

基于双目视觉的无人机自主感知与导航系统设计与实现

Design and Implementation of Autonomous Perception and Navigation System with Micro Aerial Vehicles Based on Stereo Vision

工程领域: 电子与通信工程
作者姓名: 吕华龙
指导教师: 侯永宏
企业导师: 曹玉良

天津大学电气自动化与信息工程学院
二零一九年十一月

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果,除了文中特别加以标注和致谢之处外,论文中不包含其他人已经发表或撰写过的研究成果,也不包含为获得天津大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名: 吕华龙 签字日期: 2019 年 11 月 29 日

学位论文版权使用授权书

本学位论文作者完全了解天津大学有关保留、使用学位论文的规定。特授权天津大学可以将学位论文的全部或部分内容编入有关数据库进行检索,并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

(保密的学位论文在解密后适用本授权说明)

学位论文作者签名: 吕华龙 导师签名: 侯永宏

签字日期: 2019 年 11 月 29 日 签字日期: 2019 年 11 月 29 日

摘要

多旋翼微型无人机因其灵活、便携、可垂直起降和高速巡航等特点，被广泛应用在航拍测绘、农业植保等多个领域。随着行业无人机技术的发展，无人机应用领域不断扩大，许多新兴的应用，如智慧物流、搜索救援、电力巡检等，对无人机的自主性提出了更高的要求，需要无人机在执行任务的过程中能够自主地处理各种复杂情况，例如无人机在 GPS 失效的情况下仍能精确定位、在飞行过程中自主躲避障碍物、动态规划航线、自主探测到合适的位置降落等。因此，使无人机具备自主感知与导航的能力是扩展其应用领域的关键。

本文设计并实现了一套完整的无人机自主感知与自主导航系统。主要包含基于双目立体视觉的环境感知算法、基于全局路径搜索和局部轨迹优化相结合的运动规划算法。在感知算法部分，利用双目视觉与惯导相融合的 SLAM 算法实现无人机状态估计并构建视觉障碍物环境地图；在运动规划部分，根据无人机的状态和障碍物地图信息，先利用混合状态 A* 算法搜索出到达目标点的可行路径，再采用 B 样条轨迹优化算法对搜索路径进一步局部调优，求解出满足无人机动力约束的安全平滑轨迹，最后通过轨迹服务器引导无人机贴合规划的航迹飞行。同时，基于 Gazebo 仿真器和 AirSim 仿真器，搭建了完善的无人机自主导航软件仿真平台用于算法测试和系统分析。

针对于无人机的多场景感知问题，本文提出了一种基于双目立体视觉与 IMU 和 GPS 相结合的多传感器融合无人机状态估计算法，采用紧耦合与松耦合相结合的方式，将双目视觉里程计与 IMU 和 GPS 信息融合，从而解决无人机多场景和跨场景的状态估计问题。

本文在多个公开数据集、仿真环境和真实场景中均设计了丰富的实验，详细分析各个功能算法的实时性与准确度以及整个无人机自主感知与导航系统的可行性。最后将本文所提出的无人机自主感知与导航系统部署在搭载了 NVIDIA TX2 机载计算机、MYNT-EYE 立体相机、Pixhawk 飞控的四旋翼飞行器上，在真实应用场景下验证了无人机系统自主避障飞行的有效性。

关键词：多旋翼微型无人机；自主感知与导航；即时定位与建图；运动规划；多传感器融合；无人机仿真

ABSTRACT

Micro aerial vehicles (MAVs) are widely used in aerial photography and mapping, agricultural plant protection and other fields due to their flexibility, portability, and the capabilities of vertical take-off and landing, high-speed cruising. With the development of industrial MAVs technology, the application field of MAVs expands continuously, such as smart express, search and rescue, electric inspection, etc., which put forward higher requirements for the autonomy of MAV in handling various complex situations autonomously during the execution of the mission. For example, in the case of GPS denied, the MAV can still accurately locate, autonomously avoid obstacles during flight, dynamically motion planning, and independently detect appropriate location for landing. Therefore, it is critical for MAV expended its application field that enable MAV with the capability of autonomic perception and navigation.

In this paper, a complete MAV autonomous perception and navigation solution is proposed and implemented. It includes environment perception algorithm based on stereo vision, motion planning algorithm integrating global path search and local trajectory optimization. In the perception part, the SLAM algorithm based on stereo vision and inertial sensing is used to realize accurate MAV state estimation and construct visual obstacle environment map. In the motion planning part, according to the state of the MAV and the obstacle map information, the hybrid state A star algorithm is used to search for the feasible path to the destination, and then the B-spline trajectory optimization algorithm is used to further optimize the path. A safe and smooth trajectory that meeting the dynamic constraints of the MAV is obtained. Finally, the trajectory server guides the MAV to fly along the re-planned trajectory. In order to facilitate the application deployment of the autonomous MAV system, the general Robot Operating System is adopted as the software architecture. In addition, Gazebo simulator and AirSim simulator based a complete MAV autonomous navigation software simulation platform is built for algorithm testing and system analysis.

A multi-sensor fusion state estimation algorithm based on stereo vision combined with IMU and GPS is proposed to solved the multi-scene sensing problem of MAV. The stereo vision odometry is integrated with IMU and GPS information by means of tight coupling and loose coupling. Therefore, the problem of state estimation of multi-scene

and cross-scene of the MAV is solved.

In this paper, various experiments are designed in public datasets, simulation environments and real-world scenes to analyze the real-time and accuracy of each functional algorithm and the feasibility of the whole MAV autonomous perception and navigation system. Finally, the proposed autonomous perception and navigation system is deployed on a four-rotors aircraft equipped with NVIDIA TX2 onboard computer, MYNT-EYE stereo camera and Pixhawk flight controller. The MAV system is validated by performing obstacle avoidance flights in real world environments.

KEY WORDS: Micro Aerial Vehicles; Autonomous perception and navigation; Motion Planning; Multi-sensor fusion; UAV Simulation

目 录

第 1 章 绪论.....	1
1.1. 研究背景及意义.....	1
1.2. 自主无人机技术研究现状.....	2
1.2.1. 无人机感知技术的研究现状.....	2
1.2.2. 无人机导航避障技术的研究现状.....	5
1.3. 主要研究内容与论文安排.....	6
第 2 章 自主无人机系统的总体方案设计.....	9
2.1. 无人机自主感知与导航系统总体方案.....	9
2.2. 软件架构的设计.....	10
2.3. 硬件平台的搭建.....	11
2.4. 仿真平台的设计.....	13
2.4.1. 基于 Gazebo 的无人机仿真平台设计.....	14
2.4.2. 基于 AirSim 的无人机仿真平台设计.....	17
2.5. 本章小结.....	19
第 3 章 无人机自主感知算法的设计.....	21
3.1. 引言.....	21
3.2. 多传感器融合的无人机状态估计.....	21
3.2.1. 双目视觉里程计.....	22
3.2.2. 多传感器融合.....	35
3.3. 障碍物地图的构建.....	39
3.3.1. 双目立体匹配.....	39
3.3.2. 全局地图拼接.....	41
3.3.3. ESDF 地图的转化.....	43
3.4. 本章小结.....	44
第 4 章 无人机自主导航避障算法的设计.....	45
4.1. 引言.....	45
4.2. 前端路径搜索.....	46
4.3. 后端轨迹优化.....	49
4.4. 本章小结.....	52
第 5 章 无人机系统实验分析.....	53
5.1. 无人机感知算法的实验分析.....	53

5.1.1. 视觉里程计与多传感器融合实验分析	53
5.1.2. 地图构建实验分析	58
5.2. 路径规划算法的实验分析	60
5.3. 无人机自主导航避障测试	62
5.4. 本章小结	63
第 6 章 总结与展望	65
6.1. 本文的工作总结	65
6.2. 下一步的工作方向	66
参考文献	67
发表论文和参加科研情况说明	73
致谢	75

第1章 绪论

1.1. 研究背景及意义

过去的几十年来,随着微型计算机和传感器技术的高速发展与计算机视觉和控制理论的快速迭代,无人机(Unmanned Aerial Vehicles UAVs)理论和技术也日益成熟。相应的无人机应用也逐渐渗透到人民的日常生活以及国民经济的各个领域,消费级无人机和工业无人机将是继移动智能终端之后改变人类生活生产方式的又一推动技术之一。因此对于无人机技术的研究也成为了学术界和工业界十分热门的课题。根据空气动力学原理,无人机可分为固定翼无人机和多旋翼无人机两种类型。多旋翼微型无人机因其便携、灵活稳定、可垂直起降、可高速巡航等特点,被广泛应用在诸多领域。如图 1-1 所示,目前多旋翼无人机在农业上可用于植保灌溉、在电力巡检中可用于电网的监护与检修、在抗灾抢险中可用于搜索救援与紧急物质运送、在城市复杂环境和偏远地区可用于物流快递的及时派送。

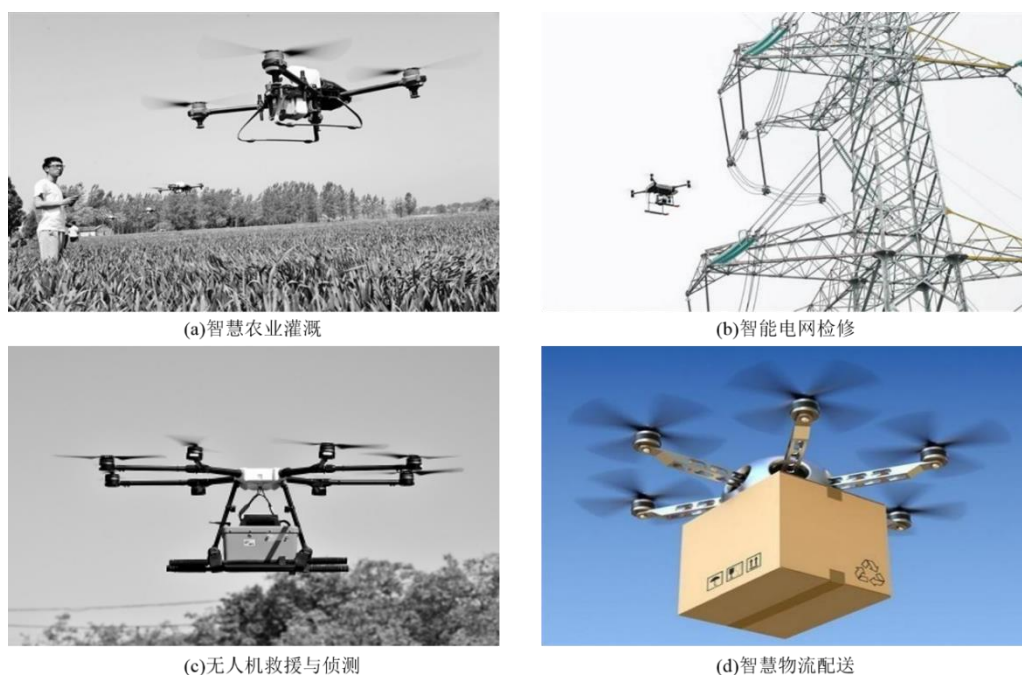


图 1-1 多旋翼微型无人机的多个应用场景

早期的无人机系统通常都配置专用的遥控器设备并配备专业的飞行操控人员,这样不仅导致无人机的活动空间和应用范围受到限制,而且成为制约无人机

普及和大规模应用的瓶颈。随着行业无人机技术的发展,应用领域不断扩大,尤其针对无人机智能物流、抗灾抢险的搜索救援、高速公路巡检、电力巡检等多个新兴应用,这些应用需要无人机具备更高自主性,要求无人机在飞行过程中能够自主地处理复杂情况,例如能够自主躲避飞行过程中的障碍物、规划航线、感知到适合降落位置和适合充电的站点等。因此在实际的无人机应用中,解决无人机的自主感知与自主导航是两个十分重要的关键技术难题。

本文针对无人机行业发展中亟待解决的问题,进行深入的研究分析并提出了相应的解决方案。针对于无人机自主感知问题,本文提出了一种基于双目立体视觉与 IMU 和 GPS 数据相结合的多传感器融合无人机状态估计算法,以满足无人机室内、室外以及跨区的定位需求,并添加了稠密地图构建单元,用于障碍物环境地图的实时重建;针对于无人机导航避障问题,本文采用了基于轨迹优化方式的无人机导航避障算法,来提高无人机路径规划的效率。最后本文构建了一套完整的无人机自主感知与导航避障系统方案,可应用部署于实际的无人机系统中。

1.2. 自主无人机技术研究现状

目前学术和工业界对于无人机的研究主要聚集在无人机控制系统、无人机智能感知导航系统两个主要方向。基于这两个方面的相互融合,衍生出多种形式的无人机应用系统。对无人机控制系统的研究主要为了实现更加鲁棒的无人机状态估计、更精准的位姿控制,以及多无人机集群协同控制。在研究中一方面优化无人机系统的控制器,例如基于经典线性控制器的方法、基于模型的非线性控制方法、基于学习的控制方法等,另一方面提升多无人机飞行器的协同控制能力,例如集中式、分散式和分布式无人机编队控制结构^[1]。本文的主要研究对象为单架次无人机自主导航系统,对于无人机自主感知导航系统的研究主要为了提升无人机对于自身状态和周围环境的感知能力,并根据定位和地图信息合理地规划出到达目标点的路径。主要包含无人机感知、无人机导航避障两个方面的研究内容。

1.2.1. 无人机感知技术的研究现状

无人机的感知系统主要依赖机载的摄像头、惯性测量单元(Inertial Measurement Unit IMU)、激光雷达、GNSS(Global Navigation Satellite System)定位模组等传感器,估计无人机的位姿运动、并且构建无人机周围的环境地图,从而实现无人机对自身姿态的估计和对于周围环境的感知,这个过程被称为即时定位与地图构建(Simultaneous Localization And Mapping SLAM)。

