

分类号: TP242.6

密 级: 公开

学校代码: 10363

学 号: 2210330155



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题目 基于语义 VSLAM 的无人机
自主导航系统设计

论文作者 何宇豪

校内导师 王冠凌（教授）

校外导师 毛德平

专业学位类别 电子信息（085400）

研究方向（领域） 控制工程与自动化技术

论文提交日期: 2024 年 5 月 31 日

分类号: TP242.6

单位代码: 10363

密 级: 公 开

学 号:2210330155

题 目 基于语义 VSLAM 的无人机自主导航系统设计

题 目 **Design of Autonomous Navigation System for Unmanned
Aerial Vehicles Based on Semantic VSLAM**

学生姓名: 何宇豪

校内导师: 王冠凌

校外导师: 毛德平

专 业: 电子信息

研究方向: 控制工程与自动化技术

论文答辩日期: 2024.5.16

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：何字豪

日期：2024年5月31日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 _____ 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：何宇豪

指导教师签名：王迎庆

日期：2024 年 5 月 31 日

日期：2024 年 5 月 31 日

基于语义 VSLAM 的无人机自主导航系统设计

摘要

在当今科技高速发展的时代，四旋翼无人机凭借其灵活性和机动性在许多领域发挥着越来越重要的作用。有着自主导航功能的无人机展现了其在提高响应速度、降低操作风险以及增强任务执行能力方面的巨大潜力。然而，传统的无人机导航系统在理解复杂环境中的语义信息上存在局限，且在动态室内环境中，这些系统的导航效率极其有限，甚至完全失效。对此，本文设计了一种基于语义 VSLAM(Visual Simultaneous Localization And Mapping)的无人机自主导航系统。本文的主要工作和创新点如下：

(1) VSLAM 技术通常假设环境是静态不变的，但这种假设在动态的室内环境中往往表现不佳，导致定位精度下降。提出了一种基于语义信息实现动态特征点剔除的视觉里程计，并与 ORBSLAM3 相结合，使用改进的 YOLOv5_GBCS 算法实现语义提取，完成动态物体特征点的剔除，有效解决动态的室内场景中性能不佳的问题并提高定位精度，可用于无人机的精确室内导航。

(2) 在复杂的环境中，传统路径规划算法所规划的轨迹往往无法适应环境变化，容易陷入局部最优解导致规划失败或效果不佳，且规划速度慢。为了解决这些问题，提出了一种改进的 PSO 路径规划算法，并实现了轨迹的平滑功能，以使飞行轨迹更符合飞控系统的要求。

(3) 在创建语义地图的过程中, 传统的做法是将二维标签直接关联到三维点云上, 这会导致辨识物体时出现模糊。针对上述问题, 本系统将 YOLOv5_GBCS 算法获取到的 2D 语义检测框内像素点映射获得 3D 点云集, 与超体素点云分割后的三维稠密点云进行几何一致性检查, 并较高效率地实现三维语义地图构建。

(4) 实验平台搭建与方法验证: 在 Linux 系统下, 部署 ROS (Robot Operating System), 测试 VSLAM 算法、路径规划算法和整个系统的可行性和稳定性。然后自主搭建了以 Pixhawk4、Intel RealSense D435i 和 Jetson Xavier NX 为组件的无人机硬件平台。

最后, 在软硬件控制平台上实现了基于语义地图的高效自主导航系统。系统的主要功能包括物体识别、语义地图构建、自主导航等。本文做了大量实验, 验证了本系统算法的准确性与有效性。

关键词: 四旋翼无人机, VSLAM, ORBSLAM3, YOLOv5, 路径规划

DESIGN OF UAV AUTONOMOUS NAVIGATION SYSTEM BASED ON SEMANTIC VSLAM

ABSTRACT

In today's era of rapid development of science and technology, four-rotor UAVs are playing an increasingly important role in many fields with their flexibility and maneuverability, and UAVs with autonomous navigation function show their great potential in improving response speed, reducing operational risks and enhancing mission execution. However, traditional UAV navigation systems are limited in understanding semantic information in complex environments, and in dynamic indoor environments, the navigation efficiency of these systems is extremely limited, or even completely ineffective. In this paper, an autonomous navigation system of UAV based on semantic VSLAM is designed. The main work and innovation of this paper are as follows:

(1) VSLAM technology usually assumes that the environment is static, but this assumption often performs poorly in dynamic indoor environment, which leads to the decline of positioning accuracy. A visual odometer based on semantic information and ORBSLAM3 is proposed. The improved YOLOv5_GBCS algorithm is used to realize semantic extraction and eliminate dynamic object feature points, which can effectively solve the problem of poor performance in dynamic indoor scenes and improve

positioning accuracy. It can be used for accurate indoor navigation of UAV.

(2) In a complex environment, the trajectory planned by the traditional path planning algorithm is often unable to adapt to the changes of the environment, and it is easy to fall into the local optimal solution, resulting in planning failure or poor effect, and the planning speed is slow. In order to solve these problems, an improved PSO path planning algorithm is proposed and the trajectory smoothing function is realized to make the flight trajectory conform to the requirements of the flight control system.

(3) In the process of creating semantic maps, the traditional approach is to directly associate 2D labels with 3D point clouds, which can lead to blurring when identifying objects. To solve the above problems, this system maps the pixels in the 2D semantic detection box obtained by YOLOv5_GBCS algorithm to obtain 3D point cloud sets, and checks the geometric consistency with the 3D dense point clouds after segmentation by super-voxel point clouds, and realizes 3D semantic maps with higher efficiency.

(4) Experimental platform construction and method verification: Under Linux System, ROS (Robot Operating System) was deployed to test the feasibility and stability of VSLAM algorithm, path planning algorithm and the whole system. Then it independently built a drone hardware platform with Pixhawk4, Intel RealSense D435i and Jetson Xavier NX as components.

Finally, an efficient autonomous navigation system based on semantic maps is implemented on the software and hardware control platform. The main functions of the system include object recognition, semantic map construction, autonomous navigation, etc. In this paper, a large number of experiments have been done to verify the accuracy and effectiveness of the algorithm.

Keywords: Quad-rotor UAV, VSLAM, ORBSLAM 3, YOLOv5, Path Planning

目录

摘要	I
ABSTRACT.....	III
第 1 章 绪论.....	1
1.1 课题研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 语义 VSLAM 技术研究现状.....	2
1.2.2 无人机目标检测神经网络研究现状.....	4
1.2.3 无人机路径规划研究现状.....	5
1.3 论文主要研究内容与章节安排.....	6
第 2 章 语义 VSLAM 的理论基础.....	8
2.1 视觉 SLAM 理论基础.....	8
2.1.1 前端视觉里程计.....	8
2.1.2 后端优化.....	10
2.1.3 闭环检测.....	11
2.1.4 地图构建.....	12
2.2 YOLOv5 理论基础.....	12
2.2.1 Backbone 主干网络.....	14
2.2.2 Neck 颈部网络	14
2.2.3 Head 输出网络	15
2.2.4 损失函数.....	16
2.3 本章小结.....	17
第 3 章 基于语义信息实现动态特征点剔除的视觉里程计.....	18
3.1 ORBSLAM3 算法	18
3.1.1 ORBSLAM3 框架	18
3.1.2 ORBSLAM3 前端视觉里程计	19
3.1.3 ORBSLAM3 后端非线性优化	20
3.2 改进 YOLOv5 的语义检测.....	21

3.2.1 改进的 YOLOv5 框架.....	21
3.2.2 网络结构轻量化设计.....	22
3.2.3 CBAM 注意力机制.....	24
3.2.4 基于 BiFPN 的特征融合网络.....	26
3.2.5 针对小目标增加预测头.....	27
3.2.6 NMS 的改进.....	27
3.2.7 对比实验分析.....	28
3.2.8 消融实验分析.....	30
3.3 基于语义实现动态特征点剔除.....	33
3.3.1 光流法追踪.....	33
3.3.2 对极几何约束.....	35
3.3.3 动态点剔除效果验证.....	37
3.4 本章小结.....	39
第 4 章 基于改进的 PSO 算法的无人机平滑路径规划	40
4.1 粒子群优化算法.....	40
4.1.1 粒子群优化(PSO).....	40
4.1.2 量子粒子群优化 (QPSO).....	42
4.2 成本分析.....	44
4.3 改进的 PSO 算法	44
4.4 平滑路径生成.....	47
4.5 实验结果与分析.....	49
4.5.1 基准函数测试.....	49
4.5.2 不同环境下的路径规划实验对比分析.....	53
4.6 本章小结.....	56
第 5 章 无人机建图与自主导航系统实验.....	57
5.1 自主导航系统设计.....	57
5.1.1 系统硬件设计.....	57
5.1.2 系统软件框架.....	59
5.2 语义 VSLAM 性能评估.....	61

5.2.1 基于 TUM RGB-D 数据集的性能评估	61
5.2.2 基于 BONN RGB-D 动态数据集的性能评估	64
5.3 语义 VSLAM 地图构建实验	66
5.4 自主导航实验	71
5.4.1 八叉树地图转换	71
5.4.2 室内导航实验	71
5.5 本章小结	75
第 6 章 总结与展望	76
6.1 论文的主要工作	76
6.2 下一步研究计划	77
参考文献	78
攻读硕士学位期间主要研究成果	86
致 谢	87

第 1 章 绪论

1.1 课题研究背景及意义

在近年来，随着计算机视觉、人工智能和机器人技术的飞速发展，智能无人机已经成为了多个领域内不可或缺的工具。在军事侦察、民用监测、工业检查、灾害响应、地图制作、农业监控等方面^[1]，无人机的使用不仅大大提高了工作效率，还扩展了人类活动的范围。特别是在需要精确地图信息、实时环境监测、灾难快速响应的场合，无人机展现出了其独特的优势和重要性。自主导航作为实现无人机广泛应用的核心技术^[2]，涵盖了无人机的自主定位、环境感知、决策规划等多个方面，其中 VSLAM 技术^[3]是其中的关键技术之一。



图 1-1 四旋翼无人机

VSLAM 技术通过捕获的视觉信息，估计设备在空间中的位置，并构建周围环境的三维地图。这一技术在机器人导航、增强现实等领域已经展现了广泛的应用潜力。然而，尽管传统的 VSLAM 技术为无人机在未知环境中的自主导航提供了基础，但其在应对复杂环境和动态变化方面仍存在明显的局限性。一方面，这些基于几何信息构建的地图缺乏对环境的深层次语义理解，这限制了无人机在复杂环境下的应用性能和适应能力。另一方面，动态环境中移动物体的存在可能会干扰关键帧的选择、特征匹配过程，导致地图构建和位置估计的准确性下降。此外，长时间跨度的环境变化对于 VSLAM 系统来说也是一个不小的挑战，因为这可能导致重定位困难，影响系统的稳定性和可靠性。

为了克服这些问题，研究者们提出了将深度学习和语义信息融入 VSLAM 的新方法，即语义 VSLAM。这种方法通过识别和理解图像中的语义信息，可

以更准确地处理动态物体和环境变化，从而提高系统的准确性和鲁棒性。通过深度学习模型，语义 VSLAM 能够有效识别图像中的动态物体，并将这些动态物体从定位和地图构建过程中排除或进行特殊处理。这不仅减少了动态环境带来的干扰，还使得无人机能够在更加复杂和不确定的环境中安全有效地导航。此外，引入语义信息还能丰富无人机对环境的理解，使其能够识别和区分不同的环境特征和对象。这种丰富的环境信息有助于无人机在执行特定任务时作出更加精准的决策和规划，如在农业监测中区分作物和杂草，在灾害响应中识别受损建筑和道路障碍等。通过这种方式，语义 VSLAM 不仅提高了无人机的定位和导航精度，还扩展了其在复杂环境下的应用能力。将语义 VSLAM 与无人机路径规划算法相结合，可以构建一个更加高效、智能的自主导航系统。这种系统能够根据环境的实时变化动态调整规划路径，有效避开障碍和动态障碍物，确保无人机安全高效地完成任务。例如，在城市环境中，无人机可以实时识别和规避行人和车辆，自动调整飞行路径；在森林火灾监测中，无人机可以根据火源的位置和扩散趋势调整监测区域，提高监测效率和精度。

综上所述，语义 VSLAM 技术的发展为无人机自主导航技术带来了新的突破，不仅显著提高了无人机在复杂环境中的导航精度和稳定性，还拓宽了其在各个领域的应用范围。未来，随着深度学习、人工智能技术的不断进步和优化，结合更先进的传感器和计算平台，无人机的自主导航技术将更加智能化、高效化，为社会的发展贡献更大的力量。

1.2 国内外研究现状

1.2.1 语义 VSLAM 技术研究现状

SLAM 技术最早由 Smith 和 Cheeseman 于 1986 年提出，至今已经发展了三十多年。2007 年，A.J. Davison 教授提出 MonoSLAM^[4]，这是第一个实时单目 VSLAM 系统，但它只能离线进行定位和地图构建。同年，Klein 等人提出了 PTAM^[5]系统，这是第一个使用非线性优化作为后端并并行化跟踪和地图构建过程的算法。然而，它只适用于小场景。2015 年 MurArtal 等人提出 ORBSLAM^[6]，基于 ORB 特征点估计姿态，采用词袋模型克服累积误差。2021 年，Campos 等

人提出了支持多摄像头的 VSLAM 算法 ORBSLAM3^[7]。该算法优化了地图初始化、重定位、闭环检测和关键帧选择等几个方面。

以上的 VSLAM 算法都假设了一个静止的环境，如果场景中有运动的物体，会降低定位和映射精度。为了解决这一问题，研究人员提出了一种基于假设大多数图像特征是静态的运动分割方法。Kim 等人^[8]基于初始图像深度构建静态背景，并在后续帧中去除非静态物体，但该方法在过滤行人方面存在局限性。随后，Sun 等^[9]提出了一种基于 RGB-D 数据作为预处理模块的运动去除方法，过滤掉运动对象。

近年来，随着深度学习的快速发展，对象检测和语义分割在准确性和速度上都有了很大的提高。Bescos 等人^[10]提出了 DynaSLAM，该 SLAM 系统首次使用 MaskR-CNN^[11]对输入图像进行语义分割，然后使用多视图几何来消除图像中的动态对象。钟等人^[12]提出了与 DynaSLAM 类似的 Detect SLAM，其中 SLAM 系统使用 SSD 对象检测神经网络来识别图像中的对象，然后通过特征匹配传播运动概率，增加影响区域以消除环境中动态点的影响。Yu 等人^[13]提出了一种 DS-SLAM 算法，通过将语义分割网络与光流方法相结合来消除动态特征的影响。吴等人使用 YOLOv3 目标检测算法来识别动态目标，并通过消除检测框中的异常值来消除动态目标的影响。Sheng 等^[14]利用 Mask R-CNN 算法对环境中高度动态的目标进行分割，并基于像素点的四个邻域确定动态区域。Soares 等^[15]直接将人视为室内环境中的主要动态目标。但这导致静态区域的错误拒绝和动态区域的拒绝不足。Dai 等^[16]使用 Delaunay 算法^[17]将特征点连接成多个三角形。然后，他们比较两个相邻帧中的相应边缘，以去除动态边缘，从而产生具有一致内部运动的多个连接域。基于空间中最大区域是静态的假设，较小的连通域被过滤掉。Yang 等^[18]计算了动态边在与特征点相连的所有边中所占的比例，将动态边占比大于 60%的点识别为动态特征点。这些方法提高了算法的精度，但它们仍然难以准确地拒绝动态区域内的动态特征。

总之，现有的 VSLAM 系统在高动态环境中运行时，鲁棒性或实时性不足。因此，在高动态室内环境下，同时增强 VSLAM 的鲁棒性和实时性是当前迫切需要解决的问题。

1.2.2 无人机目标检测神经网络研究现状

随着网络技术的快速发展,无人机图像目标检测技术具有广泛的应用范围,包括森林防火、农业信息、电力线检测、道路和桥梁损伤评估等。相较于自然环境视角下拍摄的图像,无人机航拍图像具有巨大的高空视角优势。但无人机拍摄图像却存在背景复杂度高、目标尺寸小、外观模糊等缺陷。同时,基于传统方法的目标检测技术对小目标检测精度低,对复杂背景下目标的适应性弱,容易发生漏检和误检。随着深度学习的发展,基于卷积神经网络的目标检测方法近年来迅速发展起来^[19],在检测精度和速度方面都超过了传统的目标检测方法,然而当前大多数目标检测模型仅仅适用于处理自然环境图像,如果采用同样的算法用于处理无人机航拍图像就会存在一些问题。首先,目前大多数目标检测模型无法满足实时检测需求,关键是在提高检测精度的同时减少计算量和参数量;其次,无人机航拍图像中含有大量小目标,其形态表征能力较弱,严重影响小目标的检测精度。

近些年来,YOLO 系列检测模型因其速度快,精度高而被广泛引用于无人机航拍图像目标检测。Liu 等^[20]提出 MTI-YOLO 用于无人机电力线路检查中的绝缘子等目标,在 YOLOv3Tiny 的基础上添加特征融合结构和 SPP 模块来扩展颈部网络,并且添加了主干网络的输出层。发现,这种模型在颈部网络的改进是多余的,需要优化网络结构。为了提高小目标检测精度,Oyku 等^[21]扩展了 YOLOv3 的主干输出层,以检测图像中不同尺度的物体,将原来的三个检测层增加到五个,这样的结构在特征融合部分起到一定的作用,但这也导致了检测模型参数量极大增加,从而增加了训练和计算成本。Wang 等^[22]提出了 LDS-YOLO 检测模型,考虑到无人机图像中小目标特征难以提取的问题,该模型在 YOLOv5 的基础上构造了一个新的特征提取模块,在 SPP 模块中引入 SoftPool 结构,并采用深度可分离卷积替代传统卷积。该模型的优点在于参数量较小,但由于训练时样本不足,很容易使模型无法学习到目标特征。为了解决电力巡检无人机检测速率与精度无法兼顾的问题,彭等人^[23]提出 ED-YOLO 目标检测算法,在 YOLOv4 算法上引入注意力机制和深度可分离卷积,使得模型检测速度大大提升,但伴随着精度的下降。为了提高无人机小目标检测精度,陈等^[24]提出了 AF-YOLO 模型,引入了自适应融合机制,在检测精度超过了原模型,但

是精度的提升是在牺牲检测性能的情况下获得的。

1.2.3 无人机路径规划研究现状

随着技术的不断进步,无人机已广泛应用于交通运输、救援、军事等各个领域。无人机路径规划是无人机任务规划系统中的一项关键任务,需要在特定的约束条件下获得从起始位置到预期目的地的安全高效的路径。与地面机器人路径规划相比,无人机在三维环境下的路径规划随着空间维数的增加,轨迹计算的复杂性也随之增加。无人机在飞行过程中受到建筑物、天气、自身能耗等因素的影响,如何规划一条更安全、更高效、更快速的飞行路径是目前研究的热点问题。传统算法如 A^* ^[25]、Dijkstra^[25]和快速探索随机树 RRT^[27]通过特定的策略获得可行的路径。然而,它们往往不是问题的最优解。

近几十年来,智能优化算法得到了全面发展,许多学者用启发式算法来研究这个问题,寻找合适的解决方案。如遗传算法 GA^[28]、蚁群优化 ACO^[29]、粒子群优化 PSO^[30]等,这些算法可以解决复杂环境下的飞行计划问题。Jun 等人^[31]从量子力学的角度对粒子群算法进行了研究。引入 Gbest 和 Mbest 的概念,使粒子位置和速度的更新具有随机性。Liu 等人^[32]介绍了 PSO 算法中的网络构造算子。在更新种群粒子的位置后,将算子应用于每个粒子,以克服 PSO 算法后期陷入局部最优和收敛速度慢的问题。Duan 等^[33]利用最大最小自适应蚁群优化算法,在动态不确定环境下重新规划无人机协同路径,确定无人机到达冲突点的时间,进而确定无人机的飞行轨迹和速度,避免静态威胁和弹出式威胁。Qu 等人^[34]提出了一种基于灰狼算法的强化学习算法,通过控制单个勘探、开采、几何调整和最优化调整四种操作积累的性能,自适应切换选择,效果优于其他改进的灰狼算法。刘等人^[35]在进化算法的基础上提出了一种无人机路径规划的进化算法,改变了 t 分布算法,有效解决了无人机动态路径规划中计算复杂度高、效率低的问题。麻雀搜索算法^[36]是 Xue 等人在 2020 年提出的一种新的启发式算法,是一种模仿麻雀捕食的群体智能算法。算法具有收敛快、择优能力强的特点。粒子群算法^[37]是由 James Kennedy 和 Russell Eberhart 提出的一种算法,具有收敛速度快、结构简单的特点。许多学者将其应用于各种优化场景,并取得了良好的效果。Li^[38]针对移动机器人在路径规划中存在的碰撞危险、路径不够

光滑、规划路径过长等问题，提出了一种 SLPSO 算法，实验结果表明该算法是有效可行的。Phung^[39]提出了一种用于无人机路径规划的改进离散粒子群算法 (DPSO)，利用并行计算的优势，通过定性初始化、随机变化和边缘交换来提高算法的性能。Foo^[40]提出了一种基于粒子群算法的三维无人机路径规划，该算法考虑了无人机的燃油消耗，飞过指定障碍物，跳过指定侦察目标。

尽管这些改进算法已被证明在解决优化问题上是有用的，但仍存在以下不足：改进的方法仅限于算法的局部优化，虽然在个别阶段的优化效果较好，但当优化问题的维数增加时，算法的性能会下降。此外，通常在路径规划阶段，如果不考虑运动学约束，算法规划的路径无法在实际情况下飞行，存在许多未处理的轨道拐点和缺乏连续平滑。

在无人机路径规划领域，目前的无人机仍然存在局部最优解问题，难以搜索到全局最优的平滑路径，无法获得更短的收敛路径和更快的规划速度。

1.3 论文主要研究内容与章节安排

本文着重探讨了如何通过融合语义信息于 VSLAM 技术中，进而实现语义地图的构建并改进路径规划算法，最终融合至无人机自主导航系统中，以增强无人机在复杂场景下的导航性能。并将整个系统部署到无人机平台，实现功能验证。全文共六章，各章节主要内容有如下：

第一章 绪论。介绍无人机自主导航的研究背景和意义，以及当前存在的问题和挑战，引出本文的研究目标和主要内容。分别概述了 VSLAM、无人机目标检测、无人机路径规划的研究现状，并介绍本文具体内容的章节结构。

第二章 语义 VSLAM 的理论基础。概述 VSLAM 算法的理论基础，包括前端视觉里程计、后端优化、闭环检测与地图构建，并分析 YOLOv5 算法的整体框架。

第三章 基于语义信息实现动态特征点剔除的视觉里程计。详细介绍本研究中使用的 VSLAM 算法的设计原理和实现过程，将无人机目标检测技术与 ORBSLAM3 相结合实现动态特征点剔除，并通过实验验证算法的有效性，提高 VSLAM 在动态环境下的准确性和鲁棒性。

第四章 基于改进的 PSO 算法的无人机平滑路径规划。针对复杂环境下无人

机的路径规划问题，提出了一种改进的 PSO 算法，详细说明了算法的实现流程，并且实现无人机平滑路径。实验结果证明了算法的有效性与稳定性。

第五章 无人机建图与自主导航系统实验。详细描述所设计的无人机自主导航系统的整体架构、功能模块和实现细节。包括无人机的硬件平台选择、传感器配置方案、控制系统设计思路等，并设计实验验证本文算法的实用性。

第六章 总结与展望。总结本研究的主要成果和贡献，分析研究中存在的不足之处和未来可能的研究方向。

第 2 章 语义 VSLAM 的理论基础

2.1 视觉 SLAM 理论基础

经典的 VSLAM 框架主要由前端和后端两部分组成，如图 2-1 所示。在前端，视觉里程计估计相邻图像或点的数据，形成局部地图，提供实时姿态估计，后端提供地图更新和优化。具体来说，成熟的视觉 SLAM 系统包括前端视觉里程计、后端优化、闭环检测和地图构建模块^[41]。

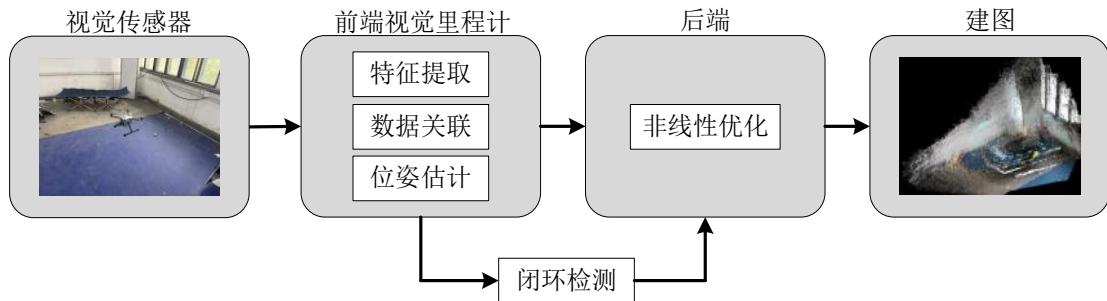


图 2-1 经典 VSLAM 框架

2.1.1 前端视觉里程计

VSLAM 的前端是对输入图像进行处理，获取摄像机运动时的运动关系，确定当前帧的位置。视觉里程计^[42]根据相邻图像的数据估计摄像机的运动，为后端提供更好的初始值。图 2-2 为视觉里程计的结构图。

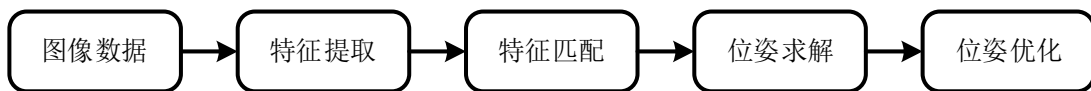


图 2-2 视觉里程计结构

在视觉里程计中，特征点的选择非常重要，因为它们是用来识别和匹配不同图像中的关键点的。选取特征点时，一般会考虑到它们的区分度、稳定性和重复性。特征描述符则是用来为这些特征点提供独特且可量化的表示，以便在不同图像中准确地匹配和识别。常见的描述符包括 Harris^[43]、SURF^[44]、SIFT^[45]、FAST^[46]和 ORB^[47]。例如 FAST 特征点提取效果如图 2-3 所示。

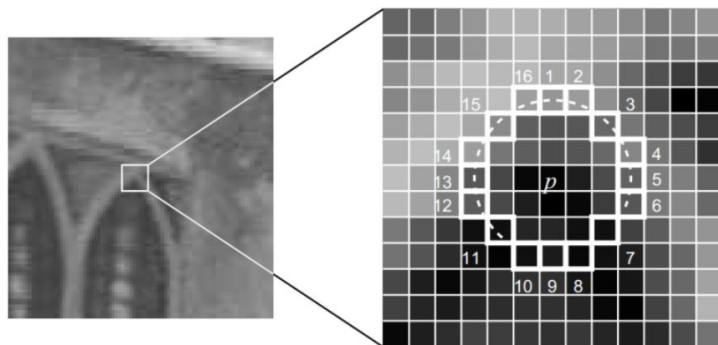


图 2-3 FAST 特征点

特征提取完成后就需要完成特征匹配，特征匹配是指在两帧或多帧图像之间找到对应的特征点。这一步骤对于后续的运动估计至关重要。特征匹配通常基于特征点的描述符，描述符是对特征点周围像素的一种编码表示，用于量化特征点的外观信息。常见的特征匹配策略包括：

暴力匹配^[48]: 对一帧图像中的每个特征点描述符与另一帧图像中所有特征点的描述符进行比较，选择最佳匹配。图 2-4 为特征点暴力匹配效果图。

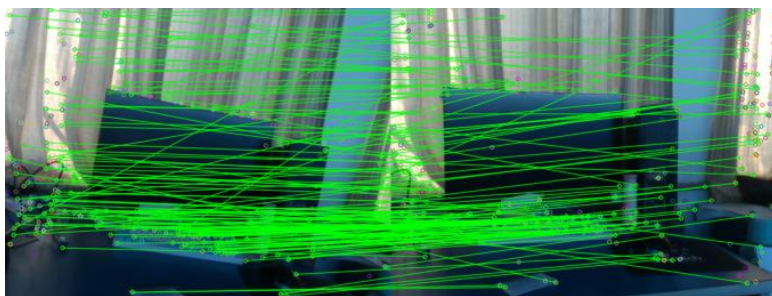


图 2-4 暴力匹配

FLANN 匹配^[49]: 一种近似最近邻搜索的方法，可以在大规模数据集中快速找到匹配点，提高匹配效率。图 2-5 为特征点暴力匹配效果图。

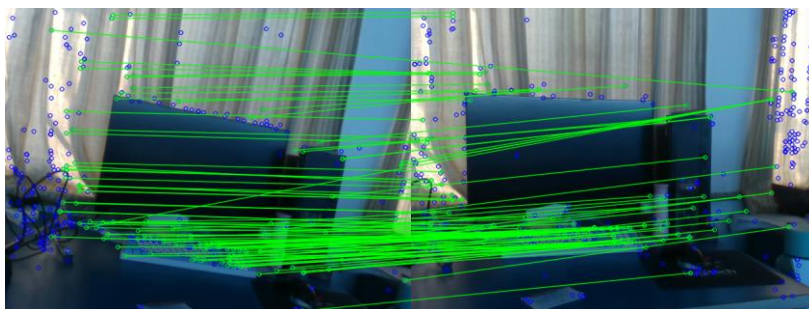


图 2-5 FLANN 匹配

匹配成功的特征点对可以用于估计相邻图像之间的相对运动，从而得到初始的相机位姿。通过解算本质矩阵或基础矩阵，结合相机内参，可以进行这一

步。然后利用初始位姿估计值，通过迭代最小化重投影误差的优化方法（如 PnP 算法），不断优化相机的位姿。在这个过程中，选取的特征点会被投影到三维空间中，并与对应的特征点进行重投影，通过最小化重投影误差^[50]来优化位姿。这一过程为后续的位姿优化提供了必要的帮助。

