

A Dissertation Submitted for the Degree of Bachelor

**Design and Implementation of SLAM System Based on
YOLOv8 Optimization**

By

Guan Bin

Hefei University of Technology

Hefei, Anhui, P.R.China

May, 2025

毕业设计（论文）独创性声明

本人郑重声明：所呈交的毕业设计（论文）是本人在指导教师指导下进行独立研究工作所取得的成果。据我所知，除了文中特别加以标注和致谢的内容外，设计（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 合肥工业大学 或其他教育机构的学位或证书而使用过的材料。对本文成果做出贡献的个人和集体，本人已在设计（论文）中作了明确的说明，并表示谢意。

毕业设计（论文）中表达的观点纯属作者本人观点，与合肥工业大学无关。



毕业设计（论文）作者签名：

签字日期：2024 年 5 月 28 日

毕业设计（论文）版权使用授权书

本学位论文作者完全了解 合肥工业大学 有关保留、使用毕业设计（论文）的规定，即：除保密期内的涉密设计（论文）外，学校有权保存并向国家有关部门或机构递交设计（论文）的复印件和电子光盘，允许设计（论文）被查阅或借阅。本人授权 合肥工业大学 可以将本毕业设计（论文）的全部或部分内容编入有关数据库，允许采用影印、缩印或扫描等复制手段保存、汇编毕业设计（论文）。

（保密的毕业设计（论文）在解密后适用本授权书）



学位论文作者签名：



指导教师签名：

签名日期：2024 年 5 月 28 日

签名日期：2024 年 5 月 28 日

摘要

同步定位与建图（Simultaneous Localization and Mapping, SLAM）是一项让机器人在未知环境中获得自主感知和自主导航能力的关键技术。SLAM 可以为未来智能交通、智慧物流和自动驾驶等领域的发展提供重要支持。当前基于传统人工特征点法的视觉 SLAM 技术虽然取得了不少的成就，但是基于传统人工特征的特征提取方法在光照、季节和其它变化复杂的场景下，会出现无法正确提取足够特征点或者匹配准确率低的问题。这一问题会导致视觉 SLAM 系统在复杂多变的环境下无法进行位姿估计与回环检测。随着深度学习技术研究的发展，当前基于卷积神经网络（Convolutional Neural Networks, CNN）的算法可以很好地解决人工特征提取方法在变化复杂场景下存在的问题。YOLO 在目标检测和语义分割领域取得了相当不错的成就，可以用于优化 SLAM 在动态场景下特征点提取的过程中动态特征点的剔除，以优化位姿估计与地图构建。本文实现了一个基于 YOLO 优化的 SLAM 系统，主要工作如下：(1) 实现训练并测试了一个基于 YOLOv8 的实例分割模型。(2) 利用多线程的思想，在基于视觉特征点提取的 ORB-SLAM3 系统的基础上，运用 YOLOv8 模型检测场景中的动态对象，并在视觉里程计中对动态特征点进行剔除。

代码已开源上传至 Github 仓库 (<https://github.com/Glencsa/YOLOv8-ORB-SLAM3/>)。

关键词：视觉 SLAM；特征提取；目标检测；场景优化

ABSTRACT

Simultaneous Localization and Mapping (SLAM) is a crucial technology for enabling autonomous perception and navigation in robots operating in unfamiliar environments. This technology holds significant potential for advancing intelligent transportation, logistics, and autonomous driving. Despite progress in vision-based SLAM methods that rely on traditional artificial feature points, these methods often struggle in complex and variable settings. Traditional feature extraction techniques face difficulties in accurately detecting and matching features under changing lighting conditions and seasonal variations, which hampers the SLAM system's ability to estimate positions accurately and detect loop closures in dynamic and evolving environments. The emergence of deep learning, particularly Convolutional Neural Networks (CNNs), presents a promising solution to the limitations of manual feature extraction in dynamic and intricate scenarios. YOLO, known for its exceptional performance in object detection and semantic segmentation, is a strong candidate for optimizing feature extraction in SLAM systems within dynamic environments. Building on YOLO's success, this dissertation proposes integrating YOLO-based optimization into SLAM systems to enhance feature point extraction, thereby improving position estimation and map construction in dynamic scenes. The proposed SLAM system, enhanced by YOLO optimization, includes the following key components: (1) Development of a YOLOv8-based instance segmentation model for training and evaluation. (2) Implementation of multi-threading techniques to integrate the YOLOv8 model with the ORB-SLAM3 system, enabling the detection and rejection of dynamic objects during visual odometry through feature point extraction. The code for this system has been open-sourced and is available in the GitHub repository (<https://github.com/Glencsa/YOLOv8-ORB-SLAM3/>).

KEYWORDS: Visual SLAM; Feature Extraction; Target Detection; Scene Optimization

目 录

1 绪论	1
1.1 研究工作的背景与意义	1
1.2 国内外研究历史与现状	2
1.2.1 特征点提取和描述符研究现状	4
1.2.2 基于 YOLO 的目标检测和语义分割发展现状	5
1.2.3 结合深度学习的视觉 SLAM 算法	6
1.3 本文的主要贡献与工作	6
1.4 论文的结构安排	7
2 视觉 SLAM 系统及 YOLO 目标检测基础	9
2.1 引言	9
2.2 YOLOv8 基本原理	9
2.3 视觉 SLAM 系统基本原理	12
2.3.1 视觉里程计	13
2.3.2 回环检测与建图	18
3 基于 YOLOv8 的实例分割研究	19
3.1 数据集与模型的训练	19
3.1.1 数据集	19
3.1.2 模型训练	20
3.2 模型的部署	23
4 动态场景下基于 YOLOv8 优化的视觉 SLAM 系统设计	24
4.1 视觉 SLAM 系统的选择	24
4.2 多线程设计	25
4.3 数据集测试及分析	27
4.4 真实场景测试与应用	31
5 全文总结与展望	33
5.1 全文总结	33
5.2 工作展望	33
参考文献	35
致谢	39

插图清单

图 1.1	视觉 SLAM 系统流程图	2
图 1.2	YOLO 发展时间线	5
图 1.3	YOLO-ORB-SLAM3 框架结构	7
图 2.1	YOLOv1 框架图	10
图 2.2	C2f 模块和 C3 模块示意图	11
图 2.3	YOLOv8 网络框架	12
图 2.4	V-SLAM 框架图	13
图 2.5	四种坐标系间的转换关系	14
图 2.6	针孔相机模型	15
图 2.7	对极几何模型	17
图 3.1	BDD100K 数据集	19
图 3.2	实例分割结果图	21
图 3.3	前景 mask 结果图	22
图 3.4	YOLOv8n-seg 在 COCO 数据集上的收敛曲线	22
图 4.1	ORB-SLAM3 框架结构 ^[1]	25
图 4.2	算法总体框架图	26
图 4.3	单帧图像对比结果图	27
图 4.4	数据集 1 测试轨迹对比, 前两个 Figure 是优化前的轨迹, 后两个 Figure 是优化后的轨迹	28
图 4.5	数据集 1 相对轨迹误差图	29
图 4.6	数据集 2 测试轨迹对比, 前两个 Figure 是优化前的轨迹, 后两个 Figure 是优化后的轨迹	29
图 4.7	数据集 2 相对轨迹误差图	30
图 4.8	教室场景测试结果	31
图 4.9	教室稀疏点云地图	32

表格清单

表 2.1 相机类型对比.....	14
表 4.1 测试数据集结果对比 1.....	30
表 4.2 测试数据集结果对比 2.....	30

1 绪论

1.1 研究工作的背景与意义

SLAM 技术使移动机器人能够在未知环境中利用环境信息确定自身运动状态，实现定位并逐步构建环境地图。视觉 SLAM 采用相机作为外部传感器，通过采集环境图像信息来定位移动相机、机器人或车辆的位置，并对探索区域进行地图构建。这项技术是实现机器人自主运动和自动驾驶的重要技术。随着机器人和自动驾驶技术的发展，SLAM 技术引起了国内外学者的广泛关注。相比于视觉 SLAM 使用的相机设备，激光雷达设备不仅昂贵且不便携，因此视觉 SLAM 逐渐成为主流。视觉 SLAM 由于采用摄像头作为外部传感器，具备较高的实用性、便携性和经济性，适合用于车辆、移动机器人、无人机及其他移动设备。此外，相机能够捕捉更丰富的场景信息，在三维重建和语义地图建模方面具有天然优势，因此，视觉 SLAM 技术现已成为 SLAM 研究的主要方向。

传统的人工特征提取方法和描述符（如 SIFT、SURF 和 ORB）经过几十年的研究和演进，在计算机视觉领域取得了显著成果，广泛应用于目标检测和图像拼接等场景^[2-4]。这些方法在旋转、尺度和亮度变化方面具备一定的不变性，能够满足大多数计算机视觉需求。然而，在面对昼夜变化、季节变化等剧烈变化的场景时，传统方法显得力不从心。视觉 SLAM 需要在多变的环境中运行，因此需要设计能够在复杂环境下高效提取和匹配特征的方法和描述符。

经过近 20 年的发展，视觉 SLAM 系统已经趋于成熟，且系统精度较高。当前，学术界的研究重心转向提升 SLAM 系统在不同环境中的鲁棒性和可靠性，以应对真实世界中的复杂场景。然而，现有的主流方案和成熟的开源框架普遍基于一个基本假设，即智能机器人所处的环境是静态的，传感器视野中的目标保持静止。这个假设在现实世界中难以满足，例如马路上疾驰的车辆、室内运动的人和宠物等动态元素。在实际应用中，这些开源框架由于无法区分自身运动和环境中物体的运动，导致相机位姿计算出现较大偏差，进而导致系统定位精度大幅下降，所构建的环境地图也因动态目标的存在而缺乏全局一致性，出现大量重影现象。

因此，为了提升视觉 SLAM 系统在动态场景中的鲁棒性和可靠性，使机器人能够从三维结构、物体运动和语义三个层面深入感知和理解整个场景，成为当前视觉 SLAM 领域的研究重点，具有重要的研究价值。随着深度学习技术的发展，许多学者在这方面进行了大量研究^[5-8]。在图像处理领域，深度学习技术在多个方面

展现了对传统算法的优势^[9]。在视觉 SLAM 系统中，许多环节仍然依赖传统图像算法，这些算法与某些特定的数学和几何问题紧密相关，利用相关的深度学习算法可以对传统的图像算法进行优化，提高算法精度。

1.2 国内外研究历史与现状

(1) 传统 SLAM 系统的发展现状

SLAM 的概念最初由 Smith RC 等人提出，作者利用统计滤波器，通过运动方程和观测方程建立 SLAM 问题模型，并通过最小化噪声来解决这一问题^[10]。早期的 SLAM 研究主要集中在激光和雷达传感器上^[11-12]，但这些传感器价格昂贵且不便携带。相比之下，视觉 SLAM 使用摄像头作为传感器，具有价格低廉、体积小巧的优点，便于搭载在小型机器人或无人机上。此外，图像数据携带丰富的环境信息，这对 SLAM 系统的优化和扩展非常重要。视觉 SLAM 系统通常由前端、后端、回环检测和建图等部分组成。根据前端方法的不同，视觉 SLAM 可以分为特征点法和直接法。与直接法相比，特征点法更为稳定，对光照等变化不太敏感，是当前主流的视觉 SLAM 方法，其系统框架如图1.1所示。

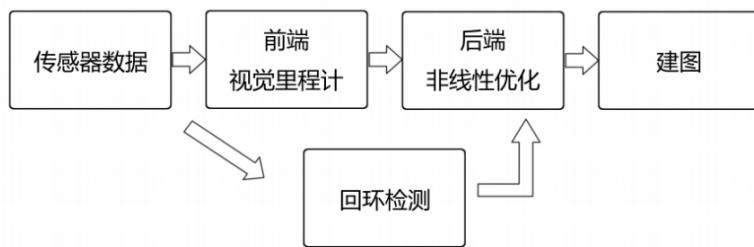


图 1.1 视觉 SLAM 系统流程图

早期的单目视觉 SLAM 系统通常采用滤波器方法实现^[13-14]，这些方法为单目视觉 SLAM 的发展奠定了基础。然而，基于滤波器的方法需要处理每一帧图像来估计相机的位置，计算量巨大，难以在普通设备上实现实时 SLAM，并且会不断积累线性误差，效果不理想。基于关键帧的方法则认为，连续两帧图像之间的信息大部分相似，因此无需处理每一帧，只需选取并处理关键帧即可。文献^[15-16]显示，基于关键帧的技术可以大幅降低计算量，同时保持准确度。H. Strasdat 等人在^[17]中证明，影响 SLAM 效果的是特征点的数量而非帧的数量，进一步说明了关键帧方法比滤波法效果更好。G. Klein 和 D. Murray 提出的 PTAM 是基于关键帧方法的经典代表之一^[16]，该方法将位姿估计和建图分别放在两个独立线程中运行，这是第一个采用多线程方法的视觉 SLAM 系统。尽管 PTAM 已经有些过时，但其设计理念

在许多 SLAM 框架中仍有体现。Raul Mur-Artal 等人借鉴 PTAM 的设计思想提出了 ORB-SLAM^[18]。ORB-SLAM 是一个基于特征点法的单目视觉 SLAM 系统，能够在 CPU 上实时运行。与 PTAM 使用的 FAST 不同，ORB-SLAM 基于 ORB(Oriented FAST and Rotated Brief)^[3]，具有更强的旋转不变性和尺度不变性，使 SLAM 系统更加鲁棒。ORB-SLAM 还增加了实时回环检测、地图自动初始化和实时相机重定位等功能，使其适用于室内小场景和室外大场景，是一个完整、高效且鲁棒的单目视觉 SLAM 系统。ORB-SLAM 提出后在业界引起了广泛关注，迅速成为当时最流行的 SLAM 系统，但其仅支持针孔相机单目 SLAM。

