSLAM 前端是使用相机来捕获周围环境，将三维立体信息捕捉到二维图像中。针孔相机成像模型（Pinhole Camera Model）是描述相机成像原理的基本模型，它通过一个理想化的简化假设来近似现实世界中的相机成像过程。针孔相机模型假设相机内部只有一个小孔，通过这个小孔，外部世界的光线投射到图像平面上，从而形成图像。

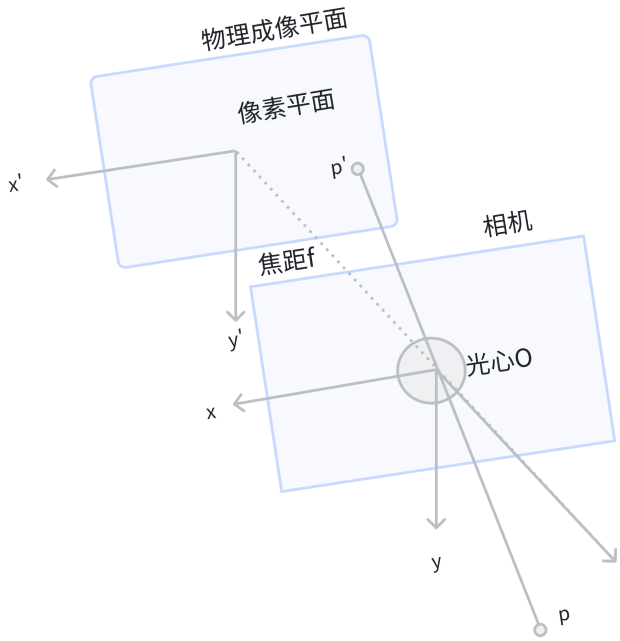


图 2-1 针孔相机成像模型

如图2-1所示，在针孔相机模型中定义了四个坐标系：世界坐标系、相机坐标系、图像坐标系以及像素坐标系，三维物体通过成像关系，从世界坐标系转换到二维的像素坐标系，形成图像。

世界坐标系是一个全局参考框架，用于描述环境中的物体和机器人在空间中的位置和姿态。由于相机在环境中是不断变化的，所以需要定义一个固定的坐标系来描述相机变化，所以它是一个固定的坐标系，通常被认为是与环境的物理特

征相关联的。世界坐标系的原点和轴的方向通常是根据实际应用需求进行定义的，一旦确定，它为机器人提供了一个统一的参考系，使得机器人能够在环境中进行定位、运动和地图构建。相机坐标系是描述相机当前位置而建立的坐标系，以相机的光心为原点，光轴为 z 轴，水平方向为 x 轴，垂直方向为 y 轴。通常可以用 $O-X_c-Y_c-Z_c$ 表示。图像坐标系是以光轴与图像平面的交点为原点、水平向右为 x 轴正方向，垂直向下为 y 轴正方向形成的二维坐标系，以 (X,Y) 形式描述图像坐标系的点坐标。像素坐标系是图像像素点的索引位置，是以图像左上角为原点，向右为 x 正方向，向下为 y 轴正方向形成的二维坐标系，通常以像素点位置为单位，用 (U,V) 表示坐标。

坐标系转换流程如图2-2所示。

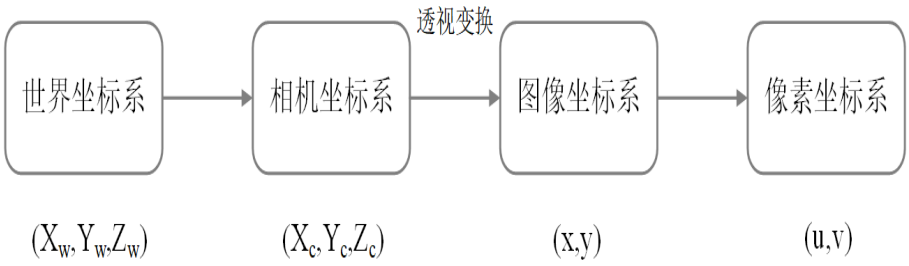


图 2-2 坐标系转换流程

假设相机坐标系下物体的表示为 (X_w,Y_w,Z_w), 通过相机光心 O 投影到二维相机平面上的像素点为 $P'(X_c,Y_c,1)$，由三角形的相似性原理可以得到式子2-1所示。

$$\begin{cases} \frac{Z}{f}=\frac{X}{X_c}=\frac{Y}{Y_c} \\ X_c=f\frac{X}{Z} \\ Y_c=f\frac{Y}{Z} \end{cases} \tag{2-1}$$

其中 f 表示相机的焦距。

将三维坐标点转换至二维平面时，最终需以像素坐标形式输出，因此还需对坐标进行平移与缩放转换。像素坐标系为 u-v 坐标系，且与图像平面坐标系平行，即 u 轴与 x 轴平行，缩放比例为 f_x；v 轴与 y 轴平行，缩放比例为 f_y，并且方向一致。图像坐标系的中心位于图像的中心，而像素坐标系的原点位于图像的左上角，因此需要引入平移量 c_x 与 c_y，其转换公式如式2-2所示。

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

其中 f_x、f_y、c_x 以及 c_y 都是相机内参，是相机固定的参数，始终保持不变。

2.1.2 双目相机

单目相机仅提供单一视角的二维图像信息，因此无法直接计算场景中物体的深度。具体而言，单目相机缺乏视差信息，无法从不同角度的图像中推导出物体到相机的精确距离。双目相机通过使用两个具有已知相对位置的相机，分别从不同的视角获取图像。通过对比两幅图像中相同场景点的位移（即视差），可以根据三角测量原理计算出这些场景点的深度信息。

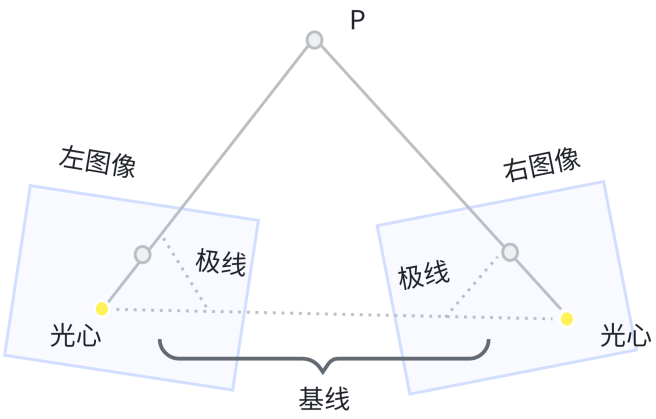


图 2-3 双目相机模型

如图2-3所示，双目相机模型两个光心的连线称为基线，物体点（空间点 P）与两个光心的连线构成的平面称为极平面；极平面与成像平面的交线为极线。

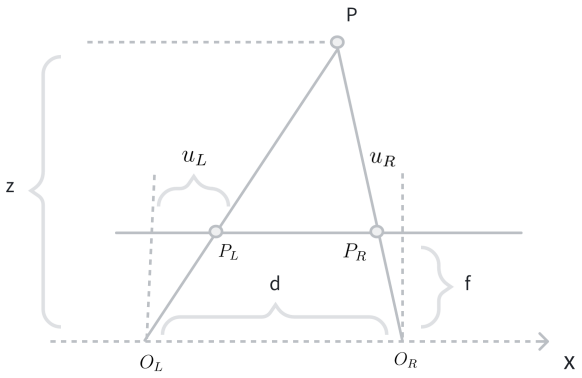


图 2-4 几何模型

如图2-4所示，点 P 是空间中需要计算的三维点。空间点 P 在双目相机左右两个相机平面成像点分别为 P_l 和 P_r，相机焦距为 f，基线长度为 d。通过三角形的

相似关系可以计算出 P 点的深度值，公式如2-3所示。

$$\frac{z-f}{z}=\frac{d-u_L+u_R}{d} \tag{2-3}$$

其中 z 表示空间点 P 的深度值。

2.1.3 深度相机

RGB-D 相机是一种能够同时捕捉彩色图像和深度图像的传感器。其深度探测原理通常基于结构光或飞行时间（Time of Flight，ToF）技术，如图2-5所示。结构光技术通过投射已知模式的光（如条纹、点阵等）到物体表面，并通过相机捕捉其变形，从而计算物体表面到传感器的距离。通过分析光的变形程度，利用三角测量原理可以恢复场景的深度信息。另一种飞行时间技术通过发射短脉冲光并测量光从传感器发射到物体并返回的时间来计算距离。光速已知，因此通过时间差即可精确计算物体与相机之间的距离。这些技术使得 RGB-D 相机能够同时提供 RGB 图像和深度图像，广泛应用于三维重建、机器人导航、虚拟现实等领域。

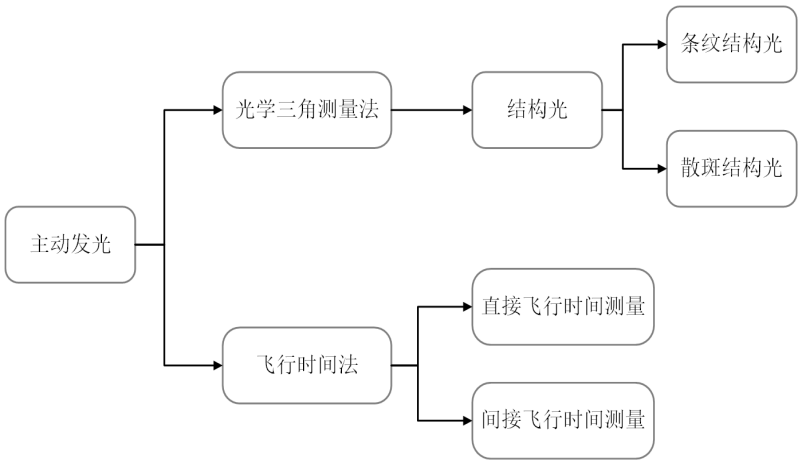


图 2-5 RGB-D 相机模型

相比于双目相机，RGB-D 相机在获取深度信息方面要简便得多，因为 RGB-D 相机已将 RGB 图像与深度图像配准了，可以直接利用图像和距离信息。然而，RGB-D 相机也存在一定的局限性，如红外光束在透明材料上的反射效果较差，以及强烈光照条件下易受干扰，这些因素在一定程度上限制了 RGB-D 相机的应用范围。

2.1.4 相机畸变模型

在现实中，为了获得更广阔的视野，相机通常会安装透镜。然而，透镜的形状和位置会使光线发生折射，从而引起成像的变化。这种由透镜形态引起的变化被

称为径向畸变。径向畸变使得原本应为直线的成像变成了曲线，且曲线的弯曲程度随着成像边缘的距离增大而愈加明显。畸变主要分为两种类型：一种是桶形畸变，另一种则是枕形畸变。如图2-6所示。

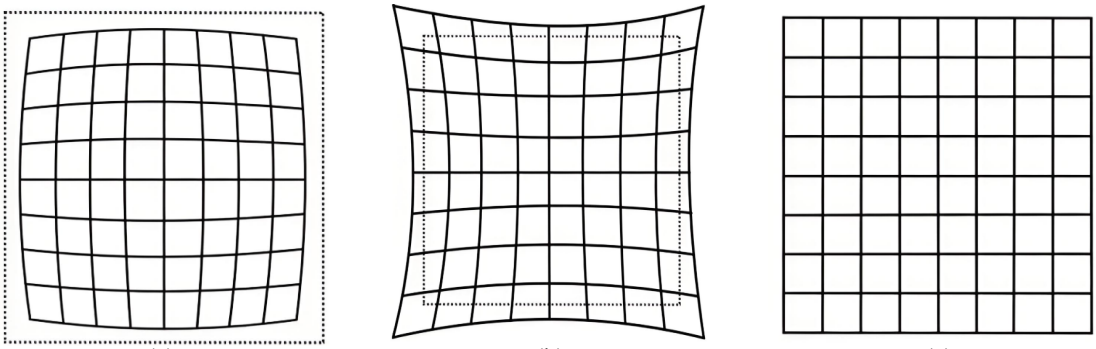


图 2-6 畸变模型。(a)桶形畸变；(b)枕形畸变；(c)正常物体

对于空间中的点 (x,y) 表示实际坐标，$x_{distored}$ 和 $y_{distored}$ 表示畸变后的图像坐标，可以由式子2-4得出。

$$\begin{cases} x_{distored} = x\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_1 xy + p_2\left(r^2 + 2x^2\right) \\ y_{distored} = y\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + p_1\left(r^2 + 2y^2\right) + 2p_2 xy \end{cases} \tag{2-4}$$

其中 r 是坐标点的极坐标表示：

$$r^2 = x^2 + y^2 \tag{2-5}$$

k_1，k_2，k_3 是径向畸变参数，p_1，p_2 是切向畸变参数。

实际应用中，会先对图像进行畸变处理，然后再将结果作为输入作用在 SLAM 系统中。

2.2 视觉 SLAM 框架

VSLAM 是一种基于视觉传感器的自定位与地图构建技术，它通过分析和处理环境中的图像信息，实时地推算出设备的位姿并构建三维环境地图。该技术的核心在于提取图像中的关键特征点，通过对这些特征点的匹配与跟踪，来实现对运动轨迹的估计，同时更新和优化环境地图。VSLAM 系统通常由前端、后端、建图以及回环检测模块组成，其中前端负责特征点的提取、跟踪和初步定位，后端则通过优化算法（如捆绑优化）进行全局地图优化，而回环检测则用于减少累积误差，确保长期运行中的定位精度。随着算法的不断更新优化，VSLAM 在多个领域中被广泛应用，如机器人导航、无人驾驶等。图2-7展示了 VSLAM 系统的整体框架，详细描绘了从图像数据采集、特征提取到地图构建与优化的整个过程。

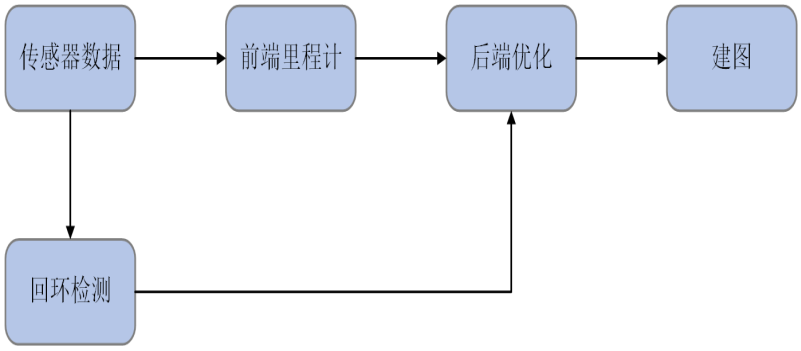


图 2-7 VSLAM 框架

2.2.1 前端视觉里程计

前端视觉里程计是 VSLAM 系统中的一个重要组成部分，负责通过对图像序列的分析，估计设备的即时运动轨迹。它通常通过对连续帧图像中的特征点进行提取与匹配，计算设备在空间中的相对位移。前端视觉里程计的主要任务是实现设备在局部范围内的精确定位，并为后端优化提供初步的轨迹估计。在实际应用中，前端里程计可分为基于特征点的里程计和基于直接法的里程计两种类型。基于特征点的里程计通过提取图像中的关键点并进行匹配来计算位移，适用于结构丰富的环境；而基于直接法的里程计则通过直接处理图像像素信息，能够在纹理较少或环境特征不明显的情况下进行有效定位。前端里程计不仅为系统提供实时的位姿估计，还能辅助判断系统的稳定性和鲁棒性，是 VSLAM 系统实现高精度定位的基础之一。

2.2.1.1 直接法

直接法通过直接利用图像中的像素值信息，借助最小化相邻图像之间的像素灰度误差，来精确估计相机的运动轨迹。这种方法不依赖于传统的特征点检测与提取，而是全面采用图像中的每一像素数据进行运动估计。具体来说，直接法通常利用光度误差或光度一致性约束来进行优化，进而精确推算相机的相对位移。其最大优势在于，能够有效应对低纹理、运动模糊等挑战性环境下的运动估计问题，因此在一些特征稀缺的场景中，直接法表现得尤为突出。然而，直接法在光照变化剧烈或视角变化较大的情况下，可能会受到一定的限制，导致运动估计的准确性下降。

2.2.1.2 特征点法

特征点法是一种通过识别和匹配图像中的独特、稳定的关键点（通常是角点、边缘点等）来进行图像分析和空间定位的方法。这些特征点通常在不同视角下具

有较高的重复性和显著性，能够在图像中准确定位和描述物体的几何信息。特征点法的步骤通常包括关键点的检测、描述符的生成以及在不同图像之间进行匹配。常见的特征点包含 FAST[43]、ORB（Oriented FAST and Rotated BRIEF）[44]、SIFT[45]和 SURF[46]。特征点法能很好的应对光照变化的场景，但是在低纹理场景下可能会受到一定的影响。

ORB 特征点被称为“Oriented FAST”，是在 FAST 角点的提取之上改进的，增加了尺度不变性和旋转不变性。主要是提取像素点周围灰度变化非常明显的点。

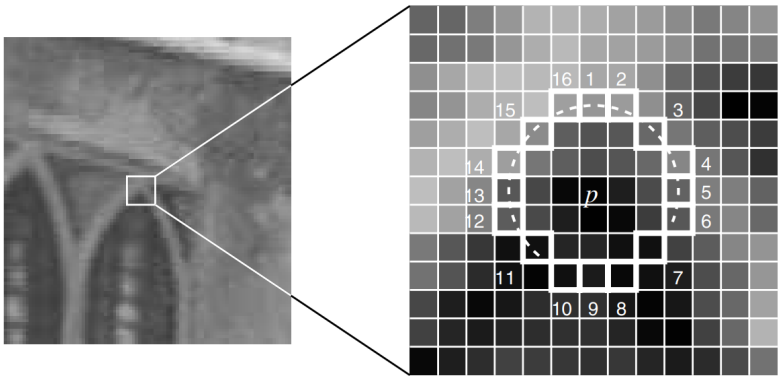


图 2-8 FAST 角点提取

如图2-8所示，在图像中选中了像素点 p，其亮度为 I_p，选取半径为 3，点 p 为圆心的圆周上的 16 个像素点，并设置一个亮度阈值 T（一般是设置 I_p 的百分比值，通常为 20%I_p）。利用这些点与亮度值 $T+I_p$ 和 $T-I_p$ 进行比较，如果连续的 N（通常取 12）个点都大于 $T-I_p$ 并且小于 $T+I_p$ 就认为这个点 p 是一个特征点。

FAST 角点检测器虽然在计算上非常高效，但其存在两个主要局限性：缺乏方向性和尺度性弱。具体而言，FAST 无法为检测到的角点分配一个旋转方向，这使得它在图像旋转时无法保持稳定；同时，FAST 对尺度变化的适应性较差，因此在不同尺度下的角点检测效果较为不稳定。为了克服尺度不变性的问题，ORB 引入了金字塔模型。通过构建图像的多尺度金字塔，ORB 能够在不同的尺度层级上提取特征点，从而使得特征点具有尺度不变性，能够在图像缩放的情况下保持稳定，如图2-9所示。这种金字塔结构使得 ORB 能够适应图像的尺度变化，并进一步增强了特征点提取的稳定性和准确性。

经过缩放之后，能够让提取的角点在不同的尺度上都有更好的表现，以此达到尺度不变性的要求。

针对角点的旋转性质，是利用灰度质心法来解决的。首先在一个小的图像块中计算图像的矩，如式2-6。

$$m_{pq}=\sum x^p y^q I(x,y) \tag{2-6}$$

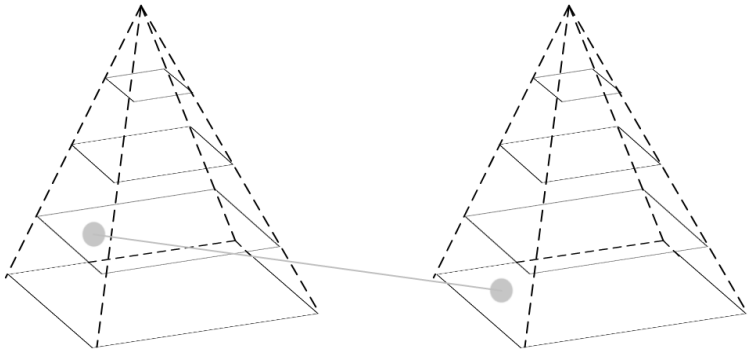


图 2-9 图像金字塔

其中 p、q 取 0 或者 1，表示矩的阶数，(x,y) 表示像素坐标点，m_{pq} 表示图像的矩，$I(x,y)$ 表示坐标 (x,y) 位置的灰度值。

通过这个图像矩可以计算图像的质心，计算公式如2-7所示。

$$C = (c_x, c_y) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{2-7}$$

最后提取的特征点的方向就可以表示为图像形心指向质心的方向向量，旋转角计算方式为：

$$\theta = \arctan 2 \left(c_y, c_x \right) = \arctan 2 \left(m_{01}, m_{10} \right) \tag{2-8}$$

在提取特征点之后，还需要在两帧之间进行特征点匹配，通过精确匹配对应的特征点，能够为相机位姿的变化提供可靠的估计。这些匹配的特征点共同作用，帮助推算出相机在空间中的运动轨迹。