2.1.2 后端优化

后端优化主要目的是通过全局地图优化来提高相机轨迹和地图点云的准确性和一致性。在 VSLAM 中，常用的后端优化算法包括图优化、束优化、因子图优化等。

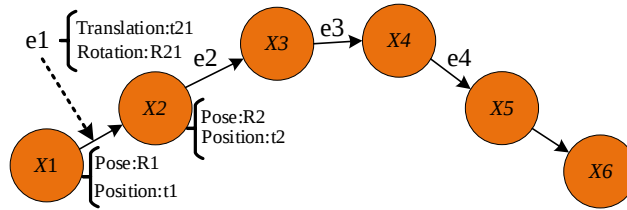


图 2-6 图优化示意图

图优化^[51]是一种基于图的优化方法，在 VSLAM 中广泛应用。该方法将相机轨迹和地图点云表示为节点，并根据它们之间的约束关系建立边。例如，两个相机之间的运动连续性约束、相机与地图点之间的投影约束等。优化过程则是最小化系统误差的过程，图优化示意图如图 2-6 所示。

光束平差法^[52]是一种经典的非线性优化算法。其核心思想是最小化相机观测数据与重投影误差之间的偏差，以优化相机姿态和地图点云的位置。

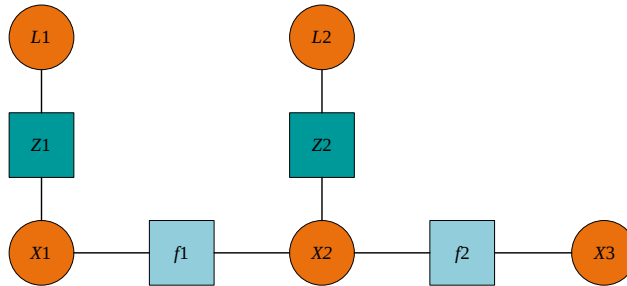


图 2-7 以因子图表示的 SLAM 过程

因子图优化^[53]是一种较新的优化方法。因子图优化示意图如图 2-7 所示。与图优化和束优化不同的是，因子图优化将相机轨迹和地图点云表示为节点，

但它们之间的约束关系不是像前两者那样构成简单的图，而是通过因子节点来表示更复杂的约束关系。

2.1.3 闭环检测

闭环检测^[54]是 VSLAM 的核心一环，用于检测相机经过的区域或轨迹是否重复。闭环检测的目标是识别并消除由于漂移或误差累积导致的位姿估计偏差，从而提高 SLAM 系统的精度和鲁棒性。

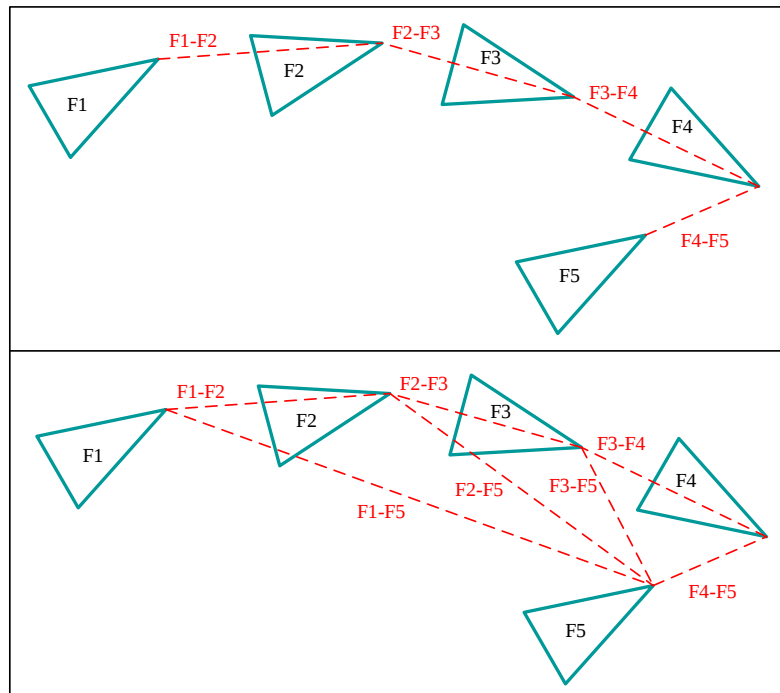


图 2-8 闭环检测

通过前端的位姿估计建立起一个初始的地图，实时 VSLAM 运行过程中，产生一系列的图像帧。闭环检测的第一步是生成闭环候选帧，通过闭环检索算法选择可能与当前帧相似的候选帧。对生成的闭环候选帧进行特征匹配或几何验证，确定闭环的存在与否。验证后，进行位姿图优化修正位姿偏差。通过非线性优化算法（如图优化、BA 优化）最小化闭环帧与其它相关帧之间的误差。闭环检测后，更新地图中的点和位姿信息融合闭环信息，提高整个 SLAM 系统的一致性。闭环检测的准确性和可靠性对于视觉 SLAM 系统的长期稳定运行至关重要。它可以提供更准确的位姿估计和地图更新，从而改善 SLAM 系统的性能和鲁棒性。

2.1.4 地图构建

地图构建模块负责将相机观测到的特征点或者地图点云进行建模，从而生成一个表示环境结构的地图。

地图构建模块负责将相机观测到的特征点或地图点云进行建模，生成表示环境结构的地图。首先对相机图像进行特征提取和描述子计算，用于相机姿态估计和地图点匹配。利用三角化等方法生成初始的地图点云，表示相机观测到的环境中的 3D 点。随着相机移动，新的观测数据会不断进入系统，根据前端视觉里程计得到的相机位姿，将新的观测数据投影到已有的地图点云上，并根据规则判断是否接受新的点云。在后端优化模块中，通过优化相机轨迹和地图点云之间的约束关系，进一步提高地图的准确性和一致性，改善地图构建的结果。

地图构建模块的实现需要结合前端视觉里程计、后端优化和闭环检测等模块的输出结果，通过特征提取与描述子计算、初始地图点云生成、地图点云更新和地图优化等步骤，逐渐构建一个准确、一致的环境地图。

2.2 YOLOv5 理论基础

R-CNN 系列^[55]和 YOLO 系列是两种主流的基于深度学习神经网络的深度图像识别算法。R-CNN 系列具有较高的检测精度，但检测速度慢于 YOLO 系列，不能满足实时要求。因为在目标检测过程中会先确定目标是前景还是背景，然后对目标进行分类。相反，YOLO 系列算法将检测问题转化为回归问题，可以更快、更准确地检测目标，更适合实时应用场景。

YOLOv2^[56]通过 k-means 聚类改进边界框预测，提升准确性。YOLOv3^[57]引入 ResNet^[58]、Darknet-53 和多尺度预测，提高检测精度和速度。YOLOv4^[59]保留 YOLOv3 头部，使用 CSPDarknet53 和 SPPF 扩展感受野，PANet 增强特征提取，进一步优化性能。

其中 YOLOv5 网络是 UltralyticsLLC 提出的模型，网络结构图如图 2-9 所示。该网络模型检测精度高、推理速度快、网络模型的权重文件较小，比 YOLOv4 小近 90%，表明 YOLOv5 模型适合部署到嵌入式设备中实现实时检测。因此，YOLOv5 网络的优点是检测精度高，同时具有轻量级的特点，检测速度

快。

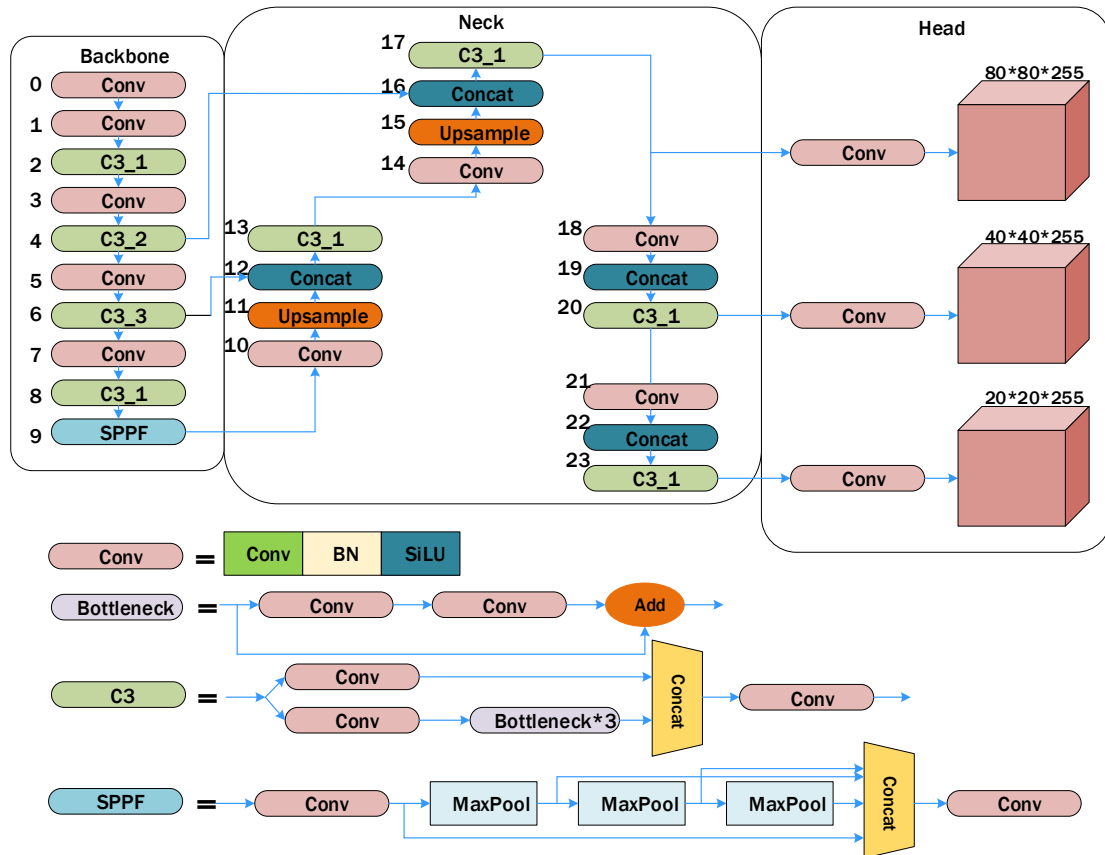


图 2-9 YOLOv5 网络结构图

与之前版本的 YOLO 系列不同，YOLOv5 架构包含四个架构，分别命名为 YOLOv5s、YOLOv5m、YOLOv5l 和 YOLOv5x。它们之间的主要区别在于特征提取模块和卷积核在网络特定位置的数量不同。在四种体系结构中，模型的大小和模型参数的数量依次增加。YOLOv5 系列网络结构参数如表 2-1 所示。

表 2-1 YOLOv5 不同 结构对比

参数	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
网络深度	0.33	0.67	1.00	1.33
网络宽度	0.50	0.75	1.00	1.25
Params	7.04	21.4	47.0	87.7
GFLOPS	16.0	51.0	115.4	218.8

YOLOv5 的基本框架由四个主要部分组成：输入(Input)、主干(Backbone)、颈部(Neck)、头部(Head)。

首先介绍输入部分：YOLOv5 中使用的数据增强方法是 Mosaic 数据增强方法^[0]。此方法采集四张图像并通过翻转、缩放和色域调整增加多样性。然后将

它们合成一个图像，以此加强小目标的表现并提升数据集的多样性。这种方法使得模型更鲁棒，并增强了其泛化能力。

2.2.1 Backbone 主干网络

YOLOv5 中使用的主干特征提取网络是 CSPDarknet 网络，从输入图像中提取丰富的信息特征，解决了深度卷积神经网络模型主干优化中梯度信息冗余的问题。通过将梯度变化集成到特征图中，CSPDarknet 减少了模型的参数大小、每秒浮点运算(FLOPS)和整体模型大小。这种优化可以提高推理速度和准确性，同时减少模型的占用空间。这些设计和优化措施使 YOLOv5 成为一个鲁棒、高效的网络模型。

Conv 卷积操作包括卷积、批量归一化和 SiLU 激活功能。C3 层通过增加 YOLOv5 的深度和感受野，从而增强特征提取能力。YOLOv5 在主干和颈部网络中使用了不同的 C3。

CSPNet 结构引入了 LeakyReLU 激活函数，它是对 ReLU 的改进版本。在训练过程中，当输入值小于 0 时，激活后的输出不为零。这确保了一些神经元的激活，并防止了参数永远不会更新的问题。LeakyReLU 激活函数引入了一个权重系数 a 。LeakyReLU 函数的表达式如下：

$$LeakyRelu(x) = \max(ax, x) \quad (2-1)$$

在式中， x 表示输入值，参数 a 通常设置为 0.01。

主干网络的最后一层是 SPPF 模块。它结合了 Conv 卷积层和三个串行 5×5 池化层来融合多尺度特征。在进一步提取特征的同时，避免了深层特征信息不完全表达的问题。

2.2.2 Neck 颈部网络

YOLOv5 采用多特征层融合进行目标检测，目的是将不同尺度的特征信息进行融合，总共需要提取三个特征层。这三个特征层分别位于特征提取网络 CSPDarknet 网络的中间层、下中间层和底层。当特征网络的输入图像为 640×640 时，三个特征层提取的网络形状分别为 80×80 、 40×40 和 20×20 。

在获得三个有效特征层后，进行 Neck 层网络的构建。YOLOv5 Neck 使用

PANet 生成特征金字塔网络，对特征进行聚合并传递给 Head 进行预测。NECK 层结合了特征金字塔网络(FPN)和路径聚合网络(PAN)结构，在 FPN 的基础上引入了自下而上的路径增强结构，FPN+PAN 的结构如图 2-10 所示。FPN 与 PAN 的结合可以聚合来自不同中继层的不同检测层的参数，大大增强了网络的特征融合能力。

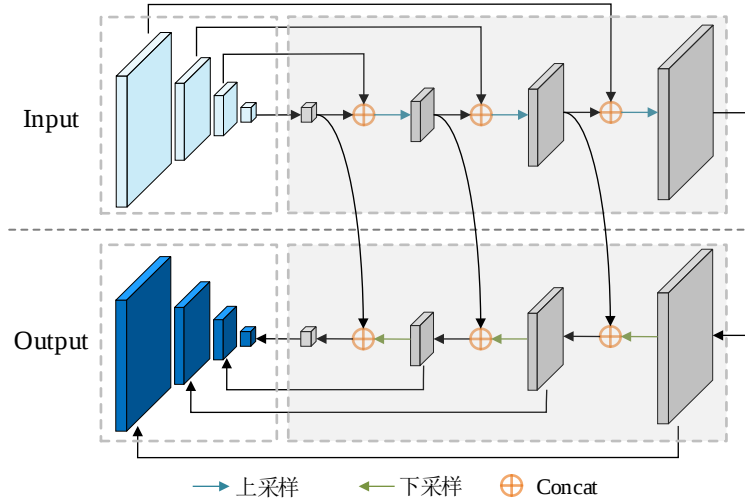


图 2-10 FPN+PAN 结构图

2.2.3 Head 输出网络

经过主干特征提取网络和颈部网络，可以得到具有三个尺度大小不同的特征层输出的结果。将这三个特征层传递到 YOLO Head 中，得到预测结果。在 YOLOv5 中，共设计了 9 个不同大小的先验框。每个特征层的每个特征点对应 3 个维数为 9 的先验框。Head 网络的任务是选择图像中的目标坐标、置信度和类别信息。

利用先验框对目标进行粗定位，然后通过式(2-2)计算预测框的中心点坐标 (b_x, b_y) 和预测框的宽度和高度 (b_w, b_h) ，确定预测框的位置。

$$\begin{aligned} b_x &= \sigma(t_x) \times 2 - 0.5 + c_x \\ b_y &= \sigma(t_y) \times 2 - 0.5 + c_y \\ b_w &= P_w(\sigma(t_w) \times 2)^2 \\ b_h &= P_h(\sigma(t_h) \times 2)^2 \end{aligned} \quad (2-2)$$

其中 σ 为 sigmoid 激活函数， t_x 、 t_y 、 t_w 、 t_h 分别为网络预测框中心点坐标的偏移量和宽高比， c_x 、 c_y 为先验框中心点坐标， P_w 、 P_h 为先验框的宽度和高

度。

在进行网络推理时，采用评分机制过滤最合适的预测框。将每个网格预测的类信息与目标框预测的置信度相乘，得到每个目标帧的类特定置信度得分，如式(2-3)所示。

$$Pr(C_i|O) \times Pr(O) \times \text{CIoU}_{\text{pred}}^{\text{truth}} = Pr(C_i) \times \text{CIoU}_{\text{pred}}^{\text{truth}} \quad (2-3)$$

其中 $Pr(C_i)$ 为每个网格预测的类别概率， $Pr(O)$ 为每个网格预测的目标概率， $\text{CIoU}_{\text{pred}}^{\text{truth}}$ 为真实框与预测框的相交比率。

得到每个目标框的类别置信度得分后，设置阈值，过滤掉得分较低的目标框。对保留的目标框进行非最大抑制(NMS)^[60]，得到最终的检测结果。

2.2.4 损失函数

YOLOv5 中的损失函数由三个组成部分组成：边界框回归损失、置信度预测损失和类别预测损失。

$$\text{Loss} = \text{Loss}_{\text{(box)}} + \text{Loss}_{\text{(conf)}} + \text{Loss}_{\text{(cls)}} \quad (2-4)$$

首先是边界框回归损失部分。边界框回归损失是使用 CIoU 损失函数计算的，该函数量化了预测边界框和真实边界框之间的差异。

$$\text{Loss}_{\text{(box)}} = 1 - \text{IoU} + \frac{p^2(b, b^{gt})}{c^2} + av \quad (2-5)$$

式中 $p^2(b, b^{gt})$ 表示预测框和真实框的质心之间的欧氏距离， c 表示同时包含预测框和真实框的最小封闭区域的对角线距离。 a 和 v 的公式如下。

$$a = \frac{v}{1 - \text{IoU} + v} \quad (2-6)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (2-7)$$

第二，是置信度预测损失部分。置信度预测损失是使用具有 Logits 的二进制交叉熵（BCEWithLogitsLoss）损失函数计算的，该函数评估对象存在的概率的损失。

$$\begin{aligned} \text{Loss}(\text{conf}) = & \sum_{i=0}^{S \times S} \sum_{j=0}^B l_{ij}^{obj} \left[\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i\right) \log(1 - C_i) \right] \\ & - \sum_{i=0}^{S \times S} \sum_{j=0}^B l_{ij}^{nobj} \left[\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i\right) \log(1 - C_i) \right] \end{aligned} \quad (2-8)$$

第三，是类别预测损失部分。类别预测损失是使用交叉熵损失函数（BECloss）计算的，该函数测量类预测的损失。

$$\text{Loss}_{(cls)} = \sum_{t=0}^{S \times S} \sum_{j=0}^B l_{ij}^{\text{obj}} \sum_{c \in \text{classes}} [\hat{p}_i(c) \log(p_i(c)) + (1 - \hat{p}_i(c)) \log(1 - p_i(c))] \quad (2-9)$$

2.3 本章小结

在本章中对于实现高效且准确的 VSLAM 系统的理论知识进行了全面而深入的介绍。通过逐一剖析 VSLAM 系统的关键要素，包括负责初步数据处理的前端视觉里程计、用于提高系统整体准确性的后端优化算法、用于纠正累积误差的闭环检测机制、以及用于构建可用于导航和任务执行的环境地图构建法，确立了 VSLAM 系统设计的理论支撑。此外，还深入探讨了 YOLOv5 这一强大的目标检测框架，详细说明了其各个组件如何协同工作以实现快速而精确的目标检测。重点分析了 YOLOv5 的 Backbone 主干网络、Neck 颈部网络、Head 输出网络的设计和函数，以及训练过程中如何通过损失函数优化模型性能。

VSLAM 和 YOLOv5 的理论基础为后续构建有效且高效的语义 VSLAM 系统提供了坚实的理论支撑。

第3章 基于语义信息实现动态特征点剔除的视觉里程计

VSLAM是一项关键技术，它可使无人机能够准确地感知周围环境并进行定位和地图构建。然而，VSLAM算法的基本假设是基于静态环境的，这往往导致在高度动态的室内场景中性能不佳。在这种环境下，准确估计相机姿态并实现精确定位是一个重大挑战。本章将YOLOv5_GBCS提取的语义信息与ORB-SLAM3的视觉里程计相结合，设计了一种用于高动态场景中动态特征点剔除的VSLAM方法。能有效提高动态室内环境下的鲁棒性和精度。在系统中，采用了YOLOv5_GBCS无人机小目标检测模型、LK光流法与对极几何约束相结合的方法来检测图像中的动态区域，去除不稳定的动态特征点，进一步提高了VSLAM系统的稳定性。

3.1 ORB-SLAM3 算法

3.1.1 ORB-SLAM3 框架

ORB-SLAM3是首个开源的SLAM系统，支持执行采用单目、立体和RGB-D相机（适用于针孔或鱼眼相机模型）的视觉、视觉-惯性融合以及多地图SLAM任务。ORB-SLAM3包括三个部分：跟踪线程、局部建图线程和闭环与地图合并线程。下图为ORB-SLAM3的系统结构图。

跟踪线程：线程接收实时相机帧输入，输出估计姿态。输入帧可作新关键帧，通知后端全局优化。使用ORB特征检测和BRIEF描述符提取关键点。当前帧特征与历史帧通过全局描述符索引匹配，实现快速特征匹配。同时进行先验运动估计，以提高跟踪精度。

局部建图线程：线程处理地图点的插入和移除，并执行地图优化。局部建图线程负责合并新的关键帧和点，修剪任何不必要的信息，并通过视觉(或视觉惯性)BA优化过程提高地图的精度。这是通过关注当前帧附近的一小组关键帧来完成的。

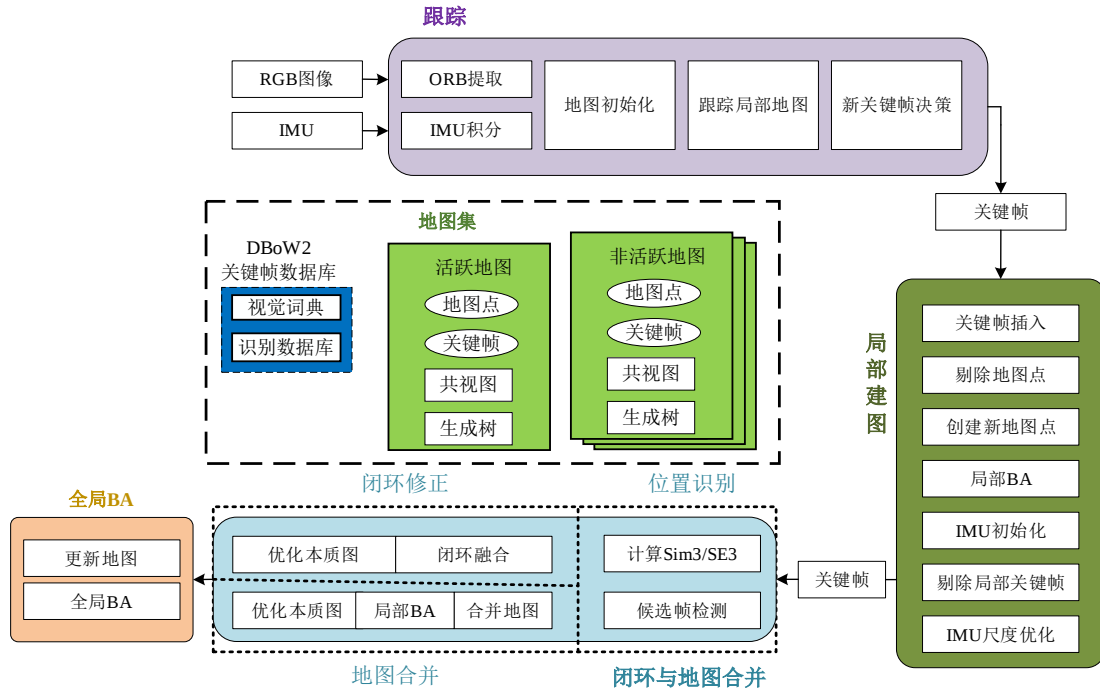


图 3-1 ORBSLAM3 系统结构图

闭环与地图合并线程：线程识别闭环。如果检测到闭环，则触发全局优化过程来更新地图点和姿势以实现全局一致性。

它最大的优点是允许匹配和使用 BA 之前的观察结果，从而实现短期、中期和长期的数据关联。然而，ORB_SLAM3 被设计为在静态环境或非常轻微的动态环境中运行。ORB_SLAM3 在跟踪线程中通过特征匹配和 g2o 算法计算当前帧的姿态。当涉及到动态环境时，两个动态特征点之间的正确匹配点数量显著增加，超过了 g2o 算法可以优化的范围。正因为如此，ORB_SLAM3 无法在动态环境中表现良好。

3.1.2 ORBSLAM3 前端视觉里程计

ORB_SLAM3 的前端视觉里程计负责定位跟踪与关键帧选取。通过特征点提取与匹配估计相机运动，这增强了重定位与闭环检测，提升了定位精度与局部稳定性。视觉里程计决定何时将帧作为关键帧，影响全局优化与建图。利用历史信息，能在长期运行中保持稳定，支持重新访问区域与现有地图合并。前端由以下关键技术构成：

特征提取：ORB_SLAM3 采用 FAST 角点检测算法从视频图像中提取候选特

征点。FAST 算法具有高速的特征点检测能力，并通过逐像素比较的方式实现了高效的角点定位。随后，针对每个检测到的特征点，ORB_SLAM3 使用 ORB 描述子进行描述。

特征匹配：ORB_SLAM3 特征匹配先用 ORB 提取旋转和尺度不变特征与二进制描述子，再用词袋模型将描述子映射至视觉词典，形成稀疏向量以快速匹配图像特征。当前帧与关键帧描述子匹配，采用哈希和汉明距离寻找最佳对应点。

姿态估计：利用匹配的关键点，算法通过解决 PnP 问题来估计相机的姿态。PnP 算法使用 2D-3D 对应关系来恢复相机相对于地图中观察到的 3D 点的位置和方向。在 ORB_SLAM3 的初始阶段，系统需要足够数量的帧来从周围环境中提取特征。这对于系统通过估计相机姿态和创建初始地图来初始化自身至关重要。

关键帧选择：ORB_SLAM3 中，关键帧选择基于几个标准平衡精确性与计算效率。选择标准包括相机与前一关键帧的运动、闭环检测表明区域重新访问、现有地图的密度和覆盖范围适中，以及捕获场景改变的能力。此外，在初始化和跟踪丢失时选取关键帧以维护建图连续性。这些关键帧对持续稳定建图和恢复跟踪至关重要。

3.1.3 ORB_SLAM3 后端非线性优化

ORB_SLAM3 后端的主要职责是全局优化和闭环检测。它涉及到对整个地图和关键帧的优化，以减少累积的误差并提供更准确的地图。后端还负责检测相机是否回到了之前的位置，并进行必要的调整以修正任何累积的误差。这个过程通常涉及到最小化所有关键帧和地图点之间的重投影误差，以确保地图的准确性和一致性。后端由以下关键技术构成：

图优化：ORB_SLAM3 通过 g2o 等图优化算法对位姿和地图进行全局优化。该过程最小化重投影误差、维护地图一致性，提升定位和地图精度。SLAM 以图的形式表示：节点代表关键帧姿态，边代表关键帧间的几何约束。闭环后，执行联合优化，精细调整地图和相机轨迹，减少误差。优化主要适用于静态环境。

闭环检测：闭环检测是 ORB_SLAM3 核心功能，辨认并纠正相机重返先前区

域时的闭环事件以减少漂移并提升精度。通过特征匹配识别潜在闭环关键帧，利用 RANSAC 等几何验证确保正确闭环。成功后，系统更新地图、优化相机姿态，降低误差。该过程独立进行，提升 SLAM 的精准度、一致性和鲁棒性，适合长期导航和建图。

尺度恢复：在 VSLAM 中，尺度恢复是一个重要的问题。ORB_SLAM3 的后端会根据已知尺度的信息（例如，知道某个物体的实际尺寸）进行尺度恢复，以确保地图的尺度与实际场景一致。尺度恢复通常在闭环检测后进行，通过将闭环中的关键帧与前期的帧进行对齐，并通过尺度因子来调整整个地图的尺度。

3.2 改进 YOLOv5 的语义检测

3.2.1 改进的 YOLOv5 框架

为了确保提升小目标检测算法的检测精度的同时，提高检测效率，在主干网络采用 GhostConv 取代后三个普通卷积，再采用由 GhostBottleneck 结果组成的 GhostBottleneckC3 取代主干网络的前三个 C3 模块，极大的减少了参数量。同时，分别在主干网络和颈部网络引入 CBAM 注意力模块，增强了网络的特征提取能力的同时保持精度；添加 P2 特征级别，用以提高了网络的深度和小目标检测的准确性；在颈部网络采用 BiFPN 结构，快速的进行多尺度特征融合。改进后的网络结果如图 3-2 所示。