随后，Raul Mur-Artal 和 Juan D. Tardós 在他们先前的研究工作 ORB-SLAM 的基础上提出了 ORB-SLAM2^[19]。这是第一个同时支持单目、双目和 RGBD 相机的完整 SLAM 系统。ORB-SLAM2 不仅支持回环检测和相机重定位，还支持地图重用。在无法建图的情况下，ORB-SLAM2 提供了一个轻量级的定位模式，有效地重用了地图。ORB-SLAM2 在 29 个公开数据集上取得了优秀的结果。Carlos Campos 等人在 ORB-SLAM1、ORB-SLAM2 和 ORB-SLAM Visual-Inertial^[20] 的基础上，提出了 ORB-SLAM3^[1] 并开源了项目代码。相比 ORB-SLAM2，ORB-SLAM3 新增了对鱼眼相机的支持，并将相机模型提取为统一接口，用户可以根据需要定义相机模型。目前该系统提供了基于 Kannala et al.^[21] 的鱼眼相机模型和基于 Tsai^[22] 的针孔相机模型。此外，ORB-SLAM3 还新增了惯性传感器模块，成为第一个同时支持单目、双目和 RGBD 的视觉惯性 SLAM 系统。该系统还提出了多地图方案，即使丢失跟踪也不影响系统运行，从而大大提高了系统的鲁棒性。

特征点法 SLAM 具有显著优势，是当前 SLAM 研究的主流。然而，它并非完美无缺。特征点法需要不断提取图像的关键帧和特征描述子，这仍是一个耗时的过程。此外，一幅图像通常包含数十万甚至上百万个像素点，而特征点法仅提取其中几百个特征点，忽略了大部分图像信息。在没有明显纹理的环境中，特征点法可能无法获取足够的特征点来估计相机的运动。直接法能够很好地克服这些问题。直接法不需要计算特征点，而是直接根据图像的亮度信息，最小化光度误差来估计相机的运动。早期研究者对直接法进行了探索^[23-24]，但直到 LSD-SLAM^[25-26] 和 SVO^[27-28] 等优秀的开源直接法 SLAM 出现后，直接法才与特征点法平分秋色。LSD-SLAM 提出了一种基于 sim(3) 的直接跟踪算法，解决了尺度漂移问题，使其在大尺度空间下也能取得良好效果。此外，LSD-SLAM 在相机跟踪过程中使用基于概率的方法处理噪声干扰，提高了准确性。然而，LSD-SLAM 在相机移动速度较快时效果较差，并且在回环检测时需要依赖特征点法。SVO 是一种半直接法的

视觉 SLAM，算法不直接处理整帧图像，而是将图像分块，通过处理图像分块来获取相机的运动信息，从而提升算法的鲁棒性。SVO 具有低抖动和在重复纹理环境中表现良好的特点。与 LSD-SLAM 无法适应相机快速移动的缺陷不同，SVO 可以每秒处理 50 帧图像信息，在 CPU 上即可实现实时 SLAM。

随着研究的深入，深度学习在许多领域取得了巨大成功。在视觉 SLAM 中，深度学习也逐渐被应用于多个模块，以获得更好的结果。例如，Kendall et al.^[29] 设计了一个端到端的卷积神经网络，利用迁移学习的方法，在训练好的分类器上进行训练，实现了一个六自由度的相机位姿估计模型。实验表明，该模型在相机位姿估计中效果显著，并解决了基于 SIFT 特征的姿态估计在长基线下失效的问题。目前，大部分视觉 SLAM 研究主要集中在单目 SLAM 上。单目 SLAM 的一个显著特点是缺乏深度信息，并且在无纹理区域难以工作。Tateno et al.^[30] 提出了一种神经网络来估计单目深度，在小基线立体匹配下有效解决了单目深度信息缺失和无纹理区域失效的问题。此外，该研究还利用语义分割和三维建模技术重建场景信息，为解决单目相机的场景理解问题提供了新思路。尽管目前机器学习与 SLAM 的结合尚未成为主流，但随着越来越多研究人员的深入探索，基于深度学习的 SLAM 方案将在未来越来越多地涌现。

1.2.1 特征点提取和描述符研究现状

当前，常用的图像特征提取和匹配方法包括 SIFT（尺度不变特征变换）^[2]、SURF（加速稳健特征）^[4] 以及 ORB（定向 FAST 和旋转 BRIEF）^[3]。SIFT 特征具有对尺度和旋转变化的不变性，同时在光照变化和噪声下也表现出相对稳定性。一些研究者在视觉 SLAM 中采用了 SIFT 特征^[31-32]，但由于 SIFT 只利用了图像的灰度信息，忽略了其他信息，导致一定程度的信息丢失。此外，SIFT 特征有 128 维，维度较大，计算速度较慢。相较之下，SURF 通过使用 Harr 特征和积分图像，在保持尺度和旋转不变性的同时大大提高了运算速度，比 SIFT 快了 3-7 倍。Zhang et al.^[33], Wang et al.^[34] 提出的 SLAM 方法采用了 SURF 特征，在运算速度上有显著提升，且对模糊和旋转的图像具有较好的鲁棒性。然而，SURF 在应对视角变化、光照变化和季节变化的场景时表现不佳。ORB 结合了 FAST 特征点检测^[35] 和 BRIEF^[36] 描述符，并进行了改进，大大提升了特征检测与匹配的速度，其速度约为 SURF 的 10 倍，SIFT 的 100 倍。ORB-SLAM^[1,18-19] 等 SLAM 系统采用了 ORB 特征，使得这些系统能够在 CPU 上实现实时 SLAM，这很大程度上得益于 ORB 特征检测与匹配的高效性。

1.2.2 基于 YOLO 的目标检测和语义分割发展现状

目标检测是计算机视觉的研究热点，广泛应用于监控安全、自动驾驶、交通监控和机器人视觉等领域^[37]。目标检测通常是识别一些预定义类别的目标实例（如人和车）^[38]。传统目标检测依赖于精巧的手工特征设计和提取，如方向梯度直方图（Histogram of Oriented Gradient, HOG）^[39]。2012 年，基于 CNN 的 AlexNet 在 ImageNet 图像识别比赛中取得显著优势，从而引发了对深度学习的广泛关注^[40]，目标检测也逐渐进入深度学习时代^[37-38]。

基于深度学习的目标检测方法分为“两阶段检测”和“单阶段检测”两类^[41]。两阶段检测是一个“从粗到细”的过程，而单阶段检测则是端到端“一步完成”^[38]。通常，两阶段检测的定位和识别精度较高，而单阶段检测速度更快^[41]。有关两阶段检测算法的详细分析，请参见文献^[37,41]。单阶段检测尝试直接将每个感兴趣区域分类为背景或目标对象，通过一个阶段直接给出物体的类别概率和位置坐标。YOLO（You Only Look Once）^[42] 是单阶段检测的典型代表，直接将图像划分为若干区域，同时预测每个区域的边界框和概率，大大提高了检测速度，但与当时的两阶段检测器相比，定位精度有所不足，特别是对小目标的检测。

图1.2展示了 YOLO 的发展时间线，从 YOLOv1 到最新的 YOLOv8，YOLO 系列不断在前代版本的成功基础上进行创新，提升性能和灵活性。YOLOv8 引入了多个新功能和改进，包括新的骨干网络、新的无锚点（Anchor-Free）检测头和新的损失函数，使其能够在从 CPU 到 GPU 的各种硬件平台上高效运行。此外，YOLOv8 还增加了实例分割和姿态估计功能，进一步增强了其在实际应用中的适用性。

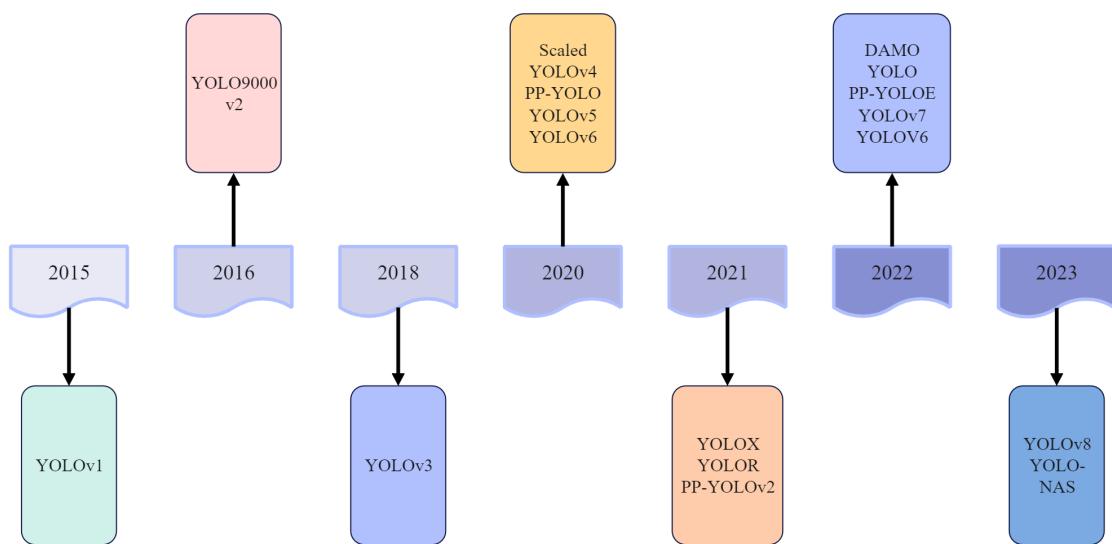


图 1.2 YOLO 发展时间线

1.2.3 结合深度学习的视觉 SLAM 算法

语义 SLAM 结合了语义信息与传统 SLAM 技术，旨在提升地图信息的丰富度和应用场景的可靠性。传统 SLAM 主要关注环境的几何信息，如地图拓扑结构和建筑轮廓。语义 SLAM 不仅包括环境的几何信息，还融合了目标检测、物体识别和语义分割等技术，以获得更丰富的语义信息。通过视觉算法，语义 SLAM 能识别周围环境中的不同物体，并估计它们的空间位置、姿态和二维分割轮廓。这些语义信息可以应用于环境理解、智能导航和机器人操作等多个领域。

1.3 本文的主要贡献与工作

针对基于特征点法的视觉 SLAM 在动态场景下的稳定性较差以及建图和定位效果较差的问题，动态物体的存在会严重影响位姿结算，产生较大的轨迹漂移，本文利用深度学习中的实例分割的方法，对动态场景下的 SLAM 算法进行优化，具体工作如下：

(1) 设计一个全新的视觉 SLAM 系统，该系统结合了传统的 ORB-SLAM3 和深度学习的 YOLO 目标检测算法，在前端特征点提取的过程中，利用 YOLOv8 对每帧进行实例分割，将分割后的结果送入 SLAM 前端，前端在提取特征点的同时，对动态物体的特征点进行剔除。该 SLAM 能够在能够在动态场景下稳定的运行，通过各项数据集和实际测试中都表现出了良好的鲁棒性和泛化性。

(2) 针对动态 SLAM 中关于语义分割时间效率的问题，本文将语义分割网络放在一个单独运行的线程之中，这样语义分割可以和 ORB 特征提取、运动一致性检测并行运行，可以提高系统运行的效率。

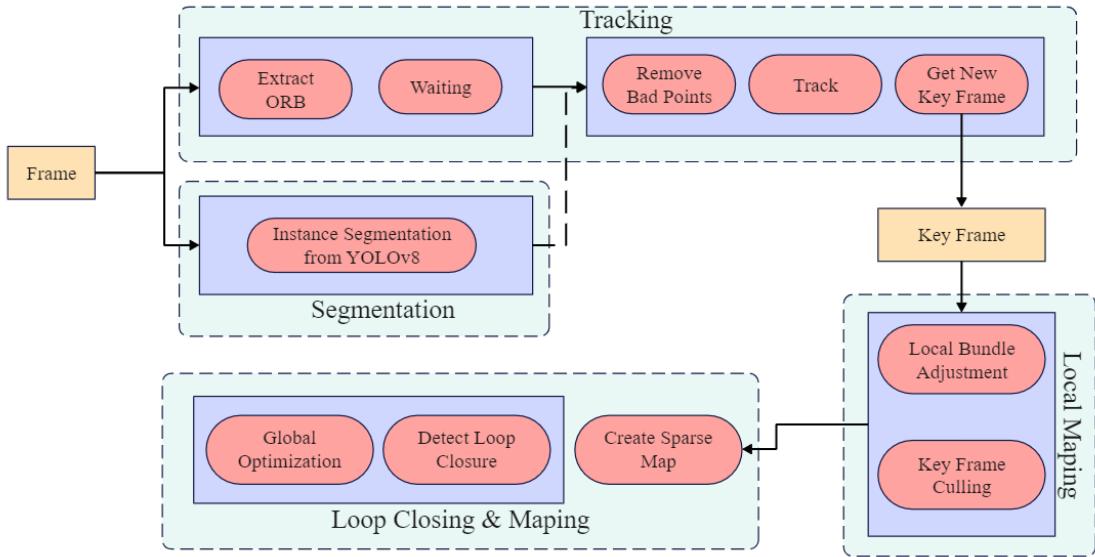


图 1.3 YOLO-ORB-SLAM3 框架结构

图1.3展示了该系统的基本框架，具体来说，我们从相机或者本地获得的每一帧图片进入系统时，分别进入 Tracking 和 Segmentation 这两个单独的线程进行处理，Tracking 线程用于对特征点的提取和检查，Segmentation 用于对每一帧图像进行语义分割，再将结果传入 Tracking 线程，在该线程中对动态特征点进行剔除，然后再进行跟踪，并生成关键帧，将关键帧传入 Local Mapping 线程当中，最后再进行基于词袋方法的循环检测机制，避免零点漂移。

1.4 论文的结构安排

本文的章节内容安排如下：

第一章为绪论，首先阐述了本文主要研究内容的背景和意义，然后介绍了视觉 SLAM 算法的国内外研究历史和现状，其中主要介绍了基于人工的特征点提取和描述符研究现状和结合深度学习后的视觉 SLAM 算法的发展和现状。除此之外，还介绍了目标检测领域 YOLO 的发展情况，以及其在语义分割上面的拓展。最后介绍了本文的研究内容及所做的工作。

第二章为视觉 SLAM 系统及 YOLO 目标检测基础，主要介绍了 YOLOv8 算法的基本原理和视觉 SLAM 系统的基本原理，其中重点介绍了前端的视觉里程计相关的内容，这和后面系统的优化息息相关。