首先对每一个特征点都生成一个 BRIEF 描述子，用来表示角点包含的信息，以用于提升匹配能力。在特征点位置取一个 $S \times S$ 大小的窗口，在其中以一定的方式选取点对进行比较，假设点对两个点的像素值分别是 p 和 q，如果 $p > q$ 则当前二进制位为 1 否则为 0。如果取 128 对点对，则特征点的描述子由 128 位的二进制向量表示。

最直接的匹配方式是对两帧的特征点进行遍历的匹配，假设两帧为 F_i 和 F_{i+1}，两帧图像上的特征点分别为 s_i^n，$n = 0, 1, \ldots\ldots\ldots, N$ 以及 s_{i+1}^n，$n = 0, 1, \ldots\ldots\ldots, M$。这种遍历匹配方法是将两帧所有的点进行组合进而计算两个特征点的 BRIEF 描述子之间的汉明距离（Hamming Distance），计算结果是两个向量之间不同位数的数量。假设两个描述子是 A 和 B，并且描述子长度为 n，每一位由 a_i 和 b_i 表示，计算两个描述子的汉明距离公式为：

$$d(A, B) = \sum_{i=0}^{n} a_i \oplus b_i \tag{2-9}$$

当汉明距离 $d(A,B)$ 小于设置的阈值 T_{brief} 时，则认为两个特征点是匹配的。然而，在匹配过程中可能出现的错误匹配情况，后续通常会进行筛选。为此，通常会设定两个阈值，当汉明距离在两个阈值区间以外时就认为是错误的匹配，进而删除这对匹配特征点。

由于遍历匹配的方式会在特征点很多的情况下导致计算量很大，所以可以在规定的范围内进行特征点的匹配，通常是规定在以特征点为圆心的一个圆形范围内进行匹配。这样能节省很多计算量。

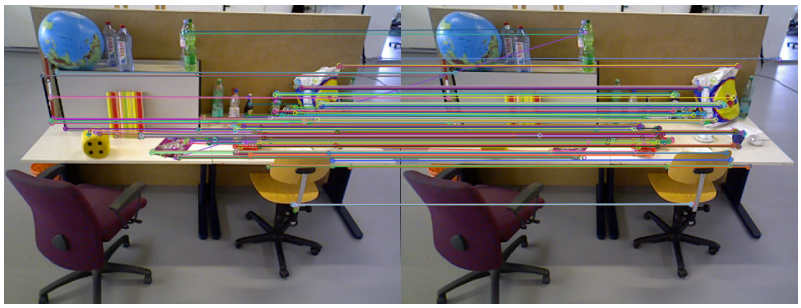


图 2-10 特征点匹配

图2-10展示了两帧之间通过连线匹配得到的特征点对，并且其中一些质量差的特征点对进行了筛选与剔除，只留下了一些质量较高的特征点对。

2.2.2 后端优化

在 SLAM 中，后端优化主要负责利用前端获取的传感器数据（如图像、IMU 数据等）进行全局优化，确保整个轨迹和地图的一致性与精确性。前端负责提取和跟踪特征点、计算局部位姿，但由于传感器噪声、观测误差以及环境变化，前端估计的位姿和地图信息可能会产生漂移和不一致。后端优化的任务就是通过全局优化方法，来调整所有的状态变量（如机器人位姿、特征点位置等），使得系统整体误差最小化，从而提高 SLAM 系统的精度和鲁棒性。

SLAM 系统的观测方程和运动方程可以表示为2-10。

$$\begin{cases} x_k = f\left(x_{k-1}, u_k\right) + w_k \\ z_k = h\left(x_k\right) + v_k \end{cases} \qquad k = 1, \ldots, N \tag{2-10}$$

其中 x_k 表示在 k 时刻的所有未知量：

$$x_k \triangleq \{y_1, \ldots, y_m\} \tag{2-11}$$

需要使用到 $t=0$ 时刻到 $t=k$ 时刻数据来对位姿进行估计，用 x_i 表示在 $t=i$ 时刻的位姿，同时使用 y_i 表示在该时刻观测到的路标信息，通过考虑 0 到 t 时刻

的所有信息数据，可以对当前状态的分布进行估计，具体的公式如2-12所示。

$$P(x_k|x_0, u_{1:k}, z_{1:k}) \tag{2-12}$$

按照贝叶斯公式，这个条件概率可以转换为：

$$P\left(x_k|x_0, u_{1:k}, z_{1:k}\right) \propto P\left(z_k|x_k\right) P\left(x_k|x_0, u_{1:k}, z_{1:k-1}\right) \tag{2-13}$$

式2-13中，前面一项是似然部分，由观测方程给出；而后一项为先验部分，是根据过去时刻得到。因此该项会受到过去信息的影响，于是将先验部分以 $t = k-1$ 时刻展开：

$$P\left(x_k|x_0, u_{1:k}, z_{1:k-1}\right) = \int P\left(x_k|x_{k-1}, x_0, u_{1:k}, z_{1:k-1}\right) P\left(x_{k-1}|x_0, u_{1:k}, z_{1:k-1}\right) \mathrm{d}x_{k-1} \tag{2-14}$$

如果仅考虑当前状态与上一时刻 $t = k-1$ 状态有关，而忽略与更早时刻状态的关系，则可以得到以扩展卡尔曼滤波为代表的滤波方法。而若考虑当前时刻之前所有状态的影响，则会构建以非线性优化为核心的优化框。

非线性优化中基于图优化的算法是通过构建一个包含机器人位姿和环境特征点的优化图，来全局调整这些位姿和特征点的位置，以最小化系统中的误差。在图优化中，图的三角形节点表示机器人的位姿，圆形节点表示地图中的特征点，而边则表示节点之间的约束关系，如图2-11所示。

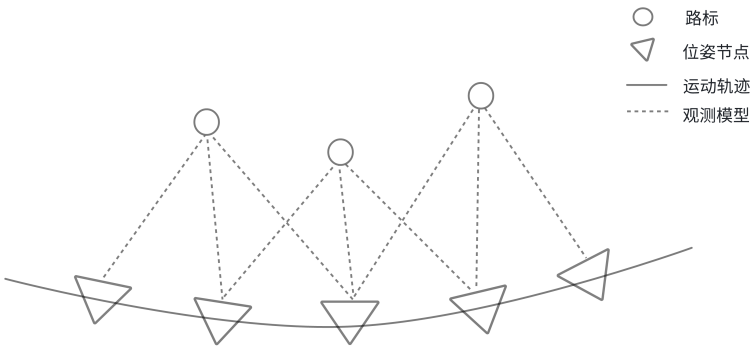


图 2-11 图表示

通过最小化图中所有边的误差，图优化算法就能找到一个最好的位姿估计值以及图估计。构建误差项 e_{ij} 为：

$$e_{ij} = \ln\left(T_{ij}^{-1} T_i^{-1} T_j\right)^{\vee} \tag{2-15}$$

其中 T_{ij} 表示预测的两相机节点之间的相对位姿，T_i 和 T_j 表示两个相机位姿。

按照李代数求导法则，求出误差关于两个位姿的雅可比矩阵：

$$\hat{e}_{ij} = e_{ij} + \frac{\partial e_{ij}}{\partial \delta \xi_i} \delta \xi_i + \frac{\partial e_{ij}}{\partial \delta \xi_j} \delta \xi_j \tag{2-16}$$

由此该问题可以转化为一个非线性最小二乘问题，并可以采用高斯-牛顿法或列文伯格-马尔夸特法进行求解。

2.3 ORB-SLAM2 算法研究

ORB-SLAM2 是一种基于特征点的 VSLAM 系统，采用了高效的 ORB 特征进行特征匹配和追踪。该系统实现了单目、双目和 RGB-D 相机的支持，具有实时性和高精度的特点。ORB-SLAM2 通过全局和局部地图优化相结合的方式，利用关键帧进行地图构建，并通过回环检测来减少累积误差。系统的核心技术包括稀疏特征点跟踪、全局优化以及回环检测等。ORB-SLAM2 能够在动态环境下稳定运行，并能够有效应对大规模场景的建图和定位任务，广泛应用于机器人、无人驾驶和增强现实等领域。

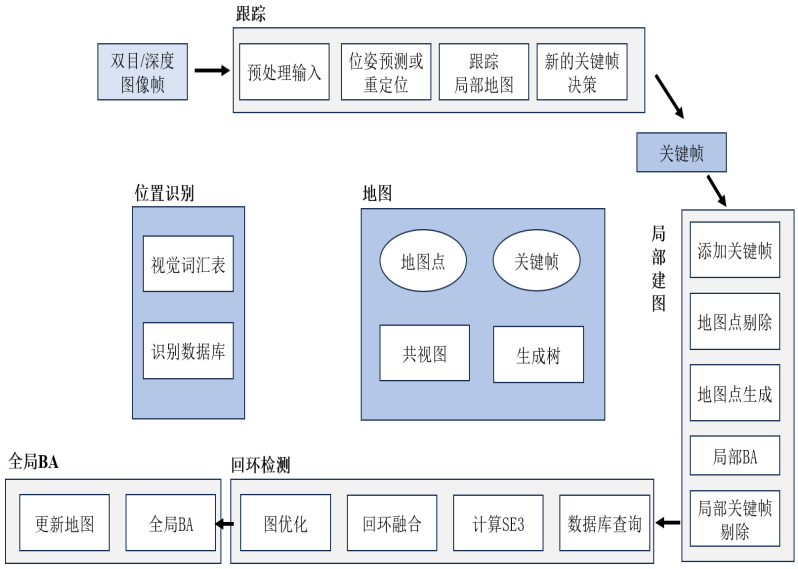


图 2-12 ORB-SLAM2 框架

2.4 八叉树地图

OctoMap[47] 是一种基于八叉树（Octree）[48] 数据结构的 3D 环境表示方法，广泛应用于机器人领域，尤其是在实时环境建图与路径规划中。其主要特点是通过递归地将空间划分为八个子立方体（即每个节点最多有八个子节点），从而高效地表示和存储稀疏的三维环境信息。OctoMap 的核心优势在于其灵活性和高效性，能够根据需要动态地细化或简化地图的分辨率，特别适合用于处理不规则且稀疏的

环境数据。

OctoMap 采用概率模型来表示每个体素（Voxel）被占据的可能性。每个体素都有一个概率值，表示该位置被障碍物占据的信置信度。这使得 OctoMap 可以处理不确定性，并且能够在传感器噪声和数据缺失的情况下保持稳健。通常，这些概率值通过传感器的测量数据（如激光雷达、深度摄像头等）更新。八叉树示意图如图2-13所示。

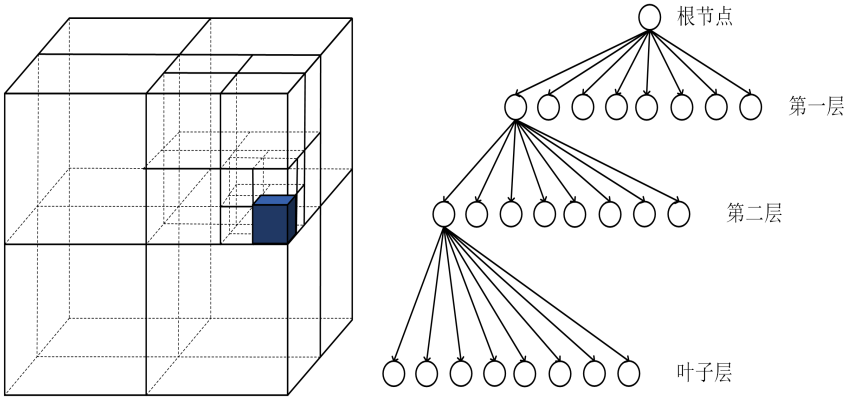


图 2-13 八叉树示意图

2.5 本章小结

本章主要介绍了语义 SLAM 所需的相关理论和技术，包括相机模型、VSLAM 框架及 ORB-SLAM2 算法的研究。

首先，介绍了 VSLAM 前端可能用到的相机工具。单目相机使用针孔相机模型进行成像，这是将复杂的三维世界映射到二维图像平面的基础。此模型介绍了四个相关的坐标系：世界坐标系、相机坐标系、图像坐标系和像素坐标系，并详细说明了它们之间的转换关系。接着，介绍了双目相机如何通过左右图像计算空间的深度信息。此外，还阐述了 RGB-D 相机的成像原理和其测距方法，以及相机畸变模型的影响和校正方法。

然后，重点介绍了 VSLAM 框架，详细探讨了前端里程计和后端优化部分。在前端里程计中，特征点提取是一个关键的步骤，特征点的质量直接影响到后续的特征点匹配和位姿估计的准确性。后端优化部分则主要涉及当前位姿的估计方法和图优化理论，强调了图优化在提高定位精度、消除误差方面的重要性。

最后，介绍了 ORB-SLAM2 算法框架以及使用到的八叉树地图表示方式。ORB-SLAM2 通过结合前端的特征提取、位姿估计和后端的图优化技术，实现了高效的 SLAM 系统。

第三章 基于目标分割的物体提取关联与位姿优化

3.1 引言

传统的 SLAM 系统在机器人和自动驾驶等领域中具有重要作用，其通过几何特征（点、线、面）的提取与匹配来实现环境的建图与位姿估计。然而，这些系统也存在明显的不足之处，限制了其在复杂场景下的性能和鲁棒性。传统 SLAM 系统主要依赖于几何特征的表示，忽略了环境中丰富的语义信息。例如，SLAM 系统可以构建出房间或走廊的几何结构，但无法理解这些结构中的物体类型或功能（如椅子、桌子、门等）。缺乏语义层级的环境理解会导致系统难以利用更高级别的环境信息，从而影响其在场景感知、地图表达和后续任务（如路径规划）的适用性。这种局限性在复杂场景中尤其明显，无法满足对环境中语义信息依赖较高的应用需求。此外，传统 SLAM 在回环检测模块上也存在一定的不足。回环检测是 SLAM 系统中至关重要的模块，其通过识别机器人是否回到了之前访问过的区域来消除累积误差并提升全局地图的一致性。然而，传统 SLAM 方法通常依赖于几何或视觉特征的匹配，这在光照条件变化较大、纹理不明显或场景相似度较高（如走廊或办公室）时，容易导致特征点提取和匹配困难问题，进而影响系统的全局优化能力。

针对上述问题，本章提出了一个基于 RGB-D 相机的物体级语义 SLAM 算法。该算法在传统 SLAM 流程的基础上引入了物体的概念，并在系统中利用物体信息。系统主要包括物体提取与关联模块、物体优化位姿模块、语义地图模块。该算法利用 RGB-D 相机获取的 RGB 图像和深度图像作为输入，结合前端的实例分割网络 YOLOv8 对物体进行识别。通过 YOLOv8 的实例分割功能，算法不仅能够检测并分类物体，还能够生成物体的掩码信息，以便进一步分析物体的空间分布和几何形状。随后，优化掩码信息，并使用从掩码和深度图中提取的物体信息加入到捆绑优化中来参与优化 SLAM 系统的位姿估计。最后由存储的物体关联关系来辅助判断 SLAM 系统是否出现回环的情况。

本章的结构如下：首先，概述了整个系统框架；其次，详细的讲解了如何在前端的图片中提取到物体并进行物体关联；接着，讨论了物体信息参与 SLAM 系统后端优化中的作用；最后，通过在开源的数据集上对提出的 SLAM 系统进行测试，并与其他的先进开源 SLAM 系统进行对比，验证提出的 SLAM 算法的有效性和可行性。

3.2 总体框架介绍

本章提出的 SLAM 算法总体框架如图3-1所示。整个算法框架主要由三个核心模块组成，物体提取与关联模块、物体优化位姿模块、语义地图模块。各模块协同工作，利用物体级语义信息提升 SLAM 系统的整体性能。

（1）首先，物体提取与关联模块接收实例分割输出的物体信息，包括物体的检测框、语义信息和掩码信息。通过融合深度图信息，SLAM 系统能够进一步优化物体分割的精度，从而获取更可靠的物体信息。此外，算法中会构建物体数据库用以存储已构建的物体。并通过物体类别、质心、颜色等信息，对观测到的新物体与构建物体进行关联匹配，从而构建精确的物体。

（2）接下来，物体优化位姿模块将物体级信息有效整合到 SLAM 系统中，用于优化位姿估计和回环检测。在优化位姿估计过程中，将已经稳定的物体信息纳入到捆绑优化中，构建新的约束项，从而进一步提升位姿估计准确性。在回环检测流程中，利用物体间的邻近关系构建物体地图，并通过计算地图相似性来判断是否存在回环。

（3）最后，语义地图模块将前两个模块生成的信息整合到全局地图中。具体而言，该模块将检测到的 3D 空间中的物体构建到语义地图中，并附加物体的语义标签和位置信息。语义地图不仅包含传统的几何信息，还通过融入物体级别的语义信息，生成更丰富、更直观的环境表示。

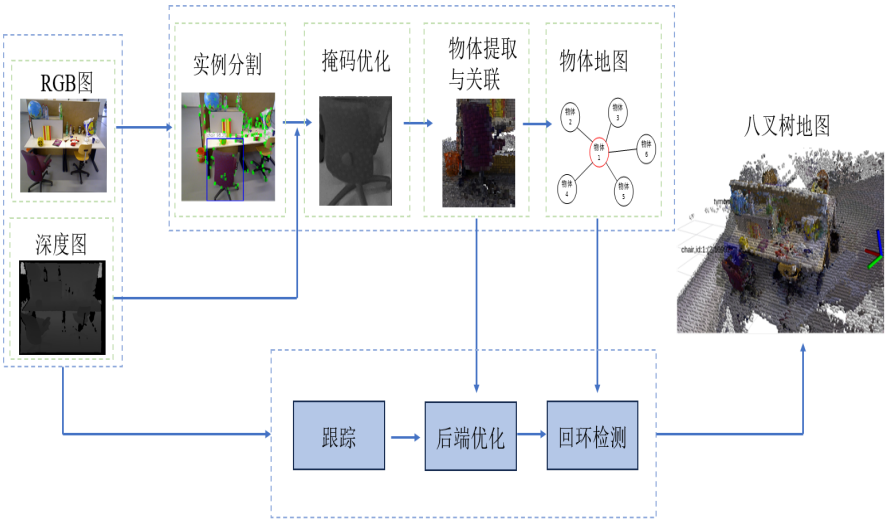


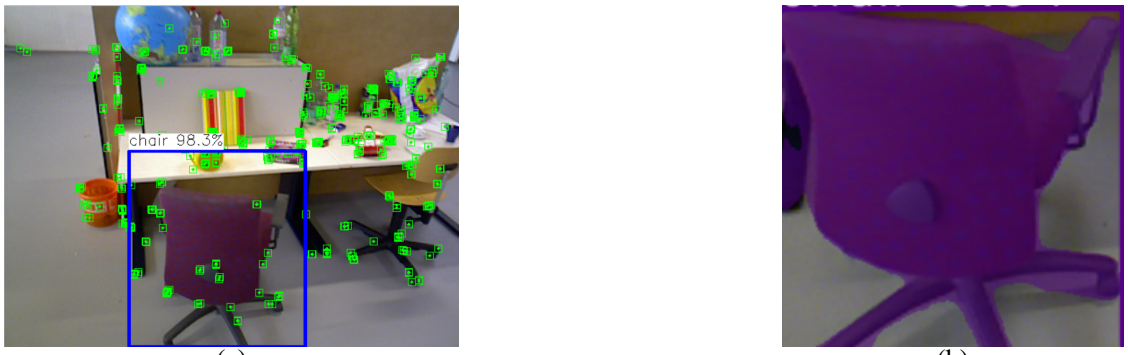
图 3-1 SLAM 系统框架

3.3 物体提取与关联

在语义 SLAM 中，解决静态物体的数据关联问题旨在对不同图像中的静态物体进行判断，确保同一物体的不同观测不会被孤立地重复构建。这一问题的核心在于通过算法分析和识别，对同一静态物体的观测点进行准确的关联融合，从而在现有地图的基础上对物体进行修改。此过程需要综合考虑几何和语义信息，以保障对静态物体的观测关联的准确性和鲁棒性，以提高系统对环境的感知和地图构建的精度。

3.3.1 物体提取

物体提取是从输入的图像中，通过 YOLOv8 实例分割网络对预训练的物体进行识别，获取物体的类别以及物体掩码。物体的识别结果以及掩码如图3-2所示，最后生成物体的三维信息。



(a) (b)

图 3-2 YOLOv8 处理结果。(a)物体类别和识别框；(b)物体掩码

在 YOLOv8 实例分割任务中，可能会出现过度分割（over-segmentation）现象，特别是在处理结构复杂或相互接触的物体时，示例如图3-3。对于被部分遮挡的人体，模型可能错误地将其上半身和下半身分别识别为两个独立的实例，而实际上它们属于同一物体。尽管 YOLOv8 在目标检测和实例分割任务中展现了较高的性能，但该问题仍然对物体的提取造成困难，并且影响后续对物体信息的使用，最终导致 SLAM 系统精度降低。

针对上述提及的情况，本文使用 IoU 结合深度的方式对检测到的物体进行校验。整体流程如图3-4所示。

首先，在同一帧中匹配出属于相同类别的物体，然后在这些匹配上的同类物体之间分别计算两个物体的 IoU 值。如果 IoU 值超过预设的阈值，则进行深度值匹配。由于认为同一个物体上的两个包围框一定会相距很近，本文设定 IoU 阈值为 0.8。接着，计算同类物体的平均深度，如果两个物体的平均深度差小于设定的