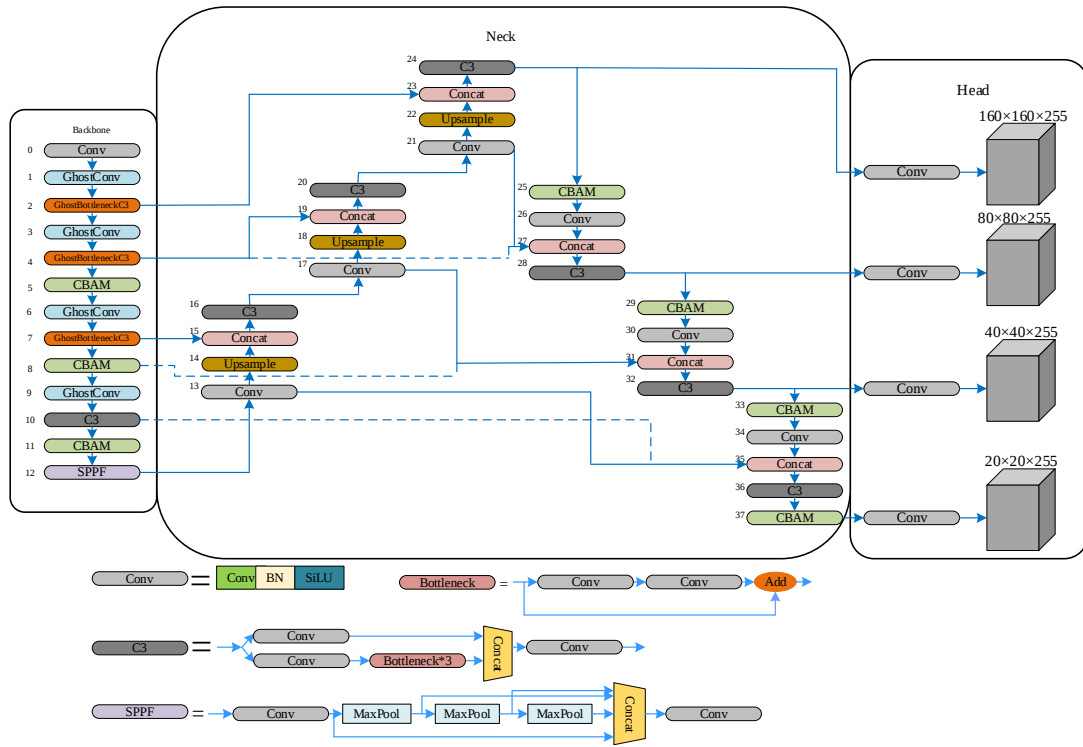


图 3-2 YOLOv5_GBCS 结构图

3.2.2 网络结构轻量化设计

GhostNet 提出了一个创新的模块 Ghost，它用更少的参数与计算量得到更多的特征图。Ghost 模块一部分是普通卷积，另一部分是线性运算。Ghost 的工作原理如图 3-3 所示。

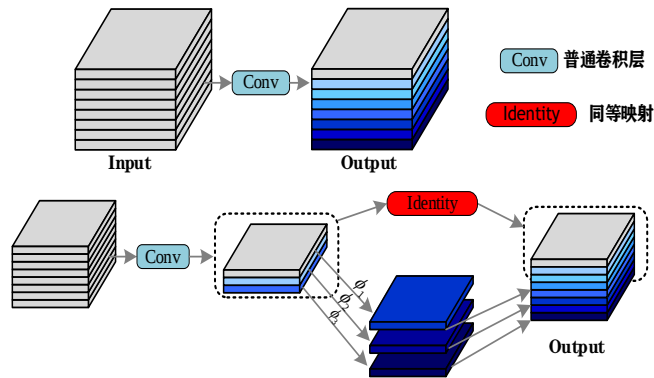


图 3-3 Ghost 卷积与普通卷积的比较

普通卷积层的操作为：

$$Y = X \cdot f + b \quad (3-1)$$

Ghost模块首先通过普通卷积生成少量的特征图；然后对第一步得到的特征图进行计算量较小的线性操作，生成Ghost特征图。最后，对两组特征图进行通道拼接，生成足够的特征图来匹配给定的输出通道数量。

Ghost卷积采用了逐步进行的策略，计算方法如所示：

$$Y' = X \cdot f' \quad (3-2)$$

$$Y_{ij} = \phi_{ij} \cdot Y'_i, i \in [1, m], j \in [1, s] \quad (3-3)$$

$Y \in \mathbb{R}^{W' \times H' \times m}$ 表示输入X在经过普通卷积 $f' \in \mathbb{R}^{k \times k \times C \times m}$ 生成的 m 个特征图 ($m \ll n$)；然后将 m 个特征图逐个进行线性操作，每个特征图生成1个特征图，共生成 $n = m \times s$ 个特征图。特征图 ϕ_{ij} 表示对第一步卷积生成的第 j 个特征图， Y'_i 表示第 i 次线性运算。为了保证CPU或GPU的效率和实用性，将每个线性操作的卷积核的大小设置为 $d \times d$ ；那么普通卷积和Ghost卷积的速度比可以通过式 (3-4) 计算得出。

$$\begin{aligned} RATE_s &= \frac{W' \cdot H' \cdot n \cdot k \cdot k \cdot C}{W' \cdot H' \cdot m \cdot k \cdot k \cdot C + W' \cdot H' \cdot (n - m) \cdot d \cdot d \cdot C} \\ &= \frac{W' \cdot H' \cdot n \cdot k \cdot k \cdot C}{W' \cdot H' \cdot (n/s) \cdot k \cdot k \cdot C + W' \cdot H' \cdot (s - 1) \cdot d \cdot d} \\ &= \frac{k \cdot k \cdot C}{(1/s) \cdot k \cdot k \cdot C + \left(\frac{s-1}{s}\right) \cdot d \cdot d} \\ &\approx \frac{s \cdot C}{s + C - 1} \approx s \end{aligned} \quad (3-4)$$

Ghost 卷积是一个更轻、更快的模块。在此基础上，主干网络采用了 GhostConv 以及基于 Ghost 模块的 GhostBottleneckC3 模块，具体结构如图 3-4 所示。

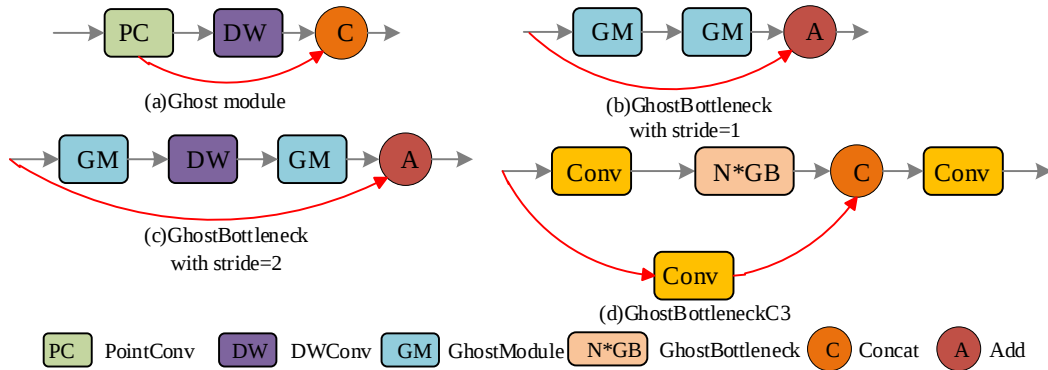


图 3-4 Ghost module 模块、GhostBottleneck 模块以及 GhostBottleneckC3 模块结构图

由于 Bottleneck 中使用了较多 3×3 标准卷积，因此用 GhostBottleneck 结构替换 C3 模块中的 Bottleneck 模块能够将模型的体积和计算量进一步压缩。用图 3-4(b)的 GhostBottleneck 结构替换 YOLOv5s 中 C3 的 Bottleneck 模块得到新的 GhostBottleneckC3 结构，如图 3-4(d)所示。GhostBottleneckC3 结构可以减少模型的计算量并压缩体积。

3.2.3 CBAM 注意力机制

在无人机航拍图像中含有大量非目标区域，可以通过引入注意机制使模型更好地关注目标区域的信息。注意力机制从大量的信息中选择少量的重要信息。改进模型中使用的 CBAM 是一种结合空间和通道的混合注意机制模块；与只关注通道本身的 SENet^[61]相比，CBAM 整合了通道注意力(见图 3-5(a))与空间注意力(见图 3-5(b))映射过程，可以保留更多的特征信息，有效地提高训练出来的网络模型的检测速度，同时提高网络的特征提取能力和检测精度。CBAM 注意力模块的结构如图 3-5(c)所示。

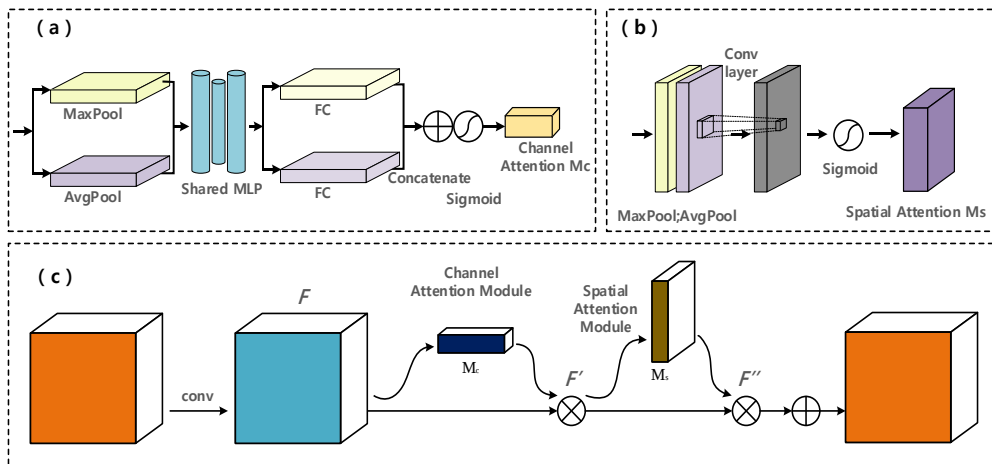


图 3-5 CBAM 模块以及子模块结构图

图3-5为CBAM注意力机制的整体流程结构，它由通道注意力模块和空间注意力模块两大模块构成。

通道注意力模块首先分别对输入特征图的宽度和高度维度进行全局最大池化和全局平均池化，以聚合特征图的空间信息，如图3-5(a)所示。然后将结果传递到多层感知机中，以便使用共享的全连接层进行处理。最后，通过Sigmoid激

活函数，生成最终的通道注意力特征图 $M_c(F)$ 。该特征图的结果与输入特征图加权后，作为空间注意力模块的输入特征。

通道注意特征图 $M_c(F)$ 的数学表达式：

$$M_c(F) = \sigma \left(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F)) \right) \quad (3-5)$$

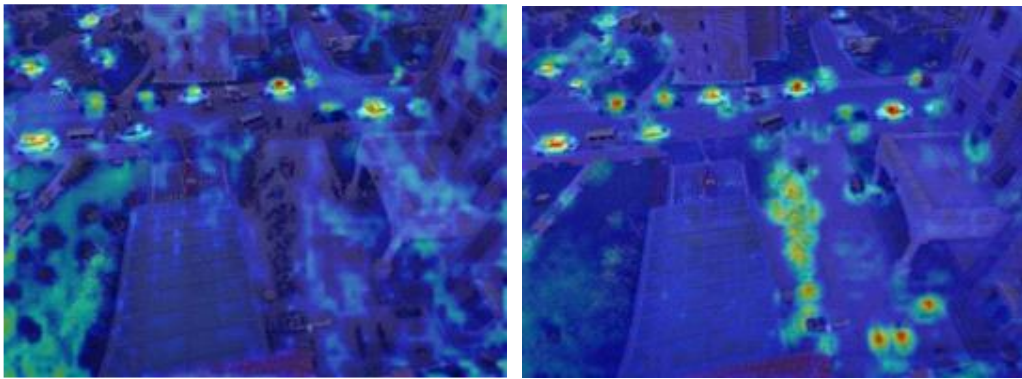
其中， σ 为Sigmoid激活函数，MLP为层间的连接权值操作，AvgPool和MaxPool分别表示最大池化操作和平均池化操作。

在通道注意力模块之后，引入了空间注意力模块，如图3-5(b)所示。与通道注意力类似，空间注意力机制压缩通道，并分别在通道维度上进行均值池化和最大池化。首先，输入特征层在每个特征点的通道上取最大值和平均值。之后将两个结果叠加。最后通过Sigmoid函数生成空间注意力特征图 $M_s(F)$ ，其数学表达式为：

$$M_s(F) = \sigma(f^{7 \times 7}([\text{AvgPool}(F); \text{MaxPool}(F)])) \quad (3-6)$$

在公式(3-6)中， 7×7 表示卷积核的大小，即对特征图进行 7×7 的卷积操作。经验表明， 7×7 的卷积核优于 3×3 的卷积核。在主干网络和颈部网络引入了CBAM注意力模块。

从图3-6中可以看出，在添加了CBAM注意力模块后，改进的算法相对于原YOLOv5s能够更多得关注目标区域，而忽略杂乱背景中的不相关因素，有利于提高目标特征的学习能力，提高检测精度。



(a) YOLOv5_GBCS的热力图可视化结果

(b) YOLOv5s的热力图可视化结果

图 3-6 热力图可视化结果对比

3.2.4 基于 BiFPN 的特征融合网络

无人机的拍摄视角更高，视野更广可以捕捉到更多的物体，但是这也考验了检测算法对小目标识别的精度，所以如何表示和处理多尺度特征是一个难题。多尺度特征处理结构，如图 3-7 所示。其中，FPN 的结构图如图 3-7(a)所示，该方法将多尺度特征相结合，使用复合特征层和更多的语义信息进行预测。根据这一想法，PANet 继续在 FPN 之上添加一个自下而上的路径聚合网络，如图 3-7 (b)所示，同时考虑了顶层的语义信息和底层的位置信息。YOLOv5 框架将多尺度特性与 FPN+PAN 结构集成在一起，但由于输入特性的分辨率不同，FPN+PAN 结构对融合输出特性的贡献往往是不均匀的，不同尺度之间的特性不能得到充分利用。因此，在 YOLOv5 框架中引入了一种简单、高效的 BiFPN 网络结构，以提高检测精度。

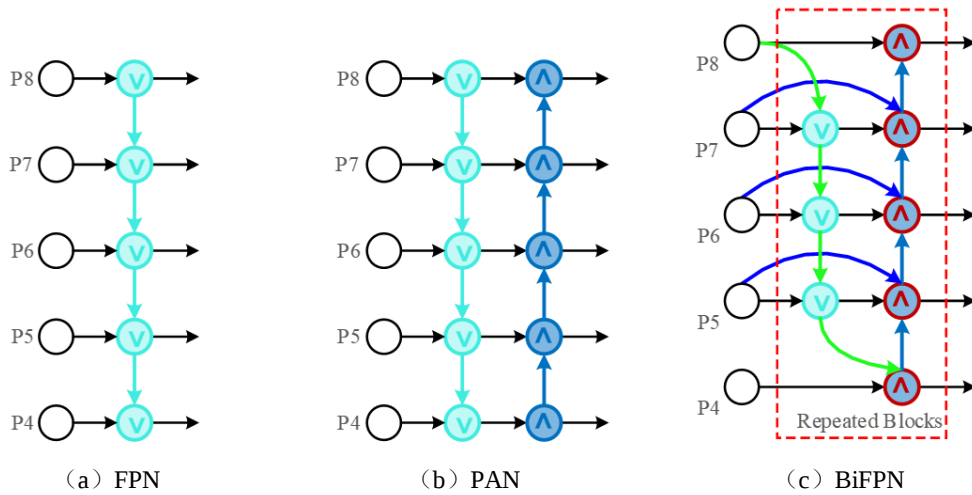


图 3-7 多种特征融合网络结构

如图3-7(c)所示，BiFPN网络加强了更高层次的特征融合，将每个双向路径作为特征网络层进行处理，并在同一层上重复多次。通过加权特征的融合，学习不同输入特征的重要性，并对不同的特征进行区分融合。

BiFPN使用快速归一化来融合加权特征，其定义如式(3-7)所示。学习权重 W_i 使用ReLU激活函数，其值为 $\epsilon=0.0001$ ，以加强该值的稳定性。为了进一步提高深度网络学习模型的检测效率，BiFPN使用了可分离的卷积融合特征，并在每次卷积后添加了批处理归一化和激活函数操作。以第6层为例，描述了两个融合特征的定义如下：

$$O = \sum_i \frac{W_i}{\varepsilon + \sum_j W_j} \cdot I_i \quad (3-7)$$

$$P_6^{td} = \text{Conv} \left(\frac{W_1 \cdot P_6^{in} + W_2 \cdot \text{Resize}(P_7^{in})}{W_1 + W_2 + \varepsilon} \right) \quad (3-8)$$

$$P_6^{out} = \text{Conv} \left(\frac{W'_1 \cdot P_6^{in} + W'_2 \cdot P_6^{td} + W'_3 \cdot \text{Resize}(P_5^{out})}{W'_1 + W'_2 + W'_3 + \varepsilon} \right) \quad (3-9)$$

其中 P_6^{td} 从上到下表示第六层的中间特征， P_6^{out} 是第六层从下到上的输出特征。通过不同层间的互连和融合，最终实现了 BiFPN 的双向交叉尺度连接和快速归一化融合。基于上述优点，引入 BiFPN，改进特征金字塔结构，增强多尺度特征融合，提高模型检测精度。

3.2.5 针对小目标增加预测头

YOLOv5_GBCS 主干网络包含连续的下采样卷积层，因此在特征信息提取过程中，特征图的尺寸随着网络的加深而减小。特征图的小尺寸影响了对图像中的小物体的检测。然而，无人机图像中包含了大量的小物体。为了增强小物体的特征融合效果，于是添加了一个额外的检测头。改进的模型如图 3-2 所示。在第 20 层之后，继续对特征图进行上采样，以确保特征图继续扩展，在第 23 层，将尺寸为 160×160 的特征图与主干网络中第 2 层的特征图进行融合，以获得小目标的更大特征图输入至第 24 层后的检测头中。将高层、低层的特征信息进行特征融合，使添加的检测头对小物体更加敏感。最后，检测头的四种不同尺度特征更能适应无人机图像物体尺度的剧烈变化。

3.2.6 NMS 的改进

NMS 被应用于大多数最先进的检测器，因为它能够显著地减少了假阳性的数量。NMS 算法的流程图如图 3-8 所示。列表中的所有检测框都根据它们的置信度得分进行排序。再将得分最高的检测框 B_M 移动到最终的检测列表 D 中，并为剩余的检测框分配一个唯一的标识符 B_i 。最后去除任何与 B_M 的重叠面积大于一定阈值 N_t 的预测框 B_i 。对其余的框 B_i 重复此过程，直到初始列表被置空为止。然而，由于无人机视角的特殊性，会导致大量小目标重叠，于是会导致 NMS 算法的检测失误。

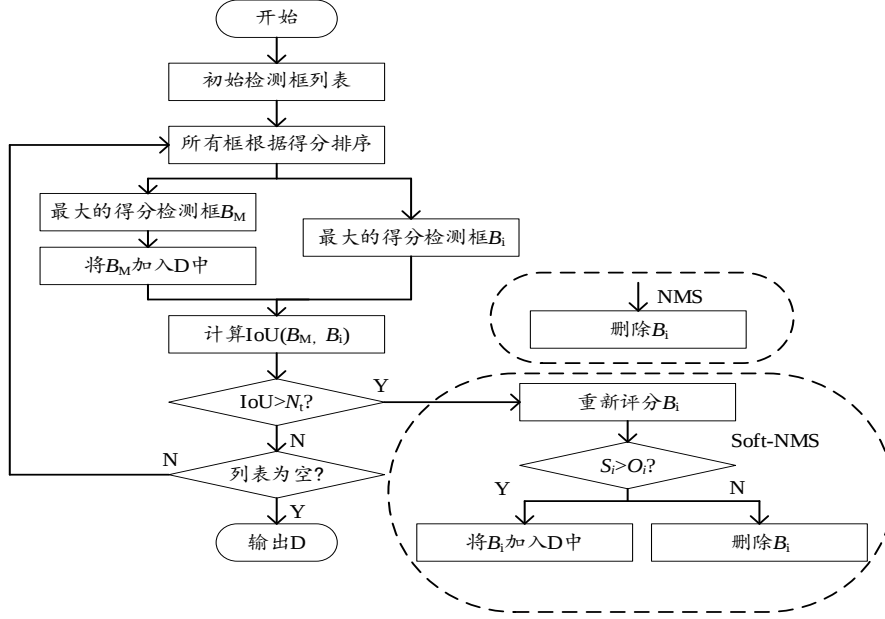


图 3-8 NMS 和 Soft-NMS 的算法流程

接下来，对于无人机的密集目标的场景，本身属于两个目标的检测框可能 IoU 很高，如果使用 NMS 算法，会将置信度较小的那个检测框删除掉，有可能造成漏检，降低了模型的召回率。因此，引入了 Soft-NMS 而不是 NMS 来处理检测结果。

Soft-NMS 的核心思想是使用一个惩罚函数来减弱与 B_M 重叠的预测框的分数，而不是将这些分数设置为零。Soft-NMS 的 S_i 如公式(3-10)所示：

$$S_i = \begin{cases} S_i, IoU(B_M, B_i) < N_t \\ S_i \cdot e^{-\frac{IoU(B_M, B_i)^2}{\sigma}}, IoU(B_M, B_i) \geq N_t \end{cases} \quad (3-10)$$

$\exp\left(-\frac{IoU(B_M, B_i)^2}{\sigma}\right)$ 是一个高斯惩罚函数，而 σ 是根据经验选择的超参数。很明显，与 B_M 重叠较大的预测框的得分会大大降低，而远离 B_M 的检测框则不会受到影响。如果预测框的得分仍然高于惩罚后的置信阈值 O_t 。然后，预测框将被保留，而不是被丢弃。在目标密集的场景中，使用 Soft-NMS 来处理小目标检测问题，可以大大降低误检率。

3.2.7 对比实验分析

在 VisDrone2019 数据集上测试了 YOLOv5_GBCS 模型，并将结果与其他具有代表性的模型进行了比较。结果如表 3-1 所示。证明了其对无人机图像的小目标检测的有效性。这些结果表明，该模型可以在保持目标检测性能的同时，提

高了小目标检测能力。从表3-1中可以看出，与YOLOv5s相比，提出的模型无论各个物体类别的精度还是mAP@0.5超过了YOLOv5s。Pedestrian的AP提升了3.2%，People的AP提升了6.6%，Bicycle的AP提升了6.8%，Car的AP提升了3.8%，Van的AP提升了4.1%，Tricycle的AP提升了13.9%，Awing-Tricycle的AP提升了3.0%，Bus的AP提升了10.9%，Motor的AP提升了4.1%，其中Trunk的AP提升的并不大，只有0.3%。最后，改进后的模型YOLOv5_GBCS的mAP@0.5比YOLOv5s提高了5.1%，与LDS-YOLO和M-YOLO这两个同样基于改进的YOLOv5的小目标检测算法相比mAP@0.5分别提高了1.7%和0.8%。由此可以得出结论，在针对无人机图像小目标检测方面，YOLOv5_GBCS模型比原YOLOv5s更加优秀。

表 3-1 不同网络模型下不同目标的 AP 与 mAP@0.5

Model	AP of different target										mAP@0.5
	Pedestrian	People	Bicycle	Car	Van	Trunk	Tricycle	Awing-t	Bus	Motor	
YOLOv5s	0.542	0.474	0.282	0.645	0.528	0.525	0.429	0.304	0.627	0.505	0.385
CenterNet ^[62]	0.173	0.105	0.056	0.483	0.263	0.185	0.084	0.055	0.309	0.155	0.336
RetinaNet ^[63]	0.084	0.031	0.002	0.414	0.154	0.100	0.040	0.030	0.134	0.074	0.191
Cascade-RCNN ^[64]	0.199	0.123	0.084	0.541	0.353	0.264	0.174	0.092	0.422	0.196	0.390
Fcos ^[65]	0.173	0.093	0.033	0.513	0.267	0.225	0.086	0.070	0.341	0.096	0.319
Tridentnet ^[66]	0.169	0.105	0.059	0.508	0.288	0.224	0.144	0.071	0.330	0.168	0.353
Faster-RCNN ^[67]	0.176	0.120	0.072	0.505	0.301	0.233	0.144	0.089	0.372	0.182	0.371
FSAF ^[68]	0.221	0.141	0.056	0.538	0.293	0.215	0.089	0.058	0.309	0.165	0.364.
ATSS ^[69]	0.197	0.065	0.074	0.544	0.310	0.245	0.142	0.087	0.370	0.181	0.363
DDOD	0.219	0.119	0.074	0.553	0.314	0.254	0.145	0.086	0.372	0.198	0.382
TOOD ^[70]	0.219	0.130	0.086	0.562	0.330	0.260	0.161	0.091	0.388	0.214	0.398
VFNet ^[71]	0.206	0.091	0.067	0.553	0.323	0.253	0.147	0.083	0.390	0.192	0.373
LDS-YOLO	0.557	0.459	0.335	0.692	0.541	0.519	0.430	0.326	0.688	0.521	0.415
M-YOLO	0.581	0.492	0.366	0.652	0.557	0.521	0.443	0.314	0.741	0.533	0.424
Yolov5_GBCS	0.574	0.540	0.350	0.683	0.569	0.528	0.447	0.334	0.736	0.546	0.432

图 3-9 中展示了来自 VisDrone2019 数据集上的 YOLOv5_GBCS 的一些具有代表性的检测结果。可以看出相对于 YOLOv5s，YOLOv5_GBCS 可以检测出更多的小目标。

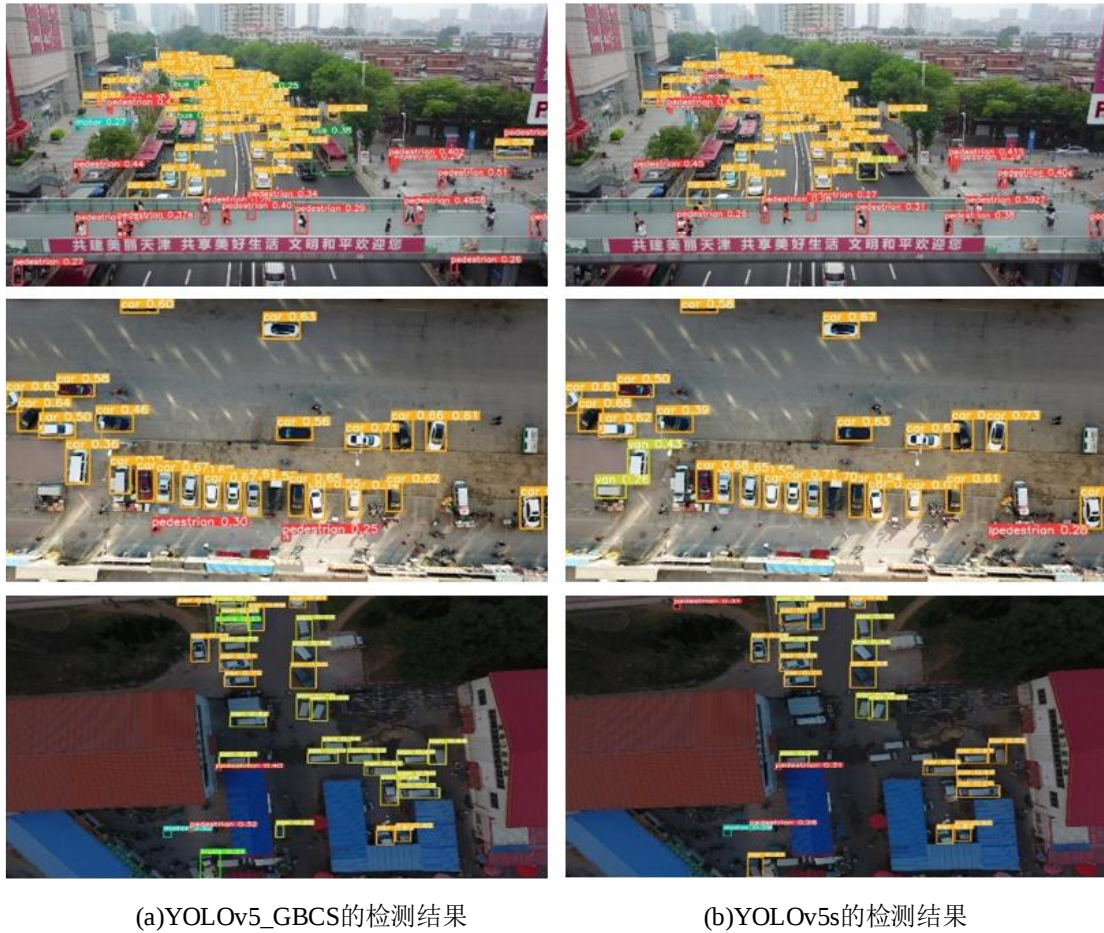


图 3-9 检测结果可视化

3.2.8 消融实验分析

为了验证五种改进方法的有效性，于是基于VisDrone 2019数据集进行了消融实验。在表3-2和图3-10中Model1为在YOLOv5s中增加P2特征级别检测头，Model2为在Model1的基础上将主干网络分别采用GhostConv卷积模块、GhostBottleneckC3模块替换baseline中的部分Conv模块和C3模块，Model3为在Model2的基础上引入BiFPN结构，Model4为在Model3的基础上加入Soft-NMS算法，Model5为在Model4的基础上加入CBAM注意力模块。

如图3-10所示，在训练过程中，损失函数值呈下降趋势，Adam算法对网络进行优化，网络权值等参数不断更新。在训练到40轮之前，损失函数值迅速下降，精度、召回率和mAP@0.5值迅速提高。当训练到100轮左右时，损失函数值的下降逐渐减慢。同样，精度、召回率和mAP@0.5的提高也有所放缓。当训