第三章为基于 YOLOv8 的实例分割研究，主要介绍了 YOLOv8-seg 模型的训练和部署过程，这里给出了自动驾驶和室内外一般场景下的两个数据集用来作为训练参考，以及训练过程中设置的一些参数设置和训练方式，然后展示了模型的

训练效果，最后介绍了模型在以 C++ 为基础的系统上如何进行高效部署，用于后续系统的优化。

第四章是动态场景下基于 YOLOv8 优化的视觉 SLAM 系统设计，阐述了训练好的 YOLOv8-seg 模型如何对传统的视觉 SLAM 系统进行动态场景下的优化。为了达到系统实时性的要求，我们为系统加入了多线程设计，使得 ORB 提取和 YOLO 预测可以并行执行，最后将设计好的系统分别在常见数据集和真实场景下进行测试和结果分析，分析系统的优缺点，验证了本文提出方法的有效性。

第五章是全文的总结和展望，对全文的工作进行总结，并反思了动态场景下视觉 SLAM 的一些问题，以及对该研究方向未来的展望。

2 视觉 SLAM 系统及 YOLO 目标检测基础

2.1 引言

YOLO (You Only Look Once) 是一种先进的目标检测算法，以其简洁高效而著称，它将目标检测任务视为一个单一的回归问题，通过在输入图像上直接预测边界框的位置和类别概率来实现。这种端到端的检测方法使得 YOLO 在实时性能方面表现突出，能够在单张图像上同时检测出多个目标，无需复杂的后处理步骤。由于其高效的设计，YOLO 被广泛应用于各种场景，包括智能监控、自动驾驶、物体跟踪等领域，目前的 YOLOv8 已经包含了许多除了目标检测以外的功能，包括语义分割，位姿估计，分类，追踪等。

视觉 SLAM 以摄像头作为传感器，最主要的优点是其价格低廉，而且体积小，便于搭载在小型机器人或者无人机上。而且图像数据携带着丰富的环境信息，这对 SLAM 系统的优化和扩展具有很重要的意义。总的来说 SLAM 技术按照传感器来进行分类可以分为激光雷达 SLAM 以及视觉 SLAM 等。而在视觉 SLAM 中又可以按照传感器的数量和型号来进行分类，如单目 SLAM 和双目 SLAM 以及针孔相机 SLAM、RGBD 相机 SLAM 以及鱼眼相机 SLAM 等。视觉 SLAM 系统本身的架构主要由前端、后端、回环检测以及建图等部分组成。在基于特征点法的 SLAM 中，前端视觉里程计的主要任务就是通过特征提取和特征匹配，计算相邻关键帧之间相机的位姿。回环检测是用于检测并发现当前相机所处的位置是否在过去某个时间曾经到达的技术。当检测到相机当前位置与之前到达的某个位置相同时，就是发现了回环，系统将回环信息反馈给后端进行处理。后端系统在 SLAM 系统运行的过程中，会接受来自前端和回环检测系统的信，在得到回环反馈后系统会对估计出的轨迹进行优化，从而消减累积漂移获得更具有全局一致性的轨迹与地图。建图是根据估计出的轨迹以及环境特征建立相应的地图的过程。本文主要研究内容主要为前端视觉里程计和 YOLO 实例分割相结合这部分，因此后续内容将对这两部分进行较为详细的介绍。

2.2 YOLOv8 基本原理

以 YOLO 系列算法为代表的一阶段模型，通过卷积网络直接提取特征，同时在预测特征图上生成边界框，并对这些边界框进行分类和回归，从输入图像到最终预测结果一次性完成。接下来将介绍 YOLO 系列算法，其中 YOLOv1 开创了该系列，

YOLOv3 进行了重大改进，YOLOv5 在训练过程中提出了许多新技巧，YOLOv8 通过修改 YOLOv5 的网络结构进一步提升了检测效果。

YOLOv1 是 YOLO 系列算法的起点，对后续算法的发展产生了重大影响，后续的算法都是在此基础上演变而来的。YOLOv1 的核心思想是通过单一的卷积神经网络，从图像输入、候选框生成到最终的类别预测和边界框回归，完成整个过程，其网络结构如图2.1所示。

算法基本原理：

在 YOLOv1 算法中将输入图像分成 7×7 个网格单元，每个网格用来预测落在此网格上的物体。

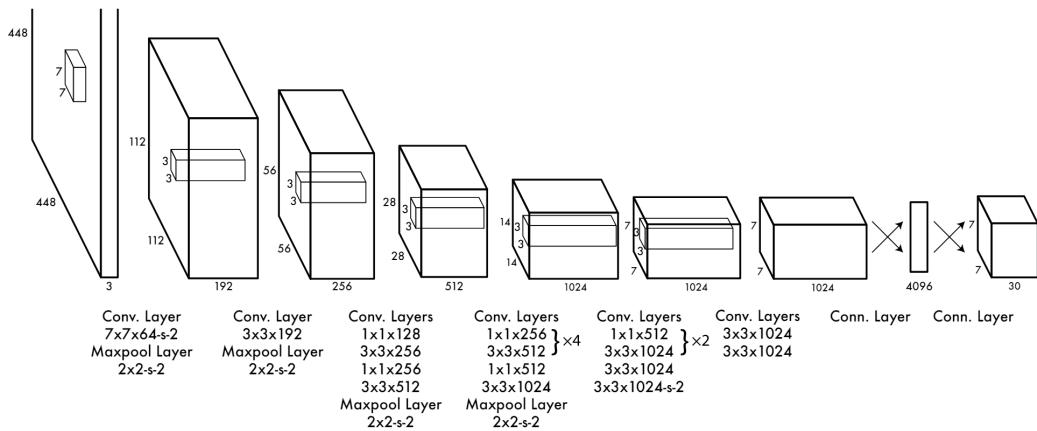


图 2.1 YOLOv1 框架图

YOLOv8 算法由 Glenn Jocher 提出，是在 YOLOv3 和 YOLOv5 的基础上发展而来的。其主要改进点如下：

(1) 数据预处理。YOLOv8 的数据预处理策略继承了 YOLOv5 的方法，采用了马赛克增强 (Mosaic)、混合增强 (Mixup)、空间扰动 (Random Perspective) 和颜色扰动 (HSV augment) 四种增强手段。

(2) 骨干网络结构。YOLOv8 的骨干网络结构可从 YOLOv5 略见一斑，YOLOv5 的主干网络的架构规律十分清晰，总体来看就是每用一层步长为 2 的 3×3 卷积去降采样特征图，接一个 C3 模块来进一步强化其中的特征，且 C3 的基本深度参数分别为“3/6/9/3”，其会根据不同规模的模型的来做相应的缩放。在 YOLOv8 中，大体上也还是继承了这一特点，原先的 C3 模块均被替换成新的 C2f 模块，C2f 模块加入更多的分支，丰富梯度回传时的支流。YOLOv8 的 C2f 模块和 YOLOv5 的 C3 模块的区别如图2.2所示。

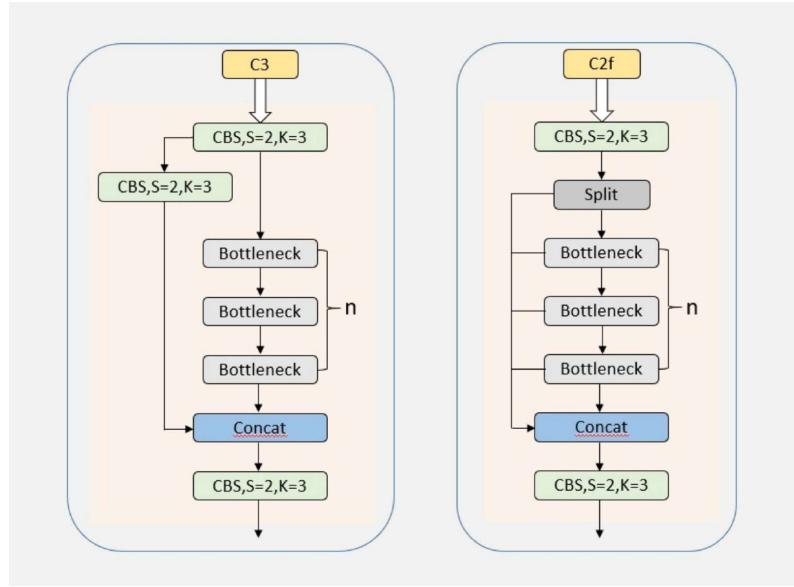


图 2.2 C2f 模块和 C3 模块示意图

(3) FPN-PAN 结构。YOLOv8 继续采用 FPN+PAN 结构构建特征金字塔，实现多尺度信息的融合。除了将 FPN-PAN 中的 C3 模块替换为 C2f 模块，其余部分与 YOLOv5 基本一致。

(4) Detection head 结构。从 YOLOv3 到 YOLOv5，检测头一直是“耦合”(Coupled)的，即通过一层卷积同时完成分类和定位任务。YOLOX 出现后，YOLO 系列首次采用“解耦头”(Decoupled Head)。YOLOv8 也采用了这种结构，使用两条并行分支分别提取类别特征和位置特征，然后通过各自的一层 1×1 卷积完成分类和定位任务。YOLOv8 的整体网络结构如图2.3所示。

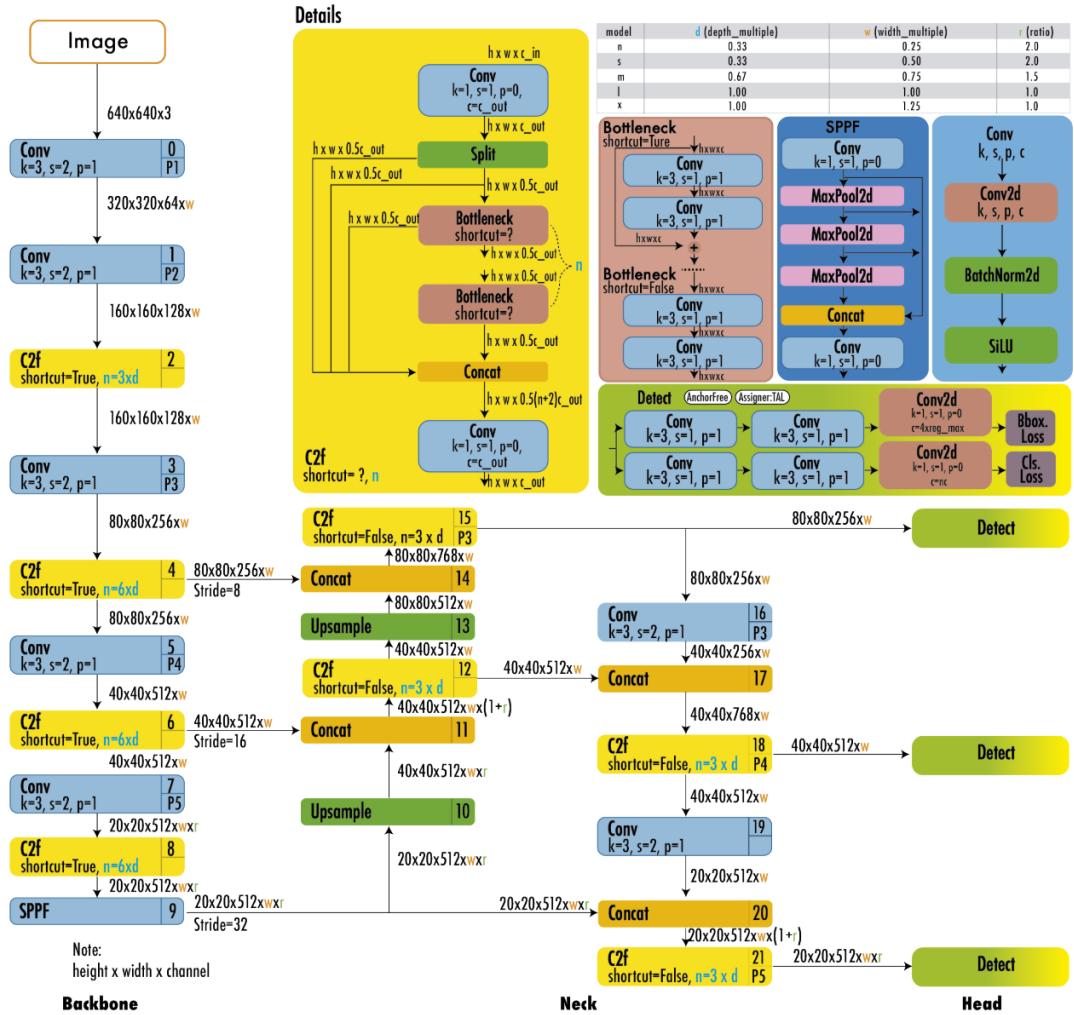


图 2.3 YOLOv8 网络框架

2.3 视觉 SLAM 系统基本原理

经过 30 多年的发展视觉 SLAM 环境感知技术的理论框架已经逐渐清晰，包括前端、后端、回环检测和建图 4 个主要的技术环节。前端的任务是估算相邻图像采集时相机的运动和计算局部地图；后端主要是基于回环检测信息对图像位姿和地图信息进行优化；回环检测则是根据图像信息识别已经出现的场景或位置，如果检测到回环，然后把信息提供给后端进行处理；建图环节主要根据估计的轨迹，建立对应的地图^[43]，整体框架如图2.4所示。