回顾 SLAM 系统中状态估计算法的发展历程,早期的无人机位姿估计算法主要利用 MEMS IMU 传感器和 GPS 模组的测量信息,采用扩展卡尔曼滤波器(Extended Kalman Filter EKF)的方式实现的^[2],使用 IMU 测量数据进行状态预测,并利用 GPS 观测数据进行状态更新。而这种无人机系统过度依赖 GPS 传感器的测量,只能在室外开阔场景中适用。Davison 等人提出 MonoSLAM 方法^[3],基于 EKF 的方法实现了单目摄像头的视觉 SLAM 系统,利用连续的图像序列能够实时估计相机的运动并构建稀疏地图。Klein 等人为了进一步提升系统的实时性,提出了基于关键帧技术的单目视觉 SLAM 系统^[4]。哈尔滨工业大学的厉茂海、洪炳熔等人进一步将 Rao-Blackwellized Particle Filter 粒子滤波的方法应用于单目视觉 SLAM 系统^[5],提升单目相机状态估计的鲁棒性。由于视觉 SLAM 随着系统的持续运行,会累积产生大量的特征点和相机位姿,导致计算量线性增长。所以早期的研究者一直采用基于滤波的方式设计视觉 SLAM 系统。直到近些年来,研究者发现了视觉 SLAM 问题中 BA(Bundle Adjustment)的稀疏性^[6],并且伴随着计算单元性能的大幅提升,使得基于图优化框架的视觉 SLAM 系统得到了快速发展并逐渐成为解决 SLAM 问题的主流方法。PTAM(Parallel Tracking And Mapping)^[7]是最早提出将特征提取追踪、位姿优化分成两个线程的 SLAM 算法,它是一种基于关键帧的单目 SLAM 算法,前端进行特征提取与追踪,后端进行局部和全局的非线性优化。ORB-SLAM^[8]在此基础上,将 SLAM 过程分为定位、建图和回环检测三个线程,基于这些理论逐步完善了 SLAM 系统的基本框架。

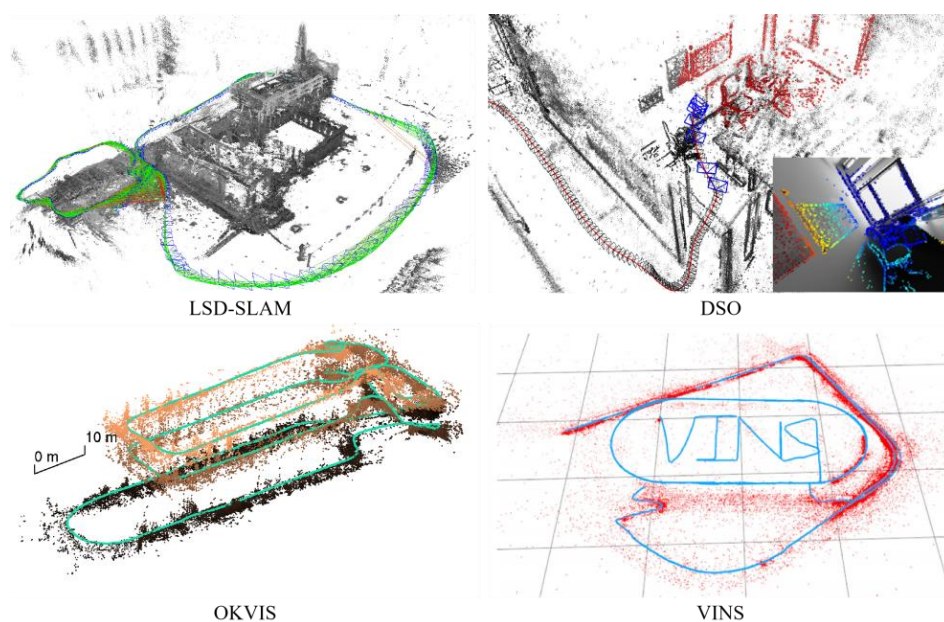


图 1-2 几种经典的 SLAM 系统运行效果图^[9,10,11,15]

上述介绍的方法主要利用关联图像帧的特征点重投影误差来构建非线性优化方程求解 SLAM 问题,除此之外,还有利用直接法,即通过连续帧图像亮度的变化构建光度约束求解相机的位姿运动,例如苏黎世理工 Scaramuzza 等人提出的 SVO^[9],慕尼黑工业大学计算机视觉组(TUM Computer Vision Group)提出的 LSD-SLAM^[10]、DSO^[11]。单目视觉 SLAM 系统虽然结构简单,但是由于单目视觉系统存在尺度不确定性和尺度漂移,无法估计出相机运动的真实尺度,因此研究者们进一步提出了基于双目的 SLAM 系统、基于视觉与惯导融合的 SLAM 系统(Visual Inertial SLAM)以及基于 RGB-D 相机的 SLAM 系统,例如 ORB-SLAM2^[12]、Stereo DSO^[13]、MSCKF^[14]、OKVIS^[15]、ROVIO^[16]、VINS-Mono^[17]等。图 1-2 为几种经典的 SLAM 系统运行效果图。

在无人机自主感知与导航系统中,障碍物地图的主要目的是在无人机飞行的过程中能够实时检测到当前飞行器位置附近的障碍物信息,当指定飞行目标点后,无人机能够基于已有的障碍物地图利用路径规划算法计算出到达目标点的可通行路径。在基于直接法和半直接法的 SLAM 系统中,后端能够直接构建相对稠密的环境地图,而使用基于特征点法 SLAM 系统,仅仅恢复了图像追踪到的特征点的深度,构建的稀疏地图仅仅能够用于 SLAM 系统的重定位与回环检测,无法直接用于导航避障,因此需要在基于特征点法的 SLAM 系统添加一个稠密建图单元。通常稠密建图使用双目摄像头,并利用双目视觉立体匹配算法恢复当前帧图像的深度,或者使用 RGB-D 相机直接获取图像的深度,然后利用每一帧相机所估计的位姿可以构建出稠密的环境地图。也有研究者直接利用单目相机连续帧之间的时间和空间信息进行稠密重建^[18]。对于环境地图的表示形式也有不同的方法,最基本的表示方式可以直接基于构建的障碍物点云地图,然后使用点云地图生成一个无人机可通行空闲区域的飞行走廊(Flight Corridor),最后通过在飞行走廊内进行路径搜索和轨迹优化^[19]。通常点云地图不利于后序路径规划算法的设计,需要根据深度估计的准确性和地图存储的精细度对地图在三维空间中进行数据结构和存储形式的映射。代表性的映射框架主要有 OctoMap 形式、有符号的截断距离场(Truncated Signed Distance Field TSDF)、有符号的欧氏距离场(Euclidean Signed Distance Field ESDF)形式。OctoMap^[20]的地图表达形式将地图以八叉树(Octo-Tree)的形式存储,并且将点云映射到三维空间的每一个体素(vertex)中,每一个体素中存储的为当前 vertex 被占用的概率。TSDF^[21]的地图表达形式是将点云映射到三维空间的每个体素中,而每一个体素中存储的当前 vertex 距离最近表面的距离,这种地图表达形式能够极快地查到当前位置最近障碍物的表面。ESDF^[22]的地图表达形式也是将点云地图映射到三维空间的体素中,

而每个体素中存储的当前 vertex 到最近障碍物的欧式距离。因为在路径规划算法中必须能够快速查询每一个体素空闲或者被占用的状态,或者将距离障碍物的距离信息传递给无人机,ESDF 具有评估障碍物距离信息和梯度变化信息,有利于路径规划算法的设计。因此在机器人导航系统中多采用 ESDF 的表达形式。Oleynikova 等人提出了一种实时的利用 TSDF 增量构建 ESDF 的方法 Voxbolx^[23]。韩路新等人提出了 FIESTA^[24](Fast Incremental Euclidean Distance Fields),用于快速增量式地构建 ESDF 地图,极大提升了稠密建图和地图表达的性能。

1.2.2. 无人机导航避障技术的研究现状

无人机的导航避障技术主要包含无人机运动规划(Motion Planning)和运动控制(Motion Controlling)两个部分。即利用感知系统输出的位姿和障碍物地图信息,动态地规划出当前无人机位置到目标点的最优可通行路径,并且将路径规划的结果生成无人机可执行的控制指令,下达给飞行控制器来完成无人机自主导航任务的执行。无人机运动控制可以利用 DJI A3、Pixhawk^[25]等工业界和学术界性能优良的飞控组件,实现精准的无人机运动控制,所以无人机导航避障系统的设计难点主要在于运动规划部分。通常运动规划算法主要包含路径搜索(Path Search)、轨迹优化(Trajectory Optimization)以及轨迹服务器(Trajectory Server)三个组成部分。

路径搜索称为运动规划的前端,主要功能是寻找一条从无人机当前位置到达目标终点的路径。前端的路径搜索也存在不同的选择策略,在机器人领域,最常用的路径搜索算法为启发式算法,其中最为典型的代表为 A*算法,通过不断估算每次移动代价和到达目标点代价的加权和,生成一条从当前点到达目标的最短路径^[26]。此外快速扩展随机树^[27](Rapidly-exploring Random Trees, RRT)算法也是一种在三维空间中有效的路径搜索算法,通过随机采样不断增加叶子节点的方式,生成一个随机扩展树,当随机树中的叶子节点包含了目标点或进入了目标区域,便可以在随机树中找到一条由树节点组成的从初始点到目标点的路径。

轨迹优化称为运动规划的后端,通常将其建模成一个带约束的二次规划问题来求解。其主要功能是基于当前的无人机状态和障碍物地图构建一些约束条件,对搜索的轨迹进行再次调优,将路径搜索的路径点变成平滑曲线的稠密轨迹点。后端的轨迹优化通过构建不同的约束也存在着不同的优化策略,比如约束轨迹经过某些路径点、设定最大加速度、设定最大速度、满足在规定时间内或空间的飞行走廊内等方式。Mellinger 等人提出的 Minimum snap^[28]轨迹优化算法就是通过约束加速度的二阶导数来最小化目标函数,从而求解最优轨迹。Ratliff 等人提出的

CHOMP^[29]轨迹优化算法主要通过约束离散的轨迹点来求解最优轨迹。高飞等人提出 Btraj^[30]轨迹优化算法,利用障碍物地图中空间点的几何信息生成飞行走廊并优化无人机航迹的控制点。轨迹服务器主要用于与飞行控制器的交互,根据无人机当前的姿态控制轨迹航点的发送,从而让无人机贴合规划的航线飞行。通常发送给无人机的控制命令形式包含位置控制、角度控制和速度控制等多种方式。

1.3. 主要研究内容与论文安排

无人机飞行器相比于其他的机器人和自动驾驶系统,具有更高维度的运动空间和更加灵活的运动范围。比如无人机在自主搜索救援的应用场景中经常需要从空中穿越丛林或者废墟建筑,在飞行过程中会导致 GPS 等传感器的状态切换,这将严重影响无人机的定位建图、路径规划和运动控制。为了解决这一问题,实现更加鲁棒的无人机自主感知与导航系统,本文提出了一种基于双目立体视觉与 IMU 和 GPS 相融合的无人机状态估计算法,用于提升无人机自主感知的稳定性。同时,设计了基于前端路径搜索和后端轨迹优化相结合的无人机导航避障算法,从而构建了一套完整的无人机自主感知与导航系统。同时搭建了基于 Gazebo 和 AirSim 的无人机仿真平台,以及基于 NVIDIA TX2 嵌入式平台的四旋翼飞行器硬件系统。此外,本文针对公开数据集、仿真平台和实际无人机系统都设计了丰富的实验,用于分析和验证各部分算法的性能以及整个系统的有效性,最后,利用仿真场景中无人机模型和真实环境下的四旋翼工程样机均实现了无人机自主导航避障的功能。图 1-3 为本文研究内容的章节脉络,论文主要的研究内容和章节结构安排如下:

第一章,绪论。首先介绍无人机自主感知与导航系统的研究背景与意义、然后分别分析了无人机自主感知技术、自主导航避障技术的国内外研究现状,最后为本文主要研究内容的概述。

第二章,自主无人机系统的总体方案设计。阐述了本文所构建的无人机自主感知与导航系统的设计方案,包括无人机的定位建图、路径规划和运动控制算法的级联、无人机系统的软、硬件设计方案和仿真设计方案。

第三章,无人机自主感知算法的设计。为了满足无人机多场景的应用需求,提升无人机自主感知的精确度和鲁棒性,本文提出了一种基于双目立体视觉与 IMU 和 GPS 相融合的无人机状态估计算法。在本章中详细阐述了多传感器融合的无人机状态估计的设计方法,此外,本章介绍了基于无人机状态估计与双目立体匹配构建稠密的障碍物环境地图的方法,从而实现无人机对自身姿态和周围环境地图的感知。

第四章，无人机自主导航避障算法的设计。本章主要阐述了本文无人机系统中自主导航避障算法的前端路径搜索和后端轨迹优化的设计方法。根据四旋翼无人机运动模型，设计了适合无人机系统的混合状态 A* 算法，用于替换前端的 3D A* 搜索算法，从而提升后端轨迹优化的性能。

第五章，无人机系统实验分析。本章通过实验对无人机的感知算法、导航算法以及整个无人机系统进行了详细的分析。在感知算法部分，在 KITTI 和 EuRoC 数据集上对本文的多传感器融合的无人机状态估计算法进行评测，并与 ORB_SLAM2、VINS_Fusion、ROVIO 等算法进行比较。同时基于 KITTI 和 EuRoC 数据集进行了室外大场景和室内小场景的建图。在导航算法部分，通过仿真实分析路径搜索和轨迹优化算法的实时性和准确性。然后分别在仿真场景中和真实的数据场景中进行前端路径搜索和后端轨迹优化的测试。最后，对于整个无人机自主感知与导航避障系统，分别在本文搭建的 Gazebo 与 AirSim 的无人机仿真平台、以及基于 NVIDIA TX2 与 Pixhawk 的实际四旋翼飞行器上进行完整的系统测试。