训练期到250轮时，训练的损失曲线几乎没有下降趋势，其他指标值也趋于稳定，网络模型基本达到收敛状态，并在训练结束时得到最优的网络权值。图3-10(a)显示，Model5模型经过约200轮训练后，mAP达到了42%，并逐渐稳定，最大值达到43.5%。图3-10(e)显示，当Model5模型训练到230轮时，精度达到63.82%，并继续增长到68.88%。图3-10(f)显示，Model5模型的召回率在40轮之前快速上涨，然后继续增长到20.3%的最高值。整体的模型性能已经达到了预期水平，甚至超出了预期水平。该模型的具体损失函数如下图3-10(b)、(c)、(d)所示。从图中可以看出，随着epoch数的逐渐增加，Model5模型算法曲线逐渐收敛，损失值越来越小。当模型迭代230次时，损失值基本稳定，网络基本收敛。与其他模型相比，Model5模型对无人机小目标具有更好的检测性能和识别效果，回归速度更快、更准确，证明了该模型的有效性。

消融实验试验结果表明，每种改进的方法都比原来的YOLOv5s模型有一定的改进。

通过比较表3-2中YOLOv5s和Model1的数据，发现增加P2特征级别检测头可以使模型的mAP@0.5从38.5%增加到40.2%，这表明针对小目标数据集增加尺度更大的检测头，保留了更丰富的小目标特征信息，在图像中的错误检测和遗漏检测得到改进，使得目标框和先验框都能更好的匹配并帮助损失函数更好地收敛，但是参数量和计算量分别增加了3.3%和20.6%，直接导致FPS明显下降。对比表3-2中Model1与Model2的数据可以看出，在引入了GhostConv模块和Ghost BottleneckC3模块后参数量、计算量分别下降了18.0%和24.3%，mAP@0.5上涨了0.9%，帧数也有45fps上涨至59fps这表面对网络结构进行轻量化处理后，在避免了网络模型过于复杂从而造成的实时性能损失的同时，检测精度也有所提升，达到速度与精度的平衡。对比表中Model2和Model3的数据，参数量和计算量都没有增加的情况下，mAP@0.5上涨了0.8%，这表明在颈部网络采用BiFPN后，弥补下采样特征信息丢失，增强多尺度特征融合，提高了模型检测精度。CBAM注意力模块旨在提高网络提取重要特征的能力，从表中可以看出，Model 4相对与Model3的mAP@0.5在FPS、参数量和浮点计算量几乎不变的情况下，上涨了0.7%，可以说明CBAM机制的增加了网络模型的深度，在保证速度不变的同时，提高了小目标的检测效果。

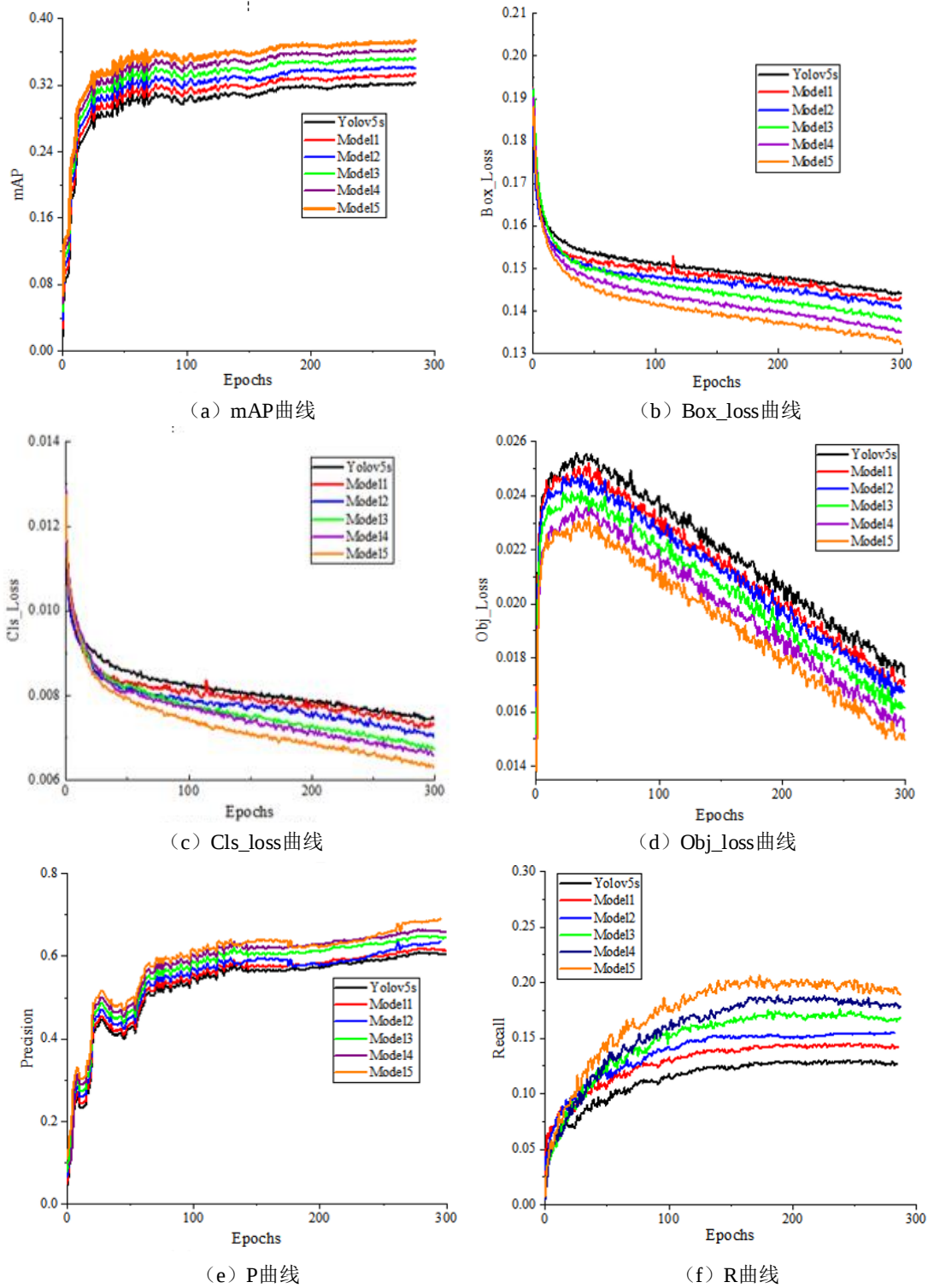


图 3-10 比较六个模型的训练结果

用 Soft-NMS 算法取代 NMS 算法后,模型有效的克服了待检测对象相互重叠时漏检的问题,模型的检测精度得到提高。YOLOv5_GBCS 相对于 YOLOv5s, mAP@0.5 方面提高到 4.7%; 在参数量、计算量方面,分别下降了 15.2%和 8.7%; FPS 从 53 提升到 59。基于上述可以得知, YOLOv5_GBCS 无论是在检测

效率方面还是检测精度方面都远远优于 Yolov5s。

表 3-2 五种改进方法的消融研究

Model	P2	Ghost	BiFPN	CBAM	Soft-NMS	Parameters	GFLOPs	FPS	mAP@0.5
YOLOv5s	—	—	—	—	—	7.04M	16.0	53	38.5
Model1	√	—	—	—	—	7.27M	19.3	45	40.2
Model2	√	√	—	—	—	5.96M	14.6	59	41.3
Model3	√	√	√	—	—	5.96M	14.6	59	42.1
Model4	√	√	√	√	—	5.97M	14.6	59	42.8
Model5	√	√	√	√	√	5.97M	14.6	59	43.2

3.3 基于语义实现动态特征点剔除

为了使系统适应动态场景，在 ORBSLAM3 的基础上设置了基于 YOLOv5_GBCS 的目标检测模块。将检测到的物体语义信息应用到动态特征点消除步骤中。在每一帧的 RGB 图像输入系统后，使用 YOLOv5_GBCS 对象检测绘制不同对象的检测框。ORB_SLAM3 将所有对象分为动态对象、静态对象。动态对象，这是图像中首先要消除特征点的部分。静态对象特征点可以直接视为静态特征点。根据这些信息，判断特征点是属于动态点还是静态点。如果它们是动态点，则需要消除它们。对于静态对象，将直接保留所有特征点。由于 YOLOv5_GBCS 中的检测框是矩形的，如果直接剔除动态对象框中的所有特征点，就会错误地删除大量的静态点。在对前一帧检测框中的点进行光流跟踪来完成匹配点对的选择，判断其之间的距离，选择一个阈值来确定它们是静态点还是动态点。

3.3.1 光流法追踪

光流法是运动检测中常用的一种方法，其中计算整个图像中部分像素的运动称为稀疏光流^[72]。稀疏光流以 Lucas-Kanade 光流为主，又称 LK 光流^[73]。对动态目标框内的特征点进行光流跟踪。为了提高计算效率，只计算去除先验高动态目标后 ORB 特征点的光流场，因此采用 LK 光流。本文的动态目标检测算法没有对稀疏光流进行运动一致性分析，而只是利用稀疏光流跟踪一些特征点。在光流跟踪过程中，可以直接获得特征点之间的对应关系，而无需使用描述符计算过程。帧间跟踪的特征点可以求解基本的相机矩阵。然后，利用对极几何

约束对动态目标进行识别；光流图如图 3-11 所示，其中 I_{t1} 、 I_{t2} 、 I_{t3} 分别为图像在 t_1 、 t_2 、 t_3 时刻的灰度值。

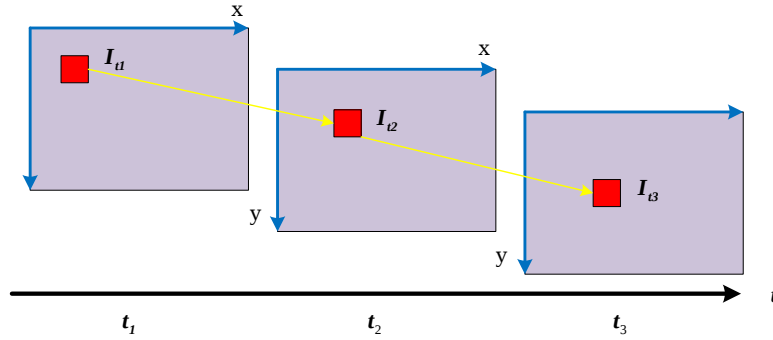


图 3-11 光流图

图像的灰度级可以看作是时间的函数。在时间 t ，位于图像中 (x, y) 的 ORB 特征点的灰度级可以写成 $I(x, y, t)$ ，而在时间 $t+dt$ ，它移动到 $(x + dx, y + dy)$ 。根据灰度级的不变性，可以获得以下结果：

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (3-11)$$

利用上面方程的泰勒展开并保留一阶项，可以得到：

$$I(x + dx, y + dy, t + dt) \approx I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt \quad (3-12)$$

由于假设灰度级恒定，则特征点在 t 时刻和 $t+dt$ 时刻的灰度值相等，可以得到：

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0 \quad (3-13)$$

两边除以 dt ，可以得到：

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} = -\frac{\partial I}{\partial t} \quad (3-14)$$

其中，像素在 X 轴和 Y 轴上的速度分别为 dx / dt 和 dy / dt (记为 u 和 v)，图像在 X 轴和 Y 轴上的方向梯度分别为 $\frac{\partial I}{\partial x}$ 和 $\frac{\partial I}{\partial y}$ (记为 I_x 和 I_y)， $\frac{\partial I}{\partial t}$ 表示灰度级随时间的变化(记为 I_t)。因此，可将上式变换为如下矩阵形式：

$$[I_x I_y] \begin{bmatrix} u \\ v \end{bmatrix} = -I_t \quad (3-15)$$

式(3-15)有两个未知数。为了求解 u 和 v ，假设像素在一个窗口中的运动相同，并设置一个 $2w$ 像素的“ $w \cdot w$ ”窗口。由于该窗口中像素的运动相同，可以得到如下 w^2 方程：

$$[I_x I_y]_k \begin{bmatrix} u \\ v \end{bmatrix} = -I_{tk} (k = 1, 2, \dots, w^2) \quad (3-16)$$

$$A = \begin{bmatrix} [I_x I_y]_1 \\ \vdots \\ [I_x I_y]_k \end{bmatrix}, b = \begin{bmatrix} I_{t1} \\ \vdots \\ I_{tk} \end{bmatrix} \quad (3-17)$$

则式(3-16)可写成:

$$A \begin{bmatrix} u \\ v \end{bmatrix} = b \quad (3-18)$$

利用最小二乘法,可以得到:

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = -(A^T A)^{-1} A^T b \quad (3-19)$$

在获得像素的运动速度 u 和 v 后,可以在多帧图像中估计出特定像素的位置,因此采用光流法对 ORB 特征点进行跟踪和匹配。光流法可以有效地避免误匹配。最后,通过多次迭代对像素点进行跟踪,得到效果最好的解。在得到相应的匹配点后,计算匹配点对的基本矩阵 F 。最后,对极几何对所有特征点进行约束,剔除出场景中的动态特征点。

3.3.2 对极几何约束

通过LK光流匹配得到基本矩阵后,利用基本矩阵和特征点的位置计算特征点对应的极线。当特征点与极线之间的距离大于某一特定值时,将其分类为动态特征点。

如图 3-12 所示,利用两帧极线之间的几何关系来检测动态特征点。

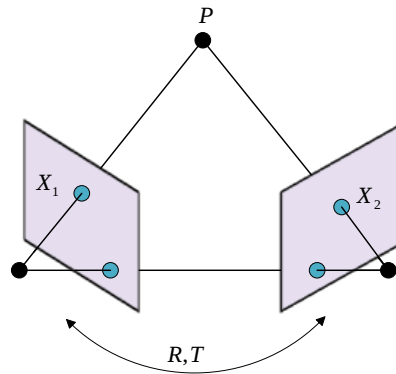


图 3-12 对极约束

假设一台照相机从不同角度观察空间中的同一点 P 。根据针孔相机模型,在 x_1 和 x_2 两帧图像中,点 $x = [u \ v]^T$ 的像素坐标符合以下约束条件:

$$\begin{cases} s_1 x_1 = Kp \\ s_2 x_2 = K(Rp + T) \end{cases} \quad (3-20)$$

式中， K 为摄像机的内参矩阵， R 和 T 分别为两个摄像机坐标系之间的旋转和平移矩阵， S 为像素的深度信息。在理想情况下，两帧图像中匹配点对的坐标符合以下约束：

$$x_2^T F x_1 = [u_2 \quad v_2 \quad 1] F \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0 \quad (3-21)$$

其中 F 是基本矩阵。将距离的计算定义如下：

$$D = \frac{|x_2^T F x_1|}{\sqrt{\|X\|^2 + \|Y\|^2}} \quad (3-22)$$

如果距离 D 大于阈值，则认为该点不满足极坐标约束。

这些点随着物体的运动而移动，产生误匹配，这就是动态点。因此，在这项研究中，对所有不符合极性约束的特征点进行了剔除。高效的动态对象处理策略可以准确地识别高动态环境中的动态和静态对象。

算法流程图如下：

表 3-3 基于 YOLOv5-GBCS 的语义动态点剔除伪代码

算法 1：基于 YOLOv5_GBCS 的语义动态点剔除	
输入：	RGB 图像
输出：	静态特征点
1：	通过 YOLOv5_GBCS 进行语义信息提取
2：	遍历每个物检测框
3：	if 是动态物体 then:
4:	遍历框内特征点进行 LK 光流追踪与对极几何约束计算，得到 D ;
5:	if $D > \text{阈值}$ then:
6:	剔除该动态特征点;
7:	else if $D < \text{阈值}$ then:
8:	保留该特征点;
9:	end if
10:	遍历结束;
10:	else if 是静态物体 then:
11:	保留语义检测框的静态特征点;
12:	end if
13:	遍历结束
14:	return 输出

基于 YOLOv5_GBCS 的语义动态点剔除的流程图如图 3-13 所示。

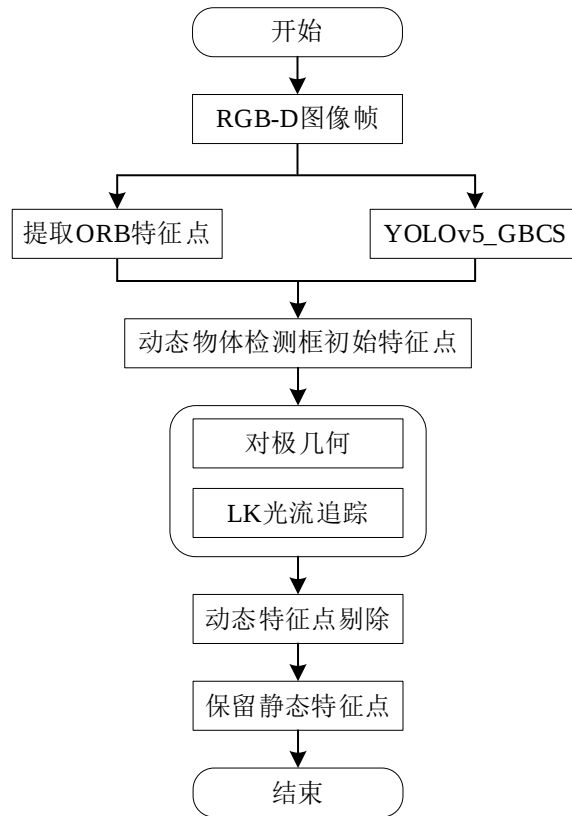


图 3-13 语义动态点剔除的流程图

3.3.3 动态点剔除效果验证

选择 fr3_walking_half、fr3_walking_xyz、fr3_walking_rpy、fr3_walking_static、fr3_sitting_half、fr3_sitting_rpy 与 bonn_crowd、bonn_person_tracking、bonn_moving_nonobstructing_box 这九组序列用于验证动态特征点剔除的有效性。

图 3-14 与图 3-15 显示了不同动态序列下特征的点检测结果，首先对这些动态序列进行特征点检测，红色检测框标出的是模型检测到的动态物体。在未采取任何特征点剔除措施前，动态物体周围的特征点仍被 SLAM 系统识别和使用，这会导致位姿估计的不准确。引入基于语义信息的动态特征点剔除方法后，这些动态物体相关的特征点被有效移除，可以减少这些动态因素对 SLAM 位姿估计的干扰，从而提高系统在动态环境中的稳定性和可靠性。这对于在复杂环境下应用 VSLAM 于无人机导航，具有重要的意义。

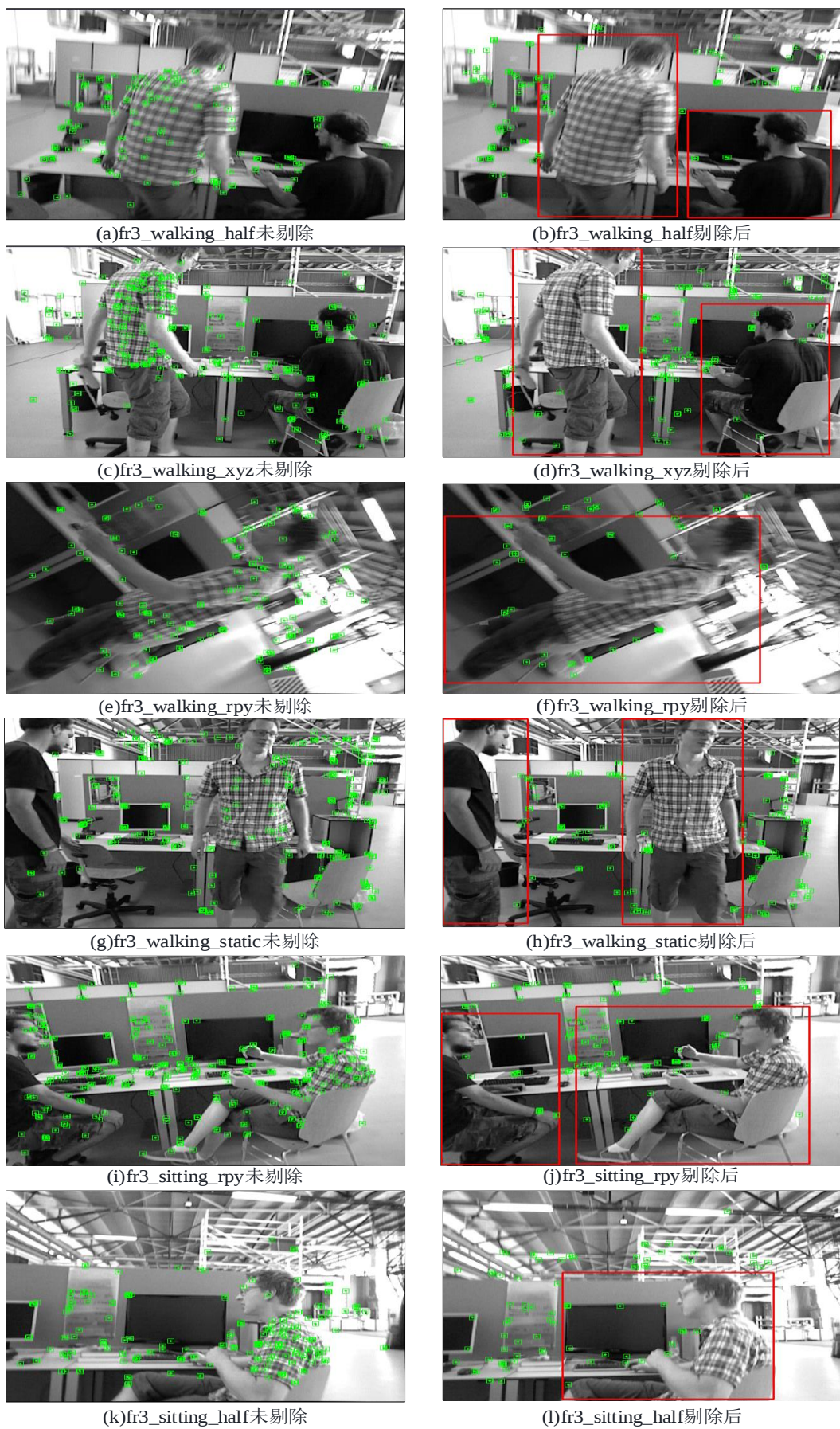


图 3-14 TUM 数据集上的动态特征点剔除效果图

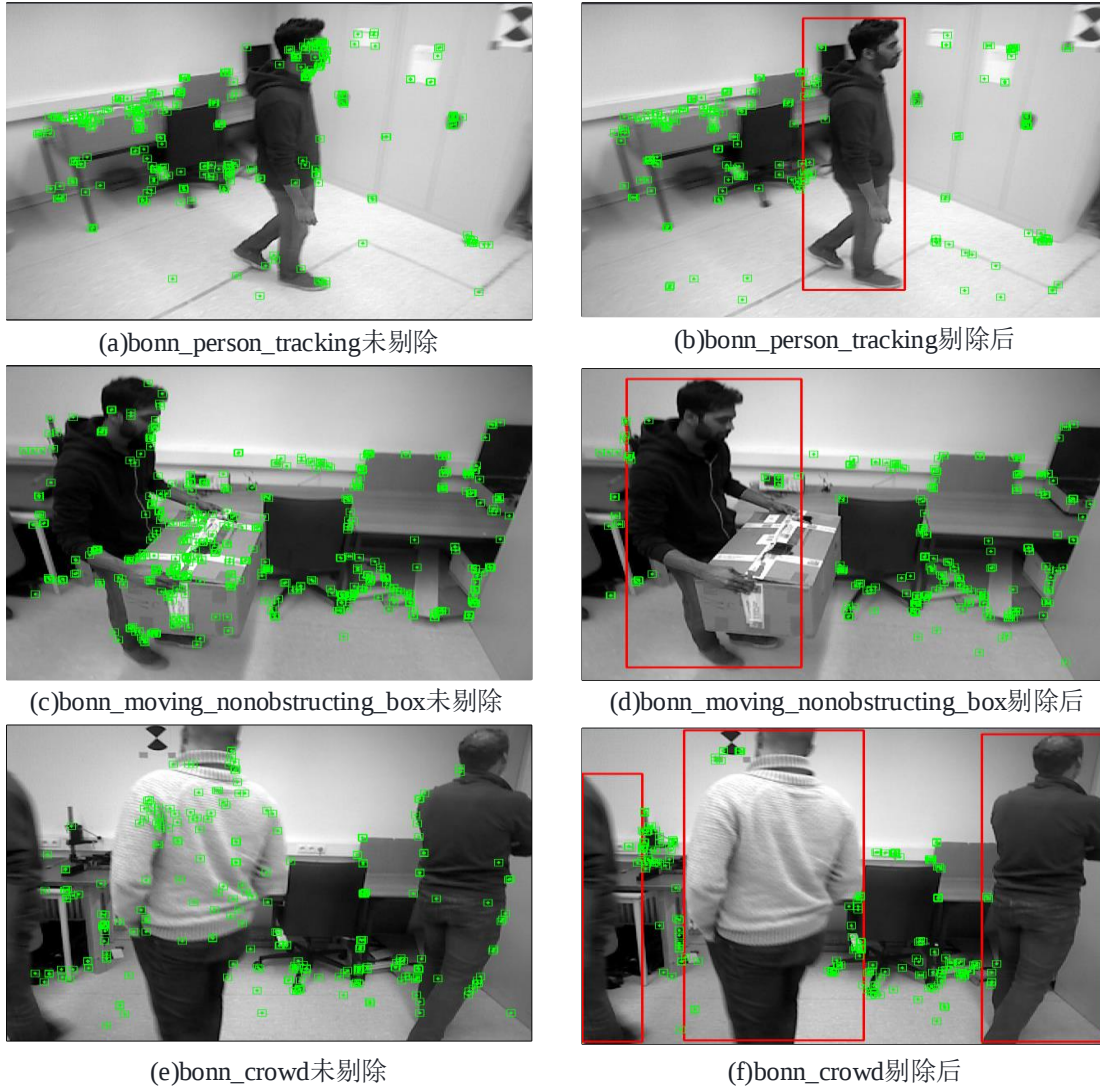


图 3-15 BONN 数据集上的动态特征点剔除效果图

3.4 本章小结

在动态室内环境中，VSLAM系统经常受到运动物体的影响，这可能导致跟踪失败，降低位置估计精度。为了解决这个问题并提高定位精度，采用YOLOv5_GBCS算法提取语义信息从而实现可以剔除动态特征点的视觉里程计，并与ORB-SLAM3相结合。在该方法中，YOLOv5_GBCS算法用于识别潜在的运动物体。随后，在动态目标的预测边界框中使用LK光流算法与几何约束来消除连续帧之间的动态特征点。旨在提高VSLAM在动态环境下的准确性和鲁棒性。

第 4 章 基于改进的 PSO 算法的无人机平滑路径规划

路径规划在无人机实现自主飞行方面具有重要意义，特别是在避免与威胁碰撞以确保安全运行方面。本章介绍了一种改进的量子粒子群优化(QPSO)算法，称为增强探索能力粒子群优化算法 E2APSO。相较于传统 PSO，该智能算法更广泛应用于实时问题求解。首先，E2APSO 基于种群多样性峰值，有效摆脱局部极小值，避免停滞，实现了快速收敛到最优路径等特性，从而减少计算量。其次，采用基于三次 B 样条曲线的路径平滑处理，使规划路径适合于旋翼无人机，实现无人机平滑路径以增强无人机飞行的稳定性，避免造成飘逸，提高在高障碍环境中确定可行且成本最低路径的速度。

4.1 粒子群优化算法

4.1.1 粒子群优化(PSO)

粒子群优化 (PSO) 被认为是一种基本但研究最广泛的群智能算法，它模仿鸟类和鱼群的社会行为。该算法使用搜索空间中种群（粒子）的随机分布进行初始化，其中每个粒子代表问题的候选解。在整个迭代过程中，它通过粒子之间的信息共享来寻找最优解。这些候选解的适用性在目标函数的帮助下不断评估，并相应地更新个体最佳解 (P_{best}) 和全局最佳解 (G_{best})。此外，粒子的当前状态 P_{best} 和 G_{best} 被不断用来更新这些粒子的位置和速度。这些位置和速度保持在可行范围内，以达到所需的解的精度。这一过程一直持续到终止条件得到满足为止，结果通常是向全局最优收敛。

设在 n 维搜索空间中，粒子总数由 N_{pop} 表示，而粒子 $l(l \in N_{pop})$ 的位置和速度分别由向量 $x_l = (x_{l1}, x_{l2}, \dots, x_{ln})$ 和 $v_l = (v_{l1}, v_{l2}, \dots, v_{ln})$ 表示。如果粒子的个人最佳位置由 $x_{P_{best}} = (x_{P_{best},1}, x_{P_{best},2}, \dots, x_{P_{best},N_{pop}})$ 描述，粒子的全局最佳位置由 $x_{G_{best}}$ 描述，则第 l 个粒子的新速度和位置可以通过方程进行数学计算。

$$v_l(t+1) = \omega v_l(t) + c_1 r_1 (x_{P_{best},l}(t) - x_l(t)) + c_2 r_2 (x_{G_{best}}(t) - x_l(t)) \quad (4-1)$$

$$x_l(t+1) = x_l(t) + v_l(t+1) \quad (4-2)$$

其中， ω 和 t 分别表示惯性权重和迭代次数。此外， c_1 和 c_2 为加速度常数，而 r_1 和 r_2 为随机数 ($r_1, r_2 \in [0,1]$)。通过式(4-3)和式(4-4)得到的粒子速度和位置保持在可行范围内。

$$v_l(t+1) = \begin{cases} v_{max}; v \geq v_{max} \\ v_{min}; v \leq v_{min} \end{cases} \quad (4-3)$$

$$x_l(t+1) = \begin{cases} x_{max}; x \geq x_{max} \\ x_{min}; x \leq x_{min} \end{cases} \quad (4-4)$$

其中， v_{max} 和 v_{min} 表示速度的上限和下限，而 x_{max} 和 x_{min} 表示位置的边界值。该算法已在算法 2 中提出给出。

表 4-1 PSO 算法伪代码

算法 2: PSO 算法流程
Input: $MaxIt, N_{pop}, d, c_1, c_2, \omega$ Output: G_{best}, F 1: 种群的随机初始化 2: for each $t(t \in MaxIt)$ do, 3: for each $i(i \in N_{pop})$ do, 4: for each $j(j \in d)$ do, 5: 通过方程 (4-1) 更新速度 6: 通过方程 (4-2) 应用速度边界 7: 通过方程 (4-3) 更新位置 8: 通过方程 (4-4) 应用位置边界 9: end 10: Fitness evaluation ($F(x_i) = J(x_i)$) 11: If $F(x_i(t+1)) < F(Pbest_i(t))$ then, 12: $Pbest_i(t) = x_i(t+1)$ 13: end 14: If $F(Pbest_i(t)) < F(G_{best})$ then, 15: $G_{best} = Pbest_i(t)$ 16: end 17: end 18: end