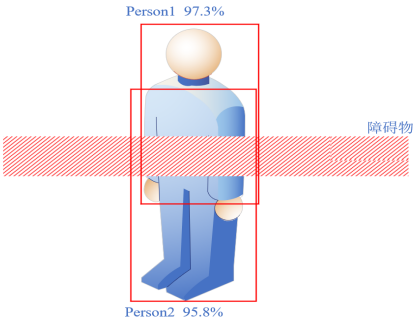


图 3-3 误检测示意图

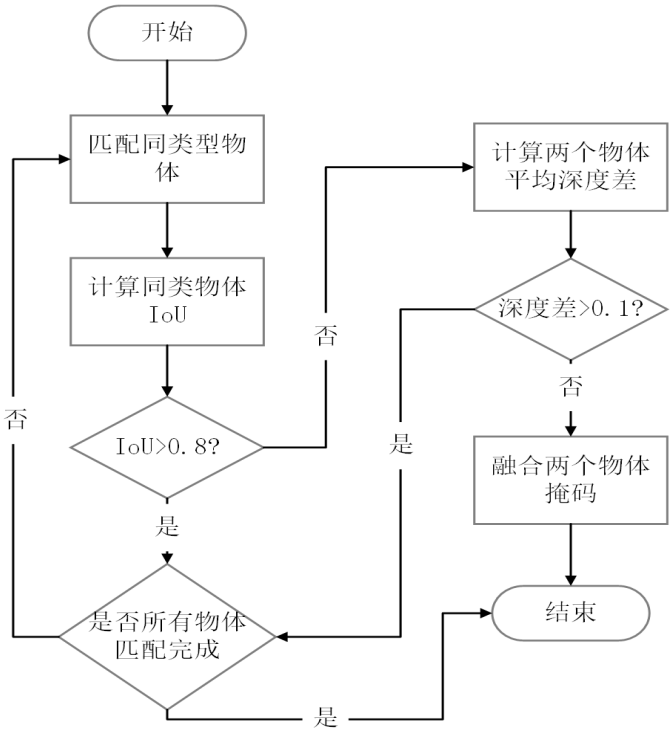


图 3-4 过度分割融合策略

阈值，则认为这两个相邻物体实为同一物体，只是在 YOLOv8 的识别过程中将同一物体误判为两个不同物体。因为同一物体的深度值变化通常较小。本文将深度阈值设置为 0.1。

IoU 是计算机视觉中常用的评估标准，衡量预测区域与真实标注区域的重叠程度，其计算公式如公式3-1所示。

$$IoU = \frac{A \cap B}{A \cup B} \tag{3-1}$$

在本文中 IoU 用来计算两个包围框的交集和并集之间的比值，以此判断两个物体之间的重叠程度。示意图如图3-5所示。

在 YOLOv8 的识别结果中，过度分割还可能会出现与上述不同的错误情况，即

(a) (b)

图 3-5 IoU 计算。(a)包围框交集；(b)包围框并集

一个物体被分割成两个不相交的部分。例如，人的头部和腿部被单独识别，而身体部分则被漏检，如图3-6所示导致误识别的情况。这种误识别会使得系统构建出两个独立的三维物体。这种情况在后续的关联处理中，可以通过计算当前观测点云是否存在重合的两块点云来解决。当下一帧成功且准确识别出该物体时，系统会检测当前的点云数据中是否有多个重叠的部分。如果检测到存在重叠区域，系统将自动将这多个误分割的物体关联为一个，从而形成一个完整的物体。

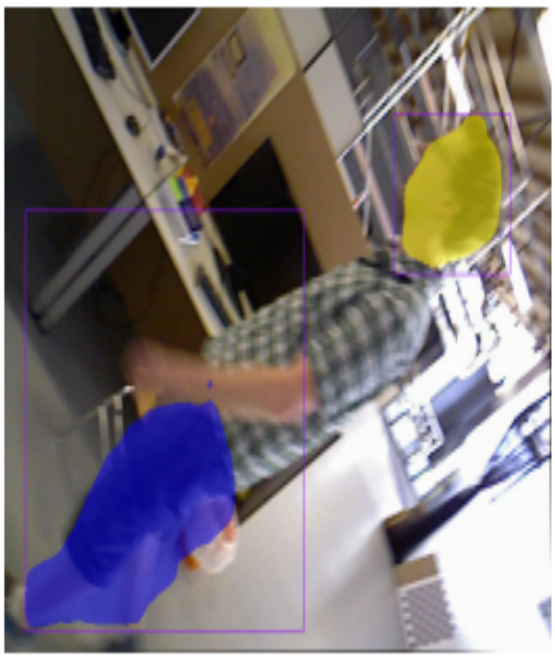


图 3-6 分割出两个部分

处理完成 YOLO 检测可能出现的问题后，还需要对物体深度图进行进一步处理。在深度数据采集过程中，RGB-D 相机可能会出现坏点（Bad Points）问题。坏点通常表现为深度图中的某些像素值异常或不准确，这些不准确的深度值往往源自相机传感器的局限性、光照变化、反射表面、视距过远或过近等因素。坏点会引

发深度数据的失真，如图3-7所示，进而导致物体深度值及点云计算的误差，严重影响物体信息的准确性。

(a) (b)

图 3-7 坏点存在。(a)坏点示意图；(b)对应 RGB 图

从图3-7中可以看出，3-7(a)为由深度图转换而来的点云示意图，其中图像左边部分是物体的点云，右侧则展示了由于深度图存在误差或量程限制，导致的一部分点云异常。3-7(b)则是对应的 RGB 图像。通过对比，明显可以发现部分点云异常突出，脱离了正常的范围，且与真实值存在较大差异。这些异常点云正是由于坏点的错误深度数据所引起的。如果以此为基础计算物体的质心、大小等信息，将会产生较大的误差，从而导致 SLAM 系统的精度显著下降。

由于使用的深度图可能存在坏点的情况，需要先处理深度图中的数据，剔除坏点，再使用剩下的数据。由于深度相机都有一个测量范围，坏点通常都被标记成了量程以外的值。本文利用量程范围设置了深度值的最大、最小阈值，只有深度在阈值之间才认为是正确的数据，其余坏点数据被设置成了 0。

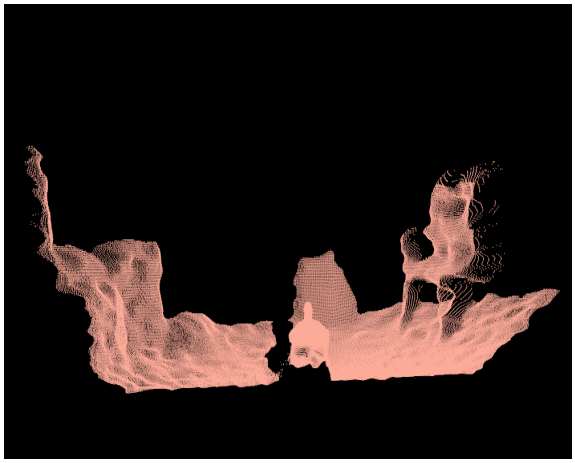


图 3-8 处理后物体的点云展示

图3-8展示了应用坏点剔除和滤波处理后的物体部分点云。可以看到，经过处理后，点云中的异常点已被有效去除，剩余点云更加平滑且符合实际场景的结构。

相比于未经处理的点云，处理后的点云在后续的 SLAM 任务中会显著提升定位精度和地图构建的稳定性。

在处理完成深度信息以及过度分割问题后，对获得的物体掩码进一步处理。物体掩码中往往包含了一些物体之外的背景信息。为了提升掩码的准确性，并使其更加精准地贴合物体的轮廓，本文结合了深度图对掩码进行优化。这个优化方法不仅能够有效去除背景干扰，还能在物体的边缘部分进一步精细化掩码，使其与物体的实际形状更加契合。通过这种深度图与掩码的结合处理，掩码的精度得到了显著提升，尤其是在物体的边缘区域，优化效果尤为突出。物体掩码优化的具体算法示意如图3-9所示。

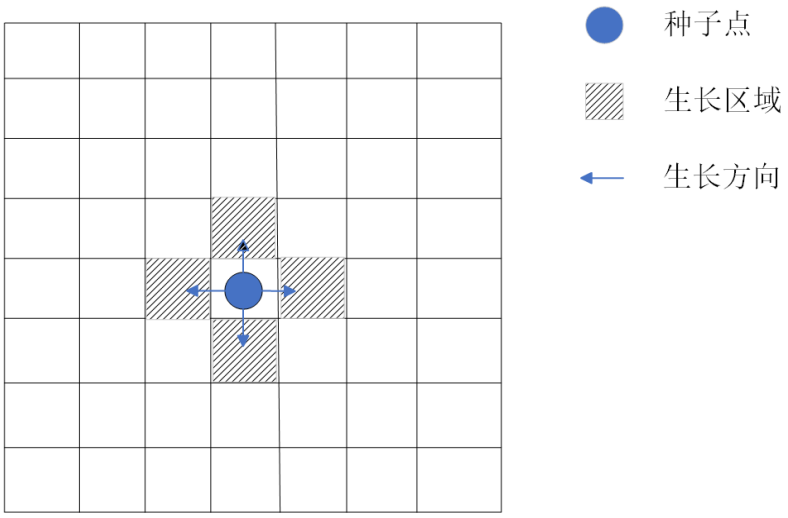


图 3-9 区域生长示意图

首先计算物体深度的平均值，并基于该值选取一个合适的像素点作为区域生长的起始点。结合深度图和掩码区域，算法进一步计算出物体区域的深度值标准差和方差。根据这两个数据，算法能够自适应地确定物体深度的最大值和最小值，从而设定精确的深度阈值。利用这一深度阈值，算法在进行区域生长时能够有效剔除与物体无关的背景像素点，确保掩码区域的精确性。

如图3-9所示，区域生长算法流程为：首先将选定的区域生长起始点加入候选点集。对于每个候选点，计算其上下左右四个相邻像素是否满足深度阈值要求。若满足，则将该候选点视为物体上的点，并将这四个相邻点加入候选点集；否则，剔除该点，仅将满足深度阈值的相邻点加入候选点集。

随后的处理步骤集中于优化物体掩码的边缘部分，使得物体的轮廓更加精准、自然。整个过程的具体实现和操作流程如图3-10所示。

在物体的边缘处，RGB-D 相机所捕捉到的深度信息往往会出现突变现象，而物体表面的深度值应当是连续且变化平缓的。因此，通过计算深度图上物体区域

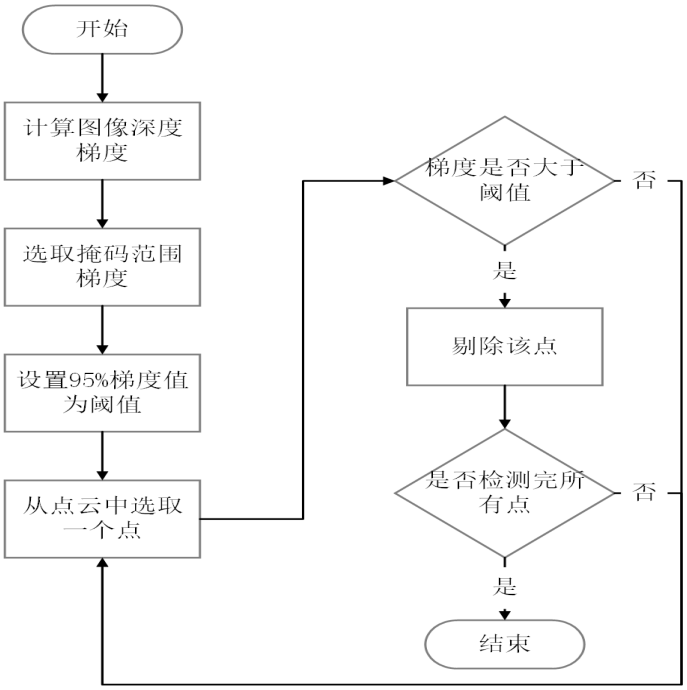


图 3-10 掩码边缘优化流程

的梯度信息，算法能够精确地识别出深度变化剧烈的区域，从而判断出物体的边缘位置。通过这种方式，可以设定一个自适应的阈值。本文对物体深度梯度进行从小到大排序，取 95% 位置的值作为阈值，确保物体边缘的检测更加准确，避免由于深度信息的突变导致错误的边界判定。

经过精细的掩码处理后，得到的掩码被视为更加准确的物体掩码。通过掩码区域并借助深度图像，进一步提取出物体的三维点云信息，为后续的分析和处理提供了重要的空间数据支持。通过像素平面与三维空间之间的对应关系，可以由公式3-2、公式3-3和公式3-4计算出当前观测物体的准确位置。

$$X = (u - c_x) \cdot \frac{d(u,v)}{f_x} \tag{3-2}$$

$$Y = (v - c_y) \cdot \frac{d(u,v)}{f_y} \tag{3-3}$$

$$Z = d(u,v) \tag{3-4}$$

其中 u 和 v 是相机平面上的像素点、f_x 和 f_y 表示相机在 x 和 y 轴方向上的相机焦距、c_x 和 c_y 表示相机的中心点坐标。

3.3.2 物体关联

物体关联是指将当前观测到的物体与已构建物体进行精确匹配与联系的过程，这是物体 SLAM 系统中的关键步骤。通过这一过程，来自不同观测的物体信息得以有效融合，显著提高了 SLAM 系统的整体精度。当前的物体信息来源于物体提取步骤中生成的三维物体，通过匹配相同类别的物体，并计算物体质心间的距离。当该距离小于预设的阈值时，进一步使用颜色信息进行对比，如果计算结果同样满足阈值，则可判定为同一物体。具体流程详见图3-11所示。

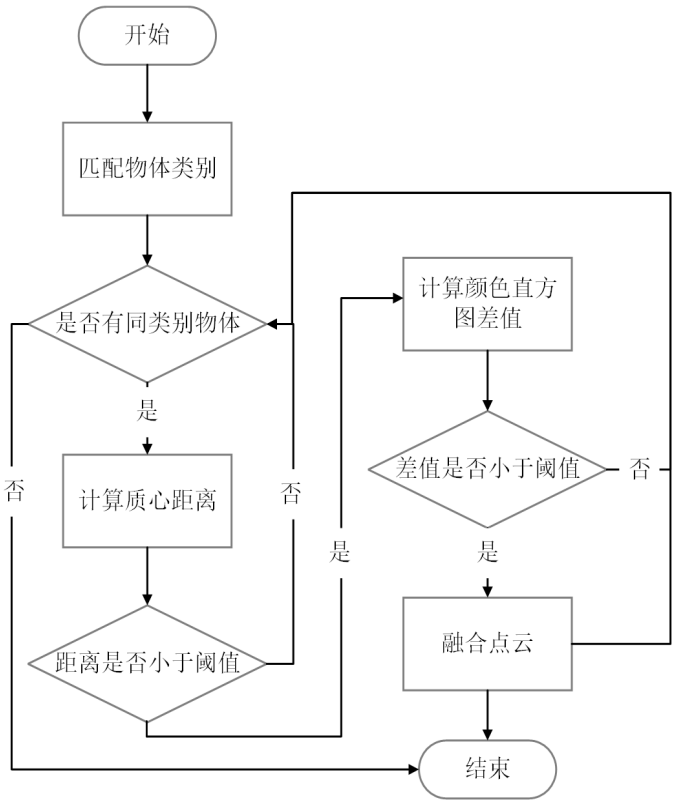


图 3-11 物体关联算法

由于每一个物体大小不一，在匹配质心距离时距离误差也应该不同，因此根据物体的不同大小，为每个物体设定了相应的距离阈值。在阈值范围内，表示两个观测物体十分接近，但由于观测的局限性、位置估计误差等因素，导致两者的距离存在一定偏差。因此，系统将进一步通过计算两个物体颜色直方图之间的巴氏距离来进行比较。如果计算结果小于预设的颜色阈值，则认为这两个观测物体为同一物体。

为了能有效的对物体进行管理和关联，SLAM 系统构建了一个全局物体数据库，用于动态维护和更新三维物体。每个物体都会被加入到该全局数据库中。此外，系统还维护着一个局部物体数据库，该数据库记录了距离最近 10 个关键帧内

的物体信息。局部物体数据库主要用于物体的关联匹配。由于系统中的运动轨迹呈缓慢旋转和平移，通常同一物体的观测会出现在相邻的几个关键帧中，因此，局部物体数据库能够有效地进行物体关联操作，并减少对物体的匹配需求，从而降低计算量。特别是在数据集场景较大时，物体数量可能众多，全局匹配会消耗大量计算资源，进而影响系统的实时性。数据库示意图如图3-12所示。

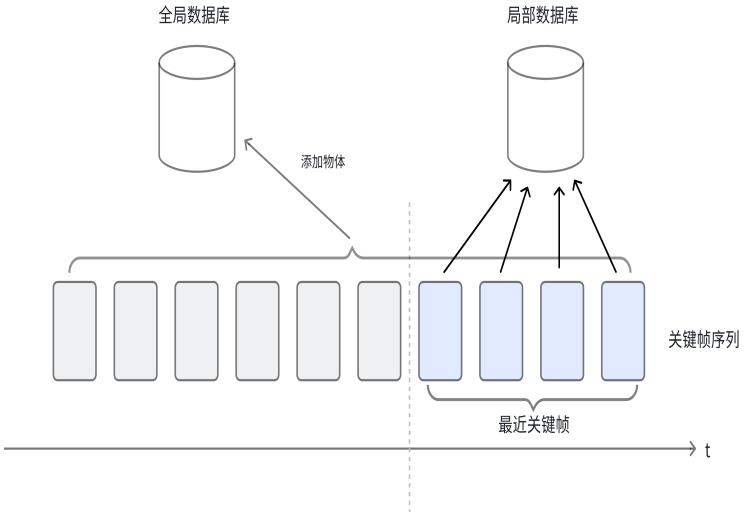


图 3-12 两个数据库内容

在构建的物体数据库中，将维护物体的多个信息，物体位置（由物体的质心和大小决定），物体质心计算公式如公式3-5所示。

$$O_c = \frac{1}{n}\sum_{i=0}^{n} x_i \tag{3-5}$$

其中 x_i 表示物体点云的三维点坐标，n 表示物体的点云数量。

此外还会记录物体的点云数量、物体类别、以及颜色信息。通过将深度图与物体掩码信息结合，将二维平面上的像素点投影到三维空间，从而得到点云表示，并最终计算出点云的数量。颜色信息通过处理后的掩码与 RGB 图像相结合获得。

由于在运动过程中多次对物体进行观测，并且不同的观测角度会导致光照效果变化，而 RGB 信息对光照的敏感度较高，因此在计算颜色直方图时，采用了图像的 H 和 S 通道，分别表示物体的色调和饱和度。这两个通道的信息对光照变化不敏感，相较于 RGB 信息，能够更加准确地反映物体的颜色特征。该算法首先将 RGB 图像转换为 HSV 色彩空间，并仅提取 S 和 H 通道的矩阵表示。随后，利用物体掩码对图像进行区域选择。通过将掩码应用于处理后的图像矩阵，提取出仅属于物体区域的像素值。接着，根据预定的区间对这些像素值进行分组，并计算

每个区间内像素值的频数，得到表示 H 和 S 的直方图。最后，将两个直方图拼接在一起，得出最终的颜色表示结果。

在运动过程中，观测到的物体会实时存储并更新至物体数据库中。随着运动的进行，同一物体可能会从不同角度被观测到，通过对比数据库中已有的物体与当前观测到的物体，一旦确认为同一物体，便进行关联操作，利用融合后的物体更新数据库中的信息；而对于不同的物体，则在数据库中创建新的物体。更新策略如图3-13所示。

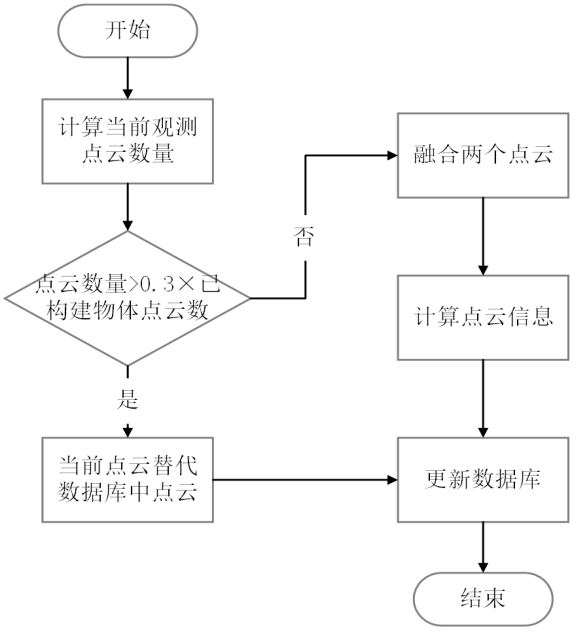


图 3-13 物体更新流程

在物体更新过程中，首先会更新物体的点云信息，将两个观测到的物体的点云融合为一个新的点云团。在这个点云团的基础上，再计算物体的质心、大小以及点云数量信息。这种先融合点云方式能够提高计算结果的准确性。最后，通过加权平均滤波方法融合两个颜色直方图，计算公式如3-6所示。

$$H_{fusion} = (1-\alpha)H_{cur} + \alpha H_{obj} \tag{3-6}$$

其中 α 是设定的融合系数，本文认为在数据库中的数据是经过多次融合的比单次观测的结果更稳定，所以将 α 设定为 0.7，H_{obj} 表示数据库中构建的物体颜色直方图，H_{cur} 表示当前观测到的物体的颜色直方图。