第六章，总结与展望。对本文的研究内容做出总结和分析，并为下一步工作指明方向。

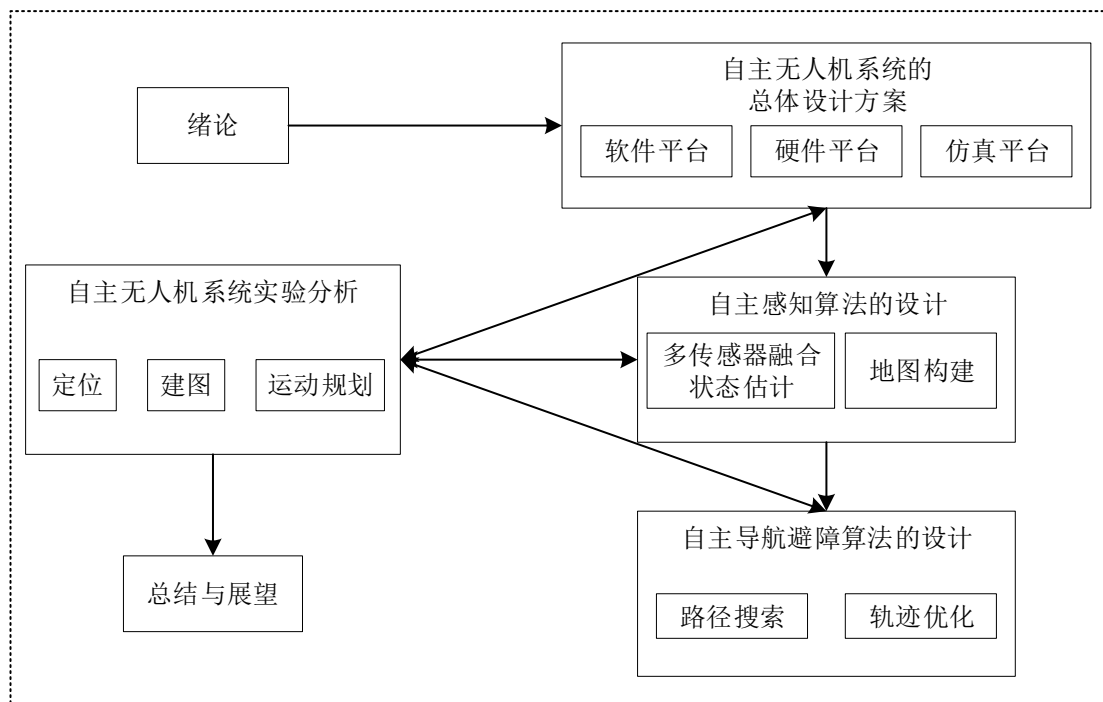


图 1-3 本文研究内容的章节脉络图

第2章 自主无人机系统的总体方案设计

本章主要阐述了课题中所设计的无人机自主感知与自主导航避障系统的整体方案，首先概述了本文的自主无人机总体设计框架，然后根据总体设计框架，介绍了系统的软件设计架构和无人机硬件平台的搭建。此外，本文针对自主无人机系统的开发与测试，设计了一套基于 Gazebo 和 AirSim 仿真器的完善的视觉无人机的仿真平台，可以实现整套自主无人机系统完全在线的软件在环仿真测试。

2.1. 无人机自主感知与导航系统总体方案

本文所设计的无人机自主感知与导航系统由传感器数据输入层、中间算法处理层、控制命令输出层组成。

传感器输入层：包含机载计算机挂载的双目立体相机，并且包括相机设备中所集成的 IMU 传感器、以及无人机飞控所挂载的 IMU 和 GPS 模组。为整个系统提供双目图像、IMU 和 GPS 测量的实时数据流。

中间的算法处理层：算法层是整个系统的核心，包含本文所提出的基于双目立体视觉与 IMU 和 GPS 相融合的无人机状态估计算法，实现对无人机系统的定位，结合稠密的障碍物地图构建，完成无人机的自主感知，同时也包含本文所改进的基于轨迹优化方式的无人机导航避障算法，完成无人机的自主路径规划。

控制命令输出层：接收状态估计算法提供的无人机位姿和导航避障算法所计算的轨迹航线，输出层将控制航点发送给无人机飞行控制器，驱动电机执行命令。

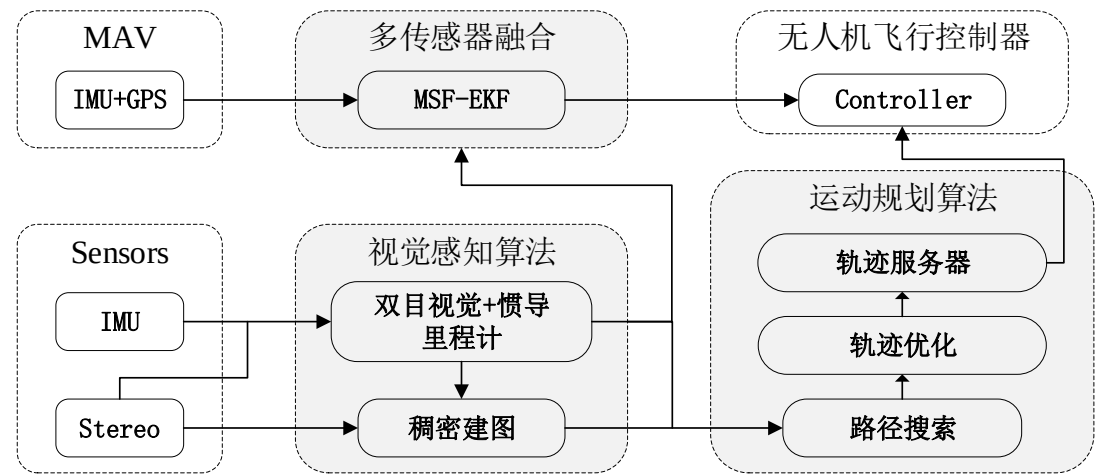


图 2-1 无人机自主感知与导航系统设计框图

如上图 2-1 所示,无人机自主感知与导航系统的设计主要集中在中间算法层,算法层主要包含无人机视觉感知、无人机运动规划两个方面。在无人机视觉感知部分中,本文提出了一种基于双目立体视觉与 IMU 和 GPS 相结合的多传感器融合无人机状态估计算法,首先,利用输入层提供的连续双目图像序列和 IMU 传感器的测量数据,通过视觉与惯导紧融合的 vSLAM 框架估计出无人机在运动过程中相机的姿态变化,同时再利用飞控提供的 IMU 传感器和 GPS 模组测量数据,将三者在外部分进行数据松耦合,并通过扩展卡尔曼滤波器估计出无人机的位姿运动。然后,通过同一时刻的左右两帧图像的双目立体匹配估计出图像中每个像素在相机坐标系下的深度,再结合当前的相机位姿就可以将图像中所有的像素点投影到世界坐标系下,从而构建出空间环境的稠密点云地图,为了使得障碍物地图能够更加适应路径规划算法的使用,本文进一步将障碍物点云地图映射到三维空间的体素中,表示成 ESDF 地图的形式,从而完成无人机在运动过程中对于自身状态的估计和障碍物环境的感知。

在无人机运动规划部分,本文采用了基于轨迹优化方式的无人机导航避障算法。首先,根据感知算法输出的位姿和障碍物地图,搜索出一条到达目的地的可通行路径。然后,根据 ESDF 栅格地图和四旋翼无人机的动力学约束对搜索的路径重新优化,生成一组更加稠密的离散轨迹点,最后通过轨迹服务器将导航命令发送给飞行控制器。最终通过视觉感知算法与运动规划算法的相互结合,实现无人机的自主感知与导航避障功能。

2.2. 软件架构的设计

本文采用 ROS^[31](Robot Operating System)机器人操作系统进行项目工程的构建与管理。ROS 机器人操作系统是一个用于编写机器人软件的灵活框架,它集成了大量的工具,提供了类似于操作系统所提供的功能^[32]。ROS 采用分布式网络结构,并使用 TCP/IP 的通信方式,实现各个功能模块间的点对点松耦合连接。通过这种形式,无人机自主感知与导航系统框架中的每个功能模块都能够被单独设计、编译,并且在运行时以松耦合的方式组合在一起。

ROS 机器人操作系统使用统一的消息接口,让模块的移植、复用更加便捷。目前 ROS 开源社区已经集成了大量的开源组件,例如 OpenCV^[33]库(Open Source Computer Vision Library)、PCL^[34]点云库(Point Cloud Library)、3D 可视化工具 RVIZ^[35]库(Robot Visualizer),利用这些丰富的组件可以快速地进行项目工程的开发迭代与测试验证。此外 ROS 支持 Linux 操作系统,这样可以直接在计算机上开发,然后直接部署在机载计算机的嵌入式平台上。

ROS 机器人操作系统设计了不同类型的节点通信机制,包括基于话题(Topic)消息的异步数据流通信、基于服务(Service)消息的同步数据流通信以及基于参数服务器和动态参数配置的形式。通过这种方式,可以灵活地实现无人机自主感知与导航系统框架中的不同功能节点的信号传递。

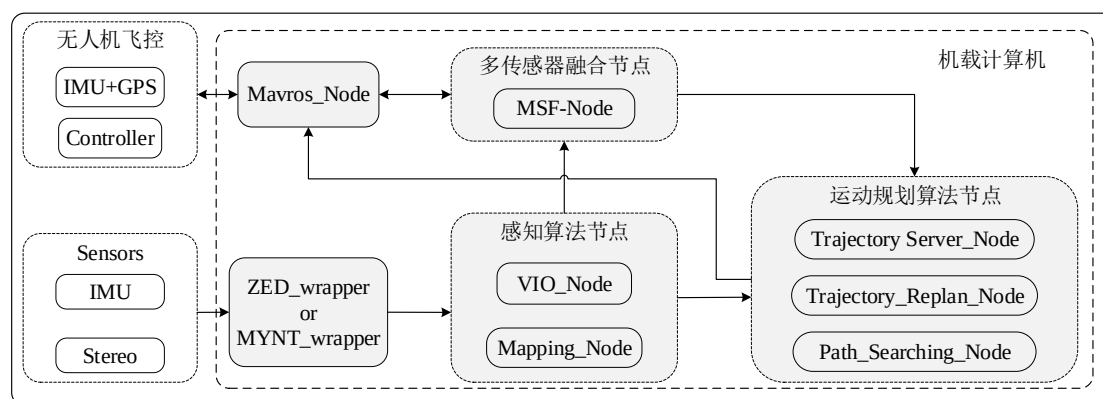


图 2-2 自主无人机系统软件节点级联图

如图 2-2 所示,为本文设计的无人机自主感知与导航系统软件节点级联图,包含发布传感器话题的驱动程序节点,无人机视觉定位和建图的节点、以及用于与飞行控制器进行数据交互的 Mavros 节点,多传感器融合的定位算法节点和用于路径搜索、轨迹优化和航迹发布的运动轨迹节点。节点之间通过话题的发布与订阅进行消息的传递。系统的处理流程为输入节点负责图像、IMU 和 GPS 数据的发布,感知部分通过视觉里程计节点订阅输入消息并发布相机位姿的话题,再根据相机位姿构建全局地图和局部地图。同时,多传感器融合节点将视觉里程计估计的姿态与飞控估计的无人机位姿相融合,并发布融合后的无人机状态信息。运动规划部分根据当前无人机位姿和障碍物地图进行路径搜索和轨迹优化,最后通过轨迹服务器节点将航迹命令发送给飞行控制器。

2.3. 硬件平台的搭建

为了在真实环境中测试本文所设计的无人机自主感知与导航系统的性能。本文搭建了一套挂载多传感器的四旋翼无人机飞行器。主要包含以下部件:具备一定载荷能力四轴飞行器机架、一个驱动无人机运动控制的 Pixhawk 飞行控制器、一个用于算法实时在线运行的 NVIDIA TX2 机载计算机、以及挂载在飞行控制器上的 GPS 模组、挂载于机载计算机上的双目立体视觉相机。为了验证本文所设计的自主无人机系统的鲁棒性,本文分别采用了 ZED 相机和 MYNTEYE 相机进

行实验测试。同时，为了方便测试，额外添加了一个无人机遥控器，用于调试和紧急制动，主要部件的清单如表 2-1 所示。

表 2-1 本文自主无人机系统的硬件参数配置表

名称	参数
无人机机架	飞越 Tarot 650
对称电机轴距	650mm
最大起飞重量	3500g
电池容量	5200mAh
遥控器	Fataba 14 通道遥控器
飞控控制器	Pixhawk v2
机载计算机	NVIDIA TX2 Module
GPS 模组	GPS M8N
双目相机	ZED / MYNTEYE 立体相机

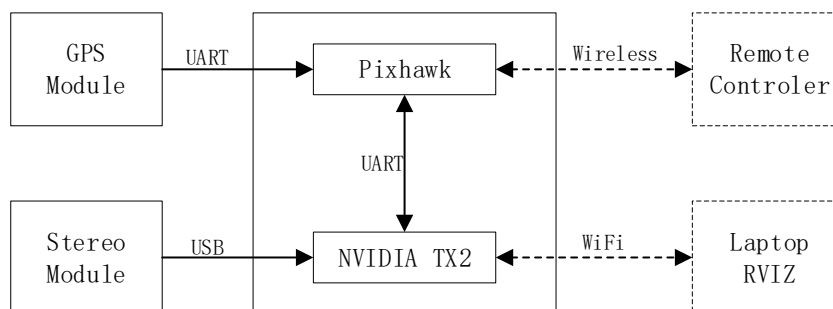


图 2-3 无人机系统硬件设备连接示意图

如图 2-3 所示，为本文无人机系统的硬件设备连接示意图。GPS 模组和双目摄像头传感器作为数据的输入层为系统提供观测数据，GPS 通过串口连接 Pixhawk 飞行控制器，在室外场景中利用 GPS 数据与飞控中 IMU 测量数据进行扩展卡尔曼滤波融合可以估计出当前无人机的状态。Stereo 双目摄像头模组通过 USB3.0 接口连接 NVIDIA TX2 机载计算机，为系统提供持续的左、右图同步的图像序列，在复杂的场景中，利用连续输入的左、右图序列使用 vSLAM 算法估计相机的位姿运动，并且可以利用双目立体匹配估计出图像中像素的深度，从而构建稠密的障碍物地图。在系统中通过无线发射器和无线接收机建立遥控器与飞控的连接，一方面使用遥控独立测试系统的定位和建图的性能，另一方面也可以在无人机自主避障测试中处理突发情况，使得无人机及时迫降。此外，在 NVIDIA

TX2 机载计算机上挂载了无线网卡，可以将笔记本电脑与机载计算机建立 WiFi 连接，方便算法的调试运行以及可视化导航避障算法的中间结果。最终本文搭建的无人机感知导航硬件系统如图 2-4 所示。



图 2-4 无人机硬件系统图

2.4. 仿真平台的设计

在无人机系统的研究与设计中，尤其是具备自主感知与自主避障等复杂功能的无人机系统，需要有无机仿真平台的支撑。一方面因为无人机系统造价昂贵，在不能保证算法正常运行的情况下贸然测试容易造成不必要的无人机坠毁，大大增加了实验成本，另一方面，无人机系统的实验具有极大的危险性，很容易由于算法设计的缺陷或者操作人员失误发生意外情况。从另一个角度分析，无人机系统的实验需要在大场景下长时间的运行，才能验证系统设计的有效性，而在实际中由于场地的限制很难有合适的测试环境、小型无人机的续航能力也无法保证其长时间的运行。因此设计一套无人机仿真系统具有十分必要的研究意义。本文针对自主无人机系统，设计了一套完善的无人机自主感知与导航的仿真系统，其中包含基于 Gazebo 仿真引擎的自主无人机仿真平台设计、以及基于 Unreal Engine4 仿真引擎的自主无人机仿真平台设计。下文分别对两种仿真系统的设计方法做出详细地阐述。