4.1.2 量子粒子群优化 (QPSO)

由式(4-1)(4-2)表示的 PSO 算法，尽管显示出良好的结果，但在可能不是局部最优的点上，存在停滞问题^[74]。为了解决这个问题。Kennedy^[75]提出，如果所有粒子都被下面方程中给出的点 p 所吸引，则可以保证局部最优收敛。

$$p_{i,d}(k) = \varphi pbest_i + (1 - \varphi)gbest \quad (4-5)$$

其中 $p_{i,d} = (p_{i,1} \ p_{i,2} \ \cdots \ p_{i,D})$ 被称为粒子 i 的局部吸引子，常数 φ 由下列方程计算得出。

$$\varphi = \frac{c_1 r_1}{c_1 r_1 + c_2 r_2} \quad (4-6)$$

根据式(4-5)和(4-6)，Sun 等人^[76]开发了一种名为量子行为粒子群优化 (QPSO) 的算法。该算法修改粒子的位置更新方程，即

$$x_{i,d}(k+1) = p_{i,d}(k) \pm L \times \ln\left(\frac{1}{u_{i,d}}\right) \quad (4-7)$$

其中， L 是粒子在一次迭代中所能移动的最大运动量的控制参数，且在每次迭代中减少，因此 QPSO 具有良好的性能。 L 由下式给出

$$L = \alpha |x(k) - mbest(k)| \quad (4-8)$$

其中 α 为：

$$\alpha = \alpha_1 + \frac{(T - k) \times (\alpha_0 - \alpha_1)}{T} \quad (4-9)$$

参数 α 应在算法的总迭代次数 T 上从初始值 α_0 减小到最终值 α_1 。而 $mbest$ 是群体所有最佳粒子适应度值的平均值，由下式计算

$$mbest(k) = \frac{1}{N} \sum_{i=1}^N pbest_i \quad (4-10)$$

由式(4-7)可以看出，QPSO 算法中没有速度更新步骤，因为该参数由 L 控制。因此，除了确保收敛外，该算法的计算成本也低于标准 PSO。QPSO 伪代码如算法 3 所示，流程图如图 4-1 所示。

表 4-2 QPSO 算法伪代码

算法 3: QPSO 算法流程	
1: Procedure QPSO	
2: 初始化: $n \leftarrow$ 种群; $t \leftarrow$ 迭代次数	
3: 用随机解初始化群	
4: for $i=1$ to t do	
5: for $j=1$ to n do	
6: 通过方程 (4-5) 计算 $p_{i,d}$	
7: 计算 $pbest_i$	
8: If $pbest_i < gbest$ then	
9: $gbest = pbest_i$	
10: 通过方程 (4-10) 计算 $mbest$	
11: 通过方程 (4-7) 计算 $x_{i,d}$	
12: end Procedure	

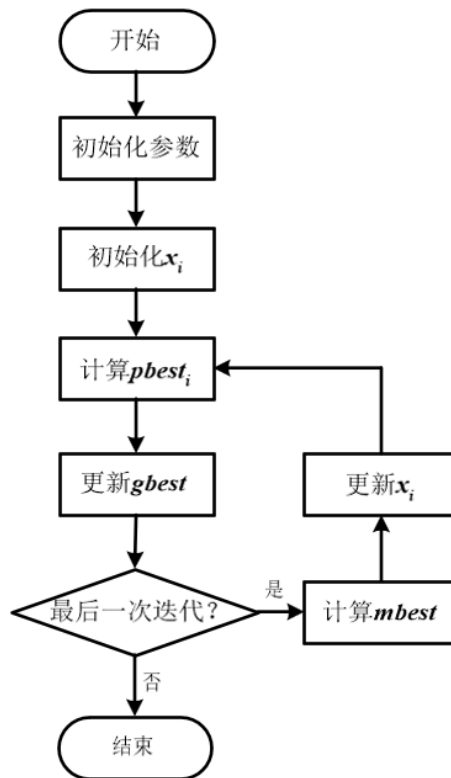


图 4-1 QPSO 算法流程图

QPSO 计算复杂度可以由创建和更新种群所需的时间 (T_{cu})、评估每个粒子的适应度所需时间(T_f) 和迭代次数(N_{ite}) 组成。因此, QPSO 算法的计算成本由

下式给出：

$$C_t = T_{cu} \times T_f \times N_{ite} \quad (4-11)$$

4.2 成本分析

轨迹优化并不一定是寻找两点之间最短路径的任务，根据任务的应用情况分析其他参数也很重要。在规划路线时，可以分析穿越某一区域的能耗、地形条件或无人机的安全性。在本小节中，成本函数使用了三个参数：路线长度、穿越的地形和避开障碍物，计算公式如下：

$$C = \vartheta C_l + \epsilon C_t + \mu C_c \quad (4-12)$$

其中 C_l 是轨迹长度成本， C_t 是穿越地形成本， C_c 是与环境中障碍物碰撞成本。 ϑ 、 ϵ 和 μ 是分别与轨迹范围、地形和碰撞相关的成本选择的参数。

参数 ϑ 的目的只是为轨迹长度成本提供权重，因为在不同环境条件下，轨迹长度成本的重要性皆不相同。在本研究中， ϑ 的值为 1。 ϵ 的目的是评估和惩罚特定地形；因此，对于完全平坦且完整的地形， $\epsilon = 1$ ；对于被认为不适合无人机飞行的环境， ϵ 需要被设置为相对较高的值。常数 μ 是一个数值较高的权重，因为避免碰撞是路径规划的主要目标之一。然而，如果该参数具有极高的值，则优化任务可能会受到影响，在本研究中，使用了 $\mu = 10$ ，此为反复测试的经验值，可以避免碰撞，但不会阻止算法在某些迭代期间找到无效解。

4.3 改进的 PSO 算法

在本文提出的算法中，粒子的位置更新通过式(4-7)进行，与 QPSO 类似。然而，QPSO 提出了使用粒子邻域的概念来计算最佳个体适应度($mbest$)的平均值。在式(4-10)中，验证了每次迭代时 $mbest$ 更新；但是，所有粒子的值都是相同的。

为了获得一个具有更多样性的粒子群，增强其探索能力，让每个粒子 i 具有不同的 $mbest$ ， $mbest$ 定义为粒子 i 与其 n 个邻居的最佳适应度值的平均值。从现在开始，这个参数被称为 $nbest_i$ 。

$$nbest_i(k) = \frac{1}{n+1} \left(pbest_i + \sum_{c=1}^n pbest_d \right) \quad (4-13)$$

其中 d 是相邻粒子相对于粒子 i 的索引。

所提出算法的另一个特点是在每次迭代中使用式 (4-7)。在其他时候，使用下列等式更新粒子的位置。

$$x_{i,d}(k+1) = x_{i,d}(k) - pbest_i + gbest + \varphi_i(k)R \quad (4-14)$$

其中 R 定义为每次迭代中取值范围为-1 到 1 的随机函数， $\varphi_i(k)$ 是一个比例因子，定义为：

$$\begin{aligned} \varphi_i(0) &= 1 \\ \varphi_i(k+1) &= \begin{cases} \gamma, & \text{if } ITE > ITE_t \\ \delta, & \text{otherwise} \end{cases} \end{aligned} \quad (4-15)$$

式(4-15)创建了一种根据每个粒子的性能来调整搜索空间的机制， $\gamma < 1, \delta > 1$ 。 f 是粒子的适应度值没有改善的连续次数。如果 ITE 的值大于阈值 ITE_t ，则降低参数 φ_i 的值以限制粒子的探测范围。当没有达到这个阈值时，希望扩展搜索空间。这部分算法基于保证收敛粒子群优化算法(GCPSO)，该算法倾向于快速收敛到局部最优。将该算法与改进的 QPSO 算法合并，旨在增强 GCPSO 算法在优化最后阶段的优势，在局部水平上利用其高收敛能力。改进的 QPSO 算法在第一次迭代时更频繁地激活，其中提出的算法的多样性旨在增加搜索范围的探索。

最后，每当优化连续停滞多轮迭代时，该算法就会引起粒子群的更新。这种更新包括在所有粒子中插入随机解，但不改变每个粒子的 $pbest$ 值以及粒子群的 $gbest$ 值。当被激活时，它负责增加粒子之间的多样性。

本研究将提出的算法称为增强探索能力粒子群优化。算法4显示了 E2APSO 伪代码，图 4-2 给出了算法的流程图。

表 4-3 E2APSO 算法伪代码

算法 4: E2APSO 算法流程
<pre> 1: Procedure E2APSO 2: 初始化: $n \leftarrow$ 种群; $t \leftarrow$ 迭代次数 3: 初始化 $ITE = 0$, $\varphi_i = 0$, ITE_t 4: 用随机解初始化群集 5: for $i=1$ to t do 6: for $j=1$ to n do, 7: 由式(4-5)计算 $p_{i,d}$ 8: 计算 $pbest_j$ 9: If $pbest_j < g_{best}$ then 10: $g_{best} = pbest_j$ 11: $ITE = 0$ 12: else 13: $ITE = ITE + 1$ 14: 由式(4-13)计算 $nbest_i$ 15: If $ITE > ITE_t$ then 16: $\varphi_i = -0.5$ 17: else 18: $\varphi_i = 2$ 19: 由式(4-14)更新 $x_{j,d}$ 20: If 经过 n 次迭代后, g_{best} 保持不变 then 21: 用随机解重新初始化群集 22: end Procedure </pre>

对于 E2APSO 算法的计算复杂度, 它由创建更新种群所需的时间(T_{cu}), 评估每个粒子的适应度的时间(T_f), 迭代次数(N_{ite}), 更新参数 φ (T_{up})的时间以及分析停滞和重新初始化群体的时间(T_a)组成。E2APSO 算法的计算代价为:

$$C_t = T_{cu} \times T_f \times T_{up} \times T_a \times N_{ite} \quad (4-16)$$

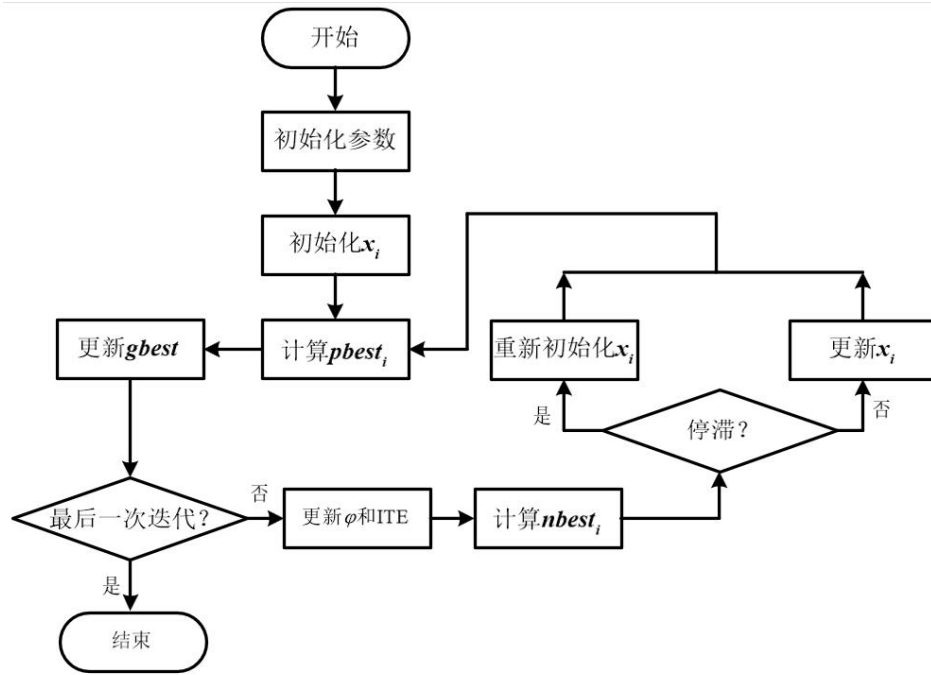


图 4-2 E2APSO 算法流程图

4.4 平滑路径生成

算法规划得到的无人机初始轨迹的轨迹点由直线直接连接，如果无人机沿着原始规划的路径移动，这将严重影响无人机飞行的稳定性，造成漂移。这显然不符合现实世界中的无人机轨迹。在平滑轨迹后，生成满足无人机性能约束的可行轨迹。

路径平滑可以考虑多种方法，例如：**b** 样条曲线、Bezier 曲线、仿线曲线、简单圆弧等。它们都需要满足无人机已知速度下的加速度约束。在许多方面，**b** 样条曲线具有局部性、几何不变性、连续性、对称性、递推性等特点。众所周知， n 条 **b** 样条曲线具有高度连续几何(Gn-h-1)特征，其中 h 表示重合控制点的个数。无人机的运动路径至少需要满足 G2 连续几何。因此，考虑的方法是三次 B 样条曲线(图 4-3)。

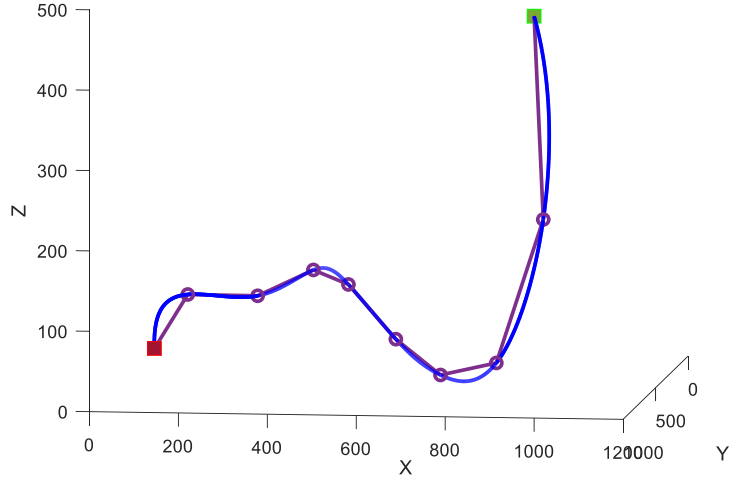


图 4-3 三次 B 样条曲线

在空间控制点上建立三次 B 样条曲线。控制点的数量总是大于 3 个。三次 B 样条基函数 $G_3(t)$ 的矩阵表示为：

$$G_3(t) = \frac{1}{6} \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}, t \in (0,1) \quad (4-17)$$

三次 B 样条曲线的矩阵表达式为：

$$\begin{cases} x(t) = G_3 X & X = [x_j, \dots, x_{j+3}]^T \\ y(t) = G_3 Y & Y = [y_j, \dots, y_{j+3}]^T, j = 1, 2, \dots, N_c - 3, \\ z(t) = G_3 Z & Z = [z_j, \dots, z_{j+3}]^T \end{cases} \quad (4-18)$$

式中 $[x_j, y_j, z_j]$, $[x_{j+1}, y_{j+1}, z_{j+1}]$, $[x_{j+2}, y_{j+2}, z_{j+2}]$, $[x_{j+3}, y_{j+3}, z_{j+3}]$ 为四个对应控制点的坐标;记下控制点的序号; N_c 为控制点总数; $x(t)$, $y(t)$, $z(t)$ 表示 t 从 0 到 1 时三次 B 样条曲线的坐标。

为了使三次 B 样条曲线适用于控制点个数小于 4 的特殊情况,增加两个控制点 x_0, x_{N_c+1} 如下:

$$\begin{cases} x_0 = x_1 \\ x_{N_c+1} = x_{N_c} \end{cases} \quad (4-19)$$

众所周知,无人机的加速度与路径曲率和无人机的速度有关。为简单起见,假设无人机的速度是恒定的。所以加速度,曲率半径和速度之间的关系是

$$a = \frac{v^2}{\rho} \quad (4-20)$$

式中 v 为无人机的速度, ρ 为空间曲线的曲率半径。考虑最大加速度约束,曲率半径需要满足:

$$\rho_{\min} = \min_{t \in [0,1]} \rho(t) \geq \frac{v^2}{a_{\max}} \quad (4-21)$$

式中， $\rho_{\min}$ 为最小曲率半径， $a_{\max}$ 为无人机的最大加速度， $\rho(t)$ 为曲率半径随 t 的变化量：

$$\begin{aligned} \rho(t) &= \frac{E(t)^{3/2}}{\sqrt{A(t)^2 + B(t)^2 + D(t)^2}}, \\ A(t) &= z''(t)y'(t) - y''(t)z'(t) \\ B(t) &= x''(t)z'(t) - z''(t)x'(t) \\ D(t) &= y''(t)x'(t) - x''(t)y'(t) \\ E(t) &= x'^2(t) + y'^2(t) + z'^2(t) \end{aligned} \quad (4-22)$$

换句话说，无人机的最大加速度取决于三次 B 样条曲线的控制点。因此，如果无人机的最大加速度是有限的，控制点也需要被限制。

4.5 实验结果与分析

4.5.1 基准函数测试

为了评估 E2APSO 的有效性，本小节使用 CEC2005 基准测试函数进行测试。CEC2005 基准测试由 23 种不同的功能的函数组成，用于将 E2APSO 与多种智能算法进行比较。这些算法是 PSO、NGO、SSA、WOA。PSO 是基于种群的优化算法，其具有快速收敛和操作简单等特点。NGO 是一种基于北方苍鹰捕猎行为的优化算法，具有全局搜索和局部搜索相结合的特点，可以处理复杂的优化问题，尤其适用于多峰值、非线性、离散和连续混合优化问题。SSA 通过模仿麻雀的觅食行为和反捕食行为实现位置寻优，以找到部分 NP 问题的局部最优值。WOA 是一种元启发式优化算法，模拟座头鲸狩猎行为的元启发式优化算法，采用随机或最佳搜索代理来模拟捕猎行为，并使用螺旋来模拟座头鲸的攻击机制。该算法具有机制简单、参数少、寻优能力强等优点。对于所有的比较算法，种群大小 N 和问题维数 D 都设置为 30，最大迭代次数为 500 或最大适应度评估次数为 25000。为了避免任何偶然引起的错误，每个算法都独立执行 30 次，为了全面分析所提出算法的鲁棒性，在表 4-4 中给出了平均误差(' mean ')和标准差('Std')。每个函数的最佳结果以黑体字突出显示。

只有 E2APSO 算法能够在 Fun12 上实现全局最优, 并在 Fun1、Fun2、Fun3、Fun4、Fun5、Fun8、Fun10、Fun12、Fun16、Fun17、Fun20、Fun21 等 12 个函数上产生最佳结果。在 Fun9 上, 所有算法产生的结果都不令人满意。具体来说, SSA 在 Fun6、Fun7、Fun14、Fun18 和 Fun19 上提供了最优结果。同样, NGO 在 Fun9 上获得最优值。WOA 在 Fun15 上得到了最好的结果。相比之下, PSO 的性能一般, 即使在某些函数上获得不错结果, 其他算法也都能取得。

总体而言, E2APSO 在 23 个功能中的 12 个功能中达到了最佳性能, 证明了它的能力, 对于更复杂的函数, 考虑粒子的分散程度可以增强优化能力。E2APSO 优于其他算法, 因此, 可以得出结论, 尽管 SSA 是一个搜索求解性能极强的算法, 但它仍然不如提出的 E2APSO。

表 4-4 基准函数测试结果

Function	Mean/Std	E2APSO	PSO	NGO	SSA	WOA
Fun1	Mean	4.8378e-114	1.1639e-27	3.8975e-87	2.1085e-55	1.6831e-74
	Std	2.6498e-113	1.9389e-27	1.2171e-86	1.1548e-54	8.9578e-74
Fun2	Mean	9.7944e-96	1.5006e-28	1.337e-88	7.2234e-58	3.7542e-73
	Std	5.3646e-95	1.5021e-28	1.9549e-88	3.4313e-57	2.0408e-72
Fun3	Mean	4.8648e-55	8.9466e-17	1.2709e-45	9.4818e-30	1.6059e-49
	Std	2.6501e-54	1.0177e-16	1.3013e-45	4.1484e-29	8.6652e-49
Fun4	Mean	3.1544e-55	1.2918e-05	8.4654e-22	7.2433e-26	39941.201
	Std	1.7277e-54	4.7113e-05	3.0331e-21	3.8702e-25	13966.1692
Fun5	Mean	2.2407e-52	9.4886e-07	1.7189e-37	1.6902e-25	48.3482
	Std	1.162e-51	1.3458e-06	2.1194e-37	9.2487e-25	32.0753
Fun6	Mean	25.7737	27.0992	25.9347	3.186e-05	27.8768
	Std	0.23384	0.73811	0.48228	9.192e-05	0.41793
Fun7	Mean	0.73169	0.67391	0.2699	1.1074e-10	0.50967
	Std	0.44523	0.21047	0.2707	4.4112e-10	0.19826
Fun8	Mean	3.1652e-227	6.0484e-51	3.1348e-170	7.5791e-128	4.151e-142
	Std	0.00e+000	1.993e-50	0.00e+000	3.995e-127	3.4874e-115
Fun9	Mean	0.0014315	0.0019145	0.00058617	0.001697	0.0044591
	Std	0.0010309	0.0010438	0.00027918	0.0015528	0.0049626
Fun10	Mean	5.8353e-112	1.2523e-94	1.8383e-179	1.5329e-45	5.4713e-103
	Std	3.1961e-111	6.7728e-94	0.00e+000	8.3961e-45	2.9614e-102
Fun11	Mean	0.72994	3.0755	0.00e+000	0.00e+000	0.00e+000

	Std	2.7779	4.9916	0.00e+000	0.00e+000	0.00e+000
Fun12	Mean	0.00e+000	1.0107e-13	6.2172e-15	8.8818e-16	3.7303e-15
	Std	0.00e+000	1.5056e-14	1.8067e-15	0.00e+000	1.7203e-15
Fun13	Mean	0.0040989	0.0050054	0.00e+000	0.00e+000	0.00e+000
	Std	0.022451	0.0084515	0.00e+000	0.00e+000	0.00e+000
Fun14	Mean	1.6004	2.1135	1.0317	2.7373e-09	14.3058
	Std	1.8896	1.0528	0.68739	4.2806e-09	18.968
Fun15	Mean	0.00044727	0.00039756	1.6198e-46	9.8814e-09	1.589e-51
	Std	0.0014301	0.00061705	1.4835e-46	2.9326e-08	7.786e-51
Fun16	Mean	0.00e+000	0.00e+000	0.00e+000	0.00e+000	0.00e+000
	Std	0.00e+000	0.00e+000	0.00e+000	0.00e+000	0.00e+000
Fun17	Mean	3.4589e-27	8.5933e-07	9.5159e-09	4.4867e-19	490.0499
	Std	1.8943e-26	2.9349e-06	2.591e-08	1.5801e-18	119.9522
Fun18	Mean	1.5311	10.8621	6.5573	2.2344e-08	0.045757
	Std	1.049	0.76691	0.6098	1.155e-07	0.18524
Fun19	Mean	2.6428	3.9736	1.0046	3.95e-11	1.3354
	Std	3.0343	2.0458	1.0783	6.1676e-11	1.1519
Fun20	Mean	1.0134e-99	1.4874e-24	2.0657e-83	1.0218e-22	3.3876e-69
	Std	4.5645e-99	1.8082e-24	3.5895e-83	4.0537e-22	1.3833e-68
Fun21	Mean	0.00e+000	0.00e+000	0.00e+000	-0.99999	0.00e+000
	Std	0.00e+000	0.00e+000	0.00e+000	1.6149e-05	0.00e+000
Fun22	Mean	0.083193	0.17484	0.099873	3.0075e-32	0.1099
	Std	0.037842	0.043069	2.1965e-13	1.5028e-31	0.071156
Fun23	Mean	0.0093921	0.03855	0.0097159	0.00e+000	0.017301
	Std	0.0017739	0.021073	9.5643e-14	0.00e+000	0.019358

为了进一步分析 E2APSO 在整个迭代过程中的性能，画出了所有优化算法的迭代曲线。图 4-4 给出了不同算法在 23 个基准函数上的收敛曲线，其中横坐标表示目标函数平均值，纵坐标表示迭代次数。

如图 4-4 所示，E2APSO 在 Fun1、Fun2、Fun3、Fun4、Fun8、Fun12、Fun20 和 Fun26 上的收敛速度比其他算法更快，说明算法有效摆脱局部极小值，避免停滞，实现了快速收敛。此外，E2APSO 在整个迭代过程中保持了相对稳定的收敛速度，优于对比的其他算法。因此，可以得出结论，即使收敛率相似，E2APSO 也比其他算法更稳定、更高效。增强多样性粒子群优化算法保证全面

的可搜索性，并且极大的提高了收敛速度。

E2APSO 在大多数情况下收敛速度最快。从 F4 的收敛曲线可以看出，大多数比较算法在迭代开始时都陷入局部最优，而 E2APSO 则继续收敛到最优值。在搜索最优解的过程的同时，E2APSO 仍然具有良好的种群多样性。

因此，基于大量的基准测试和统计分析，可以得出结论，与其他算法相比，E2APSO 算法在处理 30 个维度的基准测试方面更有效。

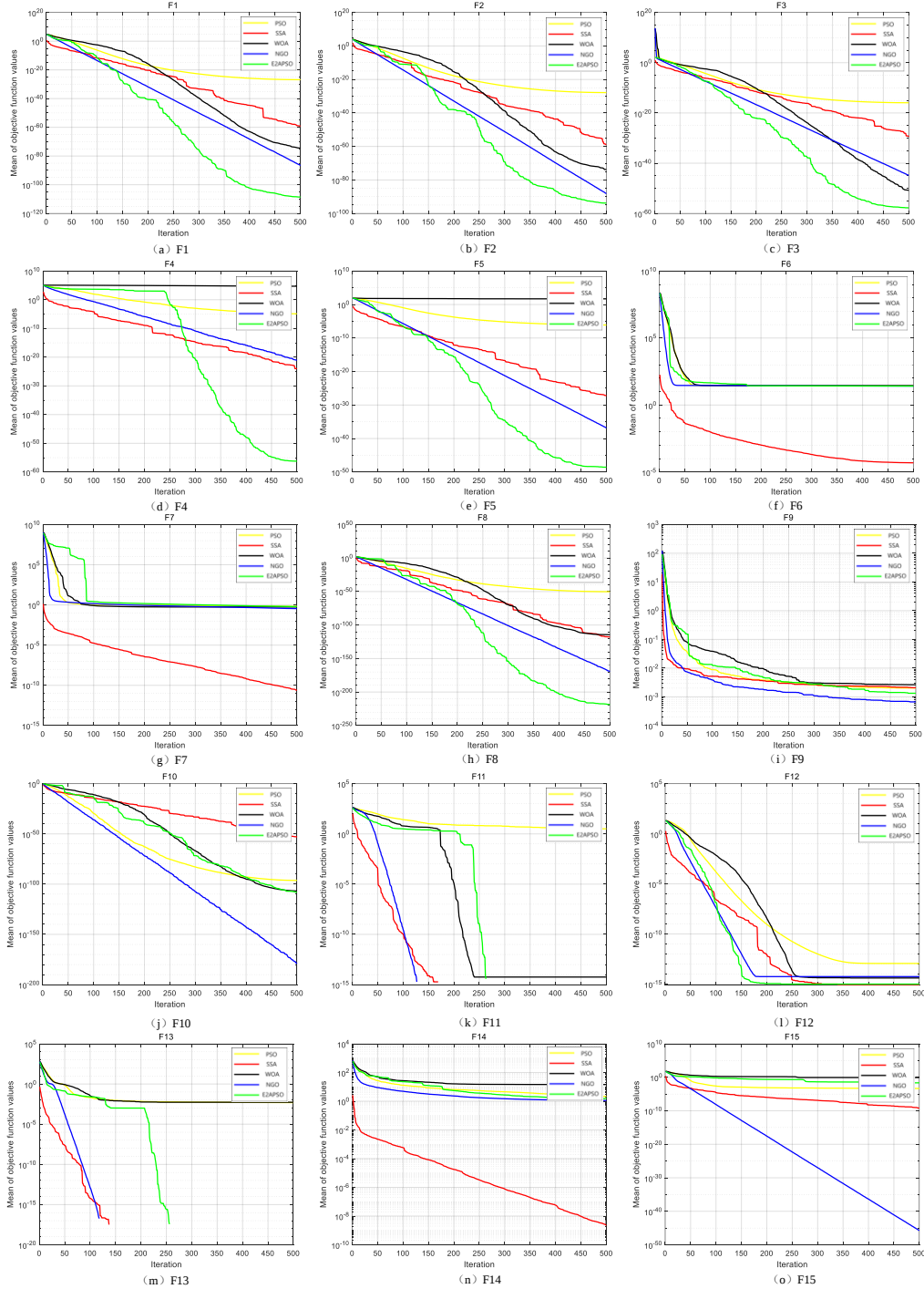


图 4-4 基准函数收敛曲线

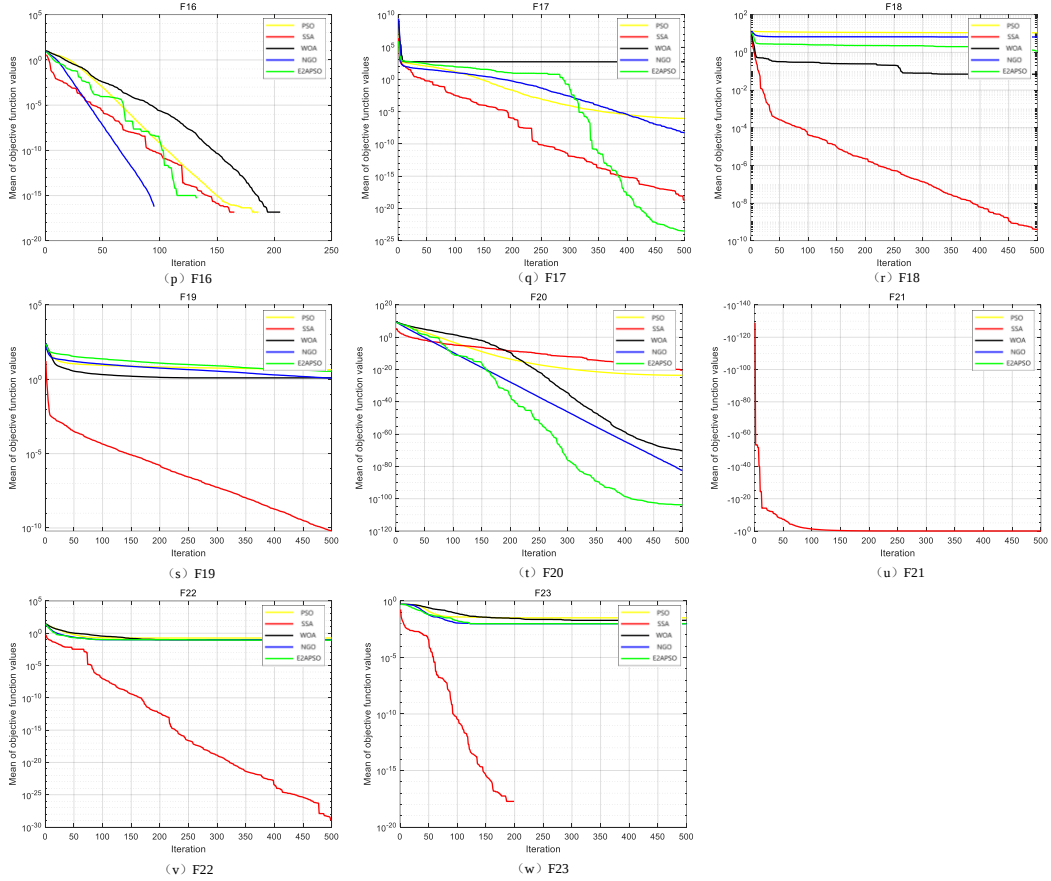


图 4-4 基准函数收敛曲线（续）

4.5.2 不同环境下的路径规划实验对比分析

为了验证 E2APSO 算法在保证客观性和公平性的前提下解决无人机路径规划问题的有效性和可行性，对提出 E2APSO 和 COA、SSA、GRO 和 GWO 进行了比较分析，全面验证了 E2APSO 算法的航迹优化性能。在三种不同的环境中进行了实验仿真。为了验证算法的性能，需要避免路径初始化的影响，并对所有比较算法进行相同的路径初始化。