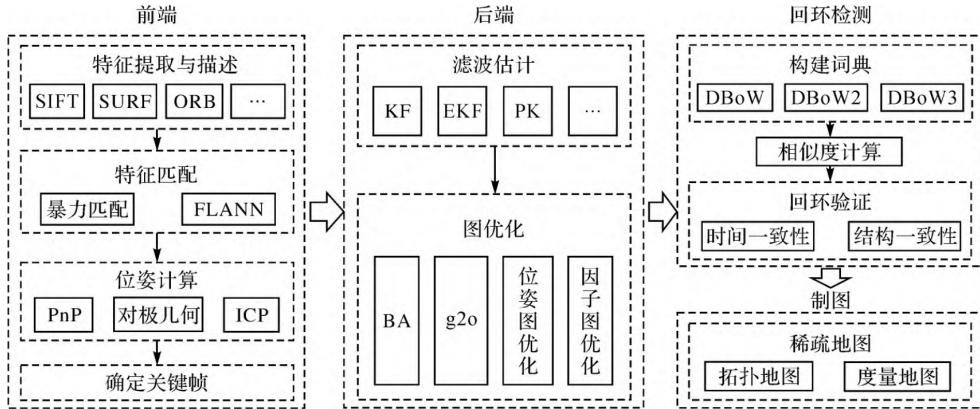


图 2.4 V-SLAM 框架图

2.3.1 视觉里程计

视觉里程计也叫 SLAM 的前端，视觉里程计的核心任务就是通过相邻的两幅图像计算出这两副图像之间相机的运动。视觉里程计通过提取相邻图像中的特征点并进行特征匹配，再根据相机的成像原理，就可以求解相机的运动（即位姿估计）。因此在视觉里程计模块核心要素就是相机模型、特征提取与匹配和位姿估计，在以下内容中将围绕这三个要素展开介绍。

(1) 相机模型

视觉 SLAM 对于相机位姿的估计算法是建立在相机成像几何模型基础之上的，因此本文首先对常见的相机成像模型进行介绍。在视觉 SLAM 中常见的相机按照类型划分可分为单目相机、双目相机和 RGBD 相机，下面将介绍本文主要研究内容普通相机的成像模型的相关基础。

① 相机类别

单目相机： 单目相机通过一个镜头将外界的三维场景投影到二维图像平面上。由于只有一个视点，单目相机本身无法直接感知深度信息，需要通过一系列计算和算法来推断三维信息。在 SLAM 当中，一般通过利用前后两帧进行三角化来估计对应特征点的深度。

双目相机： 双目相机由两个平行放置的单目相机构成，通常具有固定的基线距离。它通过两个相机捕捉到的两张视点不同的图像来推断场景的深度信息。两个摄像头可以简单解决单目相机无法直接获取深度信息的问题，但使用之前需要进行标定。双目相机标定包括单目标定和立体标定。使用张正友标定法进行内参和外参的标定，校正双目相机的内参矩阵、畸变参数和相对位姿。

RGBD 相机： RGB-D 相机结合了传统的 RGB 相机和深度传感器，能够同时捕捉彩色图像和深度信息。常见的深度传感技术包括结构光、飞行时间（ToF）和激光雷达（LiDAR）。RGB-D 相机更适合需要高精度、实时深度数据的应用场景，表2.1中对比了不同相机之间的特性。

表 2.1 相机类型对比

特性	单目相机	双目相机	RGB-D 相机
深度获取方式	通过多帧图像推断	通过立体匹配计算	直接测量深度
硬件复杂度	最简单	中等	较复杂
适用环境	光照变化小的环境	光照变化较小的环境	各种环境，光照对深度测量有影响
计算量	计算复杂度高	较高的计算需求	计算量相对较低
成本	最低	中等	较高
应用场景	机器人导航、增强现实等	机器人导航、3D 重建、工业自动化等	AR/VR、手势识别、机器人导航等
优点	低成本、轻便灵活、应用广泛	能直接获取深度信息、结构简单、无主动光源	能直接获取高精度深度信息、使用简便
局限性	无直接深度信息、计算复杂、对光照敏感	计算复杂、视差范围有限、遮挡问题	受环境影响、有限视距、功耗较高

②坐标系和坐标变换 为了清楚理解成像模型的相关概念，首先需要了解相机成像几何模型中存在的四种坐标系，分别是世界坐标系、相机坐标系、相机图像坐标系和图像像素坐标系。当三维世界中的一个物体投影成像到数字图像上时，其位置坐标映射关系通过这四种坐标系进行描述，整个映射过程如图2.5所示。

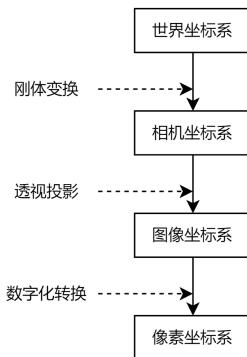


图 2.5 四种坐标系间的转换关系

世界坐标系和相机坐标系都为三维坐标系，世界坐标系一般使用人为选定参考点为坐标系原点，相机坐标系一般以相机光心为坐标系原点。刚体变换只改变物体的空间位置和朝向，而保留物体自身形状，即刚体变换只包含旋转和平移，可用两个变量来描述：旋转矩阵 R 和平移向量，世界坐标系中的点 $P_w = [X_w, Y_w, Z_w]^T$ 和相机坐标系中的点 $P_c = [X_c, Y_c, Z_c]^T$ 之间的刚体变化可表示为式2.1：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + t \quad (2.1)$$

使用齐次坐标，可由式2.2表示：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.2)$$

从相机坐标系到图像坐标系的转换涉及到相机的成像几何模型，针孔相机模型也称为直线投影模型（Rectilinear Projection Model），其成像原理与小孔成像相似，简化模型如图2.6所示。

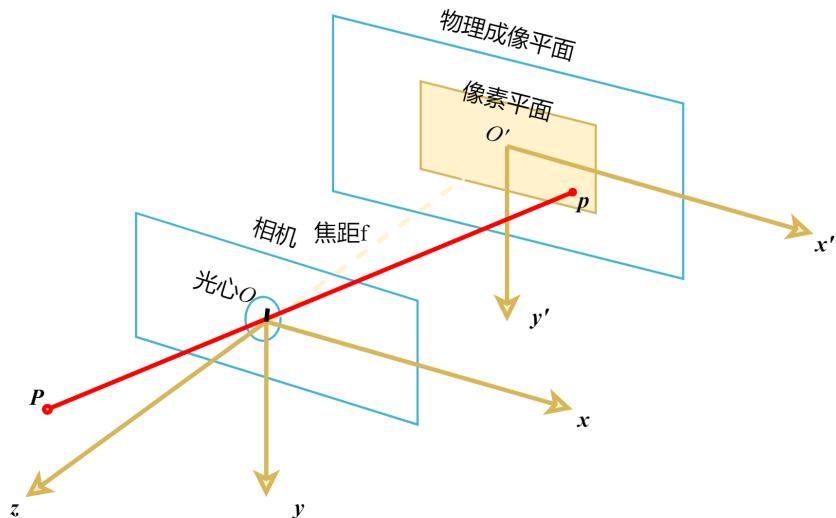


图 2.6 针孔相机模型

相机成像过程实际上就是把 O-xyz 的三维坐标系中的物体透过相机中心的透镜投影到相机传感器所形成的二维像素坐标系 O-x'y' 中。其中透镜光心到传感器中心的距离称为焦距 f ，设三维空间中物体 P 的坐标为 $P_w = [X, Y, Z]^T$ ，其在像素平面的成像点 p 在相机坐标为 $p = [x, y, z]^T$ ，根据相似三角形关系可得到相机坐标系到图像坐标系的关系如式2.3：

$$\begin{cases} x_u = f \frac{x_c}{z_c} \\ y_u = f \frac{y_c}{z_c} \end{cases} \quad (2.3)$$

在齐次坐标下可由式2.4表示：

$$z_c \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (2.4)$$

总的来看，由相机坐标系到图像坐标系的转换关系只与相机焦距有关。

光线在通过镜头后最终成像到图像传感器上，产生数字化图像，一般人们使用图像左上角作为像素坐标系的原点，因此从图像坐标系到像素坐标系的转化描述的是传感器大小和像素单位长度之间的映射关系，用 dx 和 dy 表示相机光学传感器上每个实际像素在 x 方向和 y 方向上的物理长度，则像素坐标 (u, v) 与图像坐标系坐标 (x_u, y_u) 的关系如式2.5所示：

$$\begin{cases} u = u_0 + \frac{x_u}{dx} \\ v = v_0 + \frac{y_u}{dy} \end{cases} \quad (2.5)$$

其中 (u_0, v_0) 是图像中心在像素坐标系下的坐标，表示相机坐标系原点和传感器像素坐标系原点之间的偏移距离。用齐次坐标可以将上式表示成式2.6：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} \quad (2.6)$$

综上所述，针孔相机成像模型用齐次坐标可以将整个过程表示为如式2.7所示：

$$z_c \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ o^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.7)$$

通过上面的推导，就获取了相机内参数的定义 \mathbf{K} 和相机外参数的定义 \mathbf{T} ，他们分别描述相机自身的固有参数和相机在世界坐标系中的位置参数，如式2.8所示：

$$\mathbf{K} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ o^T & 1 \end{bmatrix} \quad (2.8)$$

(2) 位姿估计

位姿估计通过输入特征信息来估算相邻两帧相机的姿态变化，包括相机的旋转 R 和平移 t 。在不同的 SLAM 系统中，根据特征点的类型，通常分为 2D-2D、3D-3D 和 3D-2D 三种情况。2D-2D 特征点常用于大多数单目视觉里程计系统中，因为缺乏深度信息，只能得到特征点在前后两帧图像中的二维坐标。在视觉 SLAM 中，利用匹配的 2D 特征点对来求解位姿通常有两种方法：对极几何（Epipolar Geometry）方法和射影几何（Projective Geometry）方法，对极几何模型如图2.7所示。

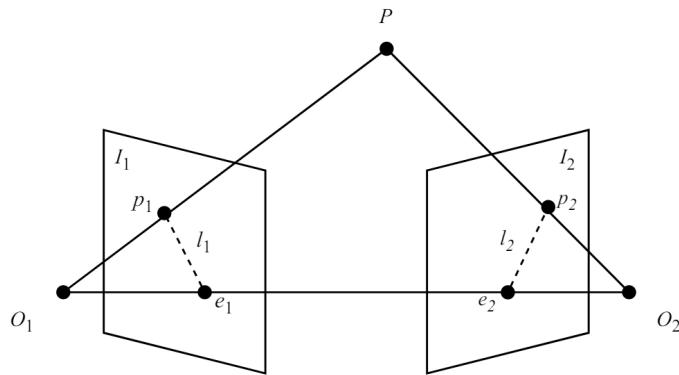


图 2.7 对极几何模型

该模型可以用来解决单目相机的深度估计问题，也可以用来解决相机的位姿估计，模型中 O_1 和 O_2 是同一相机在不同帧下的光心， I_1 和 I_2 表示相邻的两帧图像，简单来说，图像中的某一像素点，对应在另一帧图像的极线上，假设 p_1 和 p_2 是两帧图像对应的特征点，对应的空间中的三维点为 P ， O_1 和 O_2 的连线与两帧图像的交点分别为 e_1 和 e_2 ，那么 p_1e_1 和 p_2e_2 分别就是这两帧图像里面的极线。

为了表示对极约束中在两个成像平面上的点的相对关系，在数学上我们只需要加入一个矩阵（本质矩阵或者基础矩阵）就可以简洁的写出两者的等式关系，对于本质矩阵 E 其矩阵等式如式2.9所示：

$$pl^T E pr = 0 \quad (2.9)$$

对于基础矩阵 F 其矩阵等式如式2.10所示：

$$cl^T F cr = 0 \quad (2.10)$$

其中 pl 和 cl 表示空间中的点投影在左视图的位置， pr 和 cr 为其投影在右视图中的位置， p 和 c 表示视图中的点在不同坐标下的表示。根据上面的任意一个等式，当我们已知空间中一点在左右视图中的两个位置坐标和基础矩阵（或本质矩阵）这

三个中的两者，我们就能够计算得到第三个未知量（对于求 E 和 F 来说，可能需要多个点联立求解，因为这两个矩阵中的未知量比较多）。

上面两式，我们可以看到其等式形式是一模一样的，唯一的区别就在于，两种是在不同坐标系中的表达。本质矩阵 E 连接的是不同视角下摄像机坐标系下两个投影点的关系，所以 pl 和 pr 是在摄像机坐标系下的坐标。基础矩阵 F 链接的是在图像坐标系下两个视图图像中的坐标像素坐标的关系，所以 $cI\ll cr$ 是图像的像素坐标。我们知道从摄像机坐标系到图像的像素坐标系，是由摄像机的内参矩阵 K 来确定的，因此基础矩阵与旋转矩阵 R 和平移向量 T 之间的关系还需要有摄像机的内参矩阵加入，所以其与本质矩阵的关系如式2.11所示：

$$F = K l^{-T} \cdot E \cdot K r^{-1} \quad (2.11)$$

知道本质矩阵 E 以后， E 矩阵是由 3 自由度平移向量，以及 3 自由度旋转向量构成，故有 6 个自由度。此时由于一个自由度可观，故需要至少 5 对点完成 E 矩阵求解。但是当将旋转向量展开，变为旋转矩阵时，考虑 9 自由度计算。同时一个自由度可观，故至少需要 8 对点完成 E 矩阵求解。这种求解方法又分别叫做五点法或八点法，通过八个对应的特征点列出线性方程，即可解得相机的旋转矩阵 R 和平移向量 t ，由此也就得到了相机的位姿。

2.3.2 回环检测与建图

在视觉 SLAM 系统中，回环检测机制对于减少累积漂移和实现重定位起着关键作用。在许多基于特征点的方法的主流 SLAM 系统中，如 Campos et al.^[1], Yu et al.^[44]，回环检测通常采用基于词袋模型的方法^[45]。这一方法的核心在于词袋向量和字典。具体步骤是首先提取图像特征，然后将这些特征转换为“单词”，每个图像包含多个“单词”，这些“单词”组成一个词袋向量。最后，将这些“单词”保存在字典中。当进行回环检测时，通过查询字典中的“单词”来找到相似的图像作为候选回环帧，随后进行进一步的判断。

SLAM 中的三维地图主要包括了稀疏地图、半稀疏地图和稠密地图，本论文所构建的主要三维点云地图，属于稀疏地图，SLAM 中的建图工作主要是在视觉里程计计算得到位姿后，将得到的三维地图点进行点云地图的构建，并对相机位姿和所有点云进行全局优化。