2.4.1. 基于 Gazebo 的无人机仿真平台设计

Gazebo^[36]是一个功能强大的三维物理仿真器，支持多种高性能的物理引擎，如支持 ODE、Bullet、SimBody^[37]、DART 等多种物理引擎的动力学仿真。具备高质量的图形渲染，例如可以显光线、纹理、阴影等多种逼真的三维环境。拥有方便的编程与图形接口，支持单目相机、双目相机、RGB-D 相机、IMU、GPS 等多种传感器数据的仿真，同时也可以设定仿真传感器的噪声。开源的 Gazebo 社区中有功能丰富的插件，同时用户也可以自主开发插件以满足个性化需求。Gazebo 与 ROS 类似，也是通过 TCP/IP 进行数据传输与交互，这样可以根据设计需求选择在云端部署仿真系统或者在本地部署仿真系统。利用一些 ROS 与 Gazebo 交互的插件，可以将 Gazebo 中仿真传感器的数据以话题的方式发布出来，同时也可以将控制命令以输入的方式回传到仿真环境中去。通过这种方式可以设计与真实无人机系统高度一致的软件系统，保证在仿真系统和真实系统中共用同一套软件程序。如图 2-5 所示，通过 Gazebo 仿真系统可以实现移动机器人、物流仓储机器人、自动驾驶汽车、自主无人机等多种不同类型的机器人仿真系统。



图 2-5 基于 Gazebo 系统的不同类型的仿真模型

在设计基于 Gazebo 的自主无人机感知与导航仿真系统时，同样应该遵循真实场景中无人机系统的设计模式。在真实世界中，当无人机在某一个地理区域飞

行时, Pixhawk 飞行控制器接收 GPS 模组和 IMU 传感器的测量数据, 根据测量数据进行机体的状态估计, 最后驱动电机做相应的运动控制。而在仿真系统中, 由 Gazebo 提供一个虚拟世界的仿真环境, 并且提供四旋翼无人机的仿真模型和虚拟的 GPS、IMU 等传感器的数据。同时根据 Pixhawk 固件的软件在环仿真的模式(Software In The Loop SITL), 接收来自 Gazebo simulator 的虚拟传感器数据、并发送给仿真环境电机的输出响应。通过这种方式可以实现多旋翼无人机在仿真环境中的运动, 通过 QGroundControl^[38]地面站软件与 Pixhawk SITL 建立网络连接, 可以操控无人机起降并设定飞行任务。除此之外, 为了模拟基于视觉的无人机自主感知与导航算法的仿真, 需要从 Gazebo 中订阅无人机挂载的虚拟相机的图像, 通过视觉定位、稠密建图、路径规划算法处理之后将控制命令发送给 Pixhawk SITL, 最后反馈给仿真环境中无人机模型的执行单元。具体的基于 Gazebo 的自主无人机仿真系统通信接口连接图如图 2-6 所示。

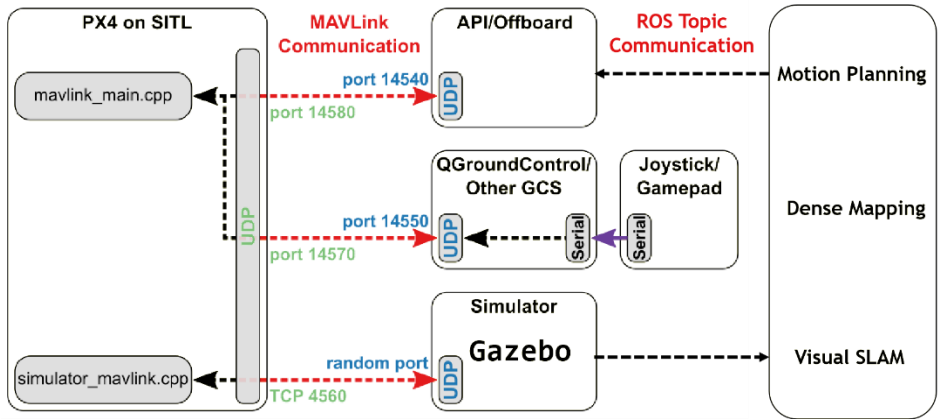


图 2-6 基于 Gazebo 的无人机仿真系统通信接口连接示意图

在基于 Gazebo 仿真器搭建自主无人机感知与导航仿真系统时, 可利用 ROS 框架搭建整个仿真系统。如图 2-7 所示, 本文所设计的基于 Gazebo 的无人机仿真系统包含三个组成部分: Gazebo 仿真引擎、Px4 SITL 软件在环仿真的飞行控制栈、以及用于无人机自主感知与导航的算法。

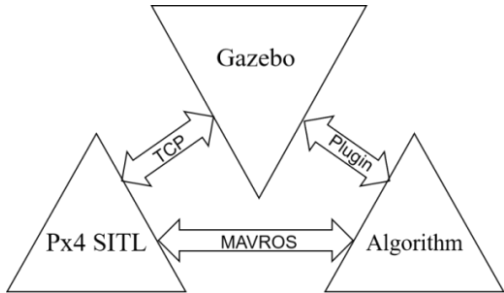


图 2-7 基于 Gazebo 的自主无人机仿真系统节点级联图

首先，通过编写仿真环境配置文件，Gazebo 仿真引擎生成一个虚拟的场景并加载一个挂载多传感的无人机模型，感知与导航算法模块通过 Gazebo 与 ROS 连接的插件获取虚拟传感器的数据，然后依次执行基于视觉的定位、建图、路径规划的算法，实时计算出能绕开所有障碍物从当前位置到达目标点的轨迹航线。Px4 SITL 一方面通过 TCP 网络连接接收来自 Gazebo 仿真环境的虚拟传感器的数据，另一方面通过 MAVROS 节点接收来自算法节点输出轨迹航线。Px4 SITL 再通过反馈控制算法产生四旋翼每个电机的执行响应，最后通过 TCP 网络回传给仿真环境。整个过程中包含的 ROS 节点如表 2-2 所示。

表 2-2 基于 Gazebo 的自主无人机仿真系统的主要功能节点

节点名称	功能描述
gazebo_ros/launch/world.launch	启动 Gazebo 虚拟环境仿真引擎
gazebo_ros/spawn_model	加载无人机仿真模型
px4/launch/px4.launch	启动 Px4 SITL 软件仿真模式
vio_slam.launch	启动视觉定位算法
dense_mapping.launch	启动稠密建图算法
motion_planning.launch	启动路径规划算法
mavros/launch/node.launch	建立 Px4 SITL 与算法节点的连接
libgazebo_ros_multicamera plugin	发布虚拟相机话题消息

最终在 Gazebo 中实现的无人机仿真系统如图 2-8 所示，左图为 Gazebo 仿真器的虚拟场景和四旋翼无人机模型，中图是从虚拟场景订阅的双目图像，右图为双目立体匹配算法估计障碍物深度，并在 ROS Rviz 组件中可视化的结果。通过仿真系统可实现无人机感知与导航算法的验证。

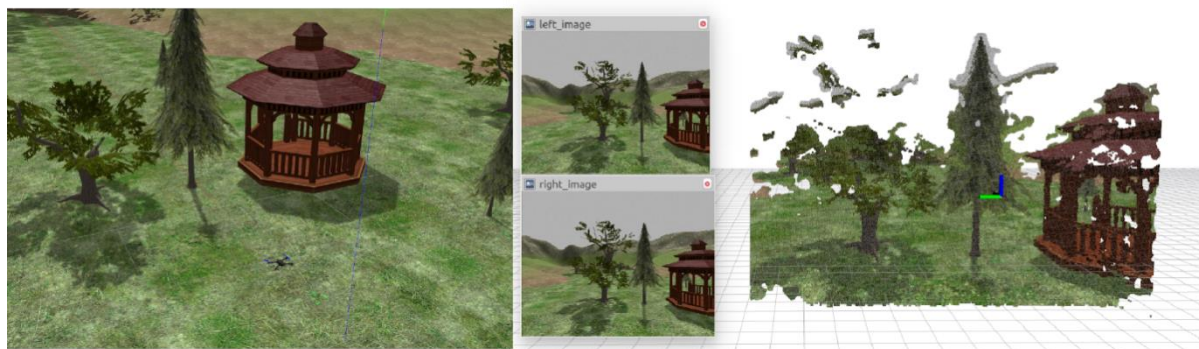


图 2-8 基于 Gazebo 的自主无人机仿真系统效果图

2.4.2. 基于 AirSim 的无人机仿真平台设计

在机器人研究领域，Gazebo 毫无疑问是应用最为广泛的仿真平台。它支持不同类型的物理引擎、拥有多种类型的传感器模型和多场景的 3D 环境，支持多种类型的机器人刚体运动(如电机转子、转轴、机械臂等)。然而 Gazebo 最大的缺陷就是无法创建具有大规模尺寸、复杂纹理的逼真仿真场景。对于本文所研究的基于视觉的无人机自主感知与导航系统，仿真系统所产生的虚拟图像数据对于整个系统至关重要，所以一个具备更逼真图像渲染的仿真平台才更加切合本文的仿真需求。因此本文进一步设计了基于 AirSim^[39]的自主无人机仿真系统。

AirSim 是专门针对于无人机和无人车所开发的仿真模拟器，它的底层是基于虚幻引擎(Unreal Engine)^[40]或者 Unity 3D^[41]实现的，可以实现在物理引擎和视觉渲染上都十分逼真的仿真效果。AirSim 支持软件在环仿真和多种硬件在环仿真(例如 Pixhawk、DJI A3 等)。在使用的过程中，将 AirSim 仿真器作为 Unreal Engine 的插件，可以放入任何一个仿真环境中，如图 2-9 所示，将 AirSim 插件添加在一个小镇街道的仿真环境中，实现无人机模型在虚拟环境中的飞行。

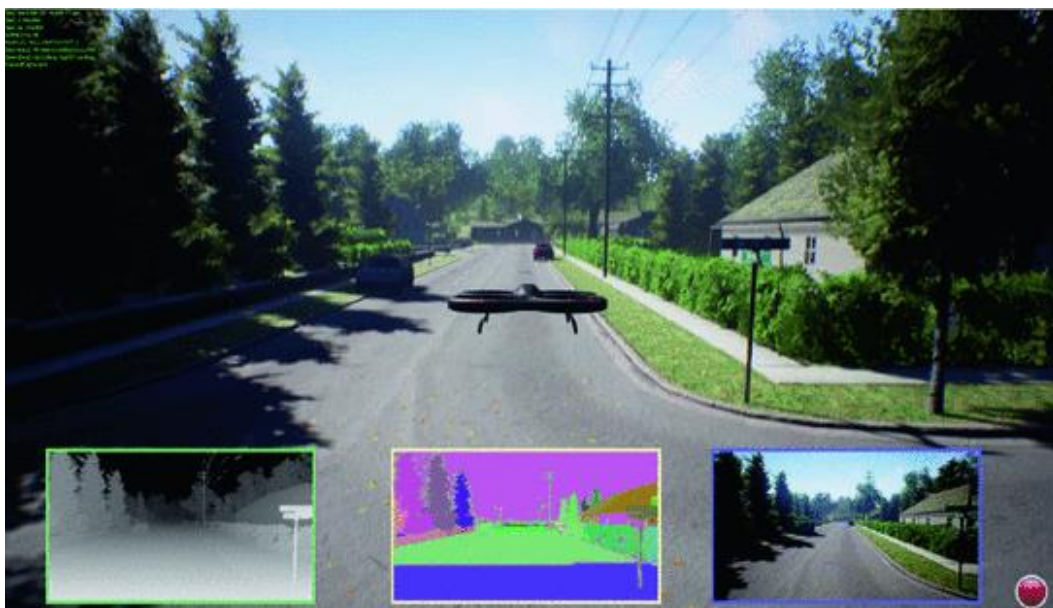


图 2-9 基于 AirSim 仿真器的无人机在 Unreal Engine 环境中飞行

本文设计的基于 AirSim 的无人机自主感知与导航仿真系统包含四个组成部分：AirSim 仿真器、Pixhawk 飞控的软件在环仿真(SITL)或者硬件在环仿真(HITL)、QGS 地面站软件以及感知与导航算法程序。四个部分的连接关系如图 2-10 所示。

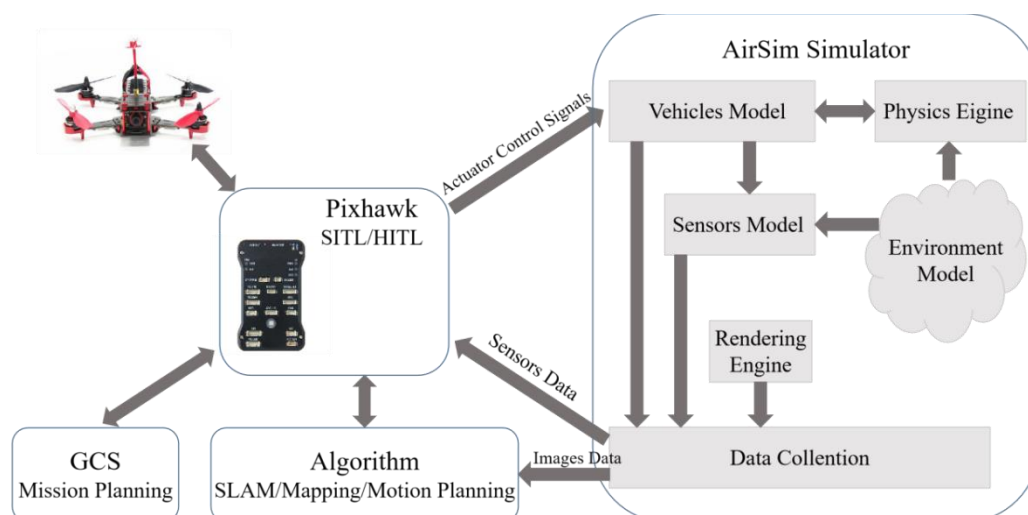


图 2-10 基于 AirSim 的自主无人机仿真平台设计框图

首先 AirSim 仿真器包括 Unreal Engine 所创建的仿真场景和 AirSim 的无人机插件，根据不同的功能可以将其细化为多个层次。Environment Model 主要提供重力、空气密度、空气阻力、电磁等物理属性的仿真；Physics Engine 和 Vehicle Model 主要描述无人机刚体运动和状态转移方程(例如无人机的位置、旋转、线速度、线加速度、角速度、角加速度)，Sensors Model 提供 GPS、IMU、磁力计等仿真传感器的数据、Rendering Engine 提供虚拟摄像头的图像数据，所有的数据都通过统一的数据接口与外界进行交互。