图 4-5 显示了五种算法在三维环境 1 中轨迹规划中的不同结果。虽然环境模型中的障碍物数量相对较少，但通过仿真可以看出，E2APSO 规划的轨迹满足路径规划的要求。SSA、GWO、COA 规划的飞行轨迹在初始阶段会产生局部不必要的轨迹，导致完整轨迹距离较长。从收敛迭代曲线可以看出，E2APSO 算法迭代速度更快，收敛性更好。E2APSO 算法只需要 120 次迭代即可获得全局最优解。SSA、GWO、COA 和 GRO 算法分别需要 150、169、236 和 250 次迭代才能获得全局最优值。迭代曲线表明，E2APSO 算法可以在 20 代内跳出局部极

值。从图 4-5 中可以看出，在环境 1 中，E2APSO 的 30 个实验的平均运行时间为 1.1863 秒，分别优于其他 4 种算法的 28% (SSA)、23% (GWO)、37% (COA) 和 41% (GRO)。与其他四种算法相比，平均路径长度分别减少了 48.1m、126m、8m、24.3m。这些数据证明了 E2APSO 在复杂场景下规划任务的显著优势。证明了 E2APSO 能够稳定地、快速地搜索到最优飞行轨迹。

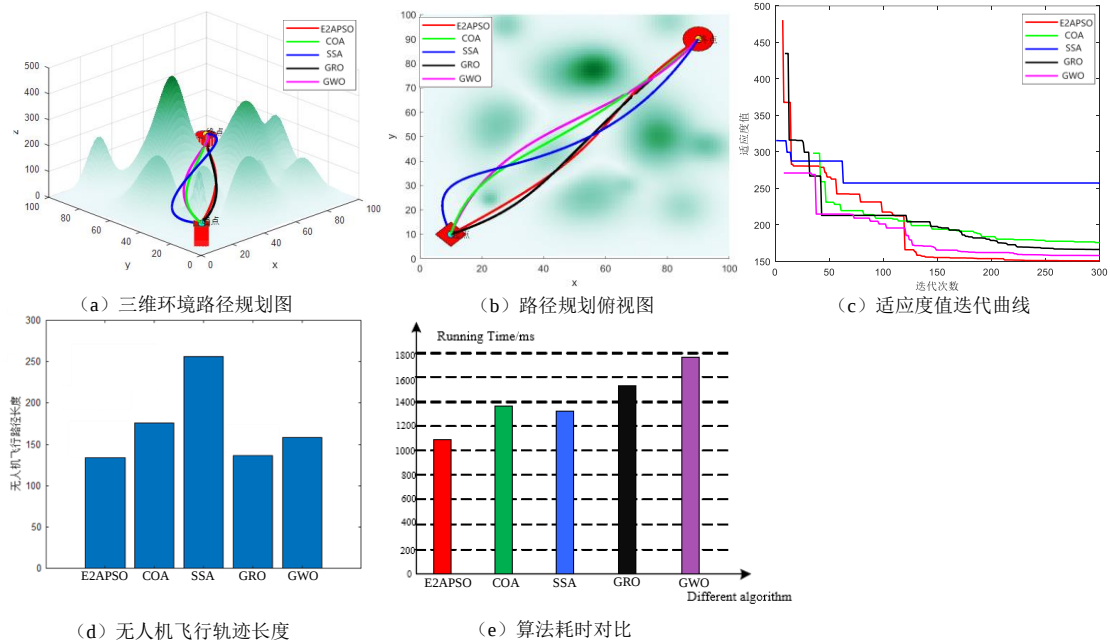


图 4-5 不同算法在环境 1 中的运行结果对比

在环境 2 中，设计了一个更复杂的环境模型来比较五种算法的性能。如图 4-6 所示，五种算法规划的航迹差异更为直观。SSA 和 COA 规划的航迹，一开始飞行方向就偏离其他算法很多，导致整个规划过程无用路径过多。GWO 和 GRO 的飞行轨迹接近，但是飞行轨迹较长。从仿真收敛曲线可以看出，E2APSO 的优化性能明显优于其他四种。E2APSO 只需要 120 次迭代即可求解全局最优值。然而，SSA 需要 130 次迭代，GWO 需要 150 次迭代，COA 需要 235 次迭代，GRO 需要 210 次迭代。在环境 2 中，E2APSO 的 30 个实验的平均运行时间为 1.2269 秒，分别优于其他 4 种算法的 21% (SSA)、17% (GWO)、19% (COA) 和 39% (GRO)。与其他四种算法相比，平均路径长度分别减少了 53m、117m、46m、101m。说明 E2APSO 具有稳定的规划性能和良好的搜索效果。

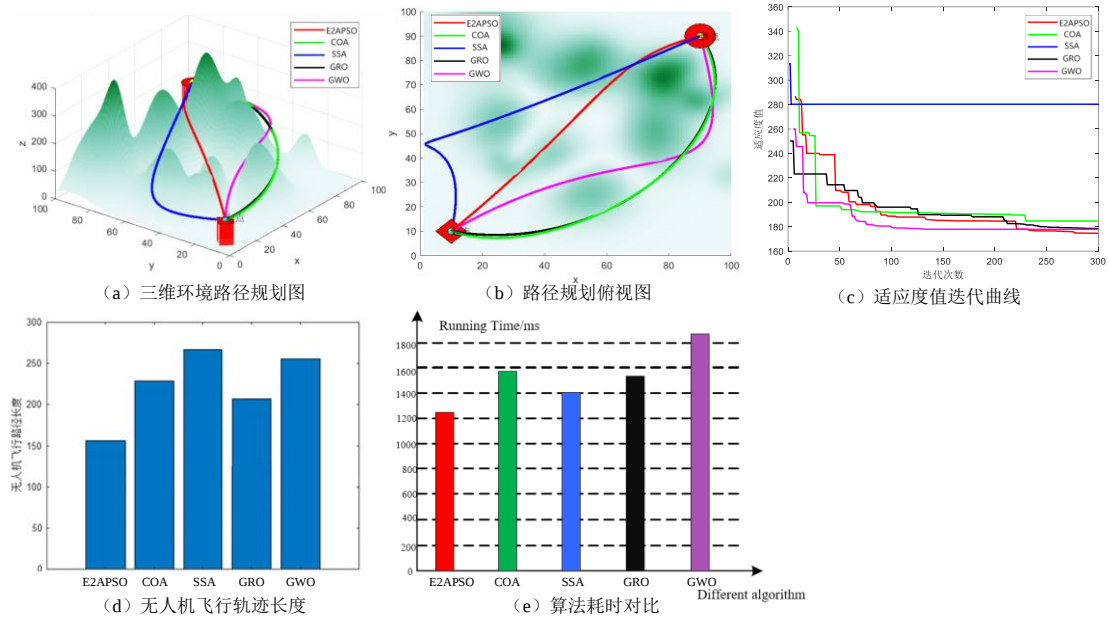


图 4-6 不同算法在环境 2 中的运行结果对比

在环境 3 中，建立了一个更复杂的飞行环境来测试五种算法的优化性能。图 4-7 显示了 5 种算法在三维环境轨迹规划中的不同结果。直观地看，E2APSO 规划的飞行轨迹可以很好地完成路径规划任务。COA 算法和 SSA 算法规划的路径距离比 E2APSO 算法规划的路径距离大。GWO 规划的航迹与 GRO 规划的航迹接近，但产生了局部不必要的航迹，增加了规划的完整航迹距离。在环境 3 的飞行轨迹中，E2APSO 算法的优化性能优于其他四种算法。图 4-7 给出了五种算法的收敛曲线，更明显地证明了 E2APSO 算法卓越的优化性能。从收敛曲线可以看出，对于复杂环境，E2APSO 只需要 56 次迭代即可获得全局最优解。而 SSA 迭代在 32 轮就陷入了局部最优解中无法摆脱，GRO 迭代接近其全局最优值的次数为 144 次，COA 迭代接近其全局最优值的次数为 163 次，GWO 迭代需要 179 次才能获得最优值。可以看出，E2APSO 算法效率高，收敛速度快。结果表明，E2APSO 算法可以有效地解决复杂环境下的无人机路径规划问题。

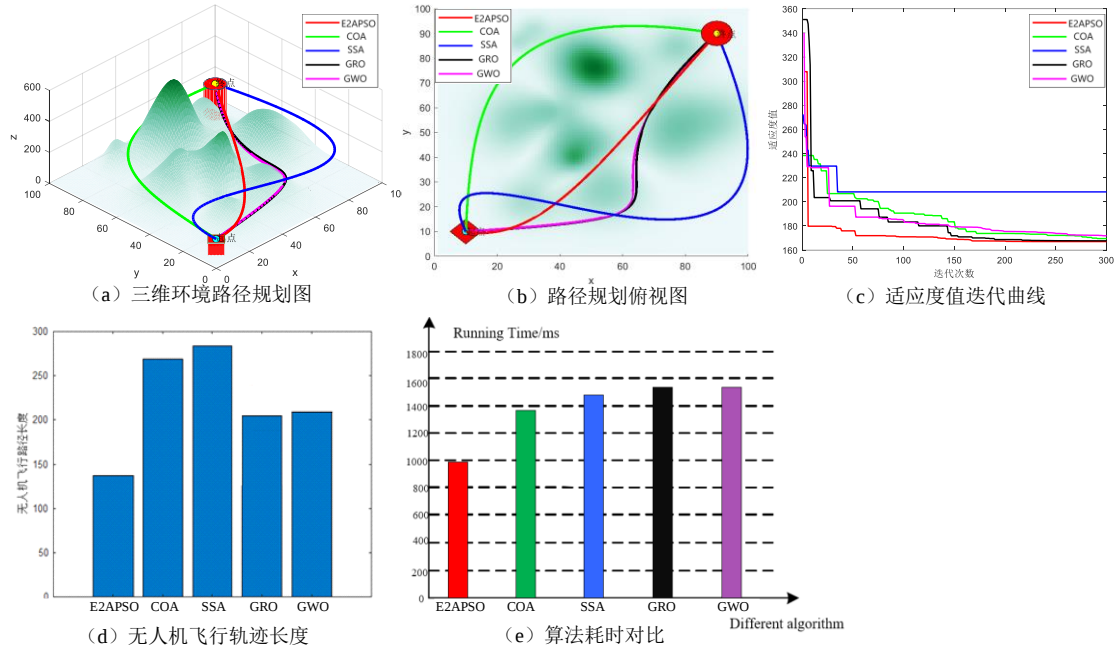


图 4-7 不同算法在环境 3 中的运行结果对比

从以上实验结果可以看出，E2APSO 算法在局部最优问题、收敛速度问题和飞行距离问题上优于其他四种算法。可以确定 E2APSO 算法可以为无人机设定安全可行、平滑、高效的飞行路径，且算法具有良好的稳定性。

4.6 本章小结

针对复杂环境下无人机的路径规划问题，提出了一种 E2APSO 算法，并设计了三种不同障碍物环境的仿真对比实验。实验结果表明，本文提出的改进算法在稳定性和收敛速度上都优于其他四种比较算法，表明本文提出的算法可以有效地解决无人机在三维空间中的轨迹规划问题。E2APSO 基于种群多样性峰值，有效摆脱局部极小值，避免停滞，实现了算法的收敛性和多样性，能够很好地应用于复杂环境下多约束的无人机路径优化问题。除此之外还减少计算量，生成高质量的解，提高了算法的运行效率。其次，采用基于三次 B 样条曲线的路径平滑处理，实现无人机平滑路径以增强无人机飞行的稳定性，避免造成飘逸，还可以大大减少无用路径的产生。

第 5 章 无人机建图与自主导航系统实验

5.1 自主导航系统设计

5.1.1 系统硬件设计

为了支持和验证基于语义 VSLAM 的无人机自主导航系统的在实际四旋翼无人机上部署的效果，构建了一个无人机硬件平台，如下图所示。下面将介绍构成这种系统的关键硬件组件。



图 5-1 无人机硬件平台

Jetson Xavier NX 代表了 NVIDIA 在 AI 计算领域的创新，提供了超小型但功能强大的计算平台，适用于机器学习和深度学习应用。这款板载计算机配备了一个 6 核 Carmel ARM 64 位 CPU 和 384 个 NVIDIA CUDA 核心，加上 48 个 Tensor 核心，为复杂的 VSLAM 任务提供了必要的计算能力。无人机通过使用 Jetson Xavier NX 进行实时图像处理和数据分析，从而能够识别和理解其周围环境的语义信息。

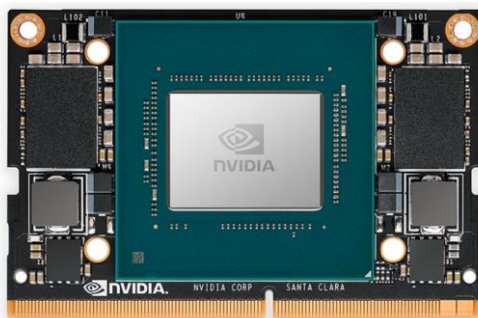


图 5-2 Jetson Xavier NX

Intel RealSense D435i 深度相机是基于语义 VSLAM 的自主导航无人机系统的另一个关键组件。它为系统提供了丰富的立体视觉信息，使无人机能够测量和理解物体的大小、形状和空间位置。具体而言，这款相机结合了深度感测、IR 传感器和全色摄像头，可以在各种照明条件下有效地捕捉高质量的三维景观映像。



图 5-3 Intel RealSense D435i 深度相机

PX4 是一个开源的无人机飞行控制软件堆栈，提供稳定、灵活和可靠的飞行控制解决方案。通过 PX4 飞行控制器，无人机能够执行精准的飞行操作和高级导航策略。此外，PX4 飞行控制器通过与 Jetson Xavier NX 的高速通信链接，可以快速响应环境变化和导航命令，为无人机在复杂环境中的自主飞行提供了坚实的基础。



图 5-4 PX4 飞行控制器

T-motor 2216 电机是专为多旋翼无人机设计的，提供高效、可靠的动力输出。这些电机通过确保无人机有稳定、精准的飞行性能，对于执行细微的操控指令至关重要，从而使无人机能够在执行 VSLAM 算法时保持最佳飞行状态。



图 5-5 T-motor 2216

MPU6000 是一款集成了 3 轴陀螺仪和 3 轴加速度计的传感器，为无人机提供关于其运动状态的精确数据。它为 PX4 飞行控制系统提供了实时的运动反馈，是实现平稳飞行和精确操控的关键。

以上硬件组件协同工作，各司其职，共同构成了一个高度集成的系统。Jetson Xavier NX 作为核心处理单元，负责执行复杂的图像处理和语义理解任务。Intel RealSense D435i 深度相机提供丰富的视觉信息，让系统能够识别和定位三维空间内的对象和障碍。PX4 飞行控制器确保了无人机的飞行性能和稳定性，而 T-motor 电机和 MPU6000 传感器保证了无人机飞行的动力输出和姿态平衡。

5.1.2 系统软件框架

构建的系统软件框架如图 5-6 所示。

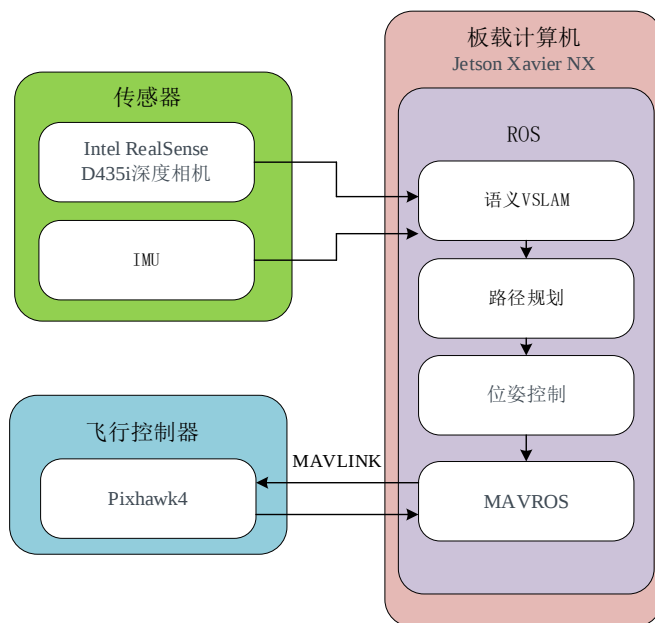


图 5-6 系统软件框架结构图

基于语义 VSLAM 的无人机自主导航系统的软件框架大体可以分为以下几个核心组件：

感知模块：使用深度相机传感器捕捉环境的 RGB 图像和深度信息。IMU 惯性测量单元为提供无人机的线速度、角速度、加速度等信息。

通信协议：MAVLINK 为无人机通信协议，实现指令和数据的传输。ROS/MAVROS 为机器人操作系统及其无人机通信接口，负责不同模块间的数据交换和指令传达。

数据处理和算法模块：解析传感器数据，执行同步定位和建图，提供环境及其语义信息；考虑到环境的语义信息制定飞行路线；结合 IMU 数据 VSLAM 输出估计无人机当前状态；基于规划路径和状态估计结果，生成控制指令控制无人机移动。

主控硬件平台：使用 Jetson Xavier NX 高性能计算模块，处理来自 IMU 和深度相机的数据，并运行 VSLAM 和控制算法。

飞行控制模块：飞行控制器执行飞行指令，维持无人机的稳定飞行。整个框架的工作流程如下：

- （1）数据采集：深度相机和 IMU 持续采集飞行环境和无人机状态的数据。
- （2）数据传输：采集的数据通过在 Jetson Xavier NX 上运行的 ROS 进行处理。
- （3）环境理解：VSLAM 算法使用摄像头数据进行环境的同步定位和建图，并加入语义理解，识别并标注重要特征或物体。
- （4）状态估计：利用 VSLAM 提供的位置数据和 IMU 数据，估计无人机的准确状态。
- （5）决策和规划：根据环境模型和当前状态，进行路径规划和飞行任务的决策。
- （6）控制指令生成：基于规划的路径，生成控制指令。
- （7）MAVROS 交互：控制指令通过 MAVROS 转化为 MAVLINK 指令。
- （8）飞行控制实施：Pixhawk4 接收 MAVLINK 指令，并控制无人机动作实现自主飞行。

5.2 语义 VSLAM 性能评估

本节将详细介绍 ORBSLAM3-ours 在动态场景中的表现。在公开的 TUM RGB-D 数据集和 BONN RGB-D 动态数据集上对 ORBSLAM3-ours 的性能进行了评估。为了定量评估该方法的性能，使用了两个常用的指标：绝对轨迹误差(ATE)和相对位置误差(RPE)。ATE 是估计姿态与实际姿态的直接差值，它直接反映了算法的准确性和轨迹的全局一致性。而 RPE 主要描述固定时间内两帧之间位姿差的精度，用于测量每秒的里程漂移，包括以 m/s 为单位的 RPE_t 平移漂移和以 %/s 为单位的 RPE_r 旋转漂移。不同序列的定量对比采用均方根误差(RMSE)和标准差误差(SD)作为评价指标。RMSE 表示估计值与真值之间的偏差，SD 表示相机轨迹估计值的离散度。

5.2.1 基于 TUM RGB-D 数据集的性能评估

本小节实验选择来自 TUM RGB-D 数据集的 4 个高动态序列和 2 个低动态序列来评估系统的定位精度。每个序列包含 640 * 480 大小的 RGB 图像和深度图像，以及高精度运动捕捉系统获得的地面真实轨迹。fr3_sitting_half 和 fr3_sitting_rpy 为低动态序列，其中两个人坐在桌子前，四肢有轻微的运动。fr3_walking_static、fr3_walking_xyz、fr3_walking_rpy、fr3_walking_half 为高动态序列，其中有两个人绕着桌子走。相机运动有四种类型：几乎静止(static)，沿着 x-y-z 轴(xyz)移动，沿着直径为 1 米的半球体(half)的轨迹移动，以及沿着滚动，俯仰和偏航轴(rpy)旋转。

在图 5-7 中，上图中蓝线为 V-SLAM 系统的轨迹，黑线为地面真实轨迹，红线段用于连接两条轨迹上相同的时间戳点。红色段越长，在那个时间点的误差轨迹就越大。从图 5-7 的结果可以看出，ORBSLAM3-ours 的误差轨迹小于 ORBSLAM3，特别是在 fr3_walking_static、fr3_walking_xyz、fr3_walking_rpy、fr3_walking_half 序列中，该方法的绝对轨迹误差显著降低。

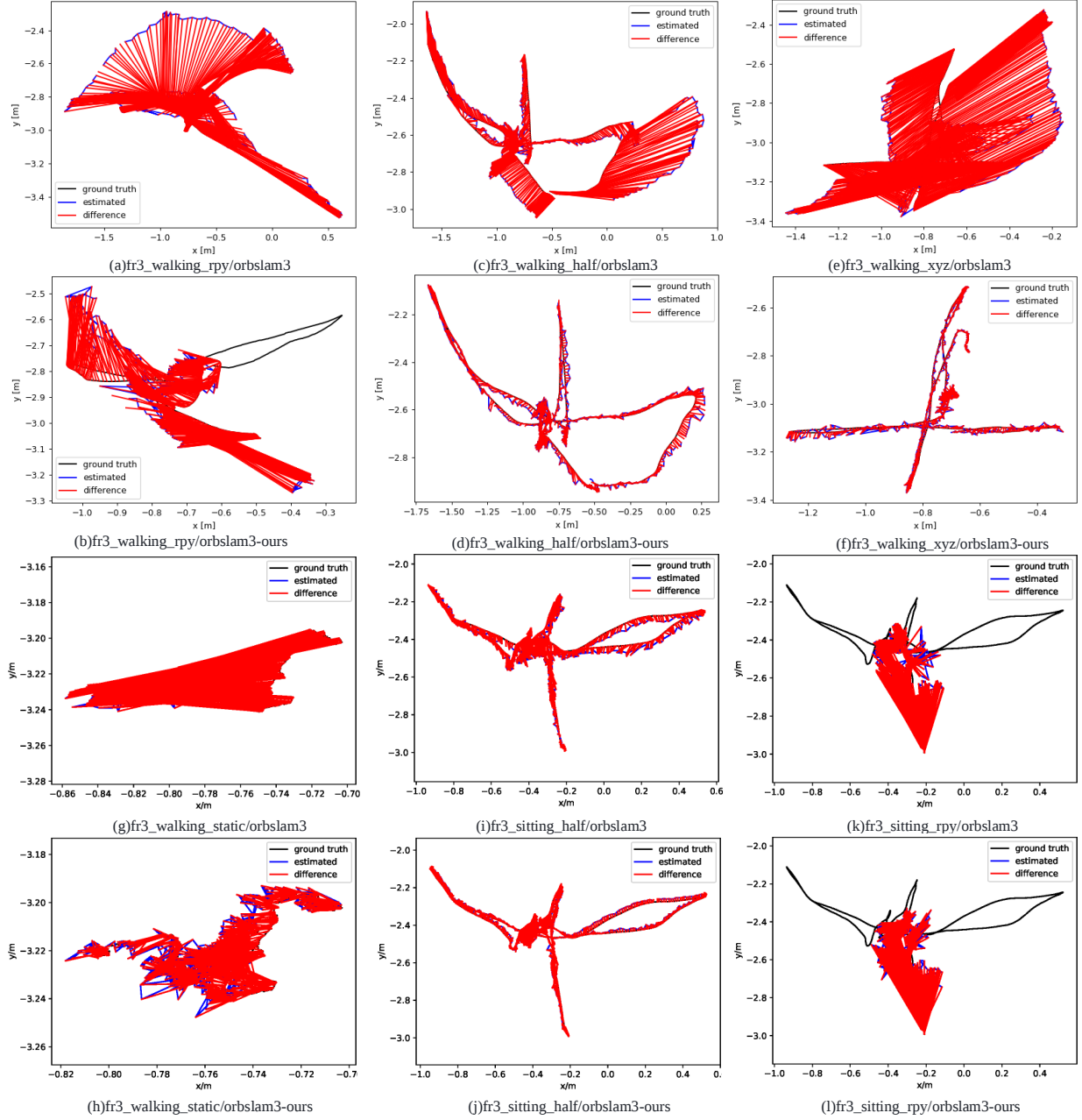


图 5-7 TUM 数据集多序列 ATE 结果

接下来，计算了 ATE 的 RMSE 和标准差（SD）。如表 5-1 所示。与原有的 ORBSLAM3 算法相比，ORBSLAM3-ours 算法在低动态和高动态环境下的 RMSE 和 STD 显著降低。特别是在高动态环境下，效果更为明显。计算了高动态数据集的平均 RMSE 和 SD，分别降低了 86.92%和 83.98%，低动态数据集的平均 RMSE 和 SD，分别降低了 51.82%和 63.13%。

表 5-1 TUM 数据集绝对轨迹误差结果[m](ATE)

Sequence	ORBSLAM3		ORBSLAM3-ours		Improvements	
	RMSE	SD	RMSE	SD	RMSE(%)	SD(%)
fr3_walking_xyz	0.4653	0.2900	0.0122	0.0120	97.40	95.93%
fr3_walking_rpy	0.9410	0.4702	0.2217	0.1669	76.44%	64.50%
fr3_walking_half	0.4901	0.1832	0.0970	0.0302	80.19%	83.51%
fr3_walking_static	0.2151	0.1198	0.0137	0.0096	93.63%	91.98%
fr3_sitting_half	0.0264	0.0124	0.0081	0.0031	69.32%	75%
fr3_sitting_rpy	0.0236	0.0160	0.0155	0.0078	34.32%	51.25%

表 5-2 和表 5-3 中的实验结果表明，与 ORBSLAM3 相比，ORBSLAM3-ours 在高动态序列中的大多数指标上都提高了 60%以上。图 5-8 显示了在六个序列下的 RPE 实验结果。如图 5-8(a)-(h)所示，ORBSLAM3-ours 在高动态场景序列中的估计结果的准确性明显高于 ORBSLAM3。综上所述，ORBSLAM3-ours 在提高高动态场景下 SLAM 系统的性能方面显示出良好的效果。

表 5-2 TUM 数据集 RPEt 平移漂移结果[m] (RPE)

Sequence	ORBSLAM3		ORBSLAM3-ours		Improvements	
	RMSE	SD	RMSE	SD	RMSE(%)	SD(%)
fr3_walking_xyz	0.6631	0.4735	0.0120	0.0115	98.17%	97.50%
fr3_walking_rpy	1.2800	0.9293	0.2821	0.3262	77.96%	64.89%
fr3_walking_half	0.8108	0.4240	0.1226	0.0621	84.87%	85.27%
fr3_walking_static	0.3041	0.1966	0.0103	0.0132	96.61%	93.28%
fr3_sitting_half	0.0398	0.0196	0.0218	0.0124	45.22%	36.73%
fr3_sitting_rpy	0.0343	0.0211	0.0226	0.0185	34.11%	16.81%

表 5-3 TUM 数据集 RPEr 旋转漂移结果[°](RPE)