由于本论文研究主要方向为视觉里程计与深度学习相结合，后端相关的细节这里就不再赘述。

3 基于 YOLOv8 的实例分割研究

3.1 数据集与模型的训练

3.1.1 数据集

再该场景中，YOLOv8 的应用场景主要根据 SLAM 的应用场景而定，如果应用场景主要为街道公路等驾驶场景，则该场景下主要的动态事物为移动的汽车，公交车，自行车，小动物和人等，在这种场景下，我们需要使用相关的自动驾驶领域的数据集对 YOLOv8 进行一个训练，这里可以使用 BDD100K 数据集进行训练。若是在室内简单普遍的场景下，可以使用 COCO 数据集进行训练。测试人员需要根据应用场景来对模型进行针对性的训练，以确保模型能在该场景下进行一个很好的目标检测和分割。

(1) BDD100K 数据集

BDD100k 数据集标注了十个类别，包括汽车 (car)、公共汽车 (bus)、行人 (person)、自行车 (bike)、卡车 (truck)、摩托车 (motor)、列车 (train)、骑行者 (rider)、交通标志 (traffic sign) 和交通信号灯 (traffic light)，共计 79863 张图片，大致内容如图3.1所示。该数据集包含大量旋转和不同光照条件的图像，有助于训练更为鲁棒的检测模型。本文实验使用的 BDD100k 数据集包括 69863 张训练集图片和 10000 张验证集图片，部分样本如图所示。由于 YOLOv8 算法对输入图片大小有限制，所有图片需调整为统一尺寸。为了尽量减少图片失真而不影响检测精度，我们将图片调整为 640x640 的大小，并保持原有的宽高比。另外，为了增强模型的泛化能力和鲁棒性，我们采用了数据增强技术，如随机旋转、缩放、裁剪和颜色变换等，以扩充数据集并减少过拟合的风险 (数据集下载地址：<https://bdd-data.berkeley.edu>)^[46]。

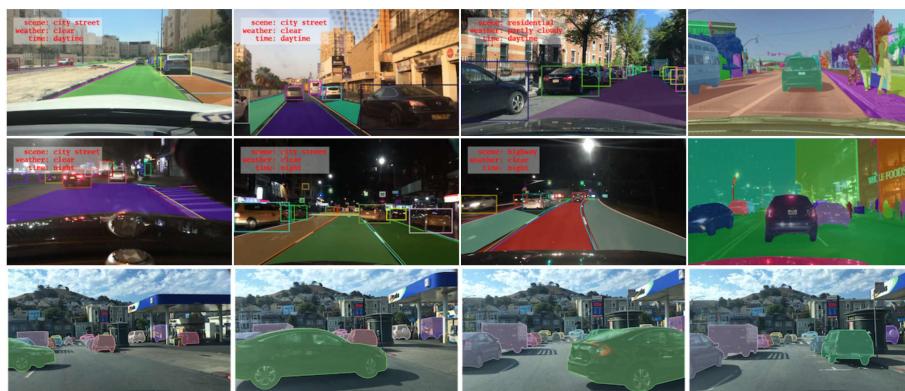


图 3.1 BDD100K 数据集

(2) COCO 数据集

COCO 数据集是一个广泛用于物体检测、分割和字幕任务的大型数据集，旨在实现对场景的深入理解。它涵盖了各种复杂的日常场景，通过准确的目标分割标注来确定图像中物体的位置。该数据集包含 91 类目标，共有 328,000 张图像和 2,500,000 个标签。作为最大的语义分割数据集之一，COCO 提供了 80 类目标，包含超过 33 万张图像，其中 20 万张图像带有标注。整个数据集中标注的个体数目超过 150 万个。(数据集下载地址：<http://images.cocodataset.org>)。

3.1.2 模型训练

在系统中，使用的是 YOLOv8 的实例分割部分，但模型训练过程和目标检测的训练过程基本上差别不大，接下来阐述模型的训练详细过程。

(1) 准备工作

在开始训练之前，需要准备以下工具和材料：

①YOLOv8 源码：从 YOLO 官方 GitHub 仓库下载最新版本的 YOLOv8-seg 源码。

②COCO 数据集或自定义数据集：COCO 数据集是一个大规模的目标检测、分割和关键点检测数据集，从其官方网站下载。自定义数据集则需要根据自己的需求进行收集和标注，这里本论文直接选择用 COCO 数据集进行训练。

③Python 环境：安装 Python 和相关依赖库，如 NumPy、Torch 等。

④GPU：为了加速训练过程，建议使用带有 CUDA 支持的 NVIDIA GPU。

(2) 数据集准备

①数据集标注：对于自定义数据集，首先需要对图像进行标注，生成 YOLO 所需的标注文件（通常为.txt 或.json 格式），使用开源标注工具，如 LabelImg、COCO Annotation Tool 等，对于现有的数据集，则可以直接使用。

②数据集划分：将数据集划分为训练集、验证集和测试集，以便在训练过程中进行模型评估和调优。

③数据预处理：为了提高模型的泛化能力，在图像输入模型前，对图像进行一系列预处理操作，如缩放、裁剪、翻转等。

(3) 模型配置

YOLOv8 具有丰富的配置选项，可以根据具体任务需求进行调整。主要配置项包括：

①锚点尺寸 (Anchor Sizes)：根据数据集中目标的尺寸分布，设置合适的锚点

尺寸，这里锚框尺寸选择默认的尺寸。

②网络结构（Network Architecture）：YOLOv8 提供了多种网络结构，我们选择 CSPDarknet 网络框架进行训练。

③训练参数（Training Parameters）：设置学习率 lr 为 0.01，训练批次 batch 为 32，训练轮数 epoch 为 250 轮。

(4) 模型训练

在配置好模型后，对模型进行训练。训练过程主要包括前向传播、计算损失、反向传播和参数更新等步骤。训练过程中，使用 YOLOv8 提供的可视化工具，实时查看训练过程中的损失、准确率等指标。

(5) 模型评估与优化

训练完成后，对模型进行评估是必要的，以了解其在测试集上的表现。评估指标主要包括准确率、召回率和 mAP（平均精度均值）等。如果发现模型性能不佳，可以通过调整模型配置、增加训练数据、使用数据增强等方法进行优化。

模型训练完成后，输入单张图片简单测试其目标检测和分割的效果如图3.2，由于我们在最终测试的算法数据集利用的是 TUM 的 rgbd_dataset_freiburg3_walking_xyz 和 rgbd_dataset_freiburg3_walking_halfsphere 这两个数据集，所以选择其中的一帧图像作为测试图片。

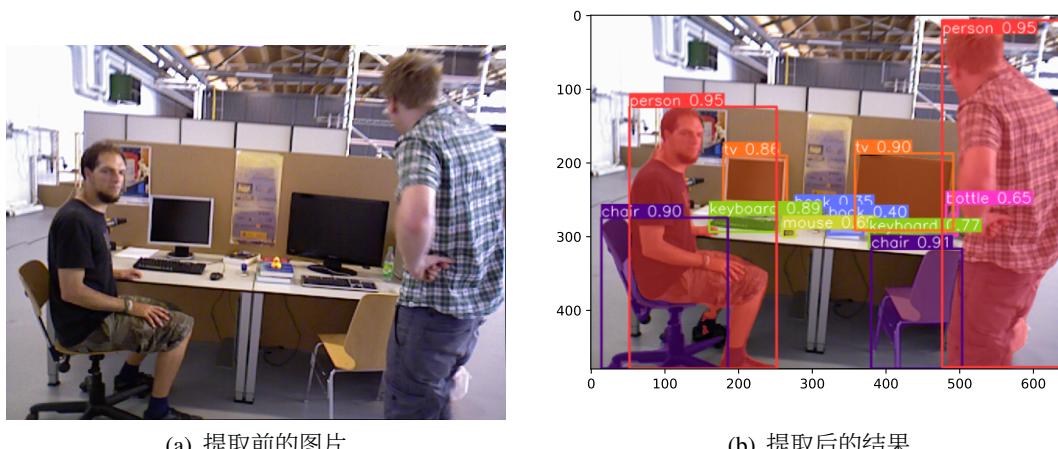


图 3.2 实例分割结果图

在结果上看，在室内的普遍场景下，利用 YOLOv8-seg 模型的检测效果表现良好，由于在系统中，只有动态的目标才会对 SLAM 系统的位姿估计有较大的影响，所以，需要对检测类别进行筛选，若检测得到的类别为静态物体，则暂时将其忽略，若检测得到物体为动态物体，如人，动物，汽车等，则将其保留，并取得其实

例分割的 mask 掩模部分，将所有的动态掩模叠加在一起，分离出前景和背景，这样在系统中只需要对前景中的动态特征点进行剔除即可，经过处理后得到的 mask 图片帧如图3.3所示。

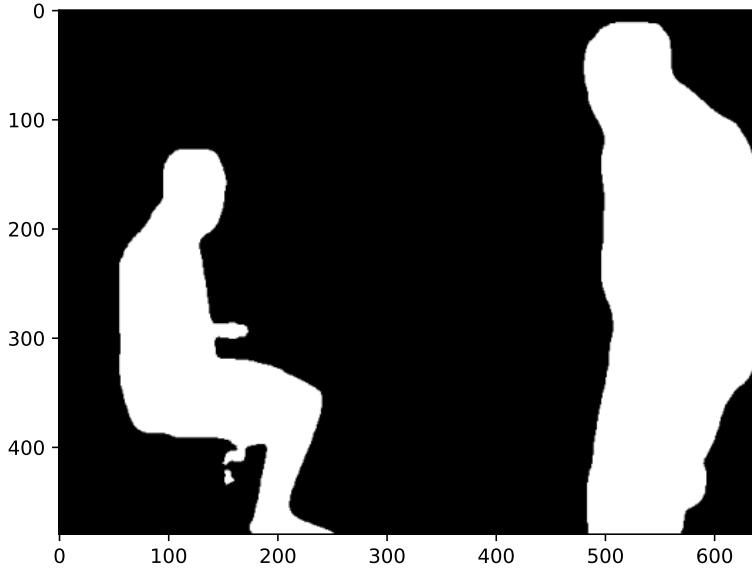


图 3.3 前景 mask 结果图

为验证本文优化后的模型在数据集上的收敛性能，以下内容展示了 YOLOv8n-seg 模型的收敛情况。图中显示了模型的边界框损失、置信度损失和类别损失的曲线，同时也展示了四个性能度量指标的收敛曲线，包括准确率、召回率、mAP@0.5 和 mAP@[0.5:0.95]，详见图3.4。

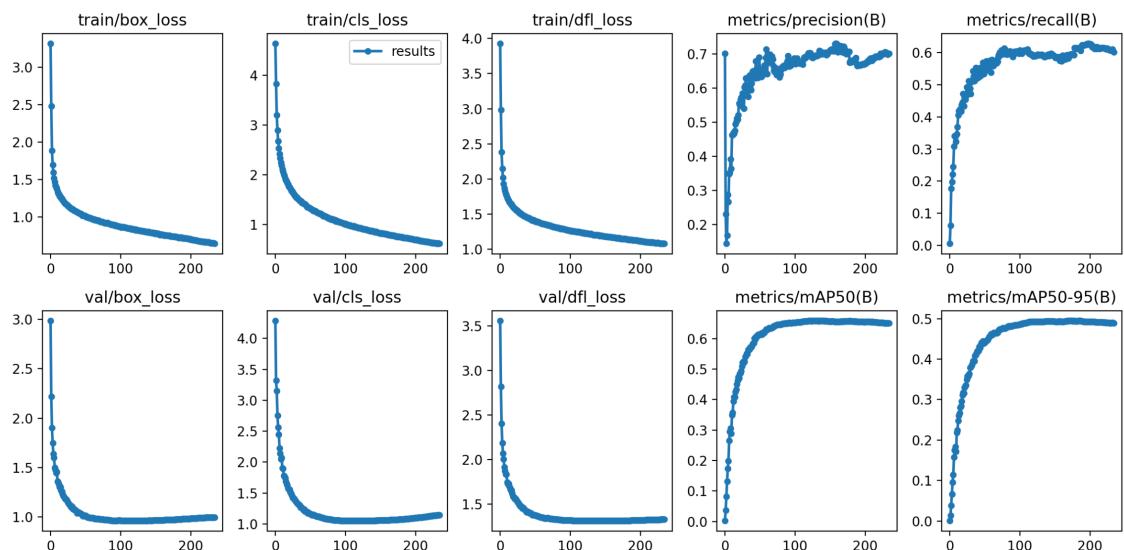


图 3.4 YOLOv8n-seg 在 COCO 数据集上的收敛曲线

通过观察模型在 COCO 数据集的训练集和验证集上的 loss 曲线可以发现 YOLOv8n-seg 模型的 loss 值，分类损失和置信度损失的值均保持在较低的状态。可以看到 YOLOv8n-seg 模型在精确率和召回率两个指标上表现很好，收敛曲线也相对平滑。

3.2 模型的部署

如何将训练好的模型部署到 SLAM 系统当中，这是一个问题，最近也存在一些用 YOLO 作为目标检测模型加入 SLAM 中的系统，但大部分都是通过离线的方式实现的，没有办法实现实时在线的进行目标检测和实时定位建图，这恰恰违背了 SLAM 的初衷，SLAM 的实时性也是系统需要考虑的主要问题之一，通过 Python 和 C++ 的通信或者通过 ROS 方法实现的 SLAM 系统在实时性上大打折扣。所以，本系统将训练好的 YOLOv8n-seg 模型转换成 ONNX 模型，并通过 opencv-dnn 进行推理，实现 YOLOv8n-seg 模型在 C++ 上的部署，为后续动态 SLAM 的优化作准备。

具体来说，在读取 ONNX 模型以后，需要利用 OpenCV(C++)，实现 YOLOv8 目标检测的整个流程，包括图像预处理、网络前向传播、结果解析、非极大值抑制和掩码生成。

①预处理输入图像： 清空输出向量，获取输入图像的宽度和高度，并对输入图像进行缩放和填充，使其适应网络输入尺寸，同时保持宽高比，最后，将预处理后的图像转换为网络输入格式。