Px4 飞行控制器可以使用 SITL 模式或者 HITL 模式与 AirSim 仿真器进行交互，当使用 SITL 时，需要启动 Px4 SITL 软件仿真固件，Px4 与 AirSim 通过网络进行连接，Px4 接收来自仿真环境的传感器数据，通过 Px4 SITL 处理后输出给 Vehicle Model 电机驱动的控制信号。而当使用 HITL 时，需要通过串口将 Px4 飞控硬件设备连接到计算机，通过串口通信接收 AirSim 的传感器数据并反馈给仿真器电机控制信号。通过两种不同的仿真形式，均可以实现无人机模型在 AirSim 模拟器中飞行。

通过地面站(Ground Control Station GCS)软件可以控制无人机模型在仿真环境中起降、执行飞行任务。但是本文的主要目的是实现基于视觉的无人机自主导航避障的仿真，因此需要在此基础上添加与本文感知与导航算法的交互接口。算法部分通过接收 AirSim 的图像数据和 Pixhawk 的状态数据，通过定位、建图、路径规划算法，计算出无人机的航迹规划，最终轨迹航点发送为 Pixhawk 飞行控制器，从而实现无人机模型在 AirSim 仿真环境中的自主导航避障的功能。

如图 2-11 和 2-12 所示，即为本文基于 AirSim 设计的无人机仿真平台运行效果图。在图 2-11 中，通过启动 AirSim 仿真器和 Px4 SITL，实现四旋翼无人机

模型在仿真环境中飞行，并可以获取无人机模型前视角的 RGB 图和深度图。在图 2-12 中，通过 ROS 接收无人机的位姿和深度图信息，将障碍物映射成三维空间点云，并在 ROS rviz 组件中可视化显示的结果。



图 2-11 AirSim 无人机仿真系统运行图

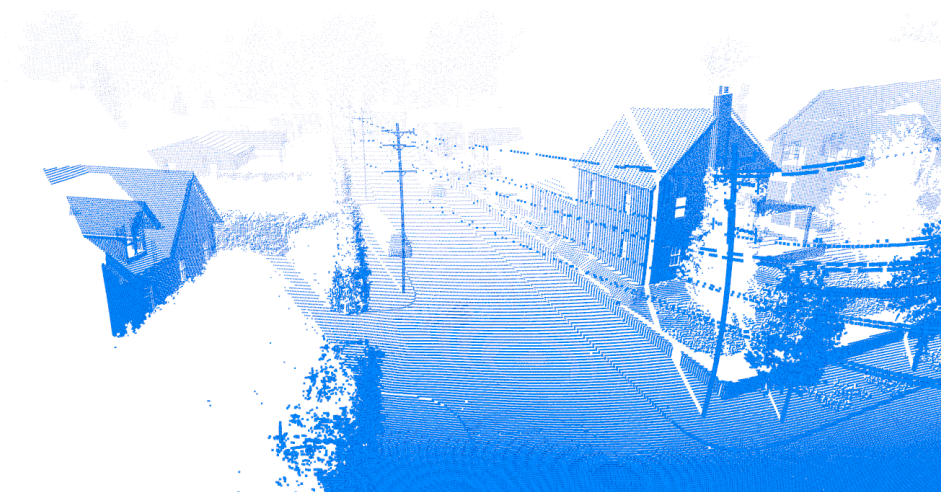


图 2-12 基于场景深度图构建的障碍物点云信息

2.5. 本章小结

本章详细阐述了无人机自主感知与导航避障系统的整体设计方案，设计了相应的四旋翼无人机硬件平台和基于 Gazebo、AirSim 的无人机软件仿真平台，并开发了基于 ROS 机器人操作系统的无人机自主感知导航软件架构。为后序的无人机感知算法和导航避障算法的设计提供总体设计和基础实验平台。

第3章 无人机自主感知算法的设计

3.1. 引言

感知算法是整个无人机自主导航避障系统的重要组成部分,因为只有系统感知到当前无人机所处的状态(如无人机 6-DoF 位姿、线速度、加速度和旋转角速度)、以及当前无人机周围的环境地图,这样才能够保证无人机在飞行过程中安全地规划出航迹并准确地做出响应。而无人机系统的感知算法不同于其他机器人和自动驾驶系统,因为无人机具有更高维度的运动空间和更加灵活的运动范围。比如无人机有高空飞行、低空巡航、或者室内外切换的多种应用场景,针对不同的场景需要利用相应的传感器设计不同的感知算法。随着行业无人机技术的发展和应用领域不断扩大,无人机在多场景下的感知问题是需要解决的关键问题。

本文针对无人机的多场景感知问题,提出了一种基于双目立体视觉与 IMU 和 GPS 相结合的多传感器融合无人机状态估计算法,采用紧耦合与松耦合相结合的方式,能够同时满足无人机在多场景和跨场景的状态估计问题。在室内、楼宇间、以及丛林等 GPS 失效的情况下,可依靠双目视觉与 IMU 紧耦合的视觉里程计估计无人机的位姿。在高空飞行时缺乏图像纹理信息,系统可依靠 GPS 与 IMU 的 EKF 融合实现无人机的定位。而在低速巡航的低空室外场景中,可结合双目图像、GPS 和 IMU 传感器的信息,用视觉和 IMU 数据进行无人机位姿预测,用 GPS 全局信息优化累积误差,从而实现更加精准、稳定的无人机状态估计。

在本章中,首先详细阐述了基于双目立体视觉与 IMU 和 GPS 相结合的多传感器融合无人机状态估计算法的设计原理,包括基于双目相机的视觉里程计、视觉与 GPS 融合的无人机状态估计两个部分。然后,结合无人机状态估计的位姿,利用双目立体匹配构建障碍物环境地图。从而实现无人机对于自身状态和周围环境的感知。

3.2. 多传感器融合的无人机状态估计

本文提出了一种基于双目立体视觉与 IMU 和 GPS 相结合的多传感器融合无人机状态估计算法,包含双目视觉里程计、基于视觉位姿与 IMU 和 GPS 融合的无人机状态估计两个部分。如图 3-1 所示。首先利用机载的双目相机、IMU 传感器,通过紧耦合的位姿图优化估计出运动相机的位姿,然后结合飞控的 GPS 和

IMU 传感器信息，进行基于扩展卡尔曼滤波方式的多传感器融合。基于视觉与 IMU 融合所估计的位姿平滑稳定，但是存在着累积漂移问题，而基于 GPS 与 IMU 融合所估计的位姿略有波动，但是具有全局一致性。因此，将两者状态信息进行互补融合，可以为整个无人机系统提供更加精确的位姿。下面分别阐述双目视觉里程计、多传感器融合两个部分的设计方法。

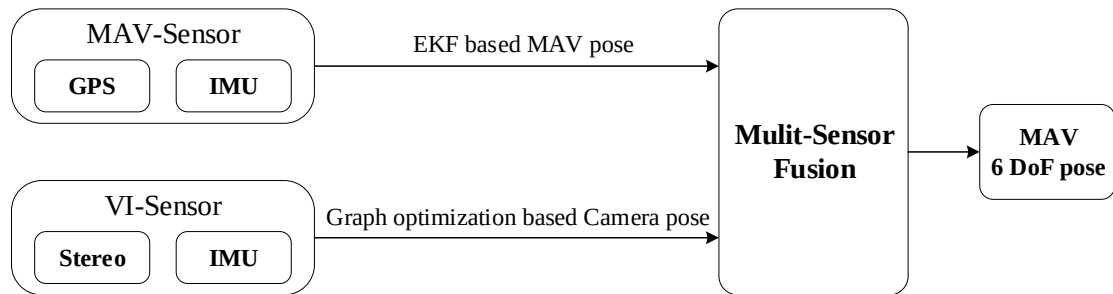


图 3-1 基于多传感融合的无人机状态估计设计框图

3.2.1. 双目视觉里程计

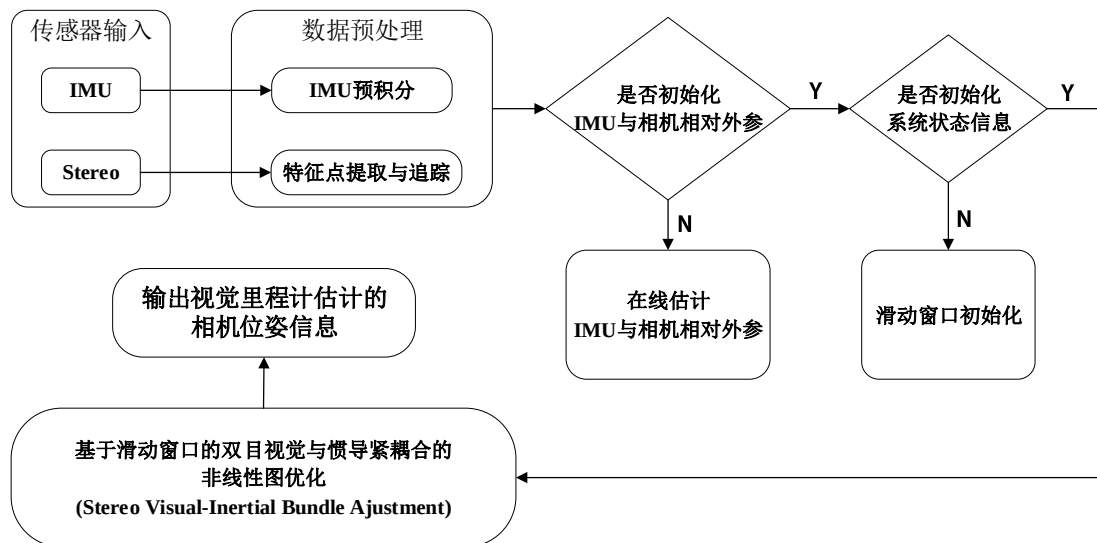


图 3-2 双目视觉里程计设计框图

本文所提出的双目视觉里程计，采用了数据紧耦合和非线性优化的设计思想。系统输入按照时间序列采集的 IMU 测量数据(三轴加速度和三轴角速度)、双目图像数据，输出运动过程中相机的位姿变化。如图 3-2 所示，本文的双目视觉里程计算法主要分三个层次：预处理阶段、初始化阶段、后端非线性优化阶段。

预处理阶段：主要对输入的双目图像和 IMU 传感器数据进行初步的处理，包括 IMU 数据的积分和预积分操作，双目图像的特征点提取与追踪；

初始化阶段：主要为状态估计系统提供一组初始状态，包括 IMU 传感器坐标系与摄像头传感器坐标系的相对外参、以及初始窗口中每个图像帧的位姿、速度、IMU 传感器的测量偏置、特征点的深度等状态信息；

后端非线性优化：根据预处理阶段的测量和预测，以及当前滑动的窗口中的状态，利用非线性优化的方法对窗口中的所有状态量进行优化调整，最终系统输出优化求解之后的无人机的运动状态(无人机的位置、姿态、速度)。下面将对本文双目视觉算法中预处理阶段、初始化阶段、后端非线性优化阶段的处理方法作详细的阐述。

3.2.1.1. IMU 与双目图像数据预处理

预处理阶段主要是对输入的 IMU 数据和双目图像数据进行初步的对齐、降维和整理，处理方式主要取决于传感器的输入特性以及模型参数化的形式。IMU 传感器的测量值为三轴加速度和三轴旋转角速度，对其每一帧数据进行相应的积分和预积分处理，得到传感器位置、速度和旋转的累积运动，而双目摄像头采集的是连续的图像序列，对其每一帧图像进行特征点提取与追踪，得到连续图像间的匹配像素点对，为 SLAM 的后端提供观测数据。下面分别对 IMU 数据的预处理过程和双目图像的预处理流程做详细的阐述。

MEMS IMU 传感器具有一个三轴的陀螺仪和三轴加速度计，能够测量物体在三维空间中的角速度和加速度，并以此解算出物体的姿态变化。但是 IMU 传感器的数据会受到噪声的影响，如式(3-1)、(3-2)所示，通常将旋转角速度测量和加速度测量建模成测量值等于在真值的基础上叠加了高斯白噪声以及测量偏置的随机游走噪声。式(3-1)、(3-2)中 w_m 、 a_m 分别表示旋转角速度和加速度的测量值， w 、 a 表示真实值， b 、 n 分别表示传感器测量的随机游走噪声和高斯白噪声。

$$w_m = w + b_w + n_w \quad (3-1)$$

$$a_m = a + b_a + n_a \quad (3-2)$$

在三维空间中，通常将刚体运动描述为位置(Position)、速度(Velocity)、旋转(Rotation / Quaternion)等参数随时间的变化。如式(3-3)、(3-4)、(3-5)所示，分别表示在 t 时刻下，IMU body 在三维空间的位置 p_{wb_t} 、速度 v_t^w 、四元素 q_{wb_t} 对时间的导数。其中加速度和旋转角速度可以由 IMU 测量真值 a 、 w 来表示。

$$\dot{p}_{wb_t} = v_t^w \quad (3-3)$$

$$\dot{v}_t^w = a_t^w \quad (3-4)$$

$$\dot{q}_{wb_t} = q_{wb_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2} w^{b_t} \end{bmatrix} \quad (3-5)$$

因此，对于每一时刻 IMU 的旋转角度和加速度的测量值使用中值积分，可以得到 IMU 传感器累积的位置、速度、旋转的变化。如式(3-6)、(3-7)、(3-8)所示，根据 k 时刻的状态，以及 k 时刻和 $k+1$ 时刻的测量，计算 $k+1$ 时刻的状态。如式(3-9)、(3-10)所示， a 、 w 分别表示 k 时刻和 $k+1$ 时刻的加速度和旋转角速度测量的平均值。其中注意加速度的测量要去掉自然重力加速度的影响。

$$p_{wb_{k+1}} = p_{wb_k} + v_t^w \Delta t + \frac{1}{2} a \Delta t^2 \quad (3-6)$$

$$v_{k+1}^w = v_t^w + a \Delta t \quad (3-7)$$

$$q_{wb_{k+1}} = q_{wb_k} \otimes \begin{bmatrix} 0 \\ \frac{1}{2} w \Delta t \end{bmatrix} \quad (3-8)$$

$$a = \frac{1}{2} [q_{wb_k} (a^{b_k} - b_{a_k}) - g^w + q_{wb_{k+1}} (a^{b_{k+1}} - b_{a_k}) - g^w] \quad (3-9)$$

$$w = \frac{1}{2} [(w^{b_k} - b_{w_k}) + (w^{b_{k+1}} - b_{w_{k+1}})] \quad (3-10)$$