物体的关联关系也能对一些错误的检测进行修正。例如，在使用 YOLOv8 检测网络时，可能会出现误检的情况，即本应没有物体存在的 RGB 图像中，YOLOv8 却错误地检测出物体，如图3-14所示。此时，系统将构建出一个虚假的三维物体。

如果后续依赖于该物体进行处理，便会将错误信息引入系统，进而影响精度。

图 3-14 YOLOv8 误检测

对于这种误检测的物体，需要及时删除其物体信息，以减少错误物体对系统的影响。由于整个数据集中的相机运动为缓慢的旋转和平移，可以假设同一物体在连续多帧中会被多次观测到。因此，如果新构建的物体在接下来的三帧关键帧中都未能再次被观测到，则可判定该物体为错误识别物体，并从数据库中删除该物体。

3.4 物体优化位姿估计

在传统的 SLAM 系统中，地图点通常由环境中的静态特征点构成，通过捆绑调整优化相机位姿和地图点的位置。然而，随着物体在环境中的出现，可以利用物体的 3D 信息来进一步优化相机位姿和地图的精度。物体信息作为额外的约束项被引入捆绑优化，可以提升对物体和场景的建模精度，使得优化结果更为准确。

由于每次观测物体时，无法全面地捕捉到物体的完整形态，因此在物体数据库中构建的物体可能仅为实际物体的一部分。在这样的情况下，当下一次观测到物体时，两个物体的质心可能并不完全重合，存在一定的偏差，如图3-15所示。

因此，若直接将所有匹配上的物体信息纳入捆绑优化中，将会引入大量偏差，导致优化结果产生显著误差。所以，需要先对物体进行筛选，确保得到更加稳定的物体后再加以使用。本章算法在记录物体信息的同时，将记录一个附加信息，用于判断物体是否足够稳定。若物体在多次观测并更新后，其大小变化不显著，即更新前后物体大小差值小于阈值 δ_{size}，则认为该物体基本稳定，可以用于捆绑优

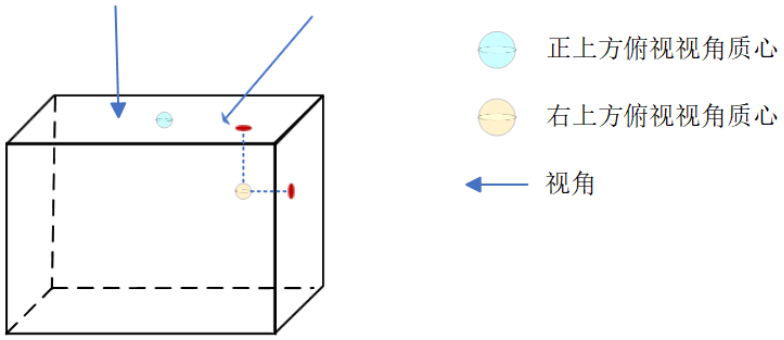


图 3-15 两次观测质心差异

化过程。

对于每个物体 i，假设其在世界坐标系中的 2D 位置为 B_{obj}，相机的位姿为 T_{cam}，通过相机投影模型将图像平面上的物体投影到 3D 坐标中，P_{obj} 为观测物体在三维空间中的坐标位置。物体的投影关系通过针孔相机模型来描述：

$$P_{obj_i} = \pi(B_{obj_i}, T_{cam_j}, K) \tag{3-7}$$

其中，$\pi(\bullet)$ 表示投影函数，K 是相机的内参矩阵。

在捆绑优化中，通常优化的是相机位姿和地图点的位置。由于系统中已构建了观测到的物体，可以利用物体的 3D 质心信息作为额外优化信息，将这些物体作为约束项加入到捆绑优化中，从而优化相机位姿和物体位置。优化目标函数包括了相机位姿和物体位置的约束。对于每个物体 i，建立一个误差项，该误差项最小化当前观测物体质心 P_{obj_i} 与物体数据库中关联上的物体的质心 Z_{obj_i} 之间的误差。优化目标可以表示为式3-8：

$$\arg\min\left((1-\lambda)\sum_{i}\|z_i-\pi(T_iX_i)\|^2+\lambda\sum_{i=1}^{n}||Z_{obj_i}-P_{obj_i}||^2\right) \tag{3-8}$$

其中，π 表示三维空间到二维平面的映射关系，T 表示相机的旋转和平移，X_i 表示二维平面像素点对应的三维点，λ 表示设置的权重值。由于物体始终会存在一定的估计误差以及观测导致的不准确性，物体构建的误差项权重值应该设置较低，这样不可以减少这些误差的影响，本文将 λ 设置为 0.3。

3.5 实验结果与分析

本小节对提出的 SLAM 算法进行了测试与评估，所有的测试是在开源数据集 TUM 上进行。TUM 由 RGBD 传感器收集，提供不同纹理、光照和结构条件下的室内图像序列，被广泛应用于 VSLAM 和里程计系统的评估。

3.5.1 实验环境与设置

本章实验使用 C++ 语言进行程序编写，同时所有实验都在 Ubuntu 系统上进行。使用了 python3.8、ROS 以及 Eigen 等作为开发环境。并使用 coco 数据集对 YOLOv8 实例分割网络进行训练。实验环境详细配置如表3-1所示。

表 3-1 实验环境配置

环境配置	型号和参数
CPU	IntelR Xeon(R) E5-2650 v4
GPU	GeForce GTX 1080 8GB
内存	64G
操作系统	Ubuntu 18.04.5 LTS
python 版本	3.8
Cuda 版本	11.0

3.5.2 TUM 数据集介绍

TUM 数据集[49] 是由德国慕尼黑工业大学（TUM）提供的，旨在支持计算机视觉和机器人技术的研究。该数据集涵盖了多种场景，包括室内外环境、动态变化以及复杂的物体识别任务，提供了高质量的标注数据和丰富的多模态信息。TUM 数据集广泛应用于物体检测、SLAM、动作识别等领域，成为该领域算法评估的标准测试集。其中大多数序列是从具有不受约束的 6-DOF 运动的手持式 Kinect 记录的，同时还有部分序列采集是安装在先锋 3 机器人上的 Kinect 采集到的，Kinect 相机如图3-16所示。

图 3-16 Kinect 相机

TUM 数据集中的 $freiburg1/room$ 序列展示了一个典型的室内环境，沿着整个办公室的轨迹进行拍摄。序列从四张桌子开始，桌面上摆放着键盘、显示器等

物品，周围还有椅子、泰迪熊以及静止的人等。这些丰富的物体提供了多样化的视觉特征，能够有效地帮助 SLAM 系统在定位和地图构建中进行更精确的识别和匹配。随后，相机沿着房间的外墙继续移动，直到回环发生。由于该序列的结构特征和回环的存在，它非常适合用来评估 SLAM 系统在处理回环问题时的表现。

序列 $freiburg3/teddy$ 是 TUM 数据集中一个物体重建的经典场景，其中传感器围绕着不泰迪熊在不同高度进行运动。泰迪熊具有柔软的皮毛以及穿着黄色光滑的衬衫，使得它在视觉上具有较强的可辨识性。由于传感器在同一物体周围从不同角度移动并采集数据，这种场景非常适合用于物体重建任务。

3.5.3 实验结果

TUM RGB-D 数据集提供的室内静态场景下的数据序列适用于本文提出的算法。在 TUM 数据集中选取了 $fr3/teddy$ 来对物体提取进行测试，其场景如图3-17所示。

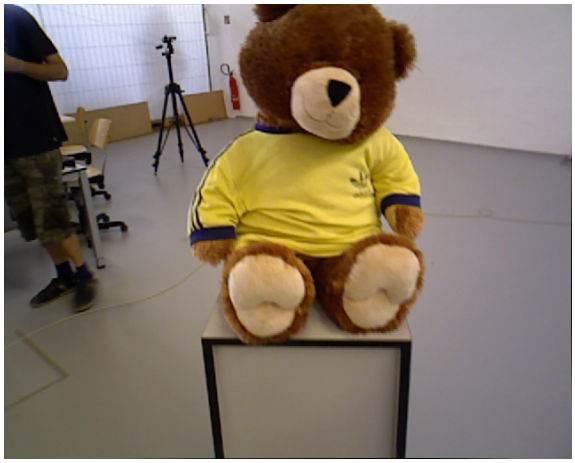


图 3-17 fr3/long_office_household 序列场景

在这一序列中，所观测到的物体都以泰迪熊为主，并且在一定时间内，物体会出现在多个关键帧中，能充分测试物体提取与关联模块的功能性。该序列的总长度为 19.807m，从起始到结束历时约 89 秒，并且在结束时回到起点，形成回环，能测试在物体关联与空间布局的精确构建。能够有效评估物体提取和关联算法对物体空间关系的处理能力。

图3-18展示了本章提出的方法在处理 TUM 数据集中的 $fr3/teddy$ 序列时，物体提取方面的优越能力。通过对生成的物体点云进一步处理，形成八叉树地图的表示方式。生成的地图能清晰地表示出物体的结构与形状，并且在物体上标有对应的标签。

为了进一步评估系统的性能，在 TUM 数据集的多个序列上与其他主流算法进

图 3-18 物体提取结果

行横向了对比，包括本文参考的 ORB-SLAM2，以及最近几年的算法 ORB-SLAM3、SG-SLAM。在本文中用于衡量算法性能的指标包括：相对旋转误差（relative rotational error，Rrpe）、相对平移误差（relative translational error，Trpe）以及绝对轨迹误差（absolute trajectory error，ATE）。ATE 用于衡量整个轨迹与地面真值的全局差异，反映了绝对定位精度；而 RPE 则聚焦于相邻位姿之间的相对精度，能够反映算法在短时间尺度上的局部误差。ATE 的计算公式如式3-9所示。

$$ATE_{all} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}||log(T_{gt,i}^{-1}T_{est,i})^{\vee}||_2^2} \tag{3-9}$$

其中 N 表示点的数量，$T_{gt,i}^{-1}$ 表示真实的位姿变换矩阵，$T_{est,i}$ 表示估计的位姿变换矩阵。

TUM 数据集中提供了真实的轨迹路径，因此算法在得出了估计的轨迹之后可以直接与真值进行计算误差。

本章节方法在大多数测试数据集上均表现出了明显的优势，展现了较低的误差值，尤其在 $fr1/room$ 和 $fr1/xyz$ 序列，本章节方法的 ATE 分别为 0.031086 和 0.007562，明显低于 ORB-SLAM2、ORB-SLAM3 以及 SG-SLAM，这表明本章节方法在这些序列上的轨迹精度更高，能够有效降低误差。但是在 $fr3/long\_office\_household$ 序列上本章算法效果较差，ORB-SLAM3 算法效果最好，该序列中桌面物品繁多且存在多个小物体聚集，导致构建物体的精度受到影响，从而使得最终结果出现较为不理想的情况。

表 3-2 在 TUM 数据集上与其他算法的 ATE 比较结果

seq	本章算法	ORB-SLAM2	ORB-SLAM3	SG-SLAM
fr1/room	**0.031086**	0.042535	0.076977	0.059916
fr1/xyz	**0.007562**	0.009379	0.010741	0.009529
fr2/rpy	**0.002053**	0.003175	0.009704	0.003099
fr3/long_office_household	0.012074	0.011615	**0.010993**	0.015816
fr3/sitting_static	**0.008537**	0.009723	0.010500	0.012673
fr3/teddy	**0.018534**	0.022663	0.476407	0.007480

表 3-3 在 TUM 数据集上与其他算法的 RPE 比较结果

seq	本章算法	ORB-SLAM2	ORB-SLAM3	SG-SLAM
fr1/room	**0.009375**	0.011798	0.021383	0.013272
fr1/xyz	**0.005721**	0.009379	0.006407	0.006148
fr2/rpy	**0.002046**	0.003175	0.002451	0.002099
fr3/long_office_household	0.13548	0.011615	**0.005403**	0.005460
fr3/sitting_static	**0.004039**	0.007208	0.005436	0.006273
fr3/teddy	0.004083	0.005375	**0.032998**	0.005143

由表3-3可知，在本章节的实验部分，RPE（相对定位误差）结果表明，本方法在大部分测试数据集上均取得了较低的误差值，显示了其较强的轨迹精度。在 $fr1/room$ 以及 $fr1/xyz$ 等数据集上，本章提出的方法有较高的准确性，尽管在 $fr3/long\_office\_household$ 序列上精准度降低，ORB-SLAM3 有最高的准确性，但是总体而言，本章提出的算法是有更好的鲁棒性和准确度。

3.5.4 消融实验

为了验证所提出的不同模块的有效性，本章在 TUM 数据集上进行了几组消融实验来比较不同模块对性能的效果。

本章提出的算法在提取二维物体到三维物体方面进行了处理，提升物体构建的准确性，以及物体参与捆绑优化对 SLAM 的定位效果有一定的影响，因此设置了相关的评估方法来验证模块的性能。（1）移除掩码优化部分测试系统位姿估计及物体构建效果（a）；（2）移除物体参与捆绑优化步骤，测试算法位姿估计效果（b）。

如表3-4所示，在未进行优化掩码处理的情况下，在建图的显示效果中，构建的物体精度却有所下降。然而，SLAM 精度的影响很小，正如图3-19所示。实验结果表明，在 b 实验过程中，SLAM 算法的精度有所降低，在序列 $fr3/teddy$ 中，不使用物体信息用在回环检测中 ATE 结果为 0.021573，对比使用物体信息结果精度

表 3-4 在 TUM 数据集上各模块消融结果

seq	本章算法	a	b
fr1/room	**0.031086**	0.033056	0.039041
fr1/xyz	**0.007562**	0.008261	0.009235
fr2/rpy	**0.002053**	0.002470	0.002862
fr3/long_office_household	0.012074	0.013516	**0.010836**
fr3/sitting_static	**0.008537**	0.009358	0.009746
fr3/teddy	**0.018534**	0.020637	0.021573

降低了大约 16%，这表明物体构建的引入及物体信息的利用对算法起到了积极作用，能够显著提升整体精确度。

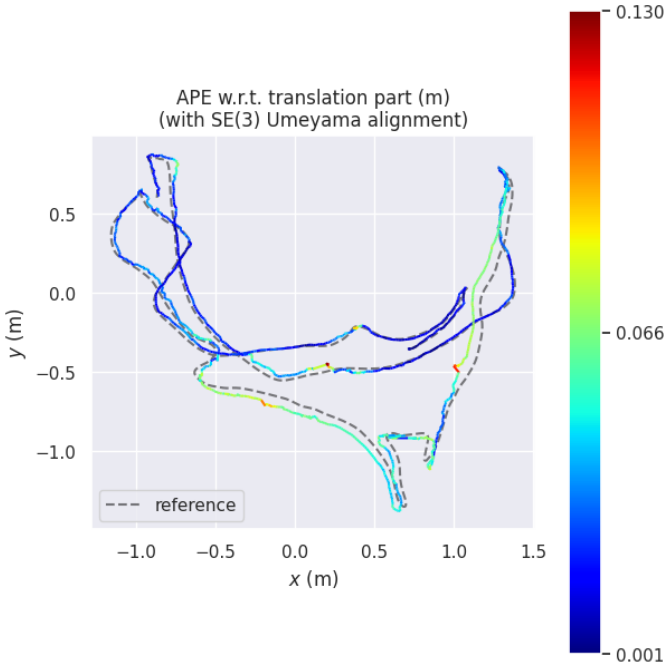


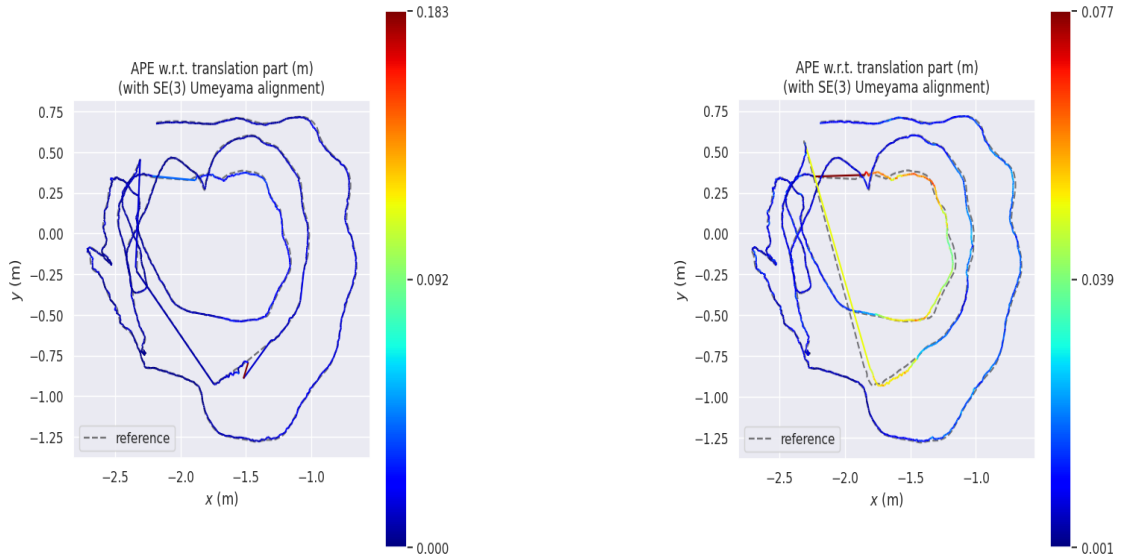
图 3-19 轨迹结果

3.5.5 可视化结果

上一节在定量的角度上对本文提出的 SLAM 系统的有效性进行了评估测试。本节图3-20展示了在数据集 TUM 的 $fr3/teddy$ 序列上本文提出的算法与 ORB-SLAM2 算法的对比结果。

图3-20结果表明，本文提出的算法比较 ORB-SLAM2 算法更贴合 TUM 数据集提供的真实值，由于本文算法对物体做了更准确的处理并且利用了更多的信息来辅助 SLAM 算法对于位置的估计，因此结果更加接近真实值。

图3-21结果为本文算法在序列 $fr3/teddy$ 上进行的物体构建。整个序列的拍摄



(a) (b)

图 3-20 轨迹结果对比。(a)本文算法；(b)ORB-SLAM2

围绕椅子上的泰迪熊展开，通过多帧图像的处理与分析，最终构建出了一个准确的物体识别结果。系统成功识别出泰迪熊这一物体类别，并通过精确的坐标定位技术标注了其在场景中的位置。最终，识别结果以八叉树地图的形式呈现。

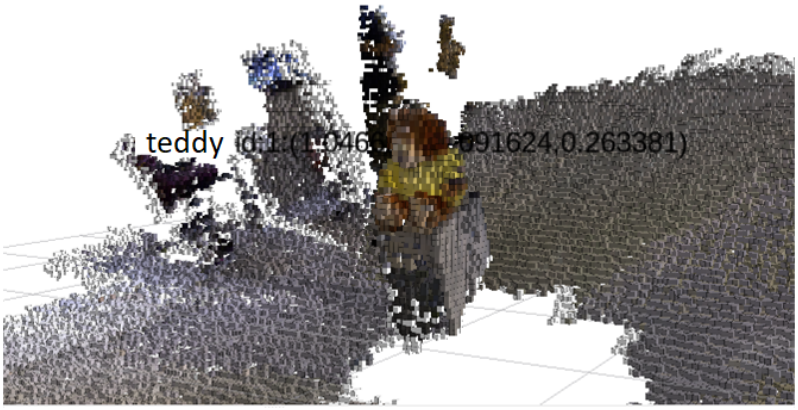


图 3-21 物体地图构建结果

3.6 本章小结

本章的研究重点在于研究物体在 SLAM 系统中的构建，并通过引入物体信息提升 SLAM 算法的精度。研究内容涵盖了物体的提取与关联，从二维图像到三维立体物体的构建，利用物体信息进行 SLAM 捆绑优化，旨在提升系统的整体有效性与鲁棒性。

本章提出了一种优化物体掩码的方法。基于 YOLOv8 识别所得的掩码，结合

深度图中的深度信息对掩码进行二次优化。通过深度一致性原理，精确提取物体边缘位置，从而优化物体边缘掩码。同时本章算法会构建一个物体数据库，在全局观测中持续维护物体信息，并确保在多视角下准确关联同一物体。为了构建更加精准与鲁棒的算法系统，本文将在算法后端融入物体信息，参与捆绑优化流程，利用更多维度的信息对算法的位姿估计进行进一步优化。

本章在公开数据集 TUM 上进行了定量与定性的实验，并与先进的 SLAM 算法进行了横向对比。在部分序列中，本文提出的算法表现出更优的性能，充分体现了其准确性与鲁棒性。最终后，通过消融实验验证了各个模块对算法有效性的贡献。