②网络前向传播： 执行前向传播，获取网络的输出结果。

③结果分析： 解析网络输出，将网络输出结果转换为方便处理的矩阵形式，初始化用于存储检测结果的向量，遍历输出结果，提取置信度和边界框，最后将结果保存到向量中。

④非极大值抑制 (NMS)： 对检测结果进行非极大值抑制，过滤掉多余的检测框。

⑤生成最终输出： 得到最终的输出结果，包括目标类别 ID、置信度、边界框、旋转矩形框、矩形框内的掩码以及姿态关键点，值得注意的是，计算 mask 的方式是在得到的矩形框中进行 mask 得到的，所以这大大加快了实例分割的速度。

我们将得到的结果进行保存并传入 ORB-SLAM3 系统当中，对动态场景下的 SLAM 进行优化。

4 动态场景下基于 YOLOv8 优化的视觉 SLAM 系统设计

4.1 视觉 SLAM 系统的选择

现在主流的视觉 SLAM 系统有两种，分别是基于特征点法的视觉 SLAM 和基于直接法的视觉 SLAM，其中前者的实时性较好但是最终的建图效果较为一般，后者在处理速度上慢很多但是最后的建图效果较好，这是由于特征点法只需要对提取的特征点进行处理而直接法需要对整个图像的像素进行处理。在本系统中，我们主要考虑 SLAM 系统的实时性，所以我们选择基于特征点法的视觉 SLAM 系统来作为载体。

具体来说，我们选择使用 ORB-SLAM3 来作为主要系统，利用在上一章训练好的 YOLOv8 模型进行动态特征点的剔除，使其能够面对更加复杂的现实场景。

ORB-SLAM3 包括以下几个部分：

①Atlas： 也就是多地图系统，由一系列离散的地图组成。其中有一个 active map，tracking 线程基于这个地图进行定位，local map 线程会把新的关键帧加到这个地图里；除此以外还有很多 non-active map；还有一个 DBoW2 关键帧数据集用于重定位，回环检测和地图合并。

②Tracking： 处理传感器数据；通过最小化重投影误差实时计算最新位姿；选择关键帧；VI 模式下，计算 body 速度和 IMU 误差；如果跟丢了，在 Atlas 多图系统中进行重定位，成功了就切换对应地图，失败了就新建一个地图。

③Local Mapping： 把新的关键帧和地图点加到 active map 里，删掉多余点和多余关键帧；在最新的滑窗内进行 BA 优化地图；初始化时估计 IMU 参数。

④Loop 和 Map Merging： 以关键帧的频率搜索 active map 和 Atlas 里其它 map 的共同区域，如果这个共同区域属于 active map，就搞一次回环矫正，如果属于别的 map，这两个 map 合并，然后变成 active map；回环矫正后，会新建一个线程进行全局 BA 优化位姿。

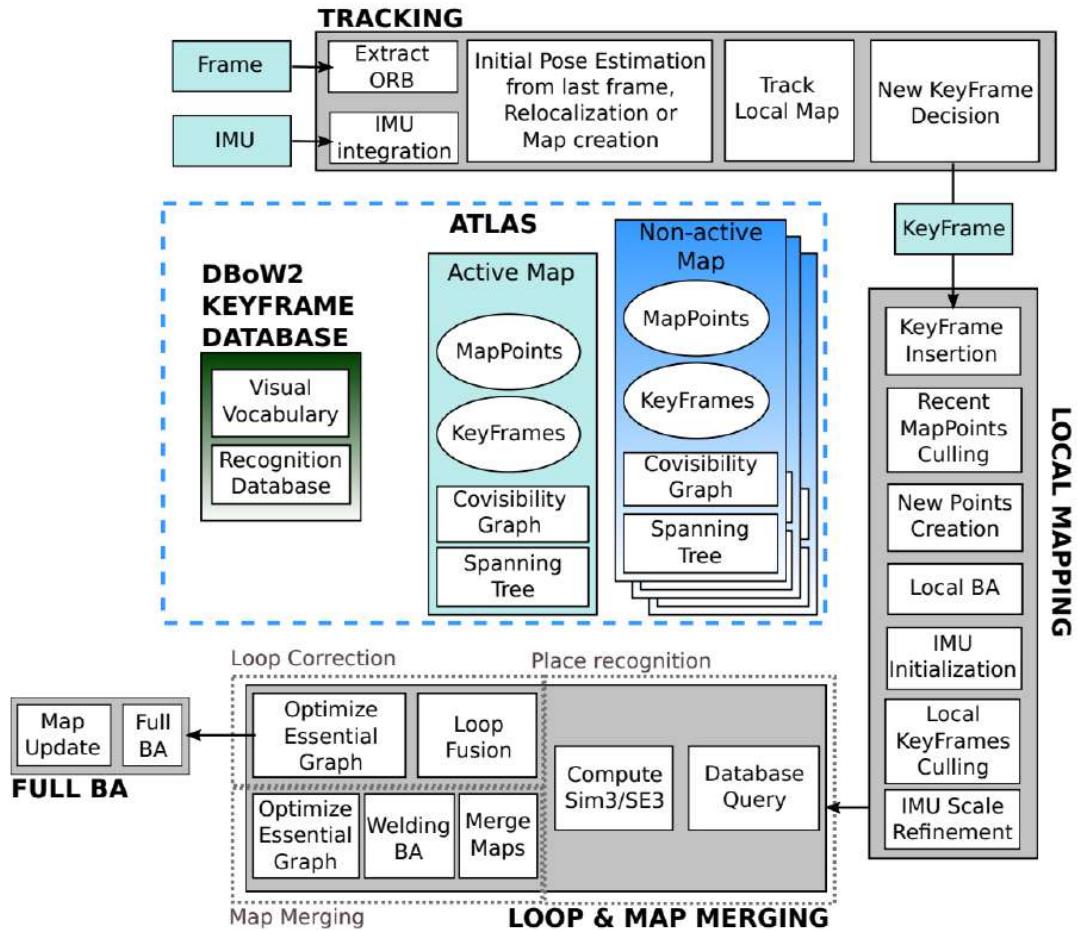
图 4.1 ORB-SLAM3 框架结构^[1]

图4.1是ORB-SLAM3的总体流程。Tracking是前端，包括IMU预积分，关键帧的构建，然后是Local Mapping，这两步就构成了最基础的里程计。中间部分是它最具特色的Atlas多图系统，包括DBoW2词袋数据库，当前正在使用的Acitive Map，和历史保存的多个Inactive Map。一旦在Loop中发现了Place recognition，就会把Active Map和Loop Map进行merging，再进行Full BA。在本系统中，主要使用单目，双目和RGBD相机作为主要测试对象，不考虑IMU的使用。

4.2 多线程设计

考虑到SLAM系统的实时性要求较高，YOLOv8-seg模型在作目标检测和实例分割的时间较长，若对输入的每一帧图像先进行实例分割后在输入SLAM系统的话，会严重影响运行速度，所以本系统受到DS-SLAM系统的启发^[44]，采用多线程并发的方式，在系统对每一帧的图像进行ORB特征提取的同时，插入一个并行线程，同时利用YOLOv8模型对该帧图片进行实例分割，并对分割后的结果进

行动态物体的筛选，保留下得到的 mask 部分，插入线程后整体模型框架如图4.2所示：

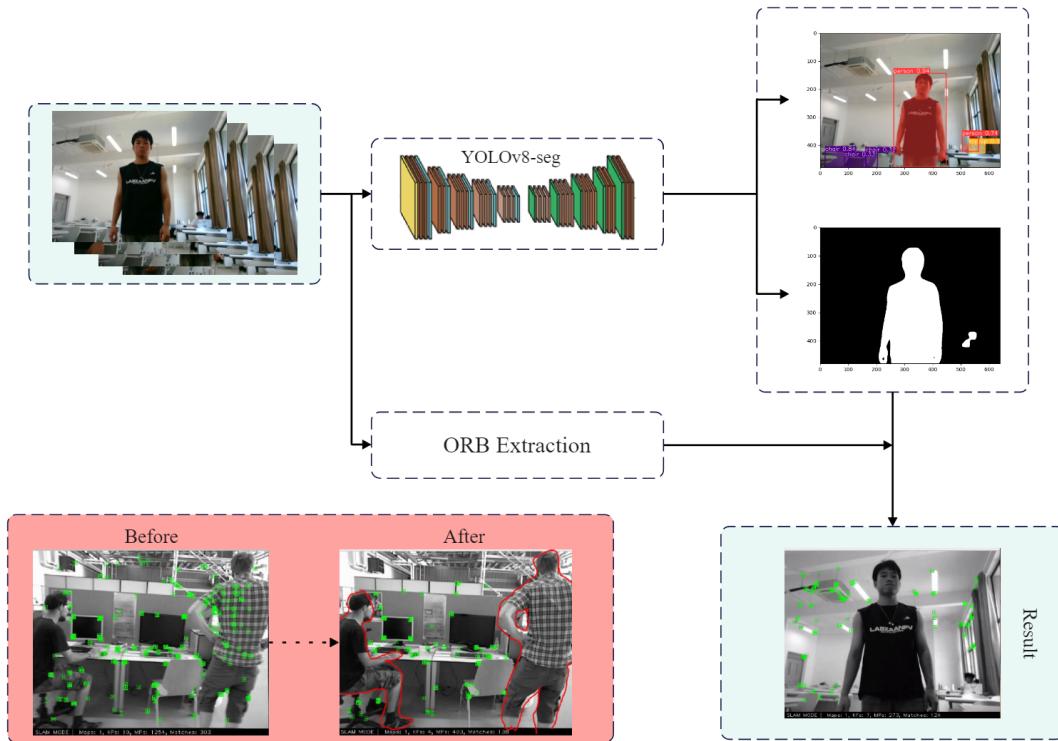


图 4.2 算法总体框架图

如图4.2所示，图片输入后，同时进行实例分割和 ORB 提取，正常来看，ORB 的提取是比模型的预测要快的，所以 ORB 提取完成后，需要等待模型预测完成之后才能够彻底结束线程，所以在系统设计的过程中，需要设置一个变量控制 ORB 线程等待 SEGMENT 线程，使其同时结束。

在上一章中得知，通过模型预测可以得到对应的掩码，接下来需要根据 mask 对该范围内的特征点进行剔除。具体做法是，在两个线程结束以后，我们得到了该帧图片的所有特征点和 mask 图片，接下来遍历该图中所有特征点，若该特征点在 mask 范围内，则将其设置为坏点，所有的坏点在后续的处理中都会被统一剔除，算法伪代码如算法4.1所示：

算法 4.1 动态点剔除算法

```

Data: ALL Feature Points
Result: Dynamic Feature Points  $S_p$ 
1 initialization;
2 for Each Frame do
3     // Processing for each frame
4     thread1: Extract ORB;
5     thread2: YOLOv8 Segmentation;
6     for each Point in Frame do
7         if Point is dynamic then
8             // Process dynamic points
9                 Add Point to  $S_p$ ;
10            end
11            else
12                // Process static points
13                Ignore Point;
14            end
15        end
16    end
17

```

4.3 数据集测试及分析

系统设计完成后，利用数据集对系统进行测试，动态场景下数据集的选择尤为重要，选择越复杂场景的数据集对 SLAM 系统的鲁棒性和泛化性考验更明显，对动态 SLAM 系统测试最常见的两个数据集是 TUM 的 rgbd_dataset_freiburg3_walking_xyz 和 rgbd_dataset_freiburg3_walking_halfsphere，所以在下面的测试中，这两个数据集作为该系统的测试数据集进行分析，在后面的阐述中，统一称其为数据集 1 和数据集 2。与此同时，未被优化过的 ORB-SLAM3 系统也将被用来测试作为对比。

首先将单帧图像拿出来进行对比，对比结果如图4.3所示。

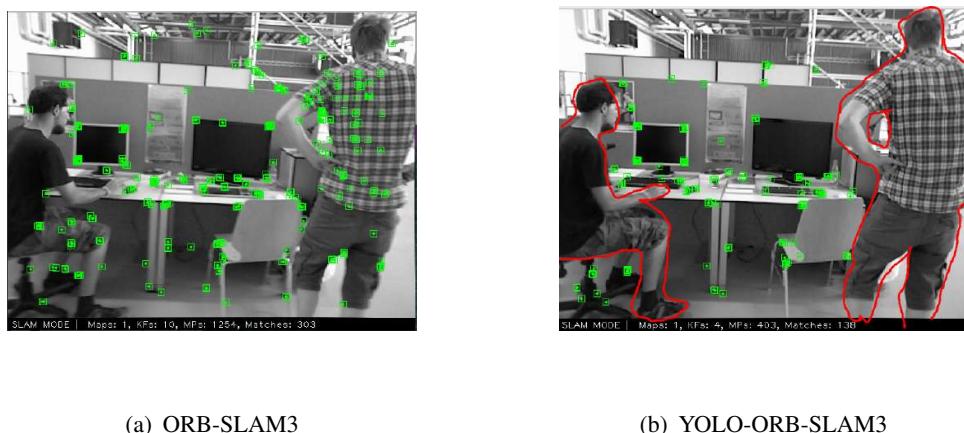


图 4.3 单帧图像对比结果图

由图4.3可以看出该帧图像中的动态景物是两个人，其中绿色的小方块代表的是提取到的特征点，红色实线代表的是被分割出来的动态景物。未优化前，人作为

动态景物却被提取了大量的特征点，这对 SLAM 的运行来说是致命的，在优化之后，两个人身上的特征点被全部剔除。

动态特征点的剔除对 SLAM 的影响最大的就是其位姿估计，图4.4和图4.6分别是 ORB-SLAM3 系统和 YOLO-ORB-SLAM3 系统在这两个数据集测试下轨迹的结果对比，图4.5和图4.7分别是其对应的轨迹的相对轨迹误差对比。