IMU 传感器可以方便地测量运动物体的旋转角速度和加速度，通过积分可以直接得到物体位置、速度、旋转的变化，但是因为 IMU 的测量存在着高斯白噪声和测量偏置的随机游走噪声，随着时间的增加，IMU 积分计算的结果会存在严重的漂移，所以经常需要将 IMU 传感器的测量与其他传感器相融合，产生更加准确的估计结果。在视觉与惯导紧耦合的 SLAM 系统中，通常计算两个连续两帧图像之间的 IMU 的积分增量，从而将视觉数据与惯导数据对齐，这个过程称之为预积分(Pre-Integration)^[42]。一方面，因为 IMU 传感器的测量频率很高，通常是图片帧率的 10 倍左右(例如 IMU 200Hz，Image 20Hz)，采用预积分可以保证每一时刻的视觉帧都有一个与之对应的 IMU 预积分增量；另一方面，预积分只是连续视觉帧之间的 IMU 测量增量，与起始时刻的位姿无关，所以当后端非线性优化更新状态量时，并不会改变预积分增量的值，并且两个连续的视觉帧的时间间隔很短，避免了严重的累积漂移问题。具体的预积分增量的推导如公式(3-11)、(3-12)、(3-13)所示，位置、速度、四元素旋转的预积分增量分别为(3-14)、(3-15)、(3-16)。

$$p_{wb_j} = p_{wb_i} + v_i^w \Delta t - \frac{1}{2} g^w \Delta t^2 + q_{wb_i} \iint_{t \in [i, j]} (q_{b_i b_t} a^{b_t}) \delta t^2 \quad (3-11)$$

$$v_j^w = v_i^w - g^w \Delta t + q_{wb_i} \int_{t \in [i, j]} (q_{b_i b_t} a^{b_t}) \delta t \quad (3-12)$$

$$q_{wb_j} = q_{wb_i} \int_{t \in [i, j]} q_{b_i b_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2} w^{b_t} \end{bmatrix} \delta t \quad (3-13)$$

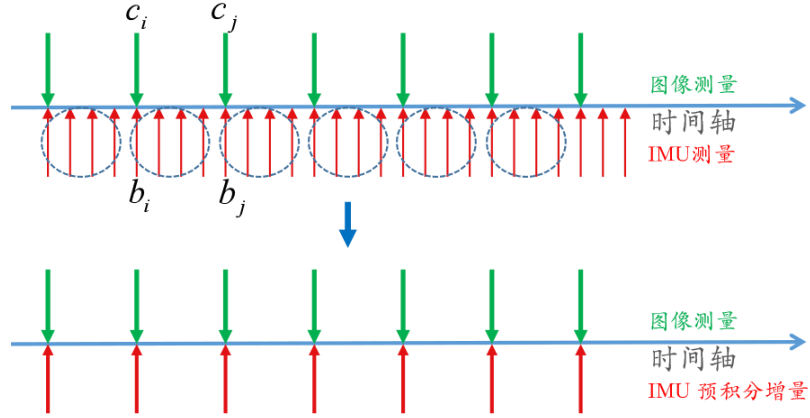


图 3-3 IMU 预积分增量描述示意图

如公式(3-11)、(3-12)、(3-13)和图 3-3 所示，两个连续的视觉图像数据 c_i, c_j 中间对应着 $b_i \dots b_j$ 多帧 IMU 测量数据， p_{wb_i} 、 v_i^w 、 q_{wb_i} 分别表示在 i 时刻 IMU body 在世界坐标系下的位置、速度、四元素，公式(3-11)、(3-12)、(3-13)分别表示以 i 时刻 IMU body 的位置、速度、四元素为参考，对 c_i, c_j 间隔内的 $b_i \dots b_j$ 多帧 IMU 测量数据积分，计算出 j 时刻 IMU body 在世界坐标系下的位置、速度、四元素 p_{wb_j} 、 v_j^w 、 q_{wb_j} 。其中 i, j 时刻所对应的预积分增量为 $\alpha_{b_i b_j}$ 、 $\beta_{b_i b_j}$ 、 $\gamma_{b_i b_j}$ ，具体形式如公式(3-14)、(3-15)、(3-16)所示。

$$\alpha_{b_i b_j} = \iint_{t \in [i, j]} (q_{b_i b_t} a^{b_t}) \delta t^2 \quad (3-14)$$

$$\beta_{b_i b_j} = \int_{t \in [i, j]} (q_{b_i b_t} a^{b_t}) \delta t \quad (3-15)$$

$$\gamma_{b_i b_j} = \int_{t \in [i, j]} q_{b_i b_t} \otimes \left[\frac{1}{2} w^{b_t} \right] \delta t \quad (3-16)$$

将预积分公式表示成离散形式，则两个连续 IMU 测量之间位置、速度、旋转的 IMU 预积分增量分别表示为式(3-20)、(3-21)、(3-18)。公式(3-17)、(3-18)中 a 、 w 分别表示以 i 时刻为基准，从 k 时刻和 $k+1$ 时刻的加速度和旋转角速度测量的平均值。

$$w = \frac{1}{2} [(w^{b_k} - b_{w_k}) + (w^{b_{k+1}} - b_{w_{k+1}})] \quad (3-17)$$

$$\gamma_{b_i b_{k+1}} = \gamma_{b_i b_k} \otimes \left[\frac{1}{2} w \delta t \right] \quad (3-18)$$

$$a = \frac{1}{2} [q_{b_i b_k} (a^{b_k} - b_{a_k}) + q_{b_i b_{k+1}} (a^{b_{k+1}} - b_{a_{k+1}})] \quad (3-19)$$

$$\alpha_{b_i b_{k+1}} = \alpha_{b_i b_k} + \beta_{b_i b_k} \delta t + \frac{1}{2} a \delta t^2 \quad (3-20)$$

$$\beta_{b_i b_{k+1}} = \beta_{b_i b_k} + a \delta t \quad (3-21)$$

双目视觉数据的处理方式是对每一帧图像进行特征点提取与特征点追踪。图像特征点的提取是指从一幅图像中选取一些有代表性的点,这些点在相机视角发生微小运动时能够保持不变,基于这种相互匹配的点的能够进行相机的位姿估计^[43]。常用的一些特征点为图像中纹理变化明显的角点、边缘和区域块。在计算机视觉领域,研究者经过长期的研究,设计出许多性能稳定的图像特征,例如具有光照、尺度、旋转不变特性的 SIFT^[44](Scale Invariant Feature Transform)特征、在 SIFT 基础上改进的 SURF^[45](Speeded Up Robust Features)特征、以及利用 FAST^[46]关键点和 BRIEF 描述子的 ORB^[47](Oriented FAST and Rotated BRIEF)特征等,不同的特征点的匹配精度和计算速度上存在很大差异, SIFT 和 SURF 算子提取的特征点具有旋转、平移不变性,但是实时性较差, FAST 和 ORB 算子所提取的特征匹配精度略差,但实时性比较好。考虑本文所设计的系统,需要利用无人机机载计算机在线处理。为了保证系统的实时性,本文选用实时性较好的 FAST 算子进行特征点提取,并且对提取的特征点使用稀疏光流法进行特征点追踪。本文对于双目图像的特征点提取与追踪流程如下图 3-4 所示。

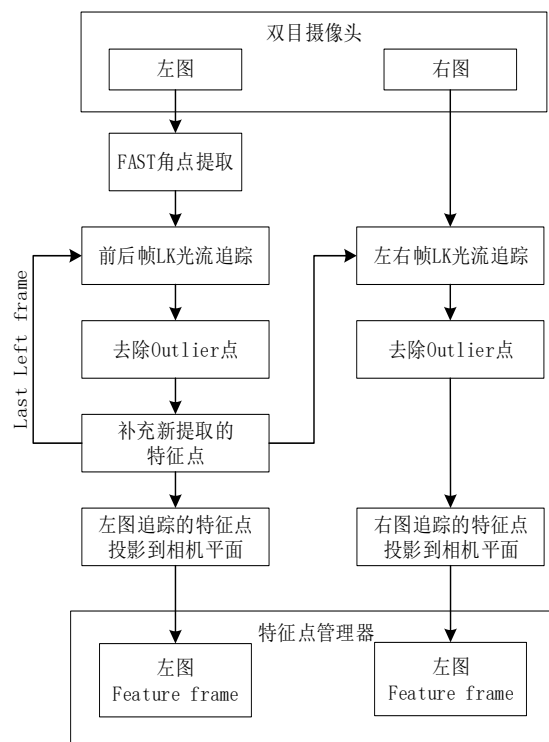


图 3-4 双目图像的特征点提取与追踪流程

当第一次获得双目相机的左图和右图时,先对左图进行 FAST 角点提取,获取 300 个分布均匀的特征点,后序每次获得双目相机的图像时,直接将左图与前一帧左图中的特征点进行 Lucas-Kanade 光流追踪^[48],并使用

RANSAC^[49](Random Sample Consensus)算法、光流逆向追踪方法去除光流追踪的误匹配。消除误匹配会去除一些错误匹配的特征点，为了使得特征点数量维持恒定，需要再次提取 FAST 角点，使得特征点的数量稳定在 300 个。因为同一时刻的左图和右图具有重叠区域，所有再次使用光流法将左图中特征点在右图中找到匹配的特征点，同样使用 RANSAC 算法和光流逆向追踪来消除误匹配的特征点。双目视觉的预处理阶段输出每一帧图像追踪到的特征点在左相机和右相机各自的归一化相机坐标系下的特征点集合，由整个系统的特征点管理器来维护和更新。

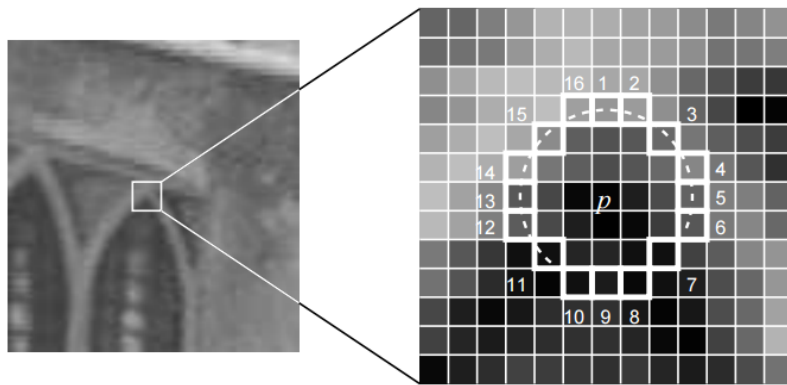


图 3-5 FAST 特征点的检测过程

FAST 算子主要利用一个像素点与其邻域内像素的灰度的差别，能够快速检测出图像中的角点和纹理变化明显的边缘。如图 3-5 所示，为一个 FAST 特征点的检测过程中。首先，对于待检测的某个像素点 p 的像素亮度为 I_p ，设置一个检测阈值 T ，然后以当前像素 p 为中心，选取半径为 3 个像素的圆周邻域上的 16 个像素点，如果其中有 N 个连续的邻域像素点的像素亮度都大于 $I_p + T$ 或者小于 $I_p - T$ ，那么则认为像素 p 为 FAST 特征点，通常 N 取值 12，相对应的特征点称为 FAST-12 特征点。图 3-6 为使用 FAST 算子对 KITTI 数据集一张图像处理之后的结果。



图 3-6 FAST 特征点提取

Lucas-Kanade 光流是一种描述图像像素随时间运动的方法。因此可以使用 LK 光流来追踪特征点在连续的图像序列中的运动。在使用光流法时，通常由两个假设条件：灰度不变假设和窗口同向假设。灰度不变假设指同一个像素点的灰度值在前后连续的两帧没有明显的变化，通常假定其像素灰度值相等。如式(3-22)所示，表示在 dt 的时间间隔内， $p(x, y)$ 像素点的灰度值保持不变。式(3-22)使用泰勒一阶公式，得到式(3-23)，再将式(3-22)带入并化简得到式(3-24)，其中 dx/dt 、 dy/dt 分别表示像素在 x 轴和 y 轴方向的移动速度， $\partial I/\partial x$ 、 $\partial I/\partial y$ 表示图像在 p 点处 x 轴和 y 轴方向的梯度。 $\partial I/\partial t$ 表示灰度对时间的变化量。组合成向量的表示形式如式(3-25)所示，这样就可以根据特征点在连续图像中灰度的变化，来估计特征点在连续图像中的运动，从而找到一个特征点在连续图像中的匹配关系。

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (3-22)$$

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + H.O.T \quad (3-23)$$

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} = -\frac{\partial I}{\partial t} \quad (3-24)$$

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = -I_t \quad (3-25)$$

窗口同向假设表示图像的某一个窗口内(例如 15×15 像素)，所有的像素点在连续两帧图像之间具有相同方向的运动。那么对于一个 $w \times w$ 大小的窗口，共有 w^2 个像素点，这些像素点具有相同的运动方程，因此可以构建 w^2 个约束方程，如式(3-26)、(3-27)所示。通过求解方程组，可以得到当前特征点的运动 v_x 、 v_y ，从而在下一帧图像中找到特征点所匹配的位置。如图 3-7 所示，表示使用 Lucas-Kanade 光流法对特征点的追踪过程。

$$\begin{bmatrix} I_x & I_y \end{bmatrix}_i \begin{bmatrix} v_x \\ v_y \end{bmatrix} = -I_{ti}, \quad i = 1, 2, \dots, w^2 \quad (3-26)$$

$$\begin{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix}_1 \\ \begin{bmatrix} I_x & I_y \end{bmatrix}_2 \\ \vdots \\ \begin{bmatrix} I_x & I_y \end{bmatrix}_{w^2} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = -\begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{tw^2} \end{bmatrix} \quad (3-27)$$



图 3-7 基于 Lucas-Kanade 光流法的特征点匹配

在实际的应用过程中，某些时刻由于相机的大幅度运动或者剧烈的旋转，会导致连续图像的灰度产生巨大的变化，使得光流追踪的假设条件不一定满足，所以这时候光流追踪就会出现追踪丢失和误匹配的情形。如图 3-8 所示，为 KITTI 数据集中汽车经过十字路口转弯的图片序列，直接使用 LK 光流追踪会出现多组特征点的误匹配，以及许多特征点的追踪丢失的情况。尤其误匹配的特征点，对后端的非线性优化会造成严重的影响。



图 3-8 存在多组特征点误匹配的光流追踪

因此在实际系统中使用 KL 光流追踪算法时，需要设计一些消除误匹配特征点的方法。常用的方法有：采用 Mask 消除一些边缘畸变较大的点、采用 RANSAC

算法减少不符合模型的 outlier 点, 以及可以采用光流的逆向追踪的方法。综合使用这些方法, 使得图像中的误匹配尽可能降低, 图 3-9 为同一场景下对 KL 光流追踪, 采取消除误匹配方法之后的实验结果, 发现图中仍有少量的误匹配, 但是总体的匹配准确度已经有大幅度的提升。