Sequence	ORBSLAM3		ORBSLAM3-ours		Improvements	
	RMSE	SD	RMSE	SD	RMSE(%)	SD(%)
fr3_walking_xyz	12.369	9.3766	2.7594	1.9947	77.69%	78.73%
fr3_walking_rpy	24.801	15.183	11.923	9.903	51.92%	34.85%
fr3_walking_half	17.076	9.8720	3.0907	1.5016	81.90%	84.80%
fr3_walking_static	5.4106	3.3954	0.6944	0.3536	87.16%	89.58%
fr3_sitting_half	2.9016	1.5908	2.2189	1.1725	23.53%	26.29%
fr3_sitting_rpy	11.786	10.726	8.758	7.8542	25.69%	26.77%

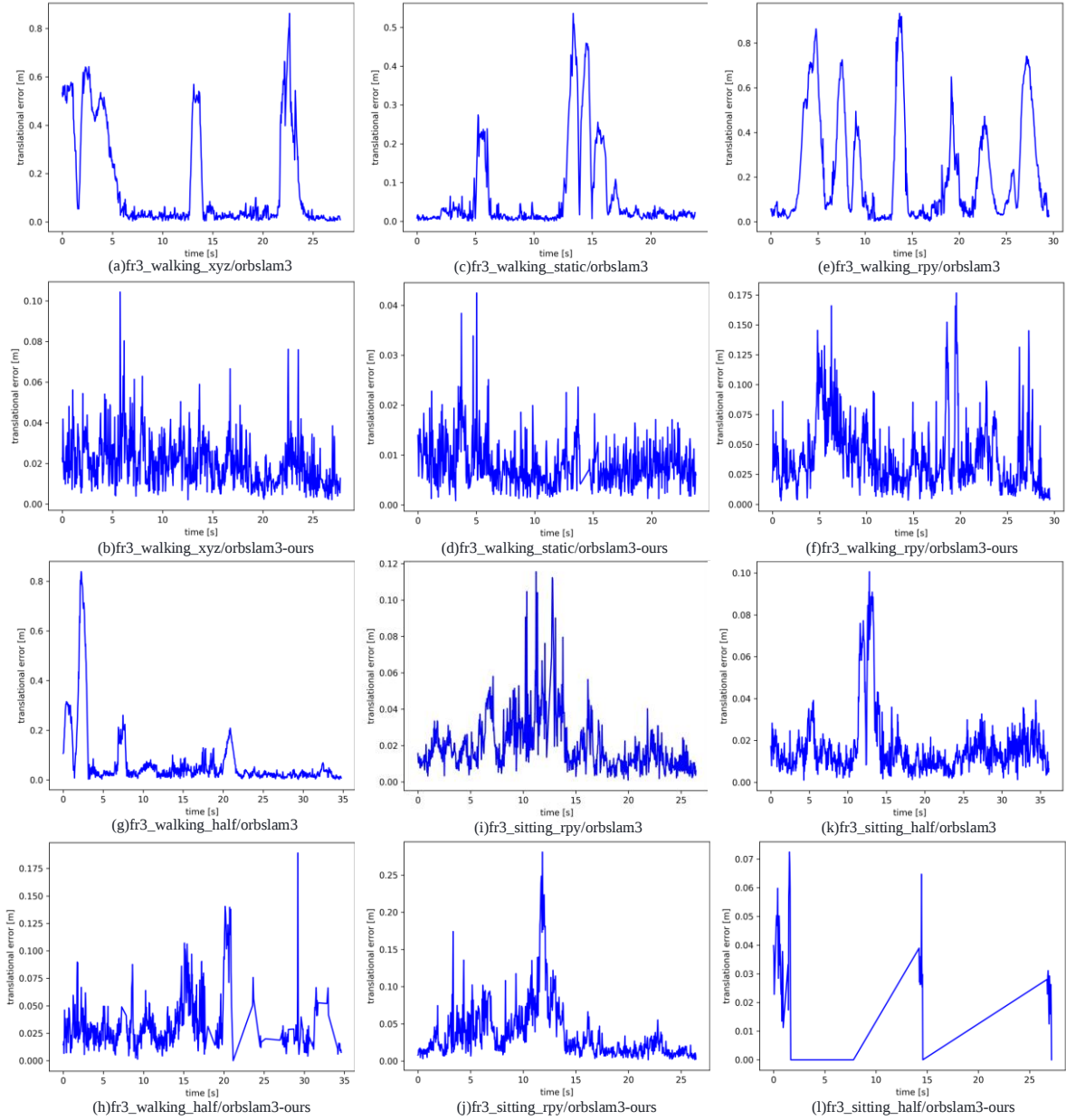


图 5-8 TUM 数据集多序列 RPE 结果

5.2.2 基于 BONN RGB-D 动态数据集的性能评估

本小节实验选择来自 BONN RGB-D 动态数据集集中的 `bonn_crowd` 和 `bonn_person_tracking` 这 2 个高动态序列和 `bonn_moving_nonobstructing_box` 这个低动态序列来评估系统性能，比 TUM 数据集的场景更具挑战性。它们具有更复杂的动力学特性，显著降低了 SLAM 系统的精度。

对比图 5-9 和表 5-4 中的 ATE 结果。可以看出，在 BONN 数据集中，动态信息更加复杂，ORBSLAM3-ours 仍然能够消除动态因素并取得很好的结果，即

使动态对象覆盖了整个区域的很大一部分，比如 `bonn_person_tracking` 序列中的一些帧，ORB_SLAM3-ours 仍然能够进行调整，取得很好的定位性能。在大多数序列中，ORB_SLAM3-ours 优于原始的 ORB_SLAM3，显著提高了轨迹估计的精度，计算得到的平均 RMSE 和 SD，分别降低了 94.81% 和 98.28%。

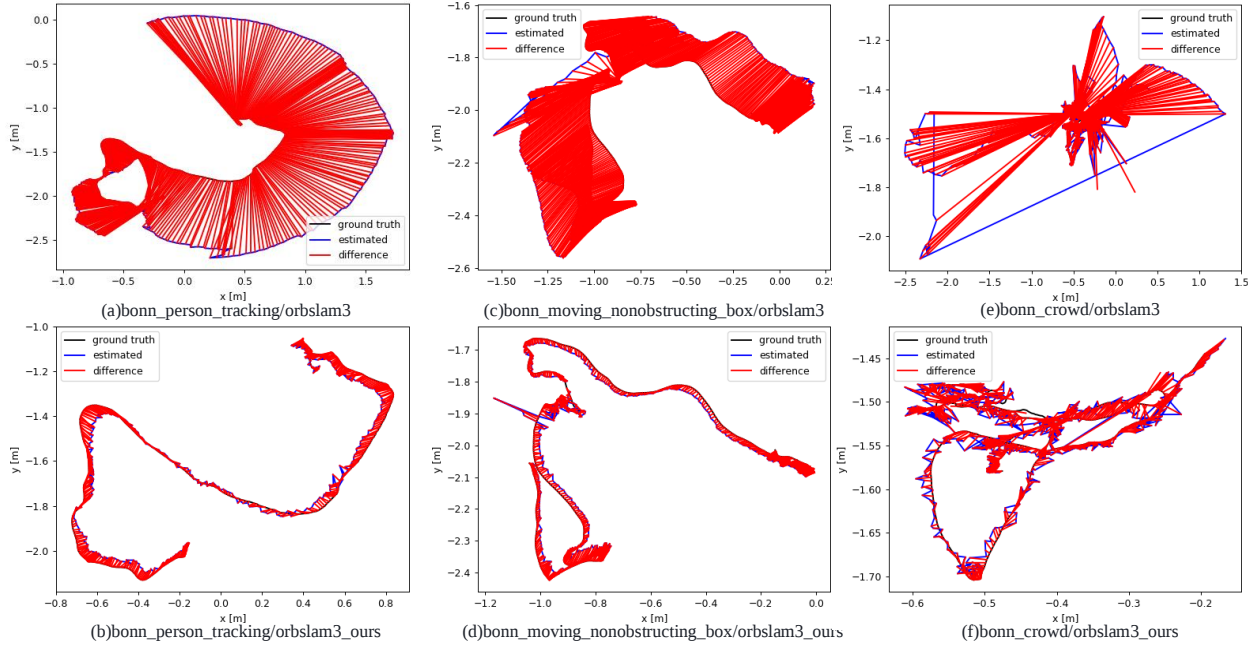


图 5-9 BONN 数据集多序列 ATE 结果

表 5-4 BONN 数据集绝对轨迹误差结果(ATE)

Sequence	ORB_SLAM3		ORB_SLAM3-ours		Improvements	
	RMSE	SD	RMSE	SD	RMSE(%)	SD(%)
bonn_crowd	0.4953	0.4131	0.0233	0.0135	95.29%	99.15%
bonn_person_tracking	0.6923	0.3389	0.0388	0.0112	94.39%	96.69%
bonn_moving_nonobstructing_box	0.3213	0.0905	0.0169	0.0009	94.74%	99.00%

根据平移和旋转测量的 RPE 对所提出的 ORB_SLAM3-ours 方法的性能进行了评估，如表 5-5 和图 5-10 所示。结果与前面提到的 ATE 研究一致，提供了系统视觉里程计性能的详细分析。RPE 反应了系统准确估计相机随时间的相对姿态的能力。所提出的 ORB_SLAM3-ours 方法在 RPE 方面表现良好，在平移和旋转测量中都提供了高精度和鲁棒性，计算得到的 RPE_t 和 RPE_r，分别降低了 65.04% 和 58.63%，证实了其在高动态场景下优于现有 SLAM 系统的优势。

结果表明，ORB_SLAM3-ours 方法显著提高了 SLAM 系统在高动态场景下的鲁棒性和稳定性。这在拥挤的场景中尤为明显。

表 5-5 BONN 数据集 RPEt 与 RPEr 漂移结果

Sequence	ORB_SLAM3		ORB_SLAM3-ours		Improvements	
	RPEt	RPEr	RPEt	RPEr	RPEt (%)	RPEr (%)
bonn_crowd	0.1321	3.1913	0.0144	0.6342	89.09%	80.13%
bonn_person_tracking	0.0306	1.5812	0.0132	0.8001	56.86%	49.40%
bonn_moving_nonobstructing_box	0.0238	0.9310	0.0121	0.4352	49.16%	46.37%

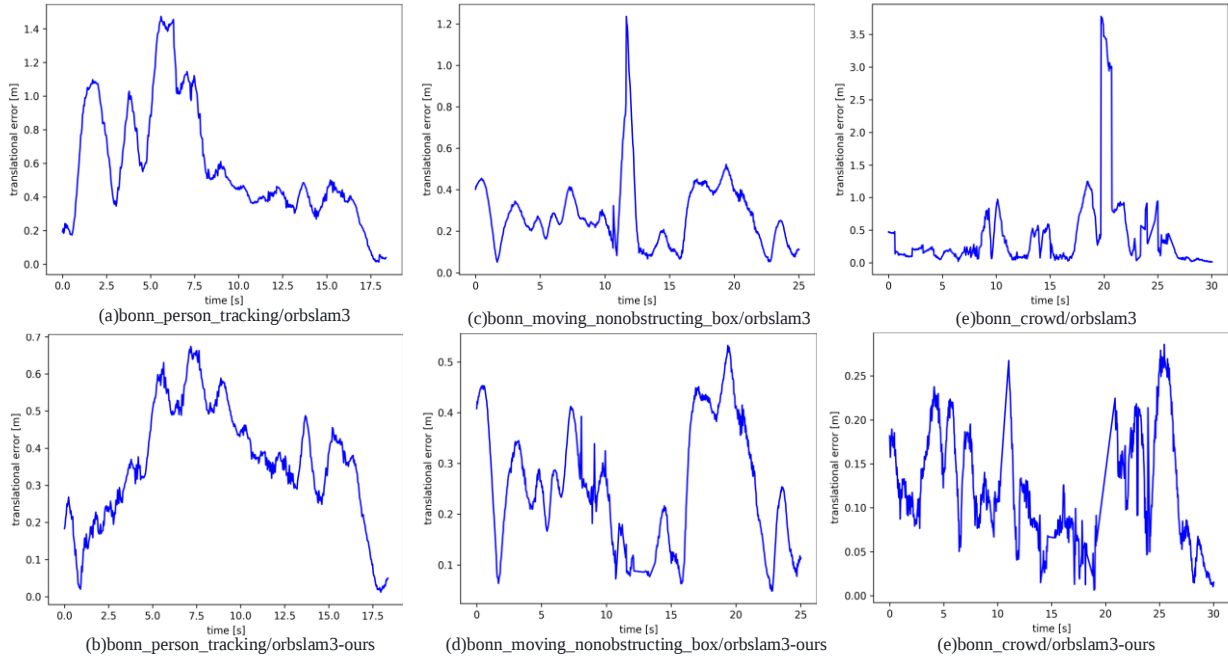


图 5-10 BONN 数据集多序列 RPE 结果

5.3 语义 VSLAM 地图构建实验

采用二维目标检测算法结合点云获取三维信息。图 5-11 给出了详细流程。

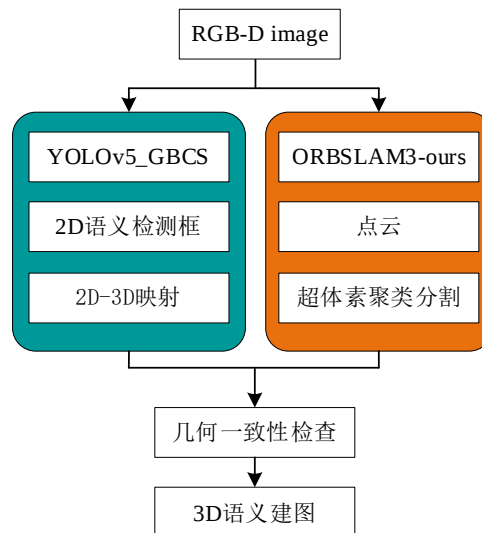


图 5-11 语义地图构建流程图

使用 YOLOv5_GBCS 提取每个关键帧中检测到的目标的类别信息以及对应的 2D 检测框。获取框内所有像素点的坐标。将 2D 像素坐标转换为相机坐标系下的归一化图像坐标。根据归一化图像坐标和深度信息，将这些点转换为相机坐标系下的三维点。根据上述映射，获得了对应于 2D 检测框的 3D 点云的一个子集。这个子集表征了检测到的对象在 3D 空间中的位置和形状。

图 5-12 展示了对构建的原始点云数据进行超体素聚类分割的具体流程。

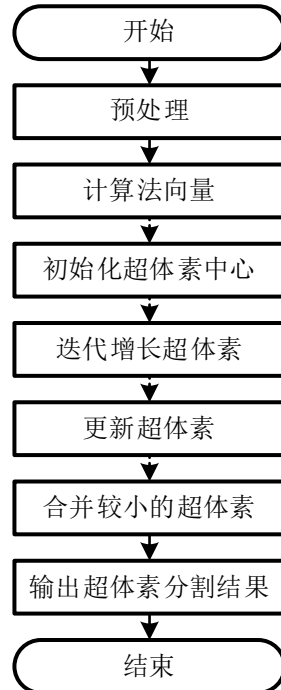


图 5-12 点云超体素分割

对构建的原始点云数据进行超体素聚类分割的具体步骤如下：

- (1) 预处理：从原始点云中移除噪声和异常点。
- (2) 计算法向量：对于每个点，计算其法向量，这是理解局部表面形状的关键信息。使用如 PCA（主成分分析）来估计法线。
- (3) 初始化超体素中心：在点云空间中均匀分布一组种子点，这些种子点是超体素中心的候选位置。
- (4) 迭代增长超体素：从每个种子点开始，基于相邻点的位置和法向量的相似性，迭代地将点加入到最近的超体素中心。这个步骤会反复进行，直至超体素的构成不再发生变化。
- (5) 更新超体素：根据已划分的超体素重新计算超体素中心。
- (6) 合并较小的超体素：由于初始化和增长步骤导致的分割结果可能存在

一些小的超体素，为了提升结果的稳健性和降低后续处理的复杂度，可以合并这些小的超体素到周围较大的超体素中。

(7) 输出超体素分割结果：最终点云数据被分割成许多超体素，为后续的几何一致性检查做准备。

将超体素分割得到的 3D 点云块与 2D 检测框的 3D 点云子集进行几何一致性检查，分析它们的几何一致性（如形状、大小和空间位置），判断每个超体素块是否可能属于检测框内部的对象。对通过检查的 3D 点云数据赋予其对应的语义信息，完成语义地图构建。

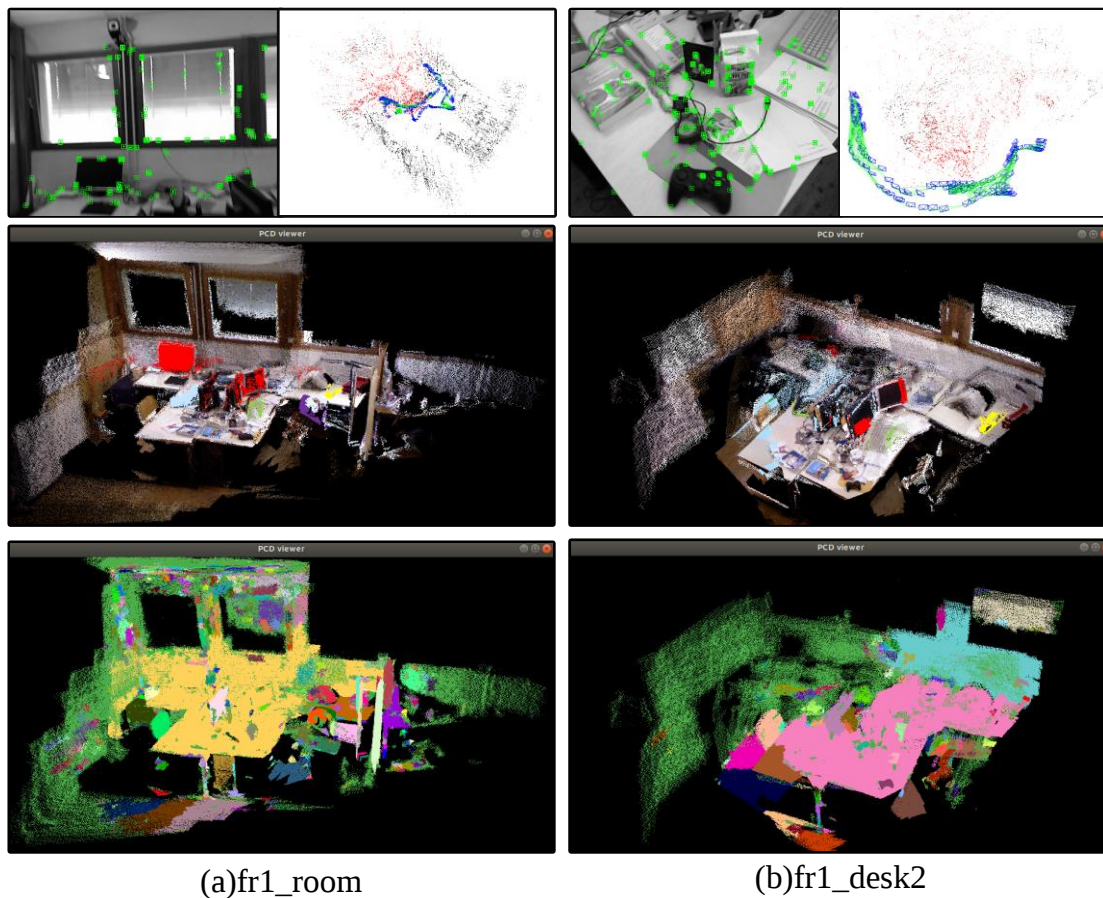
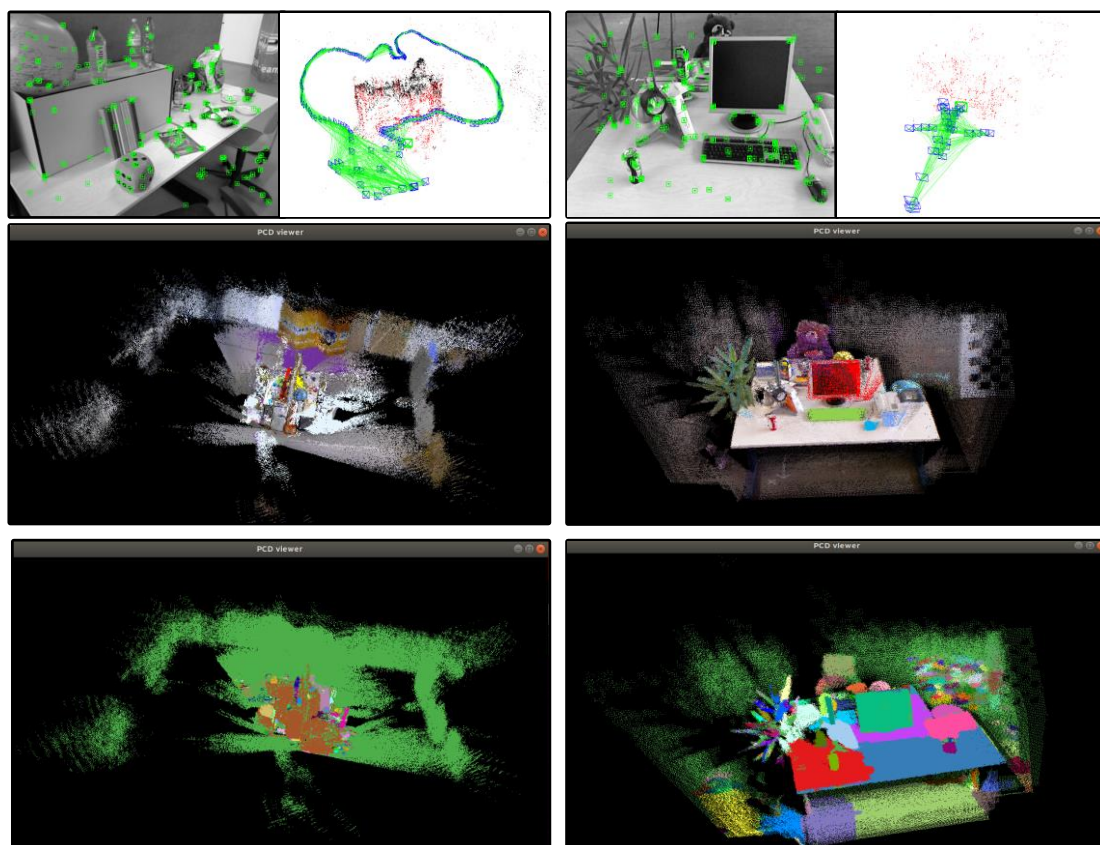
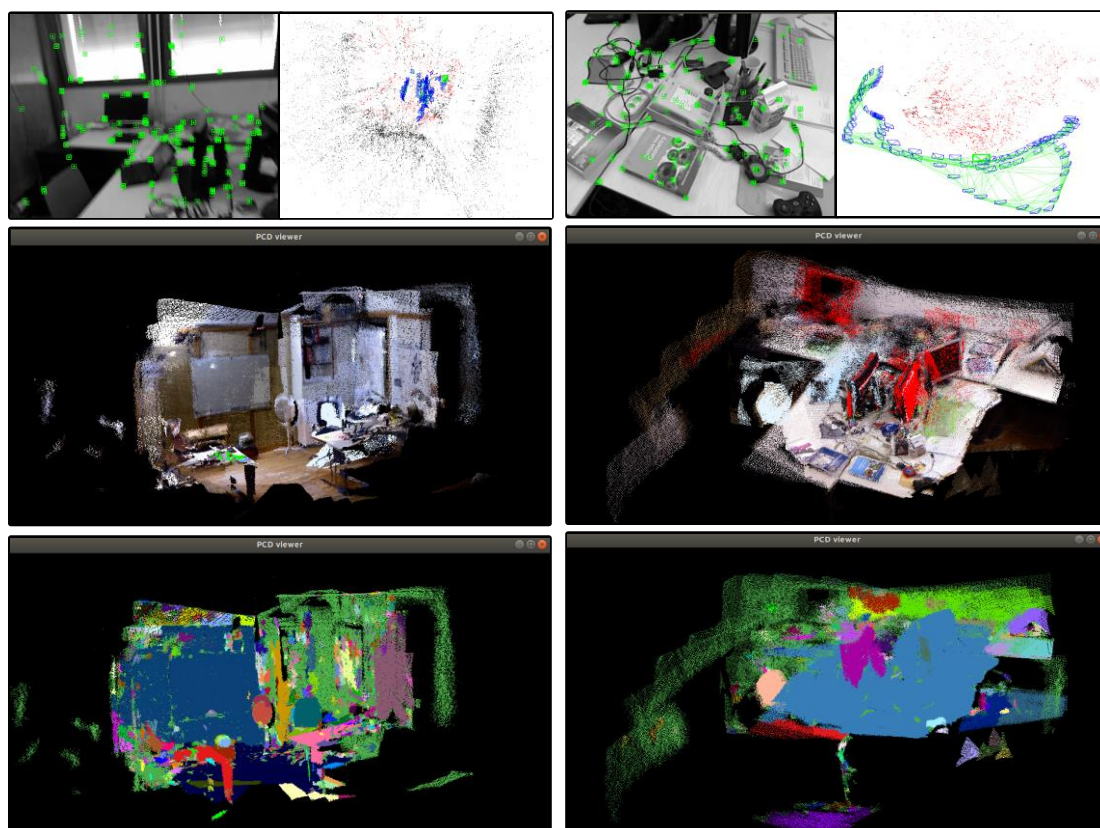


图 5-13 基于部分 TUM 数据集的语义建图实验



(a)fr3_long_office_household

(b)fr2_xyz



(c)fr1_360

(d)fr1_desk

图 5-14 基于部分 TUM 数据集的语义建图实验

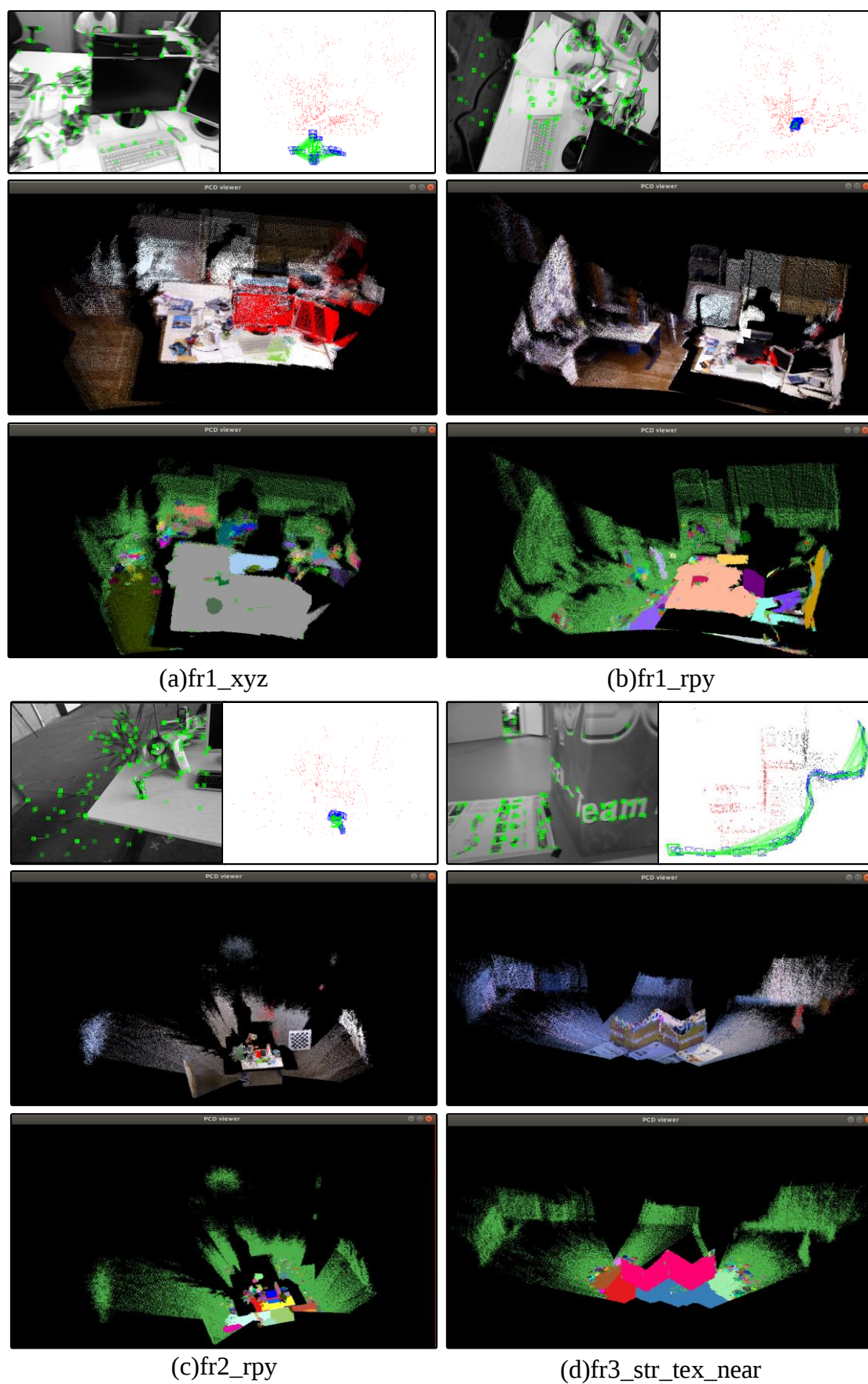


图 5-15 基于部分 TUM 数据集的语义建图实验

在 TUM 数据集场景中对语义建图进行了实验。实验结果如图 5-13、图 5-14、图 5-15 所示。每个子图的左上角为特征点提取效果图，右上角为建图过程，中间为未融合语义信息的建图效果，下方为融合语义信息的建图效果。在十种环境中进行实验，电脑桌子等物体的语义检测准确，且正确的完成了语义信息与点云的数据融合。