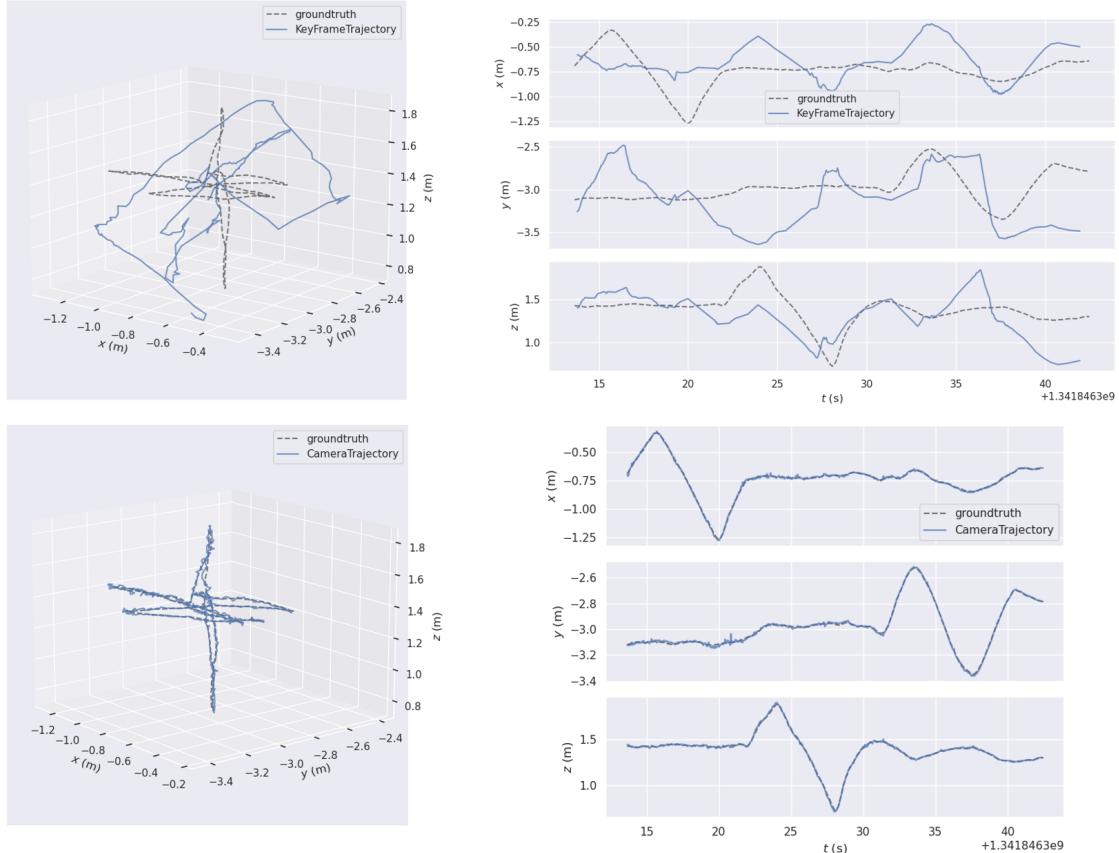


图 4.4 数据集 1 测试轨迹对比，前两个 Figure 是优化前的轨迹，后两个 Figure 是优化后的轨迹

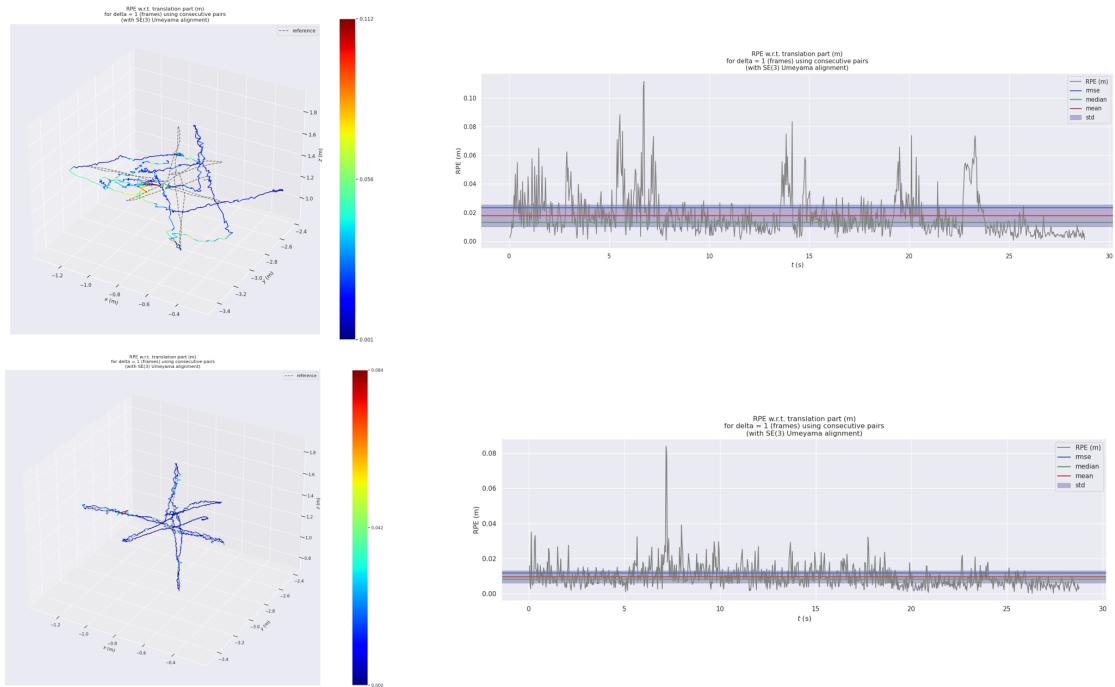


图 4.5 数据集 1 相对轨迹误差图

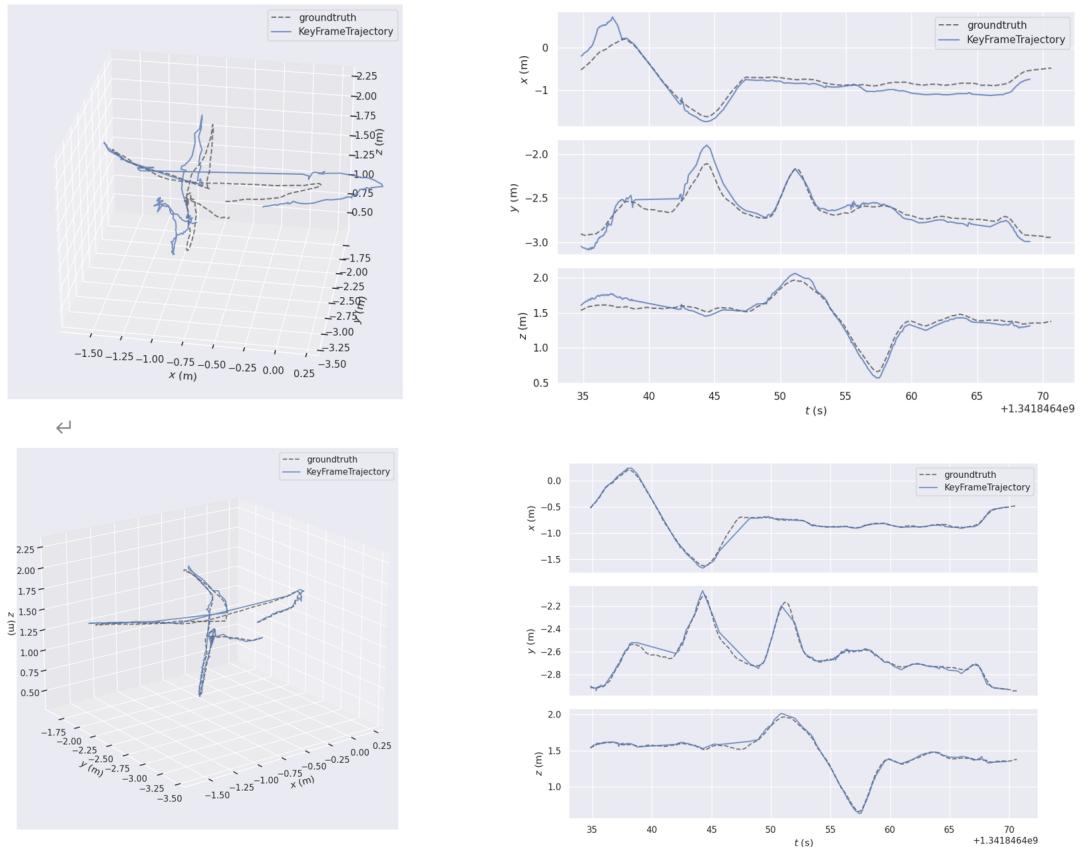


图 4.6 数据集 2 测试轨迹对比，前两个 Figure 是优化前的轨迹，后两个 Figure 是优化后的轨迹

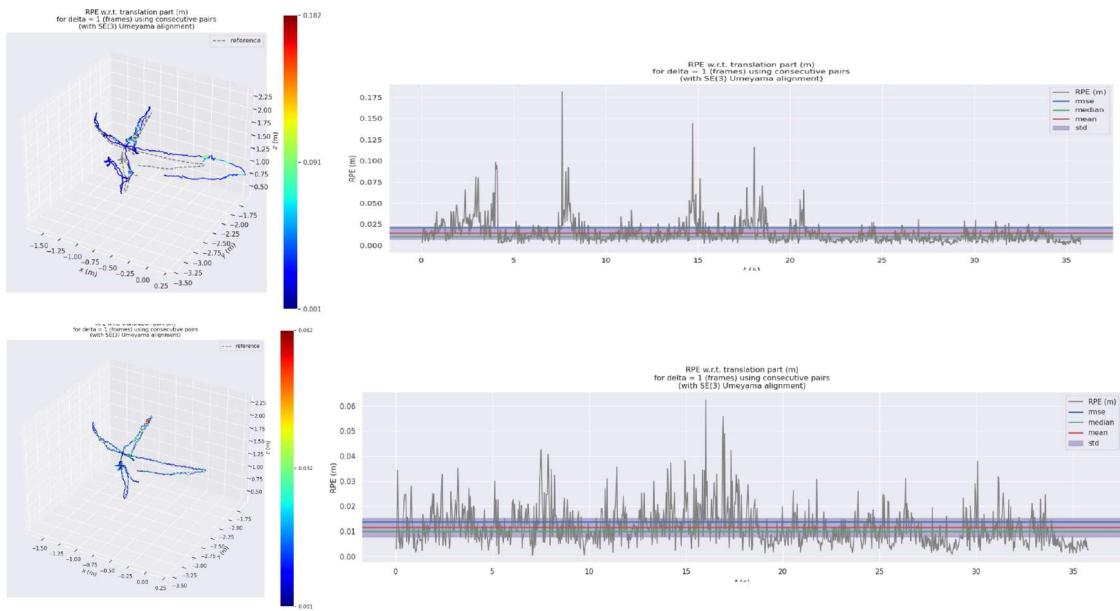


图 4.7 数据集 2 相对轨迹误差图

在轨迹对比中可以看出，本系统在剔除动态特征点后，对轨迹的优化作用非常好，未加入 YOLO 前 ORB-SLAM3 在数据集 1 和 2 上面的轨迹是乱的（数据集 2 上轨迹出现了漂移，数据集 1 轨迹错误过于明显）。但在经过优化后，轨迹和真实轨迹相差较小。表4.1和表4.2是现在主流的动态 SLAM 在这两个数据集上测试的绝对轨迹误差 (ATE) 和相对轨迹误差 (RPE) 结果对比：

表 4.1 测试数据集结果对比 1

TUM_walking_xyz	ORB-SLAM3	DS-SLAM	Dyna-SLAM	YOLO_ORB-SLAM
ATE	0.459	0.025	0.015	0.012
RPE	0.412	0.033		0.009

表 4.2 测试数据集结果对比 2

TUM_walking_halfsphere	ORB-SLAM3	DS-SLAM	Dyna-SLAM	YOLO_ORB-SLAM
ATE	0.351	0.030	0.025	0.025
RPE	0.355	0.030		0.011

从表格中可以看出，在算法轨迹精度上，YOLO-ORB-SLAM3 已经超过了 DS-SLAM 和 Dyna-SLAM^[44,47]，特别是在数据集 1 上。从各方面的测试验证表明，本系统能够应对动态场景下的各种变化情况，达到了很好的效果，在保持轨迹精度的同时，由于多线程的加入，也能够保证很好的实时性，接下来分析该系统在真实场景下进行测试的结果。

4.4 真实场景测试与应用

为了证明 YOLO-ORB-SLAM 的稳健性和实时性能，我们将该系统与双目相机设备连接，并在教室和宿舍环境中进行了大量实验。图像分辨率为 1920×1080 分辨率，由于电脑性能有限，我们将分辨率降低到 640×480 ，以方便测试。

相机参数：

- ① 双目，分辨率为 3840×1080
- ② 基线 b 为 60mm
- ③ 帧率 30 帧
- ④ 72° 小畸变，手动调焦

这里选取了摄像头提取到的几帧图像实例分割和特征点提取效果进行展示，图4.8中定性展示了异常值剔除的结果。从上行到下行的子图分别是原图像、目标检测结果、实例分割结果、ORB 特征提取结果。