图 3-9 消除误匹配特征点的光流追踪

最后, 在得到连续图像的匹配特征点之后, 需要将特征点在像素坐标系下的平面坐标转化为在图像坐标系下空间坐标, 以便后端系统的使用。在视觉 SLAM 系统中, 常用的相机类型有针孔相机(Pinhole)、鱼眼相机(Fisheye)、全景相机(Omnidirectional)等多种不同的类型, 针对不同的相机类型有不同的相机模型来描述三维空间物体到相机成像平面的投影映射关系和图像畸变的矫正^[50]。其中应用最为广泛的为针孔相机模型, 它使用 $[f_x, f_y, c_x, c_y]$ 来描述相机坐标系下点 $P(X, Y, Z)$ 投影到像素平面坐标 $f(u, v)$ 的过程, 具体表示形式如公式(3-28)所示, 其中 K 称为相机内参矩阵。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} = K \frac{1}{Z} P \quad (3-28)$$

3.2.1.2. 系统初始化

系统初始化的主要目的是为后端非线性优化提供一组初始的估计值, 然后系统才能构建残差约束, 并根据估计的初值不断地迭代直至收敛。所以需要通过对多视图几何的原理先估算出系统的待优化状态参数的初值, 其中主要包括三个部分:

- (1) IMU 传感器与双目相机的相对旋转的初值;
- (2) 初始滑动窗口中待优化变量的初始估计值;
- (3) 当前帧的位姿与路标点的空间位置的初始估计值。

为了将 IMU 与双目相机的测量结果建立联系并构成一个整体, 需要标定 IMU 传感器与双目相机传感器的相对位姿关系, 即 IMU 坐标系与相机坐标系的相对外参。IMU 与相机之间的相对外参包括旋转 R_{bc} / q_{bc} (旋转向量或者四元数表示形式) 和平移 t_{bc} , 本文中采用的双目相机中集成有 IMU 芯片, 由于相机的基线比较短, 所以两者之间的平移非常小, 假定平移的初值 $t_{bc} = (0,0,0)^T$, 而旋转的初始值由于 IMU 传感器的放置位置可能与相机之间存在较大的变化。所以首先需要估计 IMU 与双目相机之间的相对旋转 R_{bc} / q_{bc} 。如图 3-10 所示, 估计相对旋转的方法是利用连续两帧图像之间的匹配特征点对和图像间所对应的 IMU 预积分增量, 根据特征点对的对极几何约束求解视觉旋转 $q_{c_k c_{k+1}}$, 根据 IMU 预积分增量求解 IMU 旋转 $q_{b_k b_{k+1}}$, 并利用两者的几何关系建立约束方程, 求解相机与 IMU 坐标系之间的相对旋转 q_{bc} 。

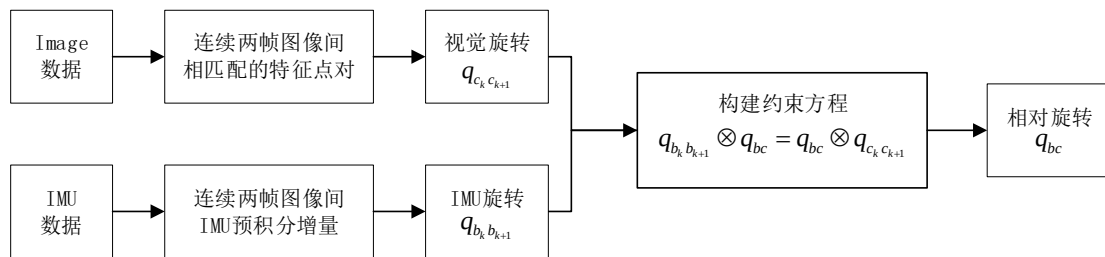


图 3-10 IMU 与相机相对旋转求解过程

对于滑动窗口中初始变量的估计, 是指在系统进行非线性优化阶段之前, 利用预处理阶段的特征点提取与追踪和双目视觉的几何约束, 求解当前滑动窗口中每一帧图像的位姿和每一帧图像中特征点的空间三维位置。在求解特征点的空间三维位置时, 利用双目视觉的三角测量的方法, 在求解每一帧图像的位姿时, 利用 PnP(Perspective-n-Point)方法。

对于当前帧状态变量初值的估计, 指在完成系统初始化之后, 利用当前帧输入的 IMU 数据和图像数据对当前的状态进行预测。首先利用 IMU 积分得到当前 IMU body 位姿、然后根据相对外参得到当前帧图像的位姿, 最后利用三角化求解出当前帧追踪的特征点在相机坐标系下的三维空间位置。

3.2.1.3. 后端非线性优化

在视觉里程计的前端中, 已经通过 IMU 积分得到每一帧图像和 IMU 传感器

的位姿，并利用三角化计算出特征点的空间位置。但是由于传感器噪声的存在，IMU 传感器的运动方程和相机的观测方程不能精确成立，所以需要从带有噪声的数据中恢复出运动的状态。而后端的非线性优化的功能即利用前端的预测与每一个时刻的观测数据建立约束方程，通过最小二乘方法来处理噪声数据，并从带噪声的数据中恢复出更加准确的状态信息。如式(3-29)所示，系统优化的状态变量包含相机与 IMU 的相对外参 $\mathbf{x}_{bc}$ 、路标点的逆深度 λ_i 、以及每一帧图像所对应的 IMU body 的位置、速度、旋转、加速度和陀螺仪的测量偏置 $[\mathbf{p}_{wb_k}, \mathbf{v}_k^w, \mathbf{q}_{wb_k}, \mathbf{b}_k^a, \mathbf{b}_k^g]$ 。并将所有的优化变量被表示成一个集合 $\mathbf{X}$ 。

$$\begin{aligned}\mathbf{X} &= [\mathbf{x}_0, \mathbf{x}_1 \dots \mathbf{x}_n, \mathbf{x}_{bc}, \lambda_0, \lambda_1, \dots \lambda_m] \\ \mathbf{x}_k &= [\mathbf{p}_{wb_k}, \mathbf{v}_k^w, \mathbf{q}_{wb_k}, \mathbf{b}_k^a, \mathbf{b}_k^g], \quad k \in [0, n] \\ \mathbf{x}_{bc} &= [\mathbf{p}_{bc}, \mathbf{q}_{bc}]\end{aligned}\quad (3-29)$$

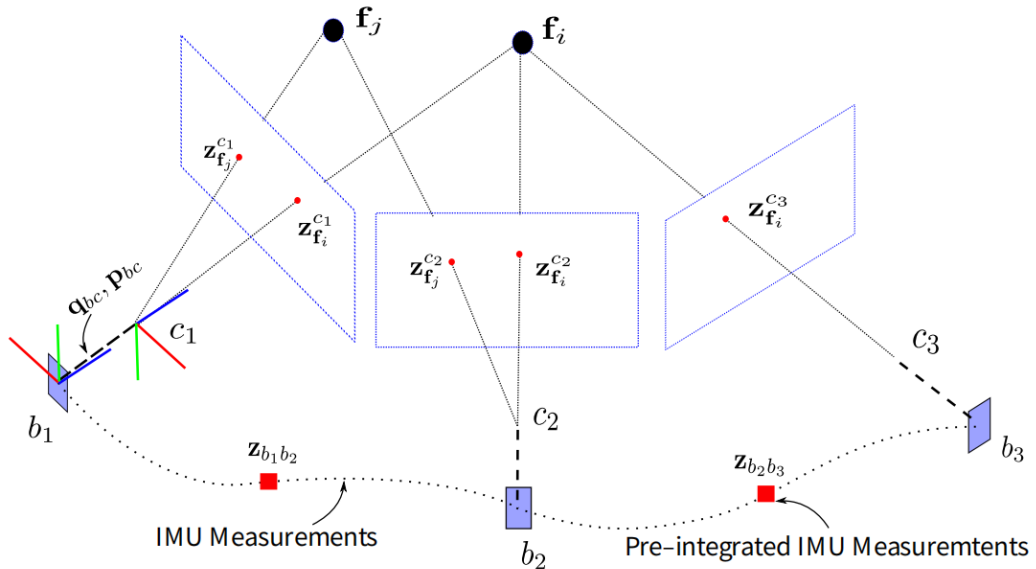


图 3-11 视觉与惯导相融合的位姿图优化

图 3-11 所示为视觉与惯导紧耦合的非线性图优化的描述示意图，每一帧图像包含相机坐标 c_i 和与其相对应的 IMU body 坐标 b_i ，连续两帧图像之间有多帧 IMU 测量，将两帧图像间的 IMU 测量进行预积分得到预积分增量 $z_{b_i b_j}$ 。并且连续的图像能够观测到相同的路标点，例如图中路标点 f_i 被 c_1 、 c_2 、 c_3 观测到，并在各自的图像中成像为像素 $z_{f_i}^{c1}$ 、 $z_{f_i}^{c2}$ 、 $z_{f_i}^{c3}$ ，路标点 f_j 被 c_1 、 c_2 观测到，并在各自的图像中成像为像素 $z_{f_j}^{c1}$ 、 $z_{f_j}^{c2}$ 。图中的顶点表示待优化变量，边表示约束条件，通过这些顶点和边可以构建视觉重投影误差和 IMU 积分误差，利用 Gauss-Newton 迭代法^[51]、Levenberg-Marquardt 迭代法^[52]等优化算法求解，使得误差最小化从而

求得状态变量的局部最优解。下文将分别阐述视觉重投影误差的构建、IMU 预积分误差的构建以及非线性优化问题的求解。

视觉重投影误差指一个特征点在 c_i 图像中被观测到，并且在后序的 c_j 图像中仍被追踪，那么利用 i 、 j 时刻的相对位姿将特征点从 c_i 相机平面投影到 c_j 相机平面(如式 3-30 所示)，则在重新投影后的归一化相机平面上，特征点的重投影值和观测值的差即为重投影误差(如式 3-31 所示)。

$$\begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \\ 1 \end{bmatrix} = \mathbf{T}_{bc}^{-1} \mathbf{T}_{wb_j}^{-1} \mathbf{T}_{wb_i} \mathbf{T}_{bc} \begin{bmatrix} \frac{1}{\lambda} u_{c_i} \\ \frac{1}{\lambda} v_{c_i} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix} \quad (3-30)$$

$$\mathbf{r}_c = \begin{bmatrix} \frac{x_{c_j}}{z_{c_j}} - u_{c_j} \\ \frac{y_{c_j}}{z_{c_j}} - v_{c_j} \end{bmatrix} \quad (3-31)$$

在公式(3-30)、(3-31)中， (u_{c_i}, v_{c_i}) 为特征点在 c_i 归一化相机平面的观测坐标， $\lambda = 1/z_{c_i}$ 为特征点在 c_i 相机平面的逆深度， $\mathbf{T}_{wb_i}$ 、 $\mathbf{T}_{wb_j}$ 变换矩阵表示 c_i 、 c_j 在世界坐标系下的位姿， $\mathbf{T}_{bc}$ 变换矩阵表示相机坐标系到 IMU 坐标系的变换。 $(x_{c_j}, y_{c_j}, z_{c_j})$ 表示特征点投影到 c_j 相机平面的坐标，将其深度归一化得到 c_j 归一化相机平面的坐标 $(\frac{x_{c_j}}{z_{c_j}}, \frac{y_{c_j}}{z_{c_j}})$ ，而 (u_{c_j}, v_{c_j}) 为特征点在 c_j 归一化相机平面的观测坐标，两者的差即为特征点的重投影误差 $\mathbf{r}$ 。将变换矩阵 $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$ 拆分成旋转矩阵 $\mathbf{R}$ 和平移矩阵 $\mathbf{t}$ ，则公式(3-30)展开形式为公式(3-32)。

$$\mathbf{f}_{c_j} = \begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \end{bmatrix} = \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} \mathbf{R}_{bc} \frac{1}{\lambda} \begin{bmatrix} u_{c_i} \\ v_{c_i} \\ 1 \end{bmatrix} + \mathbf{R}_{bc}^T (\mathbf{R}_{wb_j}^T ((\mathbf{R}_{wb_i} \mathbf{p}_{bc} + \mathbf{p}_{wb_i}) - \mathbf{p}_{wb_j}) - \mathbf{p}_{bc}) \quad (3-32)$$

在求解非线性优化问题时，需计算重投影误差对于待优化变量的导数 $\mathbf{J}_c$ ，称之为雅克比矩阵(Jacobian Matrix)。公式(3-33)表示误差对于所有待优化变量的导数，即 $\mathbf{J}_c$ 表达式。

$$\mathbf{J}_c = \begin{bmatrix} \frac{\partial \mathbf{r}_c}{\partial [\mathbf{p}_{wb_i}, \boldsymbol{\theta}_{wb_i}]} & \frac{\partial \mathbf{r}_c}{\partial [\mathbf{p}_{wb_j}, \boldsymbol{\theta}_{wb_j}]} & \frac{\partial \mathbf{r}_c}{\partial [\mathbf{p}_{bc}, \boldsymbol{\theta}_{bc}]} & \frac{\partial \mathbf{r}_c}{\partial \lambda} \end{bmatrix} \quad (3-33)$$

IMU 预积分误差是根据 IMU 的观测值和估计值求差得到的。观测值为连续两帧图像 i 、 j 间的 IMU 的位置、速度、旋转以及测量偏置的预积分增量，而估计值为 i 、 j 帧 IMU body 的状态量的相对变化，如式(3-34)，两者的差即为 IMU 预积分误差。

$$\mathbf{r}_B = \begin{bmatrix} \mathbf{r}_p \\ \mathbf{r}_q \\ \mathbf{r}_v \\ \mathbf{r}_{ba} \\ \mathbf{r}_{bg} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{wb_i}^T (\mathbf{p}_{wb_j} - \mathbf{p}_{wb_i} - \mathbf{v}_i^w \Delta t + \frac{1}{2} \mathbf{g}^w \Delta t^2) - \mathbf{a}_{b_i b_j} \\ 2[\gamma_{b_i b_j}^{-1} \otimes (\mathbf{q}_{wb_i}^{-1} \otimes \mathbf{q}_{wb_j}^{-1})]_{xyz} \\ \mathbf{R}_{wb_i}^T (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t) - \mathbf{\beta}_{b_i b_j} \\ \mathbf{b}_j^a - \mathbf{b}_i^a \\ \mathbf{b}_j^g - \mathbf{b}_i^g \end{bmatrix} \quad (3-34)$$