第四章 深度恢复与回环检测算法

4.1 引言

随着机器人技术和自动驾驶等领域的快速发展，物体地图和回环检测在 SLAM 系统中扮演着越来越重要的角色。物体信息作为环境建图的关键元素之一，能够提供更加细致和智能的地图表示，而回环检测则有助于系统在长期运行中消除累积误差，实现自我校正。因此，结合物体信息进行回环检测是提高 SLAM 系统性能的重要方向。然而，在现实环境中，物体的形状和材质对深度感知带来了挑战，特别是对于反光物体和透明物体，传统的深度传感器往往难以准确获取其深度信息。

针对上述问题，本章将详细介绍物体参与回环检测以及深度恢复的具体实现方法，并对比实验结果，展示其在实际应用中的优势与可行性。

4.2 物体地图检测回环

回环检测在 SLAM 系统中起到了关键作用，其主要功能是识别机器人是否回到了之前经历过的位置，从而为地图构建和位姿估计添加全局约束。当机器人在环境中移动时，由于传感器噪声和算法的累积误差，位姿估计会逐渐偏离真实位置，导致绘制的地图出现不准确甚至变形。回环检测通过识别已访问区域并建立当前位姿与历史位姿之间的联系，能够有效修正这些漂移误差。它通过引入回环约束，让 SLAM 系统在全局范围内调整所有位姿和地图点的位置，优化地图的一致性和精度。回环检测还可以为定位提供额外的参照点，使机器人在长时间运行或复杂环境中依然能够保持可靠的定位和导航能力。因此，回环检测不仅是消除累积误差的重要手段，也是保障 SLAM 系统鲁棒性和全局优化效果的关键环节。

语义 SLAM 方法能够将物体信息有效地引入回环检测算法中，利用物体地图提供基于特征物体的更为细致和丰富的环境表示。传统的回环检测通常依赖于全局的几何信息，而物体地图的引入则使得回环检测不仅仅局限于全局几何结构，还能结合局部物体特征进行检测。这种结合局部物体特征的回环检测方法，能够在复杂环境中提升回环检测的准确性和鲁棒性。物体地图的应用流程如图4-1所示，它展示了物体地图的应用流程。

物体地图是本文回环中的核心，它提供了一个基于物体的环境表示，其中不仅包括每个物体的几何特征，还包括其邻近物体的空间关系。在实际操作中，物体地图的构建通过从传感器数据中提取物体的几何信息（如物体的形状、位置、尺

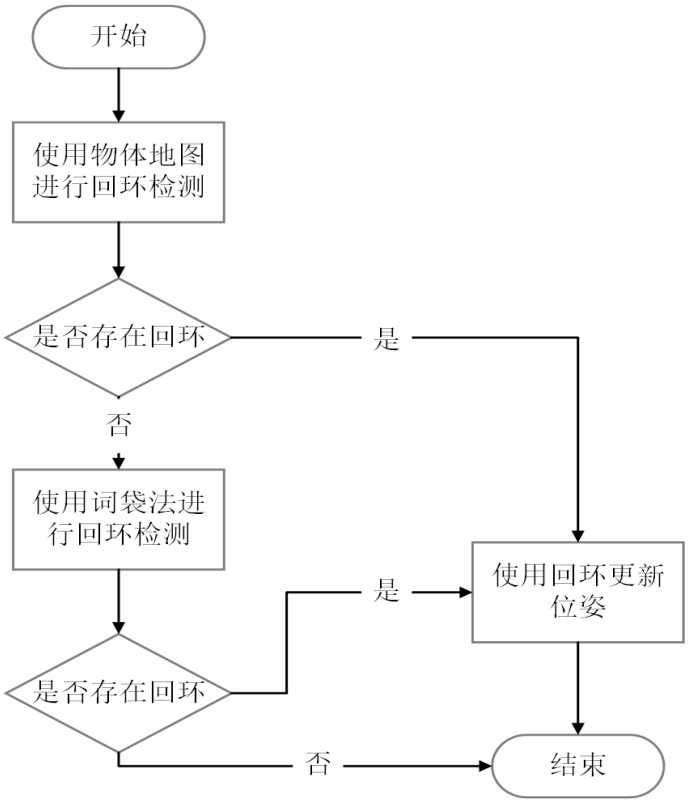


图 4-1 物体参与回环检测流程

寸等），并结合物体之间的空间关系来形成一个更加细致的环境模型。这一过程中，不仅捕捉了每个物体的局部几何特征，还通过邻近物体的空间依赖关系进一步丰富了地图的信息量。

每个物体在地图中的表示不仅仅是一个孤立的几何体，而是一个包含多个物体之间关系的集成体。具体来说，物体地图包含了每个物体的几何特征，以及该物体周围最近 5 个物体的空间位置及其相对关系，物体地图关系表示方式如4-1所示。这些邻近物体的空间关系提供了更为丰富的特征，能够有效反映物体间的相对布局。由于物体间的空间关系通常具有较高的局部一致性，因此这些关系能够帮助提高回环检测的鲁棒性和精度，尤其是在具有相似场景的环境中。

$$M_1 = \{P_1, \{(P_2, d_{12}), (P_3, d_{13}), (P_4, d_{14}), (P_5, d_{15}), (P_6, d_{16})\}\} \tag{4-1}$$

其中 M_i 表示第 i 个物体以及最近的 5 个物体形成的局部地图，P_i 表示第 i 个物体，d_{ij} 表示第 i 个物体和第 j 个物体之间的距离。

在 SLAM 系统中，由于每一个轨迹点均由系统估算得出，这不可避免地导致了误差的逐步积累，并产生了轨迹漂移的现象，即最终估计的位姿可能存在一定偏差。即使数据集最终回环，返回到初始位置，因误差的存在，观测到的物体与序列最初构建的物体难以匹配。在此情形下，回环检测技术可被用以消除这些误差，

进而优化先前的物体位置结果。由于物体相邻关系计算所依赖的信息较少，计算速度较快，本章的算法在 ORB-SLAM2 回环检测流程之前引入了物体检测回环方式。然而可能存在遮挡、检测不准确等因素，当使用物体地图检测回环失败时，不能断定一定不存在回环，所以需要进一步采用原有的回环检测方法，即通过关键帧的词袋信息匹配检测可能的回环。

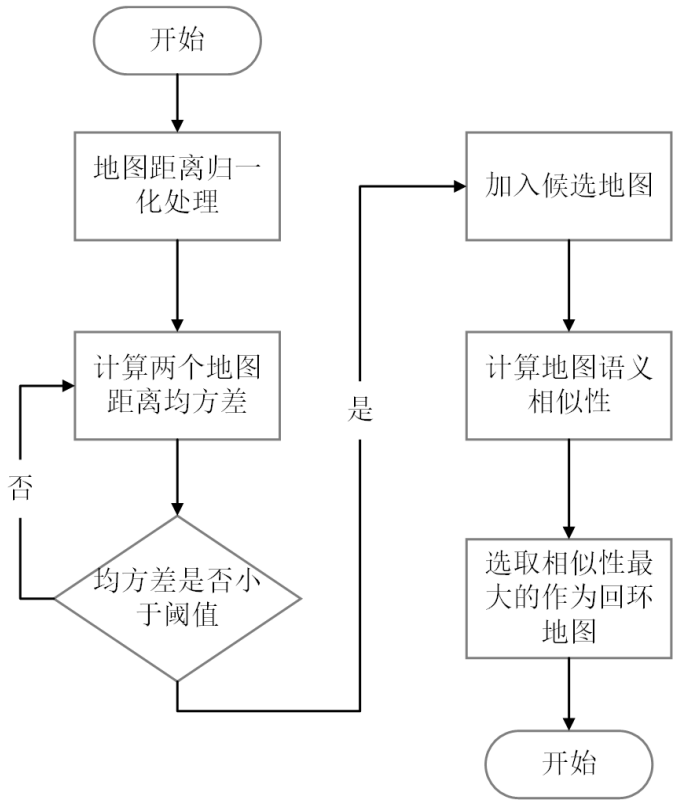


图 4-2 物体地图匹配流程图

物体地图匹配如图4-2所示。利用物体地图中物体与其相邻物体的空间关联信息，结合当前观测中的物体及其邻近物体，计算两者之间的相似度，以此判定当前观测是否与先前某一位置相匹配。为确保回环检测的准确性，预先设定一个相似度阈值 δ_{map}。当当前观测的物体关联关系与物体地图中的相似度超过该阈值时，系统会认为可能存在回环。相似度的计算不仅依赖于单个物体的几何特征，还融合了物体之间的空间依赖关系，从而提高了回环检测的精度与鲁棒性。计算两个局部地图相似度的公式如4-2所示。

$$d_{similarity} = \frac{1}{5}\sum_{i=1}^{5}(d_i)^2,\ d_i = |d_{e_i} - d_{f_i}| \tag{4-2}$$

其中 d_{e_i} 是当前观测物体地图的物体间距离，d_{f_i} 是局部地图中对应物体间距离。并且在计算相似性时需要找到两个地图中对应的物体，所以首先需要对两个物体地图中的距离分别进行排序，确定对应关系，进而得到排序后的 d_{e_i} 和 d_{f_i}。

当物体之间的距离相似度满足预定条件时，当前局部物体地图将被加入到候选地图列表中。在完成所有子地图的计算后，系统会从候选地图中进一步计算每一对地图之间的语义相似度，如式4-3所示。语义相似度的计算考虑了物体的类型，以确保候选地图的语义一致性。最后，系统会选择语义相似度最低的一组地图作为最终的回环检测结果。

$$S_{label} = \sum_{i=1}^{5}\sum_{j=1}^{5} L_{ij}, \quad L_{ij} = \begin{cases} 0 & if\ L_i = L_j \\ 1 & if\ L_i \neq L_j \end{cases} \tag{4-3}$$

其中 S_{label} 表示两个地图的语义相似度，L_i 表示在当前观测地图上物体 i 的类别，L_j 表示局部地图上对应物体的标签，L_{ij} 表示两个物体是否为相同类别物体。

当物体地图中检测到回环时，可以通过物体之间的匹配关系来进行物体匹配。首先根据物体匹配信息，修正当前观测到的物体位置。这一过程通过与物体数据库中的对应物体信息进行对比，来消除当前观测物体的定位误差。在物体匹配的基础上，对匹配的物体及其对应的帧信息进行优化，进而优化当前观测帧的位姿信息。与此同时，回环过程中所有相关帧的位姿信息也会被同步调整，以减小整个回环中的累计误差。该过程的工作流如图4-3所示。

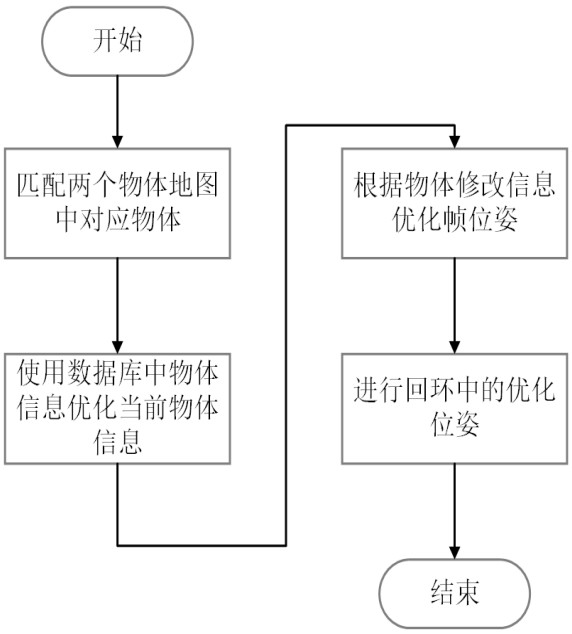


图 4-3 检测回环后优化位姿流程图

4.3 透明物体深度恢复

RGB-D 相机通过捕捉反射光来进行深度估计，但透明和反光物体（如玻璃、水面或镜面物体）由于其复杂的光学特性（如光的折射、反射和透射）而导致出现

深度测量的严重误差。具体来说，透明物体（如玻璃）可能会使深度传感器无法准确感知物体几何形状；而镜面反射物体（如镜子）会将相机的光束反射回传感器，导致不真实的深度信息。因此，RGB-D 相机在遇到这些物体时，常常会在深度图中出现不可识别或模糊的区域，这对后续的物体识别、定位、甚至抓取任务带来了显著的挑战。

语义 SLAM 算法的目标是同时构建环境的语义地图并进行自身定位。在实际环境中，SLAM 系统需要同时处理环境中的几何信息和语义信息，而透明和反光物体的存在，通常会干扰 SLAM 算法的稳定性和精度。以下是几个原因：

1. 反光物体的表面特征可能导致深度传感器产生误导性的深度估计，从而影响系统对环境的几何建模。

2. 透明物体可能会造成深度信息丢失或模糊，导致 SLAM 算法无法获取物体的正确几何信息。

3. 对反射或折射的误估计可能导致语义信息（如物体类别或位置）的错误分配。

因此，在物体 SLAM 中，如何正确处理这些物体的深度信息，并恢复或补全错误的深度数据，是确保 SLAM 系统鲁棒性和精确度的一个关键任务。

受到 CLIP[50] 网络的启发，本章提出了一种基于多尺度特征对比学习方法。具体而言，使用深度图像编码器（如卷积神经网络或 Transformer 结构）使得网络能够处理并提取深度图像的特征。并且加入了特征金字塔，以在多维度上分析训练图像特征。网络结构如图4-4所示。

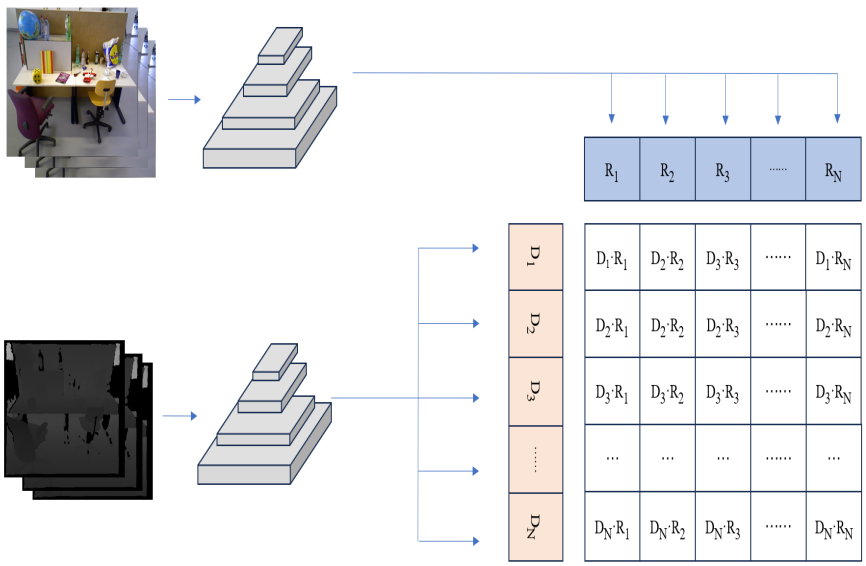


图 4-4 基于多尺度特征的对比学习网络

该方法采用了两流网络架构，其中每个流由一个基于 Transfomer 的编码器（Encoder）组成，分别用于提取输入的 RGB 图像和深度图像的特征。对于 RGB 图像 I_{rgb}，使用一个编码器 T_{rgb} 来提取其特征向量 f_{rgb}。同样地，对于深度图像 I_{depth}，我们使用另一个编码器 T_{depth} 来提取其特征向量 f_{depth}。

$$f_{rgb} = T_{rgb}(I_{rgb}),\ f_{depth} = T_{depth}(I_{depth}) \tag{4-4}$$

这样能得到两个模态的特征表示，分别是 RGB 和深度图像的高维特征向量。

为了最大化 RGB 图像和深度图像之间的特征相似性，引入了对比学习损失。目标是让同一场景的 RGB 图像和深度图像的特征在嵌入空间中尽可能靠近，而不同场景的特征则尽量分开。使用 InfoNCE 损失来实现这一目标，计算每一对 RGB 图像和深度图像的特征向量 f_{rgb} 和 f_{depth} 的相似度：

$$\sin(f_{rgb}, f_{depth}) = \frac{f_{rgb} \cdot f_{depth}}{\|f_{rgb}\| \|f_{depth}\|} \tag{4-5}$$

基于这个相似度，定义了对比损失为：

$$\mathcal{L}_{\text{contrastive}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\sin(f_{\text{rgb}}^{(i)}, f_{\text{dcpth}}^{(i)}))}{\sum_{j=1}^{N} \exp(\sin(f_{\text{rgb}}^{(i)}, f_{\text{dcpth}}^{(j)}))} \tag{4-6}$$

其中 N 是图像对的数量，i 和 j 分别表示正样本对和负样本对。该损失函数鼓励正样本对的特征相似度最大化，并最小化其他样本对之间的相似度。

通过引入深度图像与 RGB 图像特征向量的相似度损失，强化了两种图像特征之间的物体关联性。RGB 图像提供了丰富的语义信息，而深度图像则包含了物体的空间结构信息，结合这两者的特征能够更全面地表达场景的几何和语义特征。因此，通过相似度损失的优化，模型能够更准确地恢复深度信息，尤其在复杂场景中表现出较强的适应能力和更高的深度估计精度。整体网络架构如图4-5所示。

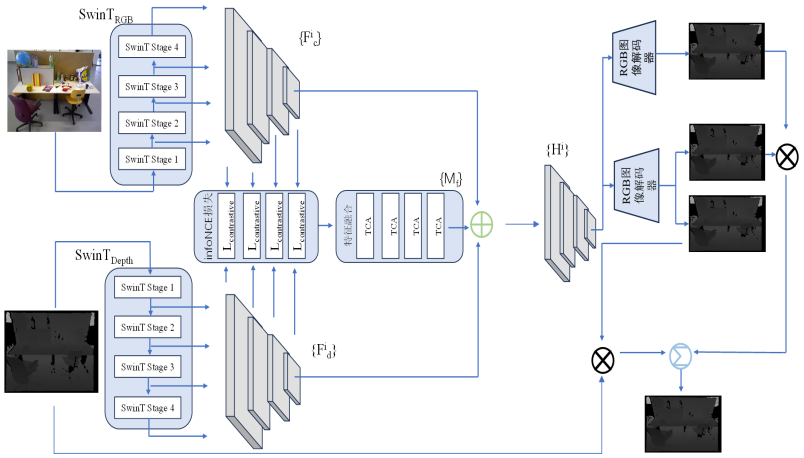


图 4-5 透明物体的解决网络图

4.4 实验结果与分析

本小节对提出的回环检测算法以及改进的深度恢复网络进行了测试与评估，同样是在 TUM 数据集上进行测试。

4.4.1 实验结果

本小节将展示将物体信息引入到 SLAM 回环检测中的实验结果。为了验证这一方法的有效性，本文进行了多组对比实验，比较了在没有物体信息和加入物体信息两种情况下的回环检测性能。实验的主要目的是评估物体信息对 SLAM 系统回环检测的影响，尤其是对系统的精度、稳定性以及计算效率的提升。

本章设计了以下几项实验：首先，使用其他 SLAM 系统进行回环检测，并评估其在常见的环境中的性能，比如 ORB-SLAM2。然后，使用本文提出的 SLAM 算法，通过物体检测与识别算法提取的关键信息，进一步提高回环检测的准确性与鲁棒性。最后，基于这些实验结果，从定位精度方面进行详细分析与对比，探讨物体信息对回环检测的潜在贡献。

图4-6展示了序列回环的结果，验证了在回环情况下，物体构建能够准确地进行关联。在序列结束时，系统成功回到最初的位置，测试了物体提取与关联在回环情境中的鲁棒性与准确性。4-6(a)展示了序列初始时刻的观测图像，4-6(b)为序列结束时的观测图像。4-6(c)表明，当重新观测到物体时，它们被准确地关联回初始阶段构建的三维物体中，保持了物体的一致性和准确的空间位置。此结果表明，算法在处理回环时，能够有效地维持物体地图的连贯性与稳定性，确保物体提取与关联在循环路径下的可靠性。

为了进一步评估提出的回环检测算法加入到 SLAM 系统中的有效性，同样将算法与其他先进的算法进行了横向比较，计算估计轨迹与真实轨迹之间的误差，可视化结果如图4-7所示。