5.4 自主导航实验

5.4.1 八叉树地图转换

为了减少地图的存储容量，使地图能够在导航项目中使用，选择了 Octomap 进行地图存储。Octomap 是一种基于八叉树结构的三维地图创建框架，可以压缩点云，节省存储空间。三维空间根据八叉树结构连续切割，直到成为最小的正方形。在八叉树中，节点存储有关其是否被占用的信息。当块的所有子节点都被占用或未被占用时，不需要扩容节点。每个小正方形使用 0 到 1 之间的浮点数来表示它被占用的概率。在环境观测过程中，由于噪声或动态对象的影响，需要对节点进行概率更新。公式如下：

$$P(n|z_{1:T}) = \left(1 + \frac{1 - P(n|z_T)}{P(n|z_T)} \frac{1 - P(n|z_{1:T-1})}{P(n|z_{1:T-1})} \frac{P(n)}{1 + P(n)} \right)^{-1} \quad (5-1)$$

其中 n 代表叶子节点， z_T 代表在时间 T 的测量值， $P(n|z_{1:T})$ 代表叶子节点的占用概率。为了确保概率值在 0 到 1 之间，上述公式以概率的对数形式编写，并带入公式中：

$$y = \log \text{it}(x) = \log \left(\frac{x}{1-x} \right) \quad (5-2)$$

得到转换后的公式：

$$L(n|z_{1:t+1}) = L(n|z_{1:t-1}) + L(n|z_t) \quad (5-3)$$

5.4.2 室内导航实验

为了深入探究基于视觉 VSLAM 和改进路径规划算法的无人机自主导航系统在实际应用中的实用性与效率，本节设计了相关实验。这些实验旨在验证无人机自主导航系统在复杂环境中的性能和可靠性，特别是系统的三维环境感知

能力、环境建模精度以及路径规划的有效性。

实验选取具有多样化布局和结构的实验室环境作为测试场景，以测试系统在实际应用场景中的适用性。实验室内设置了多种常见室内元素，如桌椅、门、锥桶、电脑等物体等，如图 5-16 所示，为真实复杂环境。



图 5-16 实际场景

实验所用无人机平台配备高性能的 Jetson Xavier NX 处理器和 Intel RealSense D435i 深度相机，为无人机提供强大的计算能力和高质量的视觉信息。PX4 飞行控制模块被用于控制无人机的飞行状态。

实验开始时，操控无人机绕实验环境周围飞行一圈，同时利用无人机上的传感器记录环境的 ROS 数据包。采集到的数据包括高质量的图像数据和惯性测量单位数据，这些数据是构建高精度三维语义点云图的基础。从中获取语义信息，提取结果如图 5-17 所示，系统精准的识别了场景中的各种物体，例如桌子、桶锥、门、床和电脑等，细致地分辨出物体的类型和位置，为构建一个详细的语义地图提供了强有力的基础。

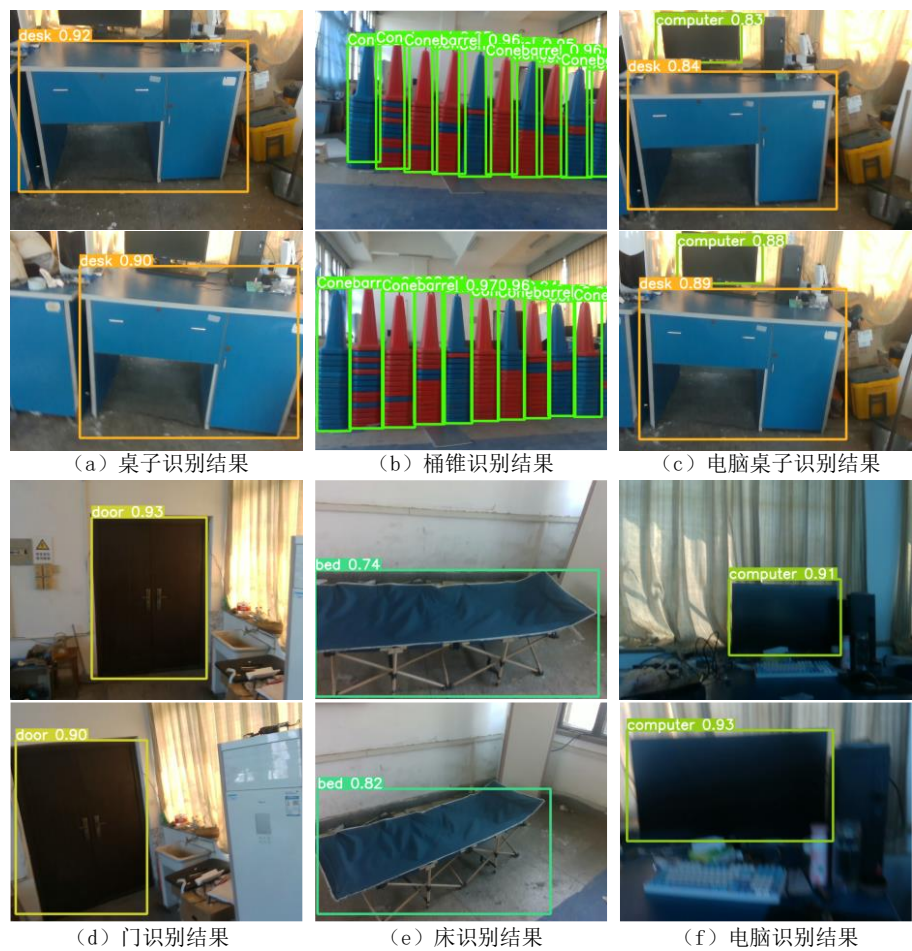


图 5-17 实际场景语义信息提取结果

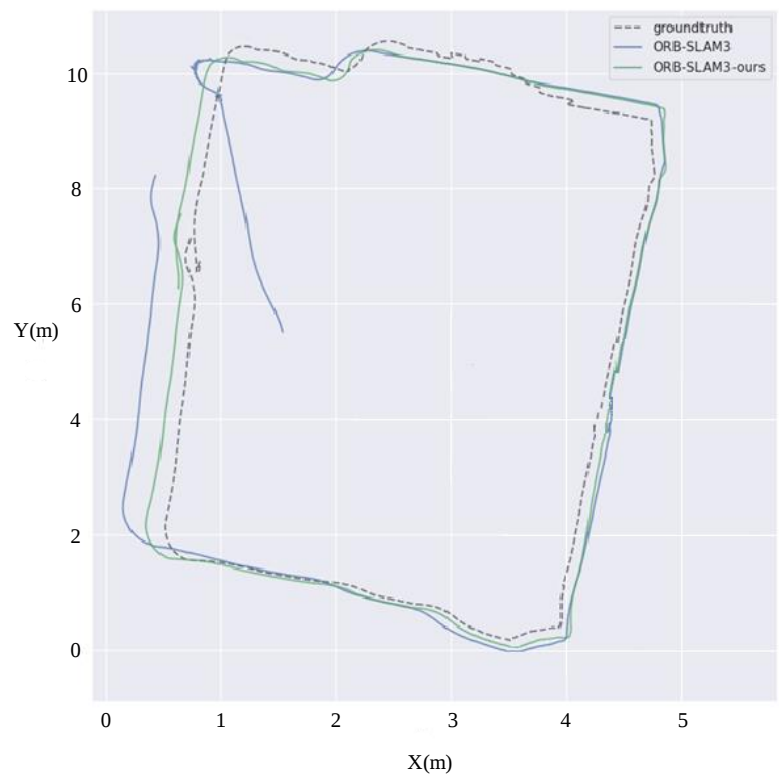


图 5-18 VSLAM 定位轨迹图

随后，通过导入 VSLAM 算法对这些数据进行处理，生成详尽的三维环境模型和对应的八叉树地图。这些地图不仅清晰地展现了环境的三维结构，还包含了环境中物体的语义信息，为后续的路径规划提供了精确的环境模型和必要的信息。建图完成后对 ORBSLAM3 和 ORBSLAM3-ours 的定位轨迹进行了评估，轨迹对比结果如图 5-18 所示。ORBSLAM3 的定位轨迹与真实轨迹存在明显的漂移现象，ORBSLAM3-ours 则更加贴合真实轨迹，很好的体现了改进后算法的稳定性与鲁棒性，定位精度大大提升。

在获得详细的三维语义点云图和八叉树地图后，启动自主导航系统，并将改进的 PSO 路径规划算法集成到系统中。通过为无人机设置明确的目标位置，启动路径规划算法，无人机便开始根据算法计算出的最优路径进行自主飞行。整个飞行过程中，系统能够实时更新环境信息和飞行轨迹，确保无人机能够有效地识别并避开新出现的障碍物，同时快速准确地达到预定的目标点。

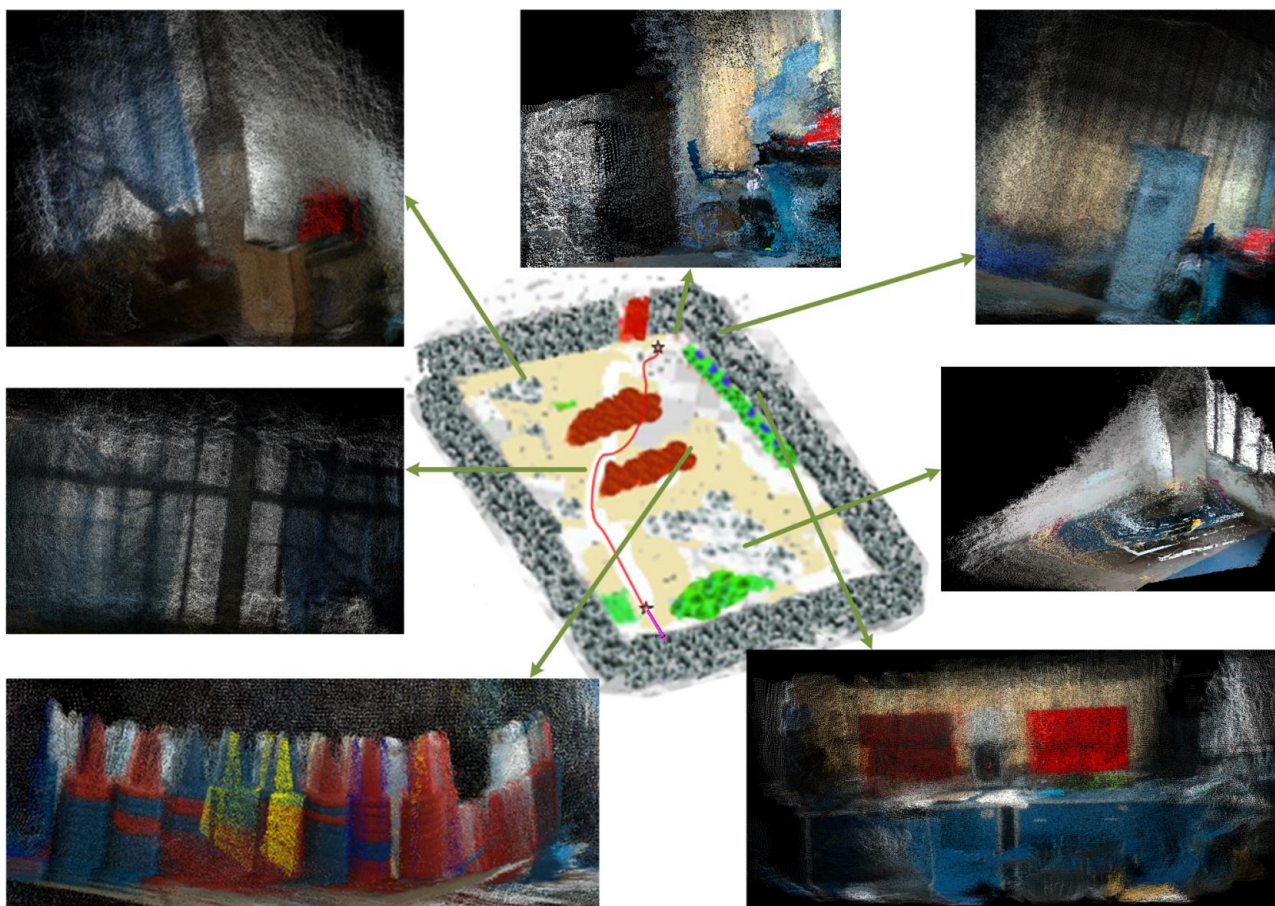


图 5-19 建图与路径规划效果

实验结果如图 5-19 所示，利用语义 VSLAM 系统构建的三维语义点云图和八叉树地图能够精确反映实验环境的三维结构和语义信息，为无人机的路径规

划提供了强有力的支持。



图 5-20 无人机路径规划飞行轨迹

无人机飞行路径如图 5-20 所示。在多次实验中，无人机能够成功地完成自主飞行任务，即使在面对复杂多变的环境时也能有效识别并避开障碍物，表现出极高的鲁棒性和可靠性。

5.5 本章小结

本章通过系列实验成功验证了基于语义 VSLAM 和改进 PSO 路径规划算法的无人机自主导航系统的实用性。实验结果表明，该系统能提供精确的环境建模，并能在复杂环境中实现高效稳定的路径规划和自主飞行，具有广泛的应用前景。未来的工作将集中于进一步提升系统的实时性能，扩展到更多的应用场景中，以及优化算法以应对更多类型的障碍物和更复杂的环境条件。

第 6 章 总结与展望

6.1 论文的主要工作

具有自动导航的四旋翼无人机因其高度的灵活性和机动性在提升效率、减少风险和执行能力方面展示出巨大潜力。但是传统无人机自主导航系统在复杂环境下的语义理解及高动态的室内环境下导航效率极其有限，甚至完全失效。针对这个问题，本文设计了一种基于语义 VSLAM 的无人机自主导航系统。全文主要工作如下：

(1) 为了克服动态室内环境中，运动物体对 VSLAM 系统的干扰，导致跟踪丢失并减低定位的精度，本文使用改进的 YOLOv5_GBCS 算法以提取场景的语义信息，从而实现可以剔除动态特征点的视觉里程计，并与 ORBSLAM3 相结合。通过 YOLOv5_GBCS 算法，可以有效地识别场景中可能移动的物体。在动态目标的预测边界框中使用 LK 光流算法与几何约束来消除连续帧之间的动态特征点。提升了 VSLAM 在动态环境中的准确度和鲁棒性。

(2) 使用 YOLOv5_GBCS 算法来识别关键帧内目标的类别，并提取相应的二维检测框信息。将框内 2D 像素坐标转换为相机坐标系下的归一化图像坐标。利用深度信息将这些归一化坐标转换为三维空间点。再对 ORBSLAM3 构建的原始点云数据进行超体素聚类分割。通过将分割得到的三维点云块与从二维检测框中得到的三维点云子集进行几何一致性检验，分析其几何一致性，判断每个超体素块是否属于检测框中的对象。对通过检查的 3D 点云数据赋予其对应的语义信息，完成语义地图构建。

(3) 在复杂环境下，传统路径规划方法可能会遇到局部最优问题导致飞行路径规划不理想甚至是规划失败。为此本文在系统中集成了经过改进的粒子群优化(PSO)算法，并引入轨迹平滑化处理，使规划出的飞行轨迹更符合飞行控制系统的要求。

(4) 为了验证本文设计的系统的 VSLAM 算法、路径规划算法和整个系统的可行性和稳定性，自主搭建了以 Pixhawk4、Intel RealSense D435i 和 Jetson Xavier NX 为组件的无人机硬件平台，并构建实际实验环境。通过软硬件控制平

台实现了基于语义地图的高效自主导航系统，进行了物体识别、语义地图构建、自主导航等实验，验证了本系统算法的准确性与有效性。

6.2 下一步研究计划

未来的研究工作在提升基于语义 VSLAM 的无人机自主导航系统性能并拓宽其应用领域方面具有广阔的前景。为了使系统能够更好地适应快速变化的环境，并满足更多复杂应用场景的需求，研究的重点将包括但不限于以下几个关键领域：

（1）提升实时性：实时性是无人机自主导航系统的核心要求之一。未来的工作将探索更先进的算法和计算方法，以减少系统处理大量传感器数据和执行复杂计算的时间。这对提升无人机在高动态、变化多端的城市环境中的自主飞行能力至关重要。

（2）扩展应用场景：随着技术的进步，无人机自主导航系统的实用性和通用性的提升将使其能够在更多复杂的环境中发挥作用。这包括在具有复杂地形的室外环境、在极端气候条件下、以及具有不同文化和地理特征的区域进行自主飞行。

（3）算法优化与调整：面对环境多样性和障碍物识别的挑战，系统中的算法需要不断的优化和调整，以提高其准确性、可靠性和适应性。对语义 VSLAM 技术的改进将集中在提高环境解析的细致度。而对路径规划算法的调整则将旨在提升算法的效率和适应多变环境的能力，如引入机器学习方法以实时地优化飞行路径决策。

随着未来研究的不断深入，基于语义 VSLAM 的无人机自主导航系统预计将在无人机技术领域发挥越来越重要的作用，带来更加安全、高效和智能的自主飞行解决方案。

参考文献

- [1] Ahirwar S, Swarnkar R, Bhukya S, et al. Application of drone in agriculture[J]. In ternational Journal of Current Microbiology and Applied Sciences, 2019, 8(01): 2500-2505.
- [2] Bijjahalli S, Sabatini R, Gardi A. Advances in intelligent and autonomous navigation systems for small UAS[J]. Progress in Aerospace Sciences, 2020, 115: 100617.
- [3] Merzlyakov A, Macenski S. A comparison of modern general-purpose visual SLAM approaches[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021: 9190-9197.
- [4] 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.
- [5] 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. IEEE, 2007: 225-234.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Sun Y, Liu M, Meng M Q H. Improving RGB-D SLAM in dynamic environments: A motion removal approach[J]. Robotics and Autonomous Systems, 2017, 89: 110-122.

-
- [10] Bescos B, Fácil J M, Civera J, et al. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.
 - [11] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
 - [12] Zhong F, Wang S, Zhang Z, et al. Detect-SLAM: Making object detection and SLAM mutually beneficial[C]//2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2018: 1001-1010.
 - [13] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]//2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2018: 1168-1174.
 - [14] Sheng C, Pan S, Gao W, et al. Dynamic-DSO: direct sparse odometry using objects semantic information for dynamic environments[J]. Applied Sciences, 2020, 10(4): 1467.
 - [15] Soares J C V, Gattass M, Meggiolaro M A. Visual SLAM in human populated environments: exploring the trade-off between accuracy and speed of YOLO and Mask R-CNN[C]//2019 19th International Conference on Advanced Robotics (ICAR). IEEE, 2019: 135-140.
 - [16] Li S, Lee D. RGB-D SLAM in dynamic environments using static point weighting[J]. IEEE Robotics and Automation Letters, 2017, 2(4): 2263-2270.
 - [17] Barber C B, Dobkin D P, Huhdanpaa H. The quickhull algorithm for convex hulls[J]. ACM Transactions on Mathematical Software (TOMS), 1996, 22(4): 469-483.
 - [18] Yang S, Fan G, Bai L, et al. MGC-VSLAM: A meshing-based and geometric constraint VSLAM for dynamic indoor environments[J]. IEEE Access, 2020, 8: 81007-81021.
 - [19] 曹家乐,李亚利,孙汉卿等.基于深度学习的视觉目标检测技术综述[J].中国图象图形学报,2022,27(06):1697-1722.
 - [20] Liu C, Wu Y, Liu J, et al. MTI-YOLO: a light-weight and real-time deep neural network for insulator detection in complex aerial images[J]. Energies, 2021, 14(5): 1426.

-
- [21] Sahin O, Ozer S. Yolodrone: Improved yolo architecture for object detection in drone images[C]//2021 44th International Conference on Telecommunications and Signal Processing (TSP). IEEE, 2021: 361-365.
 - [22] Wang X, Zhao Q, Jiang P, et al. LDS-YOLO: A lightweight small object detection method for dead trees from shelter forest[J]. Computers and Electronics in Agriculture, 2022, 198: 107035.
 - [23] 彭继慎,孙礼鑫,王凯,宋立业.基于模型压缩的ED-YOLO 电力巡检无人机避障目标检测算法[J].仪器仪表学报,2021,42(10):161-170.DOI:10.19650/j.cnki.cjsi.J2108173.
 - [24] 陈蕊,郑华飞,蒋鸿宇等.结合仿真迁移学习和自适应融合的无人机小目标检测[J].小型微型计算机系统,2023,44(08):1743-1749.DOI:10.20009/j.cnki.21-1106/TP.2021-0905.
 - [25] Liu H, Wei X, Li J, et al. A star identification algorithm based on simplest general subgraph[J]. Acta Astronautica, 2021, 183: 11-22.
 - [26] Dijkstra E W. A note on two problems in connexion with graphs[M]//Edsger Wybe Dijkstra: His Life, Work, and Legacy. 2022: 287-290.
 - [27] Wang J, Li B, Meng M Q H. Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning[J]. Expert Systems with Applications, 2021, 170: 114541.
 - [28] Sonmez A, Kocyigit E, Kugu E. Optimal path planning for UAVs using genetic algorithm[C]//2015 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2015: 50-55.
 - [29] Li Z, Han R. Unmanned aerial vehicle three-dimensional trajectory planning based on ant colony algorithm[C]//2018 37th Chinese Control Conference (CCC). IEEE, 2018: 9992-9995.
 - [30] Wang Q, Zhang A, Qi L. Three-dimensional path planning for UAV based on improved PSO algorithm[C]//The 26th Chinese Control and Decision Conference (2014 CCDC). IEEE, 2014: 3981-3985.
 - [31] Sun J, Xu W, Feng B. A global search strategy of quantum-behaved particle swarm optimization[C]//IEEE Conference on Cybernetics and Intelligent Systems, 200

4. IEEE, 2004, 1: 111-116.
- [32] Liu Y, Zhang X, Guan X, et al. Potential odor intensity grid based UAV path planning algorithm with particle swarm optimization approach[J]. Mathematical Problems in Engineering, 2016,2016: 780-798.
- [33] Duan H, Zhang X, Wu J, et al. Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncertain environments[J]. Journal of Bionic Engineering, 2009, 6(2): 161-173.
- [34] Qu C, Gai W, Zhong M, et al. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning[J]. Applied soft computing, 2020, 89: 106099.
- [35] Liu X, Du X, Zhang X, et al. Evolution-algorithm-based unmanned aerial vehicle path planning in complex environment[J]. Computers & Electrical Engineering, 2019, 80: 106493.
- [36] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. Systems science & control engineering, 2020, 8(1): 22-34.
- [37] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proceedings of ICNN'95-international conference on neural networks. IEEE, 1995, 4: 1942-1948.
- [38] Li G, Chou W. Path planning for mobile robot using self-adaptive learning particle swarm optimization[J]. Science China Information Sciences, 2018, 61: 1-18.
- [39] Phung M D, Quach C H, Dinh T H, et al. Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection[J]. Automation in Construction, 2017, 81: 25-33.
- [40] Foo J L, Knutzon J, Kalivarapu V, et al. Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization[J]. Journal of aerospace computing, Information, and communication, 2009, 6(4): 271-290.
- [41] Taheri H, Xia Z C. SLAM; definition and evolution[J]. Engineering Applications of Artificial Intelligence, 2021, 97: 104032.
- [42] Nistér D, Naroditsky O, Bergen J. Visual odometry[C]//Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. Ieee, 2004, 1: I-I.

-
- [43] Harris C, Stephens M. A combined corner and edge detector[C]//Alvey vision conference. 1988, 15(50): 10-5244.
 - [44] 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 Berlin Heidelberg, 2006: 404-417.
 - [45] David L. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60: 91-110.
 - [46] 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 Berlin Heidelberg, 2006: 430-443.
 - [47] 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.
 - [48] Jakubović A, Velagić J. Image feature matching and object detection using brute-force matchers[C]//2018 International Symposium ELMAR. IEEE, 2018: 83-86.
 - [49] Muja M, Lowe D G. Fast approximate nearest neighbors with automatic algorithm configuration[J]. VISAPP (1), 2009, 2(331-340): 2.
 - [50] Koide K, Menegatti E. General hand–eye calibration based on reprojection error minimization[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 1021-1028.
 - [51] Kümmerle R, Grisetti G, Strasdat H, et al. g 2 o: A general framework for graph optimization[C]//2011 IEEE International Conference on Robotics and Automation. IEEE, 2011: 3607-3613.
 - [52] Zach C. Robust bundle adjustment revisited[C]//Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing, 2014: 772-787.
 - [53] Wen W, Hsu L T. Towards robust GNSS positioning and real-time kinematic using factor graph optimization[C]//2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021: 5884-5890.

-
- [54] Wang J, Yang F. A Review of Vision SLAM-based Closed-loop Inspection[C]//2023 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2023: 507-512.
 - [55] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
 - [56] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.
 - [57] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
 - [58] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
 - [59] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arxiv preprint arxiv:2004.10934, 2020.
 - [60] Hosang J, Benenson R, Schiele B. A convnet for non-maximum suppression[C]//Pattern Recognition: 38th German Conference, GCPR 2016, Hannover, Germany, September 12-15, 2016, Proceedings 38. Springer International Publishing, 2016: 192-204.
 - [61] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
 - [62] Duan K, Bai S, Xie L, et al. Centernet: Keypoint triplets for object detection[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6569-6578.
 - [63] Wang H, Liu B, Liu Z, et al. LSS target threat estimation and capture prediction trajectory simulation[C]//2020 7th International Conference on Information Science and Control Engineering (ICISCE). IEEE, 2020: 806-813.
 - [64] Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6154-6162.

-
- [65] Tian Z, Shen C, Chen H, et al. Fcos: Fully convolutional one-stage object detection[C]//roceedings of the IEEE/CVF international conference on computer vision. 2019: 9627-9636.
 - [66] Li Y, Chen Y, Wang N, et al. Scale-aware trident networks for object detection[C]//roceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6054-6063.
 - [67] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
 - [68] Zhu C, He Y, Savvides M. Feature selective anchor-free module for single-shot object detection[C]//roceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 840-849.
 - [69] Zhang S, Chi C, Yao Y, et al. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9759-9768.
 - [70] Feng C, Zhong Y, Gao Y, et al. Tood: Task-aligned one-stage object detection[C]//2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE Computer Society, 2021: 3490-3499.
 - [71] Ahmed A, Tangri P, Panda A, et al. Vfnet: A convolutional architecture for accent classification[C]//2019 IEEE 16th India Council International Conference (INDICON). IEEE, 2019: 1-4.
 - [72] Timofte R, Van Gool L. Sparse flow: Sparse matching for small to large displacement optical flow[C]//2015 IEEE Winter Conference on Applications of Computer Vision. IEEE, 2015: 1100-1106.
 - [73] Yedjour H. Optical flow based on Lucas-Kanade method for motion estimation[C]//Artificial Intelligence and Renewables Towards an Energy Transition 4. Springer International Publishing, 2021: 937-945.
 - [74] Van Den Bergh F. An analysis of particle swarm optimizers[M]. University of Pretoria (South Africa), 2001.
 - [75] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in

a multidimensional complex space[J]. IEEE transactions on Evolutionary Computation, 2002, 6(1): 58-73.

- [76] Omkar S N, Khandelwal R, Ananth T V S, et al. Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures[J]. Expert Systems with Applications, 2009, 36(8): 11312-11322.

攻读硕士学位期间主要研究成果

(1) 发表论文

- [1] 何宇豪,易明发,周先存,王冠凌.基于改进的 YOLOv5 的无人机图像小目标检测[J].智能系统学报,(已录用,第一作者)。
- [2] 兰天,欧阳嘉泰,何宇豪,易方旭,易明发,王冠凌.改进 YOLOv5 框架的交通标志检测算法[J].传感技术学报,(已录用,第三作者)。

(2) 获奖情况

- [1] 2023 年第十八届全国大学生智能汽车竞赛创意组航天智慧物流全国总决赛一等奖。
- [2] 2023 年第十八届全国大学生智能汽车竞赛创意组航天智慧物流线下选拔赛一等奖。
- [3] 2023 年第十八届全国大学生智能汽车竞赛创意组讯飞智慧农业全国总决赛二等奖。
- [4] 2023 年第十八届全国大学生智能汽车竞赛创意组讯飞智慧农业安徽赛区选拔赛一等奖。
- [5] 2023 年第十八届全国大学生智能汽车竞赛室外专项赛 ROS 组全国总决赛一等奖。
- [6] 2023 年第十七届“西门子杯”中国智能制造挑战赛离散行业运动控制全国初赛二等奖。
- [7] 2022 年第十七届全国大学生智能汽车竞赛室外 ROS 无人车竞速赛(高教组)三等奖。
- [8] 2022 年第十七届全国大学生智能汽车竞赛华为 AI 组全国初赛二等奖。
- [9] 2023 年第十八届全国大学生智能汽车竞赛 ROS 室外赛全国总决赛二等奖。

致 谢

时光匆匆，三年研究生时光即将落幕。在这段充实而又短暂的旅程中，有许多人在不同的阶段给予了我帮助和支持，他们是我学术道路上的重要引路人。在此，我想对他们表达我最真挚的感谢。

首先，我要感谢我的导师，王冠凌教授，感谢您的细心指导和悉心教诲。在论文的撰写过程中，您的专业知识和丰富经验为我提供了宝贵的指导，每次讨论都让我受益匪浅，让我对研究方向有了更深刻的理解和把握。

其次，我要感谢我的同学们。在论文的撰写过程中，我们相互支持，共同探讨，共同进步。你们的陪伴和帮助让我的学术之旅充满了乐趣和动力。

此外，我还要感谢我的家人。你们的支持和鼓励是我前进的动力源泉。在我迷茫和困惑的时候，你们总是给予我最坚定的支持和最温暖的怀抱。

最后，我要感谢参与我论文评审和答辩的各位老师。您的宝贵意见和建议对我论文质量的提高有着极大的帮助。您的严谨和认真让我更加明白学术研究的重要性和责任感。

尚德敏学 唯实惟新