图 4.8 教室场景测试结果

我们可以看到通过 YOLOv8 模型的实例分割对动态特征点进行剔除，不光分割出了前景中的人物，背景当中人物也被分割得到，可以得到模型训练效果显著，

在 ORB 提取的过程中，图中两个人对应的特征点也全部被剔除，由此可知，SLAM 系统在真实环境中也能表现出更好的鲁棒性和稳定性。

最终，我们在有人物走动的情况下对教室进行相机定位和稀疏点云地图重建，结果如图4.9。

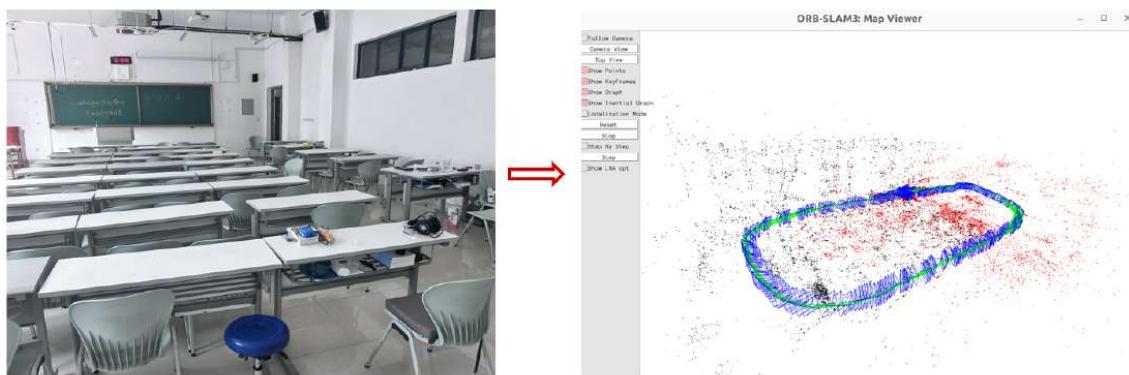


图 4.9 教室稀疏点云地图

5 全文总结与展望

5.1 全文总结

随着计算机运算能力的提升、机器人控制技术的发展以及图像和深度学习技术的进步，各种能够自主感知环境进行导航或分析建图的机器人越来越多地应用于生产和生活中。基于视觉传感器的 SLAM 算法由于成本低、复杂度低，已成为机器人自主导航和环境建图的主要方法。而深度学习在多种视觉任务中的鲁棒性和准确度远超传统方法，使得其与视觉 SLAM 的结合成为当前研究的热点。本论文主要设计了一个基于 YOLO 优化的视觉 SLAM 系统，在动态场景下提升了轨迹的精度，提升了算法的鲁棒性和稳定性。

论文主要工作如下：

- (1) 训练并部署了一个 YOLOv8 的实例分割模型，在实例分割的过程中对其进行优化，基于 YOLO 的目标检测特性，语义分割不需要在全图上面进行，而只需在已检测出的目标框中进行实例分割，这大大提升了分割速度，有利于提高动态语义 SLAM 的实时性和效率。
- (2) 基于已有的实例分割模型，在传统的 ORB-SLAM3 上进行改进，在原本的线程当中插入一个全新的线程，该线程利用训练好的 YOLOv8-seg 模型进行实例分割，并筛选出所有实例当中的动态实例，计算所有动态实例融合后的实例分割掩码，利用 Mask 掩码对其中的动态特征点进行筛选和剔除，从而避免其在追踪的过程中被使用，而导致轨迹漂移。
- (3) 该系统在主流数据集和真实场景下进行了复杂多样的测试，表现出了良好的性能和效果，系统表现出良好的实时性和稳定性。

5.2 工作展望

视觉 SLAM 作为当前工业应用的热点问题，吸引了众多学者和研发人员的关注。新的深度学习方法不断被引入视觉 SLAM 算法，不同模块和渠道的结合提升了系统整体性能。在视觉里程计中结合深度学习进行位姿估计方面，仍存在许多需要改进的问题。基于前面的研究内容，本文提出以下几点对未来工作的展望：

- (1) 在本文中提到的动态场景，其实理解起来是片面的，真实环境极其复杂，什么是动态的什么是静态的其实机器很难作出判断，人虽然是动态的，但如果是在休息的人，正在座位上工作的人，那么他应该也可以被认为是静态的，其对应

提取到的特征点应该被保留。而像椅子，杯子这种也并非是完全静态的，他可以在某个时刻被移动，所以可以称之为准静态，这类物体若被移动，那应该视其为动态的，上一时刻该物体的特征点应该被剔除，这一时刻的特征点应该被重新提取，所以说，仅靠实例分割模型果断的分割动态物体这样虽然可以解决大部分问题，但也是存在问题的。

(2) 在复杂场景中，若动态物体较多，则剔除的特征点对应也变多，这样会导致提取到的特征点偏少，也会影响位姿估计和跟踪，如何解决优化后特征点过少的问题，也是需要继续探索。

(3) 通过模型优化后的动态 SLAM 模型，运行效率同样受到影响，模型的预测速度是很大的限制因素，通常来讲 ORB 提取速度是比模型预测速度要快的，即使是加入了多线程，还是需要等待模型预测运行的时间，所以这一类 SLAM 系统要提高运行速度，根本上还是需要提高模型预测的速度，这也是这类系统的瓶颈所在。

参考文献

- [1] Campos C., Elvira R., Rodríguez J.J.G., Montiel J.M., Tardós J.D. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam[J]. *IEEE Transactions on Robotics*, 2021, 37(6): 1874-1890.
- [2] Lowe D.G. Object recognition from local scale-invariant features[A]. *Proceedings of the seventh IEEE international conference on computer vision: volume 2[C]*. IEEE, 1999: 1150-1157.
- [3] Rublee E., Rabaud V., Konolige K., Bradski G. Orb: An efficient alternative to sift or surf[A]. *2011 International conference on computer vision[C]*. IEEE, 2011: 2564-2571.
- [4] Bay H., Ess A., Tuytelaars T., Van Gool L. Speeded-up robust features (surf)[J]. *Computer vision and image understanding*, 2008, 110(3): 346-359.
- [5] Hausler S., Garg S., Xu M., Milford M., Fischer T. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition[A]. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition[C]*. 2021: 14141-14152.
- [6] Dusmanu M., Rocco I., Pajdla T., Pollefeys M., Sivic J., Torii A., et al. D2-net: A trainable cnn for joint description and detection of local features[A]. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition[C]*. 2019: 8092-8101.
- [7] Garg S., Milford M. Seqnet: Learning descriptors for sequence-based hierarchical place recognition[J]. *IEEE Robotics and Automation Letters*, 2021, 6(3): 4305-4312.
- [8] Arandjelovic R., Gronat P., Torii A., Pajdla T., Sivic J. Netvlad: Cnn architecture for weakly supervised place recognition[A]. *Proceedings of the IEEE conference on computer vision and pattern recognition[C]*. 2016: 5297-5307.
- [9] Oprea S., Martinez-Gonzalez P., Garcia-Garcia A., Castro-Vargas J.A., Orts-Escalano S., Garcia-Rodriguez J., et al. A review on deep learning techniques for video prediction[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 44(6): 2806-2826.
- [10] 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.
- [11] Biber P., Straßer W. The normal distributions transform: A new approach to laser scan matching [A]. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453): volume 3[C]*. IEEE, 2003: 2743-2748.
- [12] Diosi A., Kleeman L. Laser scan matching in polar coordinates with application to slam[A]. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems[C]*. IEEE, 2005: 3317-3322.
- [13] Davison A.J., Reid I.D., Molton N.D., Stasse O. Monoslam: Real-time single camera slam[J]. *IEEE transactions on pattern analysis and machine intelligence*, 2007, 29(6): 1052-1067.

- [14] Civera J., Davison A.J., Montiel J.M. Inverse depth parametrization for monocular slam[J]. IEEE transactions on robotics, 2008, 24(5): 932-945.
- [15] Mouragnon E., Lhuillier M., Dhome M., Dekeyser F., Sayd P. Real time localization and 3d reconstruction[A]. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06): volume 1[C]. IEEE, 2006: 363-370.
- [16] Klein G., Murray D. Parallel tracking and mapping for small ar workspaces[A]. 2007 6th IEEE and ACM international symposium on mixed and augmented reality[C]. IEEE, 2007: 225-234.
- [17] Strasdat H., Montiel J.M., Davison A.J. Visual slam: why filter?[J]. Image and Vision Computing, 2012, 30(2): 65-77.
- [18] 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.
- [19] Mur-Artal R., Tardós J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgbd cameras[J]. IEEE transactions on robotics, 2017, 33(5): 1255-1262.
- [20] Mur-Artal R., Tardós J.D. Visual-inertial monocular slam with map reuse[J]. IEEE Robotics and Automation Letters, 2017, 2(2): 796-803.
- [21] Kannala J., Brandt S.S. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses[J]. IEEE transactions on pattern analysis and machine intelligence, 2006, 28(8): 1335-1340.
- [22] Tsai R. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses[J]. IEEE Journal on Robotics and Automation, 1987, 3(4): 323-344.
- [23] Silveira G., Malis E., Rives P. An efficient direct method for improving visual slam[A]. Proceedings 2007 IEEE International Conference on Robotics and Automation[C]. IEEE, 2007: 4090-4095.
- [24] Silveira G., Malis E., Rives P. An efficient direct approach to visual slam[J]. IEEE transactions on robotics, 2008, 24(5): 969-979.
- [25] Engel J., Schöps T., Cremers D. Lsd-slam: Large-scale direct monocular slam[A]. European conference on computer vision[C]. Springer, 2014: 834-849.
- [26] Engel J., Stückler J., Cremers D. Large-scale direct slam with stereo cameras[A]. 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)[C]. IEEE, 2015: 1935-1942.
- [27] Forster C., Pizzoli M., Scaramuzza D. Svo: Fast semi-direct monocular visual odometry[A]. 2014 IEEE international conference on robotics and automation (ICRA)[C]. IEEE, 2014: 15-22.
- [28] Forster C., Zhang Z., Gassner M., Werlberger M., Scaramuzza D. Svo: Semidirect visual odometry for monocular and multicamera systems[J]. IEEE Transactions on Robotics, 2016, 33(2): 249-265.

- [29] Kendall A., Grimes M., Cipolla R. Posenet: A convolutional network for real-time 6-dof camera relocalization[A]. Proceedings of the IEEE international conference on computer vision[C]. 2015: 2938-2946.
- [30] Tateno K., Tombari F., Laina I., Navab N. Cnn-slam: Real-time dense monocular slam with learned depth prediction[A]. Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2017: 6243-6252.
- [31] Ali A.M., Nordin M.J. Sift based monocular slam with multi-clouds features for indoor navigation [A]. TENCON 2010-2010 IEEE Region 10 Conference[C]. IEEE, 2010: 2326-2331.
- [32] Zhu D.x. Binocular vision-slam using improved sift algorithm[A]. 2010 2nd International Workshop on Intelligent Systems and Applications[C]. IEEE, 2010: 1-4.
- [33] Zhang Z., Huang Y., Li C., Kang Y. Monocular vision simultaneous localization and mapping using surf[A]. 2008 7th World Congress on Intelligent Control and Automation[C]. IEEE, 2008: 1651-1656.
- [34] Wang T.C., Chen C.H. Improved simultaneous localization and mapping by stereo camera and surf[A]. 2013 CACS International Automatic Control Conference (CACS)[C]. IEEE, 2013: 204-209.
- [35] Rosten E., Drummond T. Machine learning for high-speed corner detection[A]. Computer Vision-ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9[C]. Springer, 2006: 430-443.
- [36] Calonder M., Lepetit V., Strecha C., Fua P. Brief: Binary robust independent elementary features [A]. Computer Vision-ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11[C]. Springer, 2010: 778-792.
- [37] Liu L., Ouyang W., Wang X., Fieguth P., Chen J., Liu X., et al. Deep learning for generic object detection: A survey[J]. International journal of computer vision, 2020, 128: 261-318.
- [38] Zou Z., Chen K., Shi Z., Guo Y., Ye J. Object detection in 20 years: A survey[J]. Proceedings of the IEEE, 2023, 111(3): 257-276.
- [39] Dalal N., Triggs B. Histograms of oriented gradients for human detection[A]. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05): volume 1[C]. IEEE, 2005: 886-893.
- [40] LeCun Y., Bengio Y., Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
- [41] Zaidi S.S.A., Ansari M.S., Aslam A., Kanwal N., Asghar M., Lee B. A survey of modern deep learning based object detection models[J]. Digital Signal Processing, 2022, 126: 103514.
- [42] Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection[A]. Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2016: 779-788.

- [43] 高翔, 张涛, 刘毅, 颜沁睿. 视觉 SLAM 十四讲: 从理论到实践[M]. 视觉 SLAM 十四讲: 从理论到实践, 2019.
- [44] Yu C., Liu Z., Liu X.J., Xie F., Yang Y., Wei Q., et al. Ds-slam: A semantic visual slam towards dynamic environments[A]. 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)[C]. IEEE, 2018: 1168-1174.
- [45] 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.
- [46] Yu F., Chen H., Wang X., Xian W., Chen Y., Liu F., et al. Bdd100k: A diverse driving dataset for heterogeneous multitask learning[A]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition[C]. 2020: 2636-2645.
- [47] Bescos B., Fácil J.M., Civera J., Neira J. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.

致谢

“凌晨四点钟，我看海棠花未眠。”

总觉得到致谢部分应该没有那么多条条框框了吧，毕业季总是有些伤感，想写一些大学四年来自内心的感受，毕竟要留档一辈子的，也祝贺我顺利度过了人生的这一个阶段。

当然是要先致谢了，首先，我应该感谢党和国家，我们生活在如此和平的年代，受党和国家的庇护，我深感幸福。其次，我要感谢我的母校，给了我一个无比宽广的平台，四年前，一个少年拿着沉重的行李包袱来到了这里，那是我第一次一个人出远门……四年后的我，好像什么都变了，又什么都没变。然后，我要感谢这四年对我帮助颇深的宁少尉老师，他随性自然，落落大方，给人一种特别亲近的感觉，那也是我第一次改变对大学老师的刻板印象。这四年不管是学业上出现问题，还是心理上出现问题，他都愿意开导我，鼓励我。我还要感谢我的毕设指导老师马学森副教授，这一年里他始终对我们都很负责，鼓励我做我想做的方向，给我关键性的指导，在完成毕设的整个过程中都支持我。最后，我要感谢我的父母和家人，他们是我永远坚实的后盾，无论我做什么都会无条件支持我，做错事了会安慰我，他们永远都站在了我这边。

最后呢，我想感谢一下我自己，感谢自己这四年仍旧能够如此努力不懈怠，似乎也少了些快乐。四年来的性格似乎发生了很大的转变，但我似乎更喜欢以前的自己，可能这也是一个过程吧。原来一直想要的是更加努力，更加优秀，但在这四年里，我似乎明白了一个道理，可能也不是道理，是事实，就是我们大部分人都只是普通人，而我呢，应该也就是这些普通人里面的一个吧。所以呢，我希望后面的日子能够多一些快乐，少一些焦虑，放下心中那么多的负担，顺其自然，继续往下走下去。

少年与爱永不老去，即便披荆斩棘，丢失怒马鲜衣！


作者：

2024年5月24日