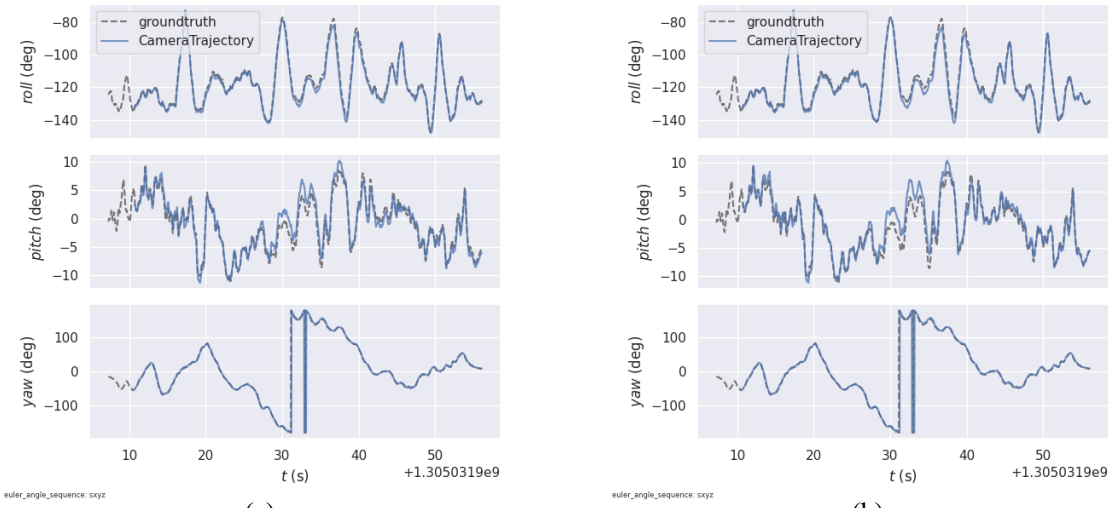
在图4-7中，4-7(a)展示了本章算法轨迹结果和真实轨迹值对比图，4-7(b)展示了 ORB-SLAM2 算法对比结果。可以看到在序列 $fr1/room$ 上，两个算法的估计值与真实值均存在一定偏差，并且在真实值上下波动，说明算法的位姿估计存在误差。然而，比较两个算法的结果可以发现，本章提出的算法更接近真实值曲线，表明其位姿估计更加准确，算法性能更优。

如图4-8所示，是在 $fr1/room$ 序列上的 ATE 结果图，其中4-8(a)表示本章算法 ATE 结果，4-8(b)表示 ORB-SLAM2 算法 ATE 结果。从图中可以看出，本章算法与 ORB-SLAM2 算法在波形变化趋势上基本一致，但在数值上有所不同。本章算法的低值更低，且最高峰值也低于 ORB-SLAM2 算法。具体而言，本章算法的高

(a) (b)

(c)

图 4-6 物体构建结果。(a)序列初始位置观测图；(b)序列结束位置观测图；(c)物体构建结果

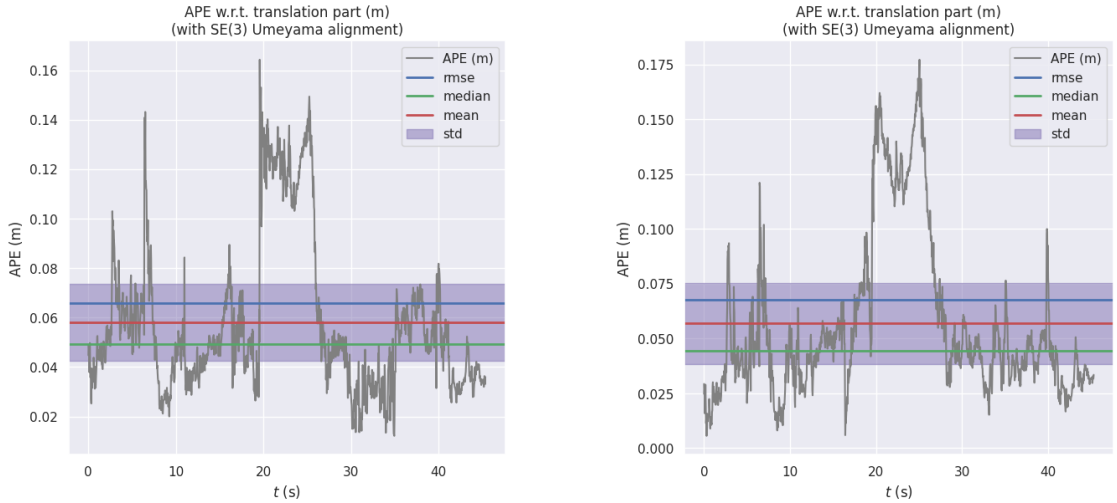


(a) (b)

图 4-7 算法轨迹结果对比。(a)本章算法轨迹；(b)ORB2 轨迹

值大约为 0.165，而 ORB-SLAM2 的峰值则超过了 0.175。在该序列上，本章算法相比 ORB-SLAM2 算法表现出更为优越的效果，具有更小的误差波动和更稳定的定位精度。

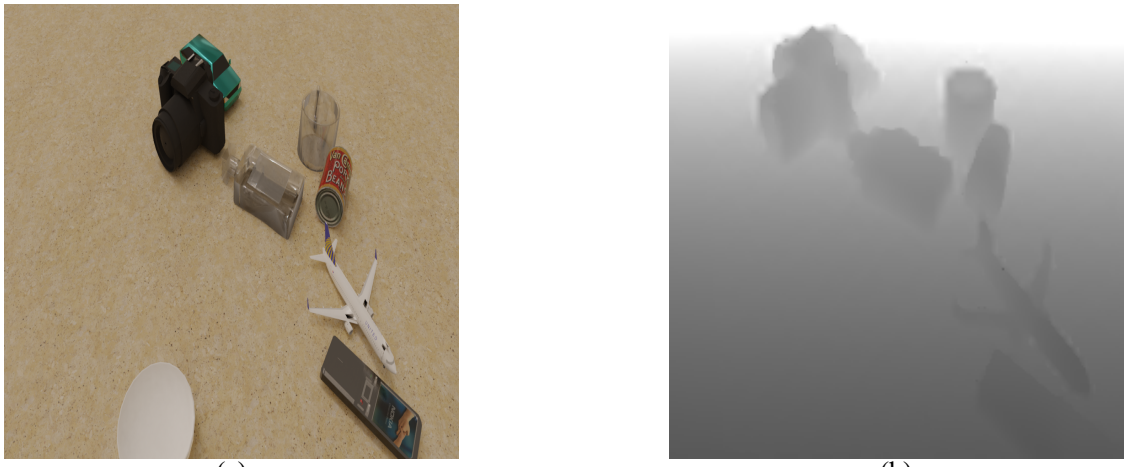
针对反光物体和透明物体带来的挑战，改进的神经网络在恢复深度值方面取得了较好的效果。通过优化网络结构和特征向量方法，该网络能够更有效地处理这些具有复杂光学特性的物体，避免了传统深度恢复方法在这些情况下的精度下



(a) (b)

图 4-8 算法 ATE 结果对比。(a)本章算法结果；(b)ORB2 算法结果

降。图4-9展示了该改进网络在处理反光和透明物体时的深度恢复结果。



(a) (b)

图 4-9 深度恢复前后对比。(a)RGB 图；(b)深度回复后

由图4-9所示，物体的深度值被估计补全。在原始的深度图中，透明物体大部分区域的深度值被检测错误，当前像素被赋值了一个错误的深度值。而在使用神经网络对深度进行恢复之后，物体像素点上的深度值被较准确的估计，能有效的给出值以方便后续对物体的使用。

4.4.2 消融实验

为了验证所提出的回环检测模块的有效性，本章在 TUM 数据集上进行了几组消融实验，通过对比模块的性能来评估它们的效果。实验结果显示，回环检测模块对于回环数据集的检测结果有显著的影响。尤其是引入物体信息后，回环检测

的准确度和鲁棒性得到了增强。

本章提出的利用物体信息参与 SLAM 系统的回环检测流程中，对有回环的数据集检测结果有一定的影响，以下设置了一组相关的评估方法来验证回环检测模块的性能。

表 4-1 在 TUM 数据集上回环检测模块消融结果

seq	本章算法	不带物体回环检测
fr1/room	**0.031086**	0.038510
fr1/plant	**0.022395**	0.026437
fr2/desk	**0.021214**	0.025381
fr3/long_office_household	0.012074	**0.011043**
fr3/teddy	**0.018534**	0.021590

如表4-1所示，将物体信息融入回环检测流程中对 SLAM 系统产生了积极影响，有效提高了算法的整体准确性。这表明，物体的空间几何信息在回环检测中发挥了显著作用。在序列 $fr1/room$ 中，启用回环检测后，由于物体信息的引入，ATE 值为 0.031086；而去除回环检测后，缺乏物体信息的支持，算法的位姿估计结果明显下降，ATE 值上升至 0.038510。使用物体信息增强回环检测使得该序列的精度提升了 19%。由此可见，物体信息在该序列中对回环检测起到了正向促进作用。

4.5 本章小结

本章的研究重点在于将提取到的物体信息引入回环检测流程中，通过利用物体的空间信息特征来优化回环检测算法。具体来说，物体信息能够为回环检测提供更多的上下文信息，帮助系统在复杂环境中更准确地识别回环。通过引入物体空间位置、物体间的相关性等特征，本章提出的回环检测算法能够更有效地识别出场景中的相似性。物体的空间信息能够帮助系统更好地处理由于光照变化等因素带来的挑战，提高回环检测的鲁棒性和准确度。

本章还探讨了透明物体、反光物体等的建模问题。尤其是反光物体在使用相机进行深度检测时，常常出现误差或完全无法检测到深度。为此，改进了神经网络对输入的 RGB 图像和深度图进行恢复，从而弥补这一缺陷。

本章在公开数据集 TUM 上进行了定量与定性的实验，与先进的 SLAM 算法进行横向对比，提出的算法在大部分序列上有更好的表现，并且通过消融实验来验证了本章的回环检测算法对 SLAM 系统有正面影响。

第五章 SLAM 系统设计与实现

5.1 需求分析

5.1.1 需求背景

随着智能机器人和自动驾驶技术的不断进步，物体级语义 SLAM 技术的需求逐渐增加。在许多应用中，机器人需要精确地识别、定位并跟踪环境中的个体物体，同时构建高质量的环境地图。物体级语义 SLAM 技术通过将物体的识别与定位集成到地图构建过程中，使机器人能够在执行任务时更好地理解和操作环境中的物体。这对于自动驾驶、智能制造、仓储管理等领域尤为重要，因为它能够提高机器人在复杂环境中的决策能力和执行效率。这一技术的应用可以显著提升机器人的自主性，使其能够高效完成任务，并在各类智能系统中发挥核心作用。

语义 SLAM 是自动驾驶、机器人导航等领域的关键技术之一。自动驾驶车辆需要精确识别和定位道路中的障碍物、行人和其他车辆，以确保安全驾驶，而语义 SLAM 技术正是实现这一目标的基础。同时语义 SLAM 对于智能制造和仓储管理等行业也具有重要应用。通过语义 SLAM 技术，机器人能够准确识别并跟踪仓库中的物品，提高物料搬运效率并减少人为错误。此外，语义 SLAM 还能为智能城市、无人机巡检等领域提供精准的环境感知支持，推动各类智能系统的进一步发展。

综上所述，语义 SLAM 技术的需求背景主要源于自动驾驶技术的不断演进、智能机器人在制造和物流领域的广泛应用，以及智能城市和无人机等领域的日益发展。为了满足这些需求，持续研究和开发更高效、稳定、精准的语义 SLAM 系统，已成为推动智能系统和自动化进步的关键。

5.1.2 功能性需求

本章对物体级语义 SLAM 系统进行需求分析，提炼了以下需要完成的功能性需求：首先用户需要通过注册和登录进入算法系统，然后选择检测的数据集，后台完成轨迹估计的同时输出可视化的物体地图构建。最后构建的地图可以保存成文件方便后续查看。此外，系统还支持用户输入 RGB 图像及其对应的深度图像进行深度恢复操作。通过深度恢复算法，系统将根据输入的 RGB 图像和深度图像信息，恢复出更加准确的深度信息，最终输出一个经过深度恢复后的结果图像。

用户管理模块功能性需求如表5-1所示。

SLAM 算法模块功能性需求如表5-2所示。

表 5-1 用户管理模块需求

需求编号	需求名称	需求描述
UserManage01	用户注册并登录	用户可以通过提供必要的个人信息进行注册，包含用户名、密码等。注册用户能够使用其用户名和密码进行登录系统
UserManage02	用户修改个人信息	注册用户能够编辑其个人信息，包括邮箱、电话等
UserManage03	用户密码重置	用户能够通过电话或其他验证方式重置密码。

表 5-2 SLAM 算法模块功能需求

需求编号	需求名称	需求描述
SLAMDet01	数据集选择	用户上传一个数据集
SLAMDet02	轨迹估计	对数据集的轨迹进行估计
SLAMDet03	物体地图构建	对观测物体进行构建

物体地图构建展示模块功能性需求如表5-3所示。

表 5-3 地图构建模块功能需求

需求编号	需求名称	需求描述
SLAMDet01	轨迹估计	对数据集的轨迹进行估计
SLAMDet02	物体地图展示	对构建的物体地图进行展示

深度恢复算法模块功能性能需求如5-4所示。

表 5-4 深度恢复模块功能需求

需求编号	需求名称	需求描述
DepthRestoration01	图像选择	用户上传 RGB 图像以及对应的深度图像
DepthRestoration02	深度恢复	对深度图中一些错误深度进行一定的恢复

为了实现上述功能性需求，将系统的主要功能设计如下：

1. 注册与登录功能。登录界面是用户与系统交互的入口，负责身份验证，确保只有授权用户可以访问系统资源。该界面应设计为简洁友好，操作流程直观。用户可以通过提供必要的个人信息，如用户名、密码、电子邮箱等进行注册，系统会验证信息的有效性，并创建相应的用户账户。

2. 轨迹估计功能。本系统需要用户上传用于检测的数据集，并且在使用数据集的过程中不断估计自身的位姿，并会在最后将轨迹输出到文件中保存。

3. 物体构建功能。在进行位姿估计的同时，系统通过 YOLOv8 检测已知类别的物体，并从二维图像中提取物体信息，进而构建出三维物体模型。这些物体将被呈现在物体地图中，提供更加直观的空间表达。

4. 地图展示功能。该功能通过八叉树的形式对物体地图进行展示，物体部分由 3D 占据栅格表示，并且标有对应的物体标签，突出物体的细节与结构，增强视觉表现力。

5. 深度恢复功能。该功能对输入的 RGB 图像及其对应的深度图像进行深度值的恢复，尤其是针对深度图中可能存在的错误或缺失点，提供较为准确的估计值。该功能通过结合图像的颜色信息和现有的深度信息，利用深度恢复网络来推断缺失或错误的深度数据，从而提升深度图的质量和精度。

6. 文件管理功能。该功能用于对位姿估计轨迹文件以及物体地图进行查看、保存和删除。

5.1.3 非功能性需求

非功能性需求对系统性能和用户体验至关重要，因此在设计物体级语义 SLAM 算法系统时必须充分考虑。主要涉及的非功能性需求包括以下几个方面：

1. 性能需求：

由于系统在运行过程中需依赖 YOLOv8 识别模块进行物体识别，并同时进行位姿估计，因此需要配备高性能计算机或专业硬件，以确保数据集的高效处理与流畅运行。

2. 可靠性需求：

系统需要保证数据处理以及数据结果的保密性。

3. 易用性需求：

系统应提供一个简洁直观，便于操作的界面，以便大多数用户群体有便捷的交互体验。

4. 安全性需求：

系统需要提供不同用户的权限管理，保证不同用户不能操作其他用户数据，对不同的用户开放不同的权限。

5. 可扩展性：

本系统设计确保每个模块具备独立的职责和清晰的功能边界，从而实现模块化管理。模块之间通过标准化接口进行通信，这种设计提高了系统的灵活性和可维护性，使得后续的功能扩展和优化更加便捷。

6. 可维护性：

该系统的技术代码和文档应遵循严格的规范，以便开发人员能够轻松理解和

使用。良好的代码风格、统一的命名约定、清晰的注释以及详细的文档能够确保系统在后期的维护、更新和扩展过程中更加高效和顺利。这不仅有助于提高开发团队的工作效率，还能减少错误和不必要的重复劳动，保障系统的长期稳定性和可扩展性。

5.1.4 系统用例设计

SLAM 算法系统主要是由四个模块组成：用户管理模块、SLAM 算法模块、深度恢复模块以及文件管理模块。如图5-1是 SLAM 系统用例图。

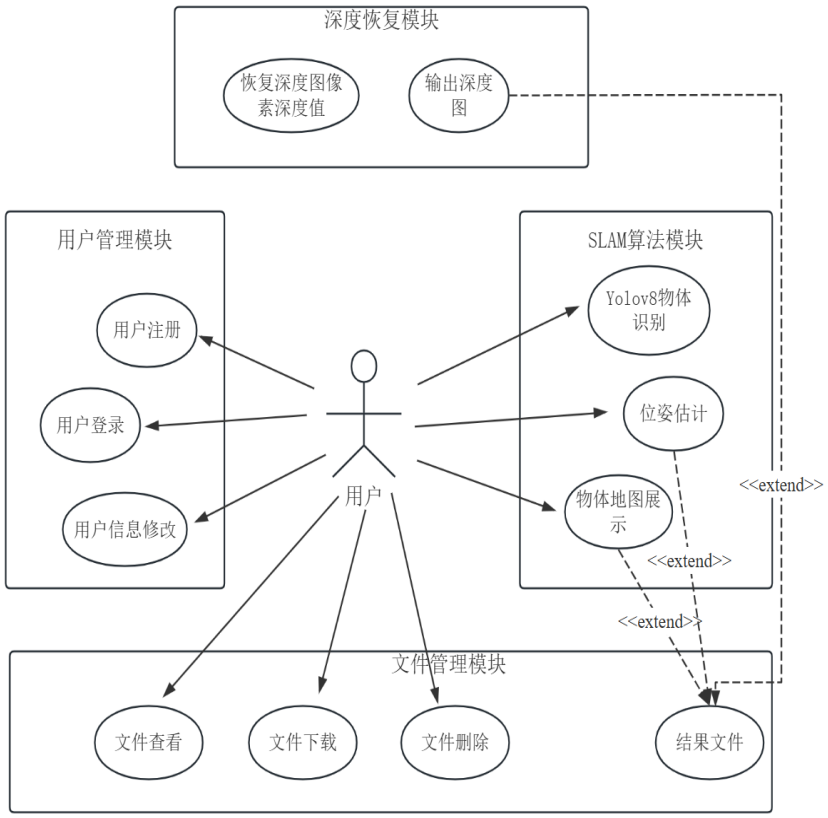


图 5-1 SLAM 系统用例图

在物体级语义 SLAM 算法系统中，用户管理模块是系统入口，提供注册和登录功能，并根据用户身份分配相应权限。同时对用户信息进行加密存储，确保其安全性。SLAM 算法模块是系统核心，负责处理和分析用户上传的数据集，实时估计数据集中每一帧的位姿，并同步对图像中的物体进行识别与重建，从二维图像中构建出精准的三维物体地图。深度恢复模块的核心功能是对用户输入的深度图进行深度值估计，特别是针对那些错误的深度值或未测量出的像素点进行修复。通过该模块，系统能够有效地修复深度图中的缺失部分，并最终输出恢复后的深度图。文件管理模块则承担着管理位姿估计数据及物体地图文件的职责，便于用

户查看、删除或操作结果文件。

5.2 总体设计

5.2.1 系统架构设计

本文的 SLAM 算法系统采用了分层架构设计，借鉴了经典的三层架构模式，涵盖了表现层、业务层与数据层。这种分层设计有效地将各个功能模块进行独立划分，每一层专注于特定的任务，从而减少了各层之间的复杂交互与依赖。SLAM 算法系统的整体架构如图5-2所示。

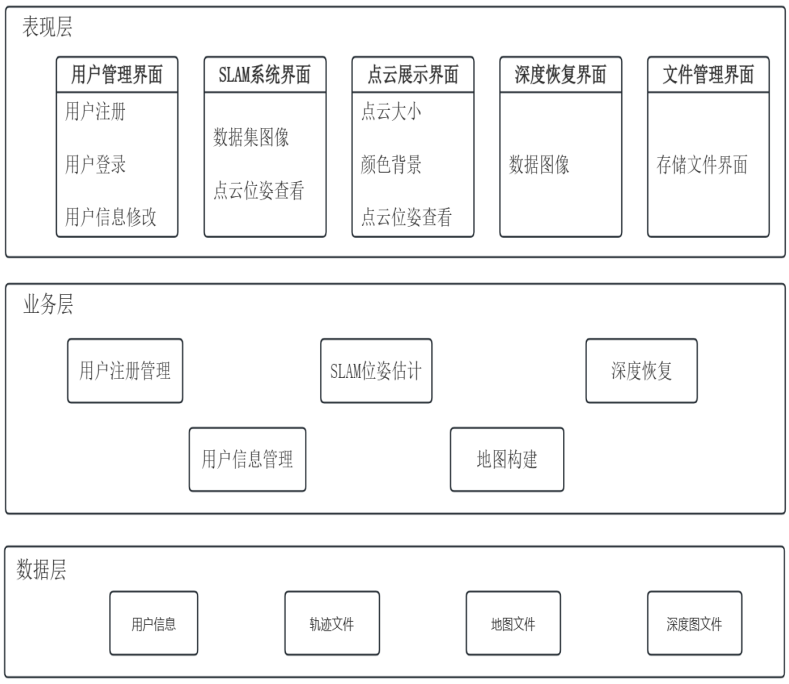


图 5-2 SLAM 系统架构图

首先表现层是整个系统的前端界面，用于接收用户的操作指令，实现与用户的操作，整体设计由 Qt 技术实现。在这一层中用户管理部分可以让用户对自己的信息进行查看、注册和删除操作。SLAM 算法结果界面是用于展示算法在运行数据集的过程中构建的八叉树地图，以及展示轨迹估计和真值的轨迹比较图像。文件存储部分是用于对算法结果的保存，支持用户对自己账号下的文件的删除和查看操作。

业务层是系统的后端部分，用于对所有业务逻辑的决策。用户管理模块专负责用户的注册、登录与信息修改等操作，确保用户数据的安全与隐私得到严格保护。SLAM 算法模块则涵盖了对数据集的逐帧处理，运用整体算法生成物体地图

和位姿信息，为后续的可视化展示提供必要的支持。深度恢复模块为输入的深度图像进行深度估计并恢复深度值。

最后，数据层负责系统中所有数据的存储与处理。用户信息包括用户名、密码等敏感数据，文件数据则包含用户上传的数据集、算法生成的轨迹结果和物体地图等关键数据。

5.2.2 系统功能模块设计

物体级语义 SLAM 算法系统的模块设计将系统划分成了四个关键的模块，如图5-3所示。主要由用户管理模块、SLAM 算法模块、深度恢复以及文件管理模块组成。

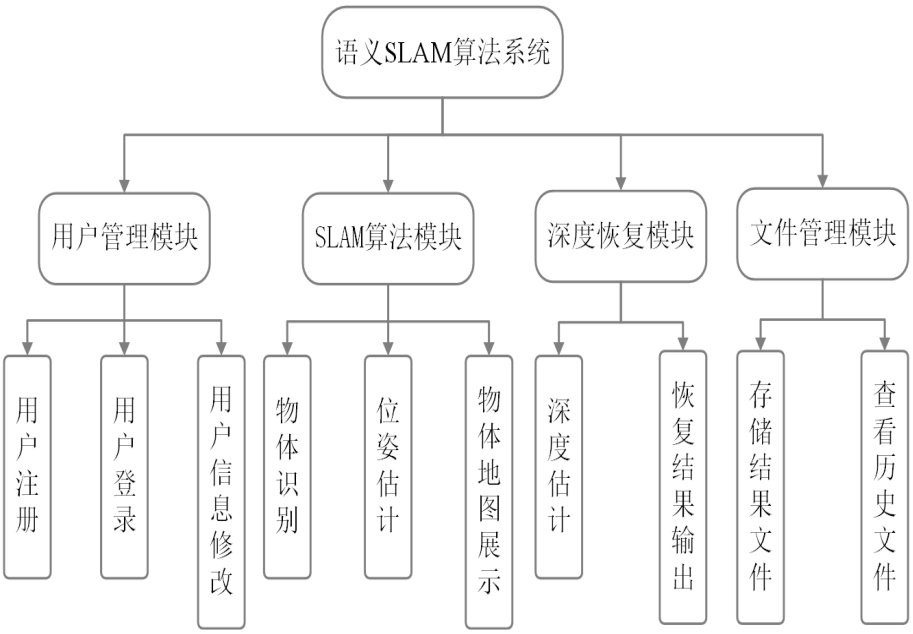


图 5-3 系统功能模块

用户管理模块是系统的入口，提供注册和登录功能，确保只有授权用户能使用系统，从而保障数据安全。同时，该模块允许用户创建新账户，确保账户间的数据独立且不可共享。

SLAM 算法模块是整个系统的核心部分。该模块提供用户在使用数据集中对数据的处理以及结果估计功能，主要需要实现以下的功能：

1.YOLOv8 识别模型构建。对用户数据集中的图像进行物体识别，输出预训练中物体类别、包围框以及物体掩码，实现物体的提取部分功能。

2. 物体级语义 SLAM 算法。对数据集中的图片观测角度进行位姿估计，实时计算每一帧的位姿，并通过后端处理优化位姿结果。同时提取物体信息，在三维空间构建物体地图，最后将结果输出到文件中进行存储。

3. 结果展示。SLAM 算法在运行过程中会根据当前的位姿实时的输出已经构建的物体八叉树地图。

深度恢复模块通过对用户提供的 RGB 图像及其对应的深度图进行处理，能够有效恢复缺失或不准确的深度信息。处理完成后，系统将生成修复后的深度图，并将其保存至用户指定的存储位置。

文件管理模块确保用户在对数据集进行检测后，能够有效地管理和保存轨迹数据及物体八叉树地图。它支持用户查看和删除已有结果的操作。

5.2.3 数据库设计

系统利用 MySQL 数据库对用户文件相关信息进行存储。MySQL 是一种开源的关系型数据库管理系统，采用结构化查询语言来管理数据，支持数据的存储、检索、更新和删除等基本操作。MySQL 具有高性能、可靠性和易用性。

表 5-5 用户数据表

字段	类型	是否为空	键类型	默认值
UserID	INTEGER	否	PRIMARY KEY	NULL
UserName	VARCHAR(15)	否		NULL
Passwd	VARCHAR(20)	否		NULL
Email	VARCHAR(25)	是		NULL
Telephone	VARCHAR(11)	是		NULL

表5-5为用户数据表，用于存储用户账户信息及个人资料。表中的 UserID 由系统根据注册顺序自动生成。UserName 为用户自行设定的账号名称，并且在全表范围内唯一，不允许多个用户使用相同的账号名。Passwd 字段用于存储加密后的用户密码，在确保可以验证用户登录信息的同时，有效保护用户的隐私。最后，Email 和 Telephone 字段存储用户的个人信息，这两项为可选内容，用户可根据需求决定是否填写。

表 5-6 文件管理表

字段	类型	是否为空	键类型	默认值
UserID	INTEGER	否	FOREIGN KEY	NULL
FileID	INTEGER	否	PRIMARY KEY	NULL
Path	VARCHAR(50)	否		NULL
FileType	VARCHAR(10)	否		NULL

表5-6为文件管理表，用于存储用户的结果文件信息。UserID 为外键，关联至用户 ID。FileID 为系统自动生成的文件标识符，用于记录文件的唯一标识。Path

字段记录文件的存储路径，便于快速定位文件位置；FileType 则用于标明文件的类型，通常轨迹文件为 txt 格式，而八叉树结果文件则为 ASCII 格式。

表 5-7 日志表

字段	类型	是否为空	键类型	默认值
UserID	INTEGER	否	FOREIGN KEY	NULL
LogID	INTEGER	否	PRIMARY KEY	NULL
Operation	VARCHAR(50)	否		NULL
Time	VARCHAR(10)	否		NULL

表5-7是日志表，用于记录用户的操作，方便维护和查找问题。UserID 项为外键，关联进行操作的用户 ID。LogID 项是系统按照操作顺序自动生成的标识符，Operation 是字符串，用于存储用户的操作行为。Time 记录用户该操作执行的时间。

5.3 详细设计

5.3.1 用户管理模块

用户注册：在注册页面，用户填写个人信息，如姓名、邮箱、电话等，并设置密码以确保账号安全。信息填写完成后，用户点击确认，系统将处理用户信息并分配相应权限。具体实现流程如图5-4所示。

用户登录：在登录页面，用户输入用户名和密码，并点击登录按钮。系统验证用户信息，确认用户名和密码是否匹配，完成身份验证。验证通过后，用户将获得相应权限，能够执行相应的操作。

5.3.2 SLAM 功能模块

SLAM 算法功能模块流程图如图5-5所示，详细描述了用户在使用 SLAM 系统时的操作步骤。

首先，用户在登录页面完成身份验证后，进入算法检测页面。在该页面，用户可选择需要检测的数据集，点击确认按钮后，所选数据集将被投入到 SLAM 算法中进行处理。数据集中的每一帧图像将依次传入算法，进行位姿估计、物体提取与管理等操作。处理结果将返回至前端进行展示，物体将通过占据栅格的形式显示，并标明其对应类别，以便用户清晰地观察与分析结果。

5.3.3 深度恢复模块

深度恢复模块流程图如图5-6所示，详细描述了用户在使用深度恢复功能是的操作步骤。首先，用户在算法检测页面上，选择需要进行深度恢复的单张图像或者

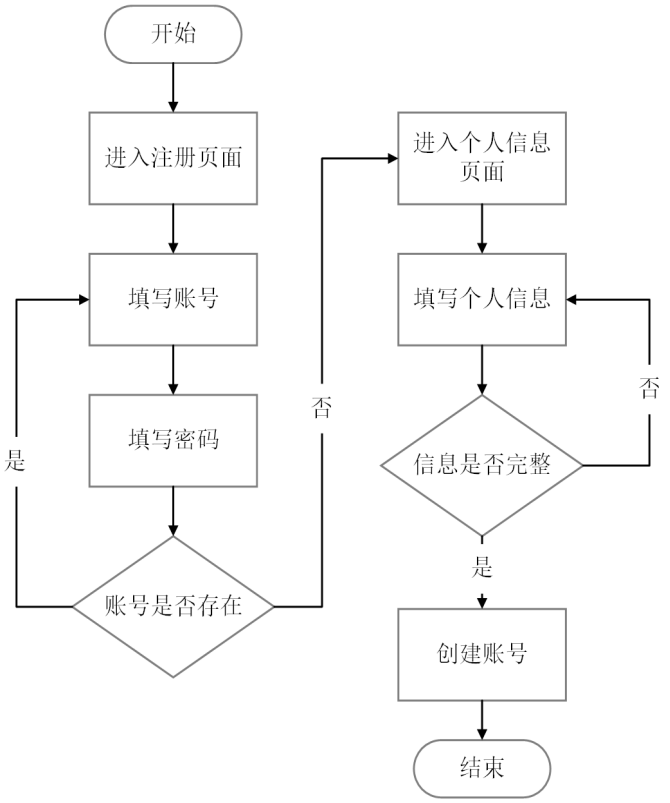


图 5-4 用户注册流程图

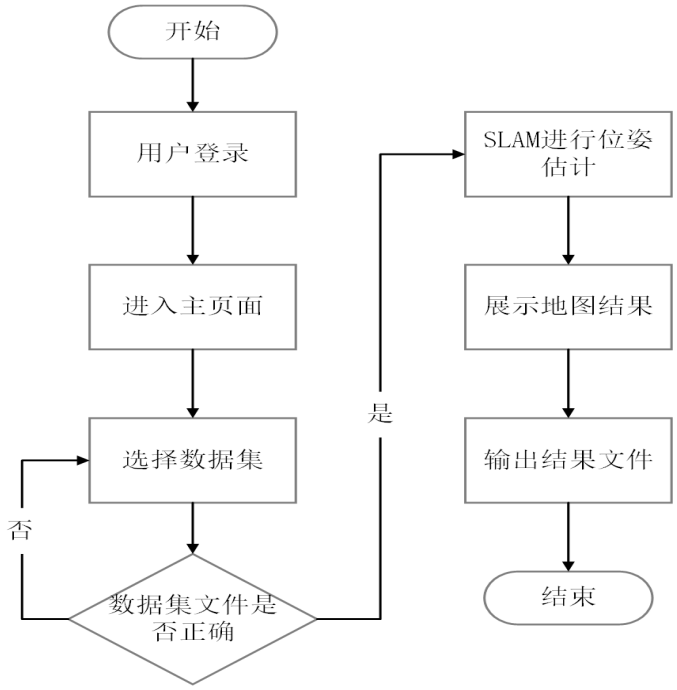


图 5-5 SLAM 算法流程图

数据集，点击确认按钮后，所选的内容会被检测是否是 RGB 图像以及对齐的深度图像。如果满足输入要求，则对图像进行深度恢复的处理，最终输出恢复后对应的深度图像。

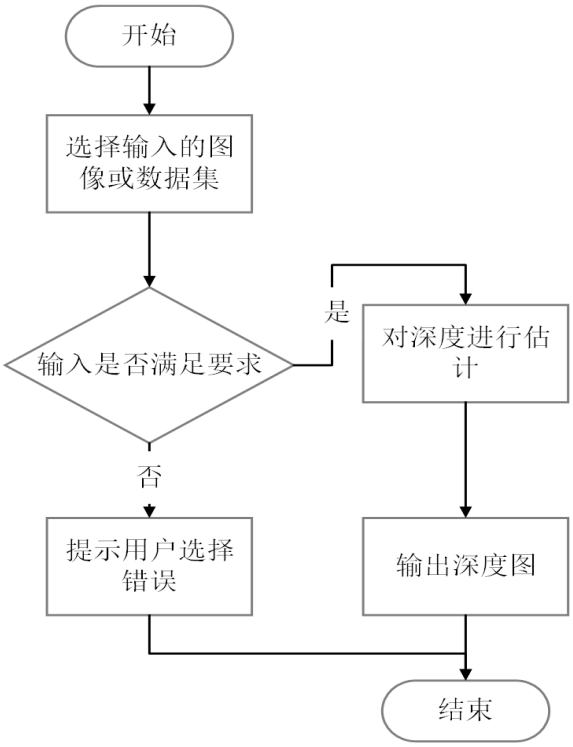


图 5-6 深度恢复流程

5.3.4 文件管理模块

SLAM 算法系统集成了文件管理模块，允许用户对结果进行二次查看、删除等操作。结果文件包括轨迹文件和物体地图文件。在使用该模块时，用户需根据登录信息浏览与管理相应文件。若用户不再需要某些结果文件，可轻松进行删除操作。整个流程简便，并能够高效处理文件。

5.4 系统各模块实现

5.4.1 开发技术与环境

物体级语义 SLAM 算法系统是一个基于深度学习网络结合 SLAM 算法设计的系统，用于在未知的环境中估计自身的轨迹并构建环境中的各类物体。为了开发和部署该系统，需要选择合适的开发工具和技术。以下是这些工具和技术的详细说明：

1. 开发语言 C++：

C++ 是一种高效的编译型语言，广泛应用于需要高性能和低延迟的系统开发，适合 SLAM 算法的实现。SLAM 算法开发以及对 YOLOv8 神经网络的调用都是使用 C++ 来完成的。

2. 数据库 MySQL：

MySQL 是一种广泛使用的开源关系型数据库管理系统，具有高性能、可靠性和灵活性。它支持复杂的查询和事务处理，具备强大的数据完整性和安全性。SLAM 系统使用 MySQL 数据库进行文件和用户信息的管理能确保高效管理数据。

3. 前端框架 Qt：

Qt 是一个跨平台的应用程序开发框架，支持 C++ 和 QML 语言，能够构建高性能、响应迅速的图形用户界面。其强大的图形渲染能力、丰富的控件库及跨平台特性，使得开发者能够轻松创建在 Windows、Linux 和 macOS 等多个平台上运行的应用程序。本文使用 Qt 进行前端展示，能提搞开发效率，提升用户体验。

4. 集成开发环境 VS Code：

VS Code 是一款轻量级且高度可定制的集成开发环境，支持多种编程语言和插件扩展。其具有丰富的社区插件和简洁的界面，并且拥有代码补全、差错检查等功能。在开发过程中主要使用 VS code 进行代码的开发管理，有效提高了开发效率。

本文通过使用 C++、MySQL、Qt、VS Code 等技术与工具，成功搭建了一个高效的物体级语义 SLAM 算法系统。这些工具在开发过程中极大地简化了操作，降低了开发与管理的难度，确保了代码质量并提升了开发效率。通过不断的优化与调整，系统功能得以完善，呈现出更加便捷与高效的成果。

5.4.2 用户管理模块实现

用户需先在注册界面完成账号注册，填写信息后，在登录界面进行登录，随后即可开始使用语义 SLAM 算法系统。前端界面采用 Qt 进行设计与开发。

在用户未登录之前，所有功能均被禁用，只有在用户成功登录后，系统后端才会根据用户权限解锁相应的功能。例如，只有 root 用户才有权限对其他用户进行操作。用户登录时，系统会将用户输入的账号和密码提交给后端进行验证，与数据库中存储的加密信息进行对比。若验证通过，系统将返回登录成功的提示；若验证失败，则提示账号或密码错误，并要求用户重新输入。

在算法5-1中，所有按钮都被设置了 Unable 的属性，表示其不可用，在界面上的展示结果为按钮变灰，用户不可点击，只有在用户进行登录之后才会根据用户可以进行的权限操作将对应操作按钮属性设置成 Enable，用户才能点击这个按钮。

5.4.3 SLAM 功能模块实现

如图5-8所示，5-8(a)为用户选择数据集时弹出的对话框展示，5-8(b)则呈现了一个八叉树地图。系统生成的位姿估计文件将自动以 txt 格式保存，并存入对应用户的数据库中。

图 5-7 系统登录

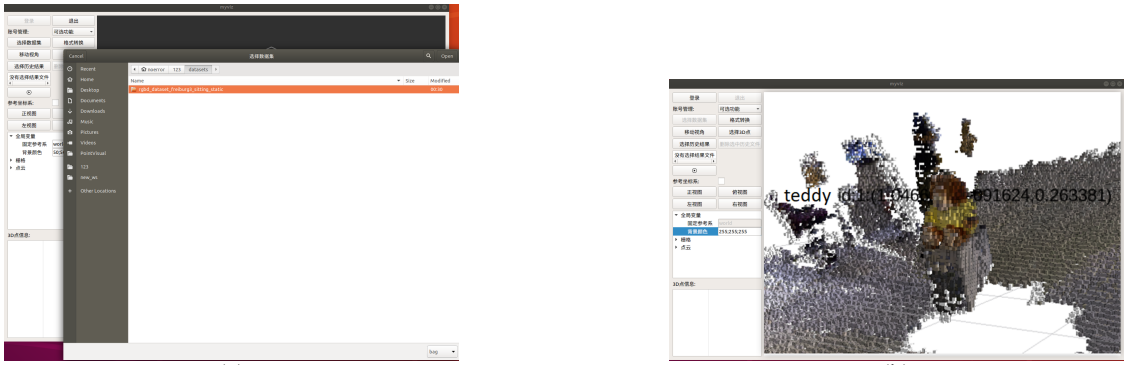
算法 5-1: 用户管理功能模块伪码

输入: 1) 账号;
密码。
输出: 进入页面

```
1  connect(but_login, &QPushButton::clicked, this, &AglSLAM::Login);
2  Login
   if !UserNameempty() & !PasswdEmpty() then
3      getUserName();
4      getPasswd();                                  /* 获取用户输入 */
5      if VerifyUser() then                          /* 验证用户信息 */
6          EnableButton();                           /* 解锁按钮 */
7      else
8          QMessageBox(" 账号密码错误");
9  else
10     QMessageBox(" 账号密码错误");
11 return
```

用户首先需在主页面选择所需测试的数据集，随后将数据集文件信息传输至后端进行验证，以确认数据集文件是否符合 SLAM 算法的使用标准。若验证成功，系统将启动 SLAM 算法流程；若验证失败，则会提示用户数据集错误，并要求重新选择合适的数据集。最终，系统界面右侧将展示构建完成的八叉树地图。

在算法5-2中，在使用 SLAM 算法模块时，系统利用了 QMessageBox 类型的弹



(a) (b)

图 5-8 SLAM 算法模块展示。(a)数据集选择；(b)八叉树地图展示

算法 5-2: SLAM 功能模块伪码

输入: 1) 数据集文件。
输出: 1) 轨迹文件；
2) 地图文件

```
connect(but_play, &QPushButton::clicked, this, &AglSLAM::startSLAM);
startSLAM
if CheckFile() then                                  /* 检测数据集文件 */
    exec(SLAM);                                      /* 开始 SLAM 算法 */
    ReceiveTopic();                                  /* 接收地图信息 */
else
    QMessageBox(" 数据集文件错误");                    /* 提示文件错误息 */
if SLAMend then
    SaveTajectory();                                 /* 保存轨迹文件 */
    SavePoint();                                     /* 保存地图文件 */
    SavePth2Sql()                                    /* 保存路径到数据库 */
return
```

窗，这种弹窗通常用于向用户显示信息、警告、错误或提示，并允许用户与之交互（如点击按钮）。这使得它非常适合用于提示用户当前系统状态或出现的异常情况。系统通过接收 SLAM 模块通过 ROS 话题（Topic）发布的八叉树地图消息来进行地图的展示。其中，ROS（Robot Operating System）是一个常用于机器人开发的框架，话题是 ROS 中传递消息的机制。通过这些地图点数据，系统能够实时展示和更新地图。最后，在 SLAM 结束后，系统会将生成的地图结果和轨迹结果保存到文件中，并将文件路径存储到数据库中。

5.4.4 深度恢复模块实现

如图5-9所示，系统对深度恢复结果图像进行显示，方便用户对结果进行查看。最终系统会将深度恢复的图像自动保存，并将文件路径等信息加入到数据库中。

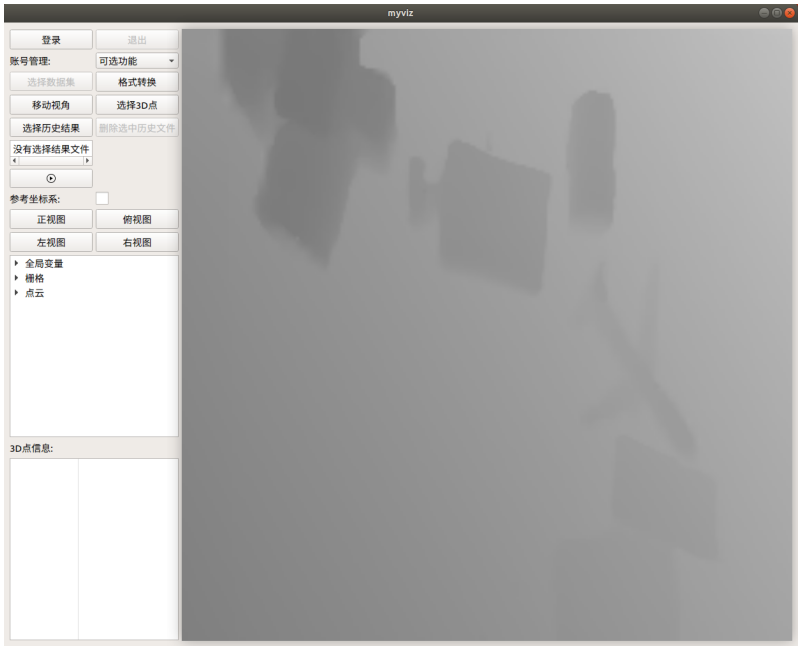


图 5-9 深度恢复模块展示

用户首先需要在系统弹出的文件目录中选择其数据集或图像文件。选定文件后，系统将检查文件是否符合深度恢复模块的输入要求。如果验证失败，会弹出提示窗口提醒用户；如果验证通过，系统将继续执行深度恢复处理。最终，处理后的深度恢复图像将展示在系统界面的右侧。

算法 5-3: SLAM 功能模块伪码

输入: 1) 数据集文件。
输出: 1) 恢复后的深度图

```
1 startDepthRestoration
  if CheckFile() then                                  /* 检测数据集文件 */
2     exec(DepthRestoration);                          /* 开始深度恢复算法 */
3 else
4     QMessageBox(" 文件选择错误");                     /* 提示文件错误息 */
5 if DepthRestorationEnd then
6     SaveDepthImage();                                /* 保存恢复后的深度图 */
      ShowFirstDepthImage();                           /* 展示第一张深度图 */
      SavePth2Sql()                                    /* 保存路径到数据库 */
7 return
```

5.4.5 文件管理模块实现

文件管理模块主要是对用户所使用的算法结果进行存储管理，后端连接到MySQL 数据库中，这个数据库存储对应用户账号以及结果文件路径，通过查询用户账号 ID 返回文件路径，最后在文件系统中展示文件。如图5-10所示。

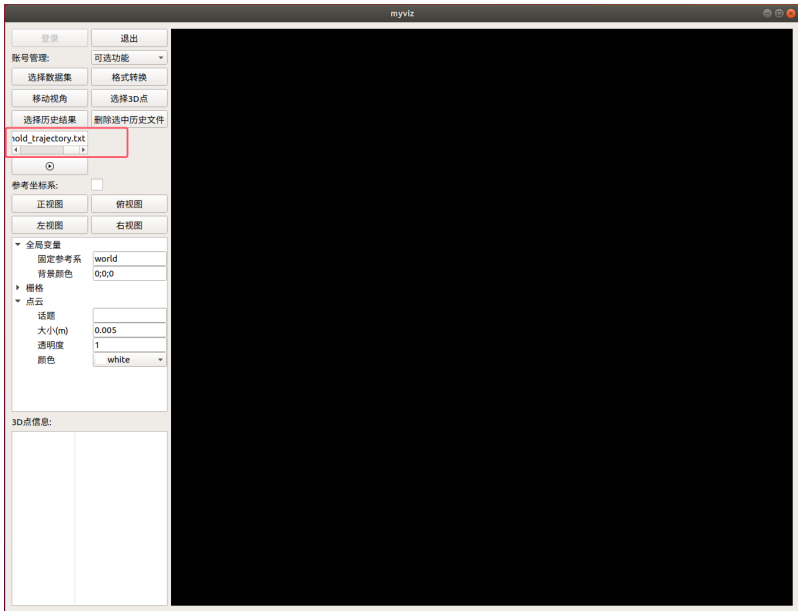


图 5-10 删除文件

如图5-10所示，当用户选择文件后，文件路径将显示在下方的文本框中，用户可以查看该路径，确认是否为需要删除的文件。确认无误后，点击“删除选中历史文件”按钮，即可删除所选文件。

在算法5-4中，系统首先通过查询数据库中当前用户的结果文件路径来获取相关信息，并将查询结果返回给用户。系统使用了 QFileDialog 窗口，来实现展示文件。QFileDialog 是 Qt 框架中用于文件选择的标准控件。通过 QFileDialog，用户可以轻松选择一个文件，系统会根据选择的文件路径进行后续的处理。如果用户没有选择文件并直接返回到主页面，系统会弹出提示，告知用户没有选择文件。这时，为了避免误操作，删除按钮会被禁用，防止用户在没有选择文件的情况下执行删除操作。当用户选择了历史文件后，系统会使用 QTextEdit 控件来展示所选文件的路径。这个控件属性被设置为不可修改和横向滚动，这样用户可以方便地查看已经选择的文件路径，同时确保文件路径不被误修改。通过这种方式，用户可以清楚地看到自己的选择并避免意外编辑。在用户完成操作后，系统会更新数据库。比如，如果用户决定删除某个文件，系统会删除文件本身，并在数据库中删除与该文件对应的记录。这种方式保证了数据的一致性，并防止了数据冗余，确保数据库中的记录与实际文件系统保持同步。

算法 5-4: 文件管理模块伪码

输入: 文件选择。
输出: 操作结果。

```
connect(but_selectFile, &QPushButton::clicked, this, &AglSLAM::selectFile);
selectFile
SelectInSql();                                   /* 数据库中查找用户文件 */
QFileDialog();                                   /* 显示文件 */
if 用户选择文件 then
    QTextEdit(" 选择文件路径");                   /* 展示选择的文件路径 */
else
    QMessageBox(" 未选择历史文件");
if 用户选择文件 then
    but_delFile(enable);                         /* 可使用删除文件按钮 */
else
    but_delFile(unable);                         /* 不可使用删除文件按钮 */
if Operation() then                              /* 用户进行操作 */
    Update2Sql()                                 /* 将操作结果更新到数据库 */
return
```

5.5 系统测试

5.5.1 测试环境

本系统的测试工作在 Linux 操作系统环境下进行，并采用本地部署的 MySQL 数据库进行用户相关信息的存储。系统开发使用了 Python、Qt 和 C++ 等编程语言和开发框架。具体配置参数详见表5-8所示。

表 5-8 系统性能测试

环境配置	型号和参数
CPU	IntelR Xeon(R) E5-2650 v4
GPU	GeForce GTX 1080 8GB
操作系统	Ubuntu 18.04
数据库	MySQL
开发工具	VS Code

5.5.2 功能性测试

将整个系统视作黑盒进行测试，模拟用户操作系统中的各个功能模块，全面检验各项功能和模块的表现，确保其运行结果符合预期。

1. 用户管理模块测试

模拟用户进行注册流程，对个人信息、账号密码等信息的验证，保证用户在正确的情况下注册账号。用户管理模块测试用例如表5-9所示。

表 5-9 用户账号注册测试用例

测试功能		账号注册		
测试目的		用户注册功能正常		
用例 ID	用例描述	输入	预期结果	实际结果
1	输入有效注册信息	username=’NoError’ email=’123@uestc.com’ passwd=’123456’ checkpasswd=’123456’	成功	符合预期
2	邮箱未填	username=’NoError email=” passwd=’123456’ checkpasswd=’123456’	成功	符合预期
3	两次密码前后不一致	username=’NoError email=’123@uestc.com’ passwd=’123456’ checkpasswd=’123456789’	失败	符合预期

2.SLAM 算法模块测试

SLAM 算法模块测试输入是否正常上传数据集，检测系统是否正常返回预期结果。算法模块测试用例如表5-10所示。

表 5-10 SLAM 算法模块测试用例

测试功能		SLAM 算法功能		
测试目的		算法模块功能正常		
用例 ID	用例描述	输入	预期结果	实际结果
1	选择正确数据集	在打开的文件系统中选择正确的数据集并确认	显示地图并保存结果文件	符合预期
2	选择非数据集文件	在打开的文件系统中选择非数据集文件并确认	提示用户数据集错误	符合预期

3. 深度恢复模块测试

深度恢复模块测试输入的数据集是否正确，检测系统是否展示预期结果。深度恢复模块测试用例如表5-11所示。

表 5-11 用户账号注册测试用例

测试功能		深度恢复模块功能		
测试目的		模块功能正常		
用例 ID	用例描述	输入	预期结果	实际结果
1	选择正确数据集	在打开的文件系统中选择正确的数据集并确认	保存深度图并展示第一张深度图像	符合预期
2	选择非数据集文件	在打开的文件系统中选择非数据集文件并确认	提示用户数据不符合输入	符合预期

4. 文件管理模块测试

对文件管理模块的各项功能进行测试，主要包括用户查看和删除文件的操作。文件管理系统的测试用例见表5-12。

表 5-12 用户账号注册测试用例

测试功能		SLAM 算法功能		
测试目的		算法模块功能正常		
用例 ID	用例描述	输入	预期结果	实际结果
1	用户查看历史文件	在打开的文件系统中选择历史结果文件	成功显示文件内容	符合预期
2	用户删除历史文件	在打开的文件系统中选择结果文件并确认删除	成功删除文件	符合预期
3	用户保存当前结果	用户选择保存并对文件命名点击确认	成功保存结果	符合预期

5.5.3 性能测试

对整个系统的性能进行全方面的评估，主要指标是系统的运行时间。如表5-13所示。具体内容包括选择文件后系统启动到系统得出结果时间。

5.6 本章小结

本章在前两章提出的物体级语义 SLAM 算法基础上，开发了一个完整的 SLAM 系统。首先，进行了需求分析，并通过用例图展示了系统的三个核心模块：用户管理、SLAM 算法和文件管理模块。随后，进行了总体设计，明确了系统架

表 5-13 系统性能测试

编号	测试功能	输入描述	测试结果
1	系统启动	算法系统启动	0.5s
2	用户登录	用户输入账号密码点击登录	0.1s
3	文件选择加载	选择数据集到算法启动	5s
4	SLAM 算法检测	数据集文件	30fps
5	文件查看	用户查看历史文目录	0.2s

构，并对各功能模块进行了详细设计，包括用户管理、SLAM 算法检测以及文件管理的整体流程图。随后，介绍了开发过程中所使用的工具和技术，并展示了系统的具体实现过程。最后，通过功能性测试对系统进行了验证，并通过各模块的运行时间进行了定量性能测试。

第六章 总结与展望

6.1 本文工作总结

物体级语义 SLAM 是自动驾驶和机器人领域中的一项重要技术，它将传统的 SLAM 技术与语义信息相结合，以实现对环境的更高层次理解。通过语义标注，物体级语义 SLAM 不仅能构建环境地图，还能识别和定位场景中的具体物体，如车道线、路标、行人、车辆等。与传统的 SLAM 方法不同，物体级语义 SLAM 不仅关注位姿估计和地图构建，还考虑了对象的语义信息，这使得系统能够对复杂环境中的物体进行实时感知和追踪。通过深度学习和计算机视觉技术，物体级语义 SLAM 可以提高系统的鲁棒性，适应各种复杂环境，如动态障碍物、变化的天气条件等，从而为自动驾驶系统提供更加智能和精确的决策支持。

基于上述研究背景，本文结合 YOLOv8 实例分割网络，提出了一种全新的物体级语义 SLAM 算法。该算法能够有效估计轨迹，并同时构建精确的三维物体地图，从而提供丰富的环境语义信息。本文的主要工作如下：

1. 针对 YOLOv8 识别网络的不足，设计了进一步的处理算法。由于 YOLOv8 可能出现过度分割和误识别等问题，本文根据这些情况提出了物体掩码融合机制和物体剔除机制，确保所构建的物体模型准确无误，避免出现错误构建物体或将一个物体误构建为两个的情况。此外，结合深度图信息，对掩码结果进行了进一步处理，消除误检测点，并依据深度一致性原理对物体边缘部分进行了精细化修正。

2. 基于上述提取的物体信息，构建全局数据库，动态跟新物体信息。设计物体关联与更新算法，该算法通过物体的类别和距离信息判断两个物体是否为同一物体的不同观测视角。同时，利用颜色信息作为补充判断依据，进一步提高识别的准确性。在此基础上，对于匹配成功的物体进行融合操作，从而提升物体的表征精度和系统的整体感知能力。

3. 利用物体信息对 SLAM 系统进行优化处理。首先，基于物体信息构建局部地图，并通过物体地图的相似性检测回环，从而增强系统对回环的识别能力。回环检测后的信息可用于优化先前的轨迹估计，有效减小累积误差。其次，筛选稳定的物体并将其纳入 SLAM 系统的后端优化过程中，这有助于提高关键帧的位姿估计精度。通过整合更多的环境信息，优化后的系统能够更精确地重建环境，并提升整体定位和建图性能。

4. 针对 RGB-D 相机在探测反光和透明物体时存在的深度感知问题，优化改进了深度恢复网络。通过对深度图中难以准确检测的反光和透明物体进行深度估计

与补全，确保系统能够获得更完整的深度信息。这些优化后的深度数据被传递到SLAM 系统，有助于提高物体重建的准确性，从而增强系统对复杂环境的适应能力和感知精度。

5. 对于本文提出的语义 SLAM 算法，设计了一个完整的 SLAM 系统，该系统包括三个主要模块：用户管理模块、SLAM 算法模块和文件管理模块。用户管理模块主要用于处理用户信息的注册与登录验证。SLAM 算法模块用于对数据集进行轨迹估计、物体地图展示以及输出结果文件。文件管理模块用于存储用户信息、历史结果文件和其他相关数据。用户可以通过该模块查看、管理、删除历史结果文件，确保数据的便捷存取与管理。

6.2 未来工作展望

本文针对室内静态场景设计了物体级语义 SLAM 算法系统，并在 TUM 数据集测试中相较于近期的一些先进 SLAM 方法取得了一定的进展。综上所述，本文的工作虽已取得一定成果，然而仍有以下几点值得改进：

1. 尽管本文提出的算法在精度上有所提升，但使用了 YOLOv8 识别网络以及改进的深度恢复网络，需要大量的计算资源，尤其是在大规模场景中。未来需要优化算法的计算效率，提升系统的实时性，以便能够适应自动驾驶、机器人等应用中的实时要求。

2. 物体识别与融合策略：当前物体识别和匹配主要依赖于类别、距离和颜色信息的简单关联。未来可以探索更多高级的物体特征（如纹理、形状等）和深度学习技术，提高物体匹配的准确性与鲁棒性，尤其是在物体遮挡或变化较大的场景中。

3. 本文的研究主要聚焦于室内静态场景下的语义 SLAM，尽管算法在 TUM 数据集上取得了较为理想的结果，但由于实验条件的限制，未能在实际场景中进行验证。因此，未来可在真实的室内场景中进行应用实现，并根据实际结果对算法进行进一步优化改进。

致　谢

在此论文即将完成之际，回望三年的研究生生涯，时光飞逝，感慨万千。这段旅程让我成长、让我蜕变。经历过无数的挑战与磨砺，也收获了宝贵的经验与友谊。

首先，我要感谢我的导师 XXX。感谢您的悉心指导与无私奉献，您严谨的学术态度、丰富的专业知识和严密的逻辑思维让我受益匪浅。在论文的选题、修改过程中，您总是耐心指导，提出宝贵的意见，帮助我不断完善。您的教诲将是我人生中重要的财富。

其次，我要感谢我的家人。感谢你们二十多年来的支持与鼓励，正是因为你们无私的付出，我才能无后顾之忧地专注于学业，成长为今天的我。无论未来如何，我都会心怀感恩，继续前行。

我要特别感谢我的朋友们，感谢你们三年来的陪伴与支持，和你们一起度过的每一刻都让我充实且难忘。无论未来我们身在何方，愿你们都能前程似锦，勇敢追梦。

此外，我要特别感谢我的师兄。感谢您在论文修改和研究过程中的悉心指导与耐心帮助。在我遇到难题时，您总是给予我宝贵的建议和鼓励，帮助我克服了许多困难。您的无私分享和经验传授让我受益匪浅。

最后，感谢学院的各位老师，尤其是我的辅导员，感谢你们在这三年里给予我的帮助与支持。无论是学术上的指导，还是生活中的关怀，您们的悉心教诲都让我受益匪浅。

尽管时光匆匆，告别在即，但我相信，未来我们会再度相见，愿大家各自安好，前程似锦。

参考文献

[1] Khairuddin A R, Talib M S, Haron H. Review on simultaneous localization and mapping (slam)[C]. 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2015: 85-90.

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

[3] 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.

[4] Qin T, Li P, Shen S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE transactions on robotics, 2018, 34(4): 1004–1020.

[5] Qin T, Pan J, Cao S, et al. A general optimization-based framework for local odometry estimation with multiple sensors. arxiv preprint[J]. arXiv preprint arXiv:1901.03638, 2019.

[6] Song S, Lim H, Lee A J, et al. Dynavins: A visual-inertial slam for dynamic environments[J]. IEEE Robotics and Automation Letters, 2022, 7(4): 11523–11530.

[7] 夏琳琳, 沈冉, 迟德儒, et al. 一种基于光流-线特征的单目视觉-惯性 slam 算法 [J]. 中国惯性技术学报, 2020, 28(05): 568-575.

[8] 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.

[9] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 779-788.

[10] Xia L, Cui J, Shen R, et al. A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots[J]. International Journal of Advanced Robotic Systems, 2020, 17(3): 1729881420919185.

[11] Cadena C, Carlone L, Carrillo H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age[J]. IEEE Transactions on Robotics, 2016, 32(6): 1309-1332.

[12] Fujii K. Extended kalman filter[J]. Refernce Manual, 2013, 14: 41.

[13] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part i[J]. IEEE Robotics & Automation Magazine, 2006, 13(2): 99-110.

[14] Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (slam): Part ii[J]. IEEE robotics & automation magazine, 2006, 13(3): 108–117.

[15] Dissanayake G, Huang S, Wang Z, et al. A review of recent developments in simultaneous localization and mapping[C]. 2011 6th International Conference on Industrial and Information Systems, IEEE, 2011: 477–482.

[16] Macario Barros A, Michel M, Moline Y, et al. A comprehensive survey of visual slam algorithms[J]. Robotics, 2022, 11(1): 24.

[17] 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.

[18] Klein G, Murray D. Parallel tracking and mapping for small ar workspaces[C]. 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007: 225-234.

[19] Newcombe R A, Lovegrove S J, Davison A J. Dtam: Dense tracking and mapping in real-time[J]. IEEE, 2011.

[20] 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.

[21] 王劭靖. 基于 orbslam2 改进的单线程双目 slam 系统 [J]. 智能计算机与应用, 2023, 13(01): 84-90+99.

[22] 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.

[23] Kaveti P, Nir J S, Singh H. Towards robust vslam in dynamic environments: a light field approach[C]. 2021 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI), IEEE, 2021: 1–8.

[24] 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.

[25] Zou D, Tan P. Coslam: Collaborative visual slam in dynamic environments[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 35(2): 354–366.

[26] Kim D H, Kim J H. Effective background model-based rgb-d dense visual odometry in a dynamic environment[J]. IEEE Transactions on Robotics, 2016, 32(6): 1565–1573.

[27] Esparza D, Flores G. The stdyn-slam: A stereo vision and semantic segmentation approach for vslam in dynamic outdoor environments[J]. IEEE Access, 2022, 10: 18201-18209.

[28] Wu W, Guo L, Gao H, et al. Yolo-slam: A semantic slam system towards dynamic environment with geometric constraint[J]. Neural Computing and Applications, 2022: 1–16.

[29] 黄晓涛, 冯桑, 张宁, et al. 一种基于 orb-slam2 的动态视觉 slam 方法 [J]. 机械工程与自动化, 2025, 54(01): 20-24.

[30] 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.

[31] 陈春朝, 刘雪飞, 李恒宇, et al. 动态环境下基于语义信息和多视图几何的视觉 slam 算法研究 [J]. 中国测试: 1-11.

[32] 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.

[33] Rünz M, Agapito L. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects[C]. 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017: 4471–4478.

[34] 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.

[35] Runz M, Buffier M, Agapito L. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects[C]. 2018 IEEE international symposium on mixed and augmented reality (ISMAR), IEEE, 2018: 10–20.

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

[37] Nicholson L, Milford M, Sünderhauf N. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam[J]. IEEE Robotics and Automation Letters, 2018, 4(1): 1–8.

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

[39] Zins M, Simon G, Berger M O. Oa-slam: Leveraging objects for camera relocalization in visual slam[C]. 2022 IEEE international symposium on mixed and augmented reality (ISMAR), IEEE, 2022: 720–728.

[40] 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.

[41] 白素琴, 诸皓伟, 吕宗磊, et al. 一种椭球模型表示的对象级动态语义 slam 方法 [J]. 中国惯性技术学报, 2025, 33(01): 46-54.

[42] Jocher G, Chaurasia A, Qiu J. Ultralytics yolov8, 2023. https://github.com/ultralytics/ultralytics.

[43] 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.

[44] 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.

[45] 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.

[46] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features[C]. Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9, Springer, 2006: 404–417.

[47] 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.

[48] Meagher D. Octree encoding: A new technique for the representation[J]. Manipulation and Display of Arbitrary, 1980.

[49] 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.

[50] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]. International conference on machine learning, PmLR, 2021: 8748–8763.