IMU 预积分误差 $\mathbf{r}_B = (\mathbf{r}_p \ \mathbf{r}_q \ \mathbf{r}_v \ \mathbf{r}_{ba} \ \mathbf{r}_{bg})^T$ 包含位置、旋转、速度、角速度和加速度偏置 5 个向量，共 15 维，而每个误差向量都会作用于 i 、 j 时刻 IMU body 的状态向量，将其分为四组，分别是 i 时刻的位姿 $[\mathbf{p}_{wb_i} \ \mathbf{q}_{wb_i}]$ 、 i 时刻的速度和 IMU 偏置 $[\mathbf{v}_i^w \ \mathbf{b}_i^a \ \mathbf{b}_i^g]$ 、 j 时刻的位姿 $[\mathbf{p}_{wb_j} \ \mathbf{q}_{wb_j}]$ 、以及 j 时刻的速度和 IMU 偏置 $[\mathbf{v}_j^w \ \mathbf{b}_j^a \ \mathbf{b}_j^g]$ 。则 IMU 预积分误差对于优化变量的导数，即雅克比矩阵 $\mathbf{J}_B$ 如下式(3-35)所示。

$$\mathbf{J}_B = \begin{bmatrix} \frac{\partial \mathbf{r}_B}{\partial [\mathbf{p}_{wb_i}, \mathbf{q}_{wb_i}]} & \frac{\partial \mathbf{r}_B}{\partial [\mathbf{v}_i^w, \mathbf{b}_i^a, \mathbf{b}_i^g]} & \frac{\partial \mathbf{r}_B}{\partial [\mathbf{p}_{wb_j}, \mathbf{q}_{wb_j}]} & \frac{\partial \mathbf{r}_B}{\partial [\mathbf{v}_j^w, \mathbf{b}_j^a, \mathbf{b}_j^g]} \end{bmatrix} \quad (3-35)$$

根据视觉约束，一个特征点 f_l 被图像 c_j 追踪到之后，会建立一个视觉重投影误差 $\mathbf{r}_c(f_l, c_j)$ 。在 IMU 约束中，连续两帧图像 c_i 、 c_j 间的预积分增量 $\mathbf{z}_{b_i b_j}$ (也称为 IMU 观测)，与 c_i 、 c_j 两帧图像对应的 IMU body 状态量 $\mathbf{x}_i$ 、 $\mathbf{x}_j$ 可建立一个 IMU 预积分误差 $\mathbf{r}_B(\mathbf{z}_{b_i b_j}, \mathbf{x}_i, \mathbf{x}_j)$ 。当系统随着时间不断运行时，图像序列随着时间不断累积，前端追踪的特征点的数量也不断增加，所以系统会随时间构建大量的视觉重投影误差和 IMU 预积分误差。滑动窗口也随着时间不断向后移动，在固定一些优化变量的同时，也会引入一些先验信息，产生先验误差约束。对当前滑动窗口中所有的误差项加权求和，即得到系统残差的损失函数 $F(\chi)$ ，如式(3-36)所示。系统优化的目标是找到状态向量 χ^* ，使得损失函数 $F(\chi)$ 取得局部最小值，那么状态向量 χ^* 即为当前滑动窗口中待优化变量的最优解。

$$\min_{\chi} \left\{ \|\mathbf{r}_p - \mathbf{J}_p \chi\| + \sum_{(l,j) \in C} \rho(\|\mathbf{r}_c(\mathbf{z}_{f_l}^{c_j} - \chi)\|_{P_{f_l}^{c_j}}^2) + \sum_{(i,j) \in B} \|\mathbf{r}_B(\mathbf{z}_{b_i b_j} - \chi)\|_{P_{b_i b_j}^{b_i}}^2 \right\} \quad (3-36)$$

在优化的求解过程中，将 VIO-SLAM 状态量的优化问题建模成一个非线性最小二乘求解问题。如式(3-37)所示，其中 f_i 为残差函数，需要找到一个 n 维向量 $\mathbf{x}^* \in \mathbf{R}^n$ ，使得损失函数 $F(\mathbf{x})$ 取得局部最小值，即对于任意 $\|\mathbf{x} - \mathbf{x}^*\| < \delta$ ，有 $F(\mathbf{x}^*) \leq F(\mathbf{x})$ 。

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^m (f_i(\mathbf{x}))^2 \quad (3-37)$$

令 $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}) \cdots f_m(\mathbf{x})]^T$ ，则损失函数的向量表达形式如公式(3-38)所

示，每个残差函数对于优化向量的导数如公式(3-39)所示。

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^m (f_i(\mathbf{x}))^2 = \frac{1}{2} \mathbf{f}^T(\mathbf{x}) \mathbf{f}(\mathbf{x}) \quad (3-38)$$

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \left[\frac{\partial f_1(\mathbf{x})}{\partial \mathbf{x}}, \frac{\partial f_2(\mathbf{x})}{\partial \mathbf{x}}, \dots, \frac{\partial f_n(\mathbf{x})}{\partial \mathbf{x}} \right] \quad (3-39)$$

在式(3-40)中，对残差函数 $\mathbf{f}(\mathbf{x})$ 进行一阶泰勒近似，并带入损失函数 $F(\mathbf{x})$ 中。得到损失函数的近似，如式(3-41)所示。

$$\mathbf{f}(\mathbf{x} + \Delta \mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J} \Delta \mathbf{x} \quad (3-40)$$

$$\begin{aligned} F(\mathbf{x} + \Delta \mathbf{x}) &= \frac{1}{2} \mathbf{f}^T(\mathbf{x} + \Delta \mathbf{x}) \mathbf{f}(\mathbf{x} + \Delta \mathbf{x}) \\ &\approx \frac{1}{2} \mathbf{f}^T \mathbf{f} + \Delta \mathbf{x}^T \mathbf{J}^T \mathbf{f} + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{J}^T \mathbf{J} \Delta \mathbf{x} \\ &= F(\mathbf{x}) + \Delta \mathbf{x}^T \mathbf{J}^T \mathbf{f} + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{J}^T \mathbf{J} \Delta \mathbf{x} \end{aligned} \quad (3-41)$$

在式(3-41)中，损失函数近似成一个对于 $\Delta \mathbf{x}$ 的二次函数，并且当雅克比矩阵 $\mathbf{J}$ 为满秩矩阵时， $\mathbf{J}^T \mathbf{J}$ 为正定矩阵，损失函数有最小值。因此对式(3-41)求导，令其导数等于零，得到式(3-42)，即高斯-牛顿法(Gauss-Newton Method)的迭代公式。

$$\mathbf{J}^T \mathbf{J} \Delta \mathbf{x} = -\mathbf{J}^T \mathbf{f} \quad (3-42)$$

在公式(3-36)中构建了本文 VIO-SLAM 系统的非线性最小二乘模型的损失函数，并推导了每一项视觉重投影残差函数 r_{C_i} 的构建，和残差函数对于优化变量的导数 $\mathbf{J}_{C_i}$ ，以及每一项 IMU 预积分残差函数 r_{B_i} 的构建和残差函数对于优化变量的导数 $\mathbf{J}_{B_i}$ 。最后利用高斯-牛顿法不断迭代 $\Delta \chi$ ，并更新状态向量 $\chi \leftarrow \chi + \Delta \chi$ ，使得损失函数取得最小值，即可得到当前窗口内所有状态量的最优值 χ^* 。

3.2.2. 多传感器融合

双目视觉里程计可以计算出无人机运动过程中相对于起始点的位姿变化，但不具有全局一致性。随着系统的运行，由于旋转和平移的误差会导致位姿的累积漂移，因此为了设计更加鲁棒的无人机位姿估计系统，本文进一步融合了视觉里程计估计的位姿以及 GPS 和 IMU 传感器的信息，通过多传感器融合实现无人机位姿估计。本文基于扩展卡尔曼滤波器将 IMU 数据与视觉里程计和 GPS 数据进行松耦合，以 IMU 数据进行状态预测、视觉里程计数据和 GPS 数据对预测量进行更新。下面具体阐述本文基于 EKF 多传感器融合的无人机位姿估计设计方法。

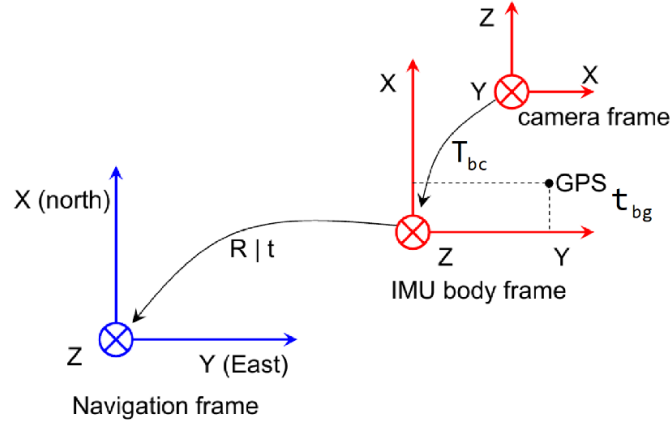


图 3-12 多传感器融合坐标变换图

如图 3-12 所示，本文以 NED(North-East-Down)坐标系为基准，作为无人机定位、导航的全局坐标系，以 IMU body 作为无人机的机体坐标系，则无人机的当前位姿可以用 IMU body 到 Navigation frame 的旋转矩阵 R 和平移向量 t 表示。无人机所挂载的双目相机到 IMU 的相对外参用变换矩阵 T_{bc} 表示。IMU 与 GPS 方向相同，所以只存在相对平移，用 t_{bg} 表示。

首先，在 EKF 融合的过程中，以 IMU 传感器测量值的积分作为无人机状态量的预测，则在 IMU body 坐标系下，定义 EKF 系统的状态量为：

$$\mathbf{X} = (\mathbf{p}, \delta\boldsymbol{\theta}, \mathbf{v}_b, \delta\mathbf{b}_g, \delta\mathbf{b}_a)^T \quad (3-43)$$

在式(3-43)中，状态量 $\mathbf{X}$ 包含导航坐标系下的位置 $\mathbf{p}$ ，机体坐标系下的旋转向量误差 $\delta\boldsymbol{\theta}$ 、速度 $\mathbf{v}_b$ 以及 IMU 陀螺仪和加速度偏置误差 $\delta\mathbf{b}_g$ 、 $\delta\mathbf{b}_a$ 。把状态量对时间求导，得 $\dot{\mathbf{X}}$ ：

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{R} \exp([\delta\boldsymbol{\theta}]^\wedge) \mathbf{v}_b \\ \dot{\delta\boldsymbol{\theta}} &= \exp([\delta\boldsymbol{\theta}]^\wedge) (\mathbf{w} - \delta\mathbf{b}_g - \mathbf{n}_g) \\ \dot{\mathbf{v}}_b &= -[(\mathbf{w} - \delta\mathbf{b}_g - \mathbf{n}_g)^\wedge \mathbf{v}_b + (\mathbf{R} \exp([\delta\boldsymbol{\theta}]^\wedge))^T \mathbf{g} + \mathbf{a} - \delta\mathbf{b}_a - \mathbf{n}_a] \\ \dot{\delta\mathbf{b}}_g &= \mathbf{r}_g \\ \dot{\delta\mathbf{b}}_a &= \mathbf{r}_a \end{aligned} \quad (3-44)$$

在式(3-44)中， $\mathbf{a}$ 、 $\mathbf{w}$ 表示 IMU 传感器的加速度和旋转角速度测量值， $\delta\boldsymbol{\theta}^\wedge$ 表示旋转向量误差的反对称矩阵， $\mathbf{n}_a$ 、 $\mathbf{n}_g$ 为 IMU 测量噪声，服从于零均值高斯分布。 $\mathbf{b}_a$ 、 $\mathbf{b}_g$ 为 IMU 测量偏置，服从于随机游走噪声分布，其导数服从于零均值高斯分布。则 EKF 系统的状态量的输入噪声表示为：

$$\mathbf{W} = (\mathbf{n}_g, \mathbf{n}_a, \mathbf{b}_g, \mathbf{b}_a) \quad (3-45)$$

根据公式(3-44)和(3-45)，EKF 状态量的雅克比矩阵如式(3-46)所示。

$$\begin{aligned} \mathbf{J}_X &= \frac{\partial \mathbf{X}'}{\partial \mathbf{X}} dt + \mathbf{I}_{15 \times 15} \\ \mathbf{J}_W &= \frac{\partial \mathbf{X}'}{\partial \mathbf{W}} dt \end{aligned} \quad (3-46)$$

在本文基于多传感器融合的 EKF 系统中, 包含了视觉里程计、GPS 和 IMU 传感器三种信息。使用 IMU 进行状态量的预测, 使用视觉里程计和 GPS 信息进行状态更新, 而 GPS 提供在 NED 坐标系下的绝对位置和速度的测量信息, 视觉里程计提供相对位姿信息, 为了在状态更新能够同时处理相对测量信息和绝对测量信息, 本文采用了 stochastic cloning^[53]方法, 在原来的状态量的基础上添加了 delayed pose $\mathbf{p}_l, \boldsymbol{\theta}_l$ 。

$$\tilde{\mathbf{X}} = (\mathbf{X}^T, \mathbf{p}_l, \boldsymbol{\theta}_l)^T \quad (3-47)$$

在公式(3-47)中, 状态量的增广矩阵为 $\tilde{\mathbf{X}}$, 其中 delayed pose $\mathbf{p}_l, \boldsymbol{\theta}_l$ 在 IMU 预测阶段保持不变, 即表示 $\mathbf{p}_l, \boldsymbol{\theta}_l$ 对时间的导数为零, 因此状态量 $\tilde{\mathbf{X}}$ 的雅克比矩阵为:

$$\tilde{\mathbf{J}}_X = \begin{pmatrix} \mathbf{J}_X & \mathbf{0}_{15 \times 6} \\ \mathbf{0}_{6 \times 15} & \mathbf{I}_{6 \times 6} \end{pmatrix} \quad (3-48)$$

$$\tilde{\mathbf{J}}_W = \begin{pmatrix} \mathbf{J}_W \\ \mathbf{0}_{6 \times 12} \end{pmatrix} \quad (3-49)$$

根据状态量的增广矩阵为 $\tilde{\mathbf{X}}$, 最终 EKF 系统的在预测阶段的协方差传递函数表示为:

$$\tilde{\mathbf{P}}(k+1|k) = \tilde{\mathbf{J}}_X \tilde{\mathbf{P}}(k) \tilde{\mathbf{J}}_X^T + \tilde{\mathbf{J}}_W \tilde{\mathbf{Q}}(k) \tilde{\mathbf{J}}_W^T \quad (3-50)$$

在 EKF 状态更新阶段, 首先考虑 VIO 融合的视觉里程计所提供的 6-DoF 相对位姿。根据视觉里程计的观测, 可以计算出连续两帧图像的相对位姿和旋转为 $\Delta \mathbf{p}_{vo}$ 、 $\Delta \boldsymbol{\theta}_{vo}$ 。同时, 根据 EKF 状态的预测, 可计算出连续两帧图像的位姿和旋转的预测增量:

$$\begin{aligned} \Delta \mathbf{p} &= \exp(-[\boldsymbol{\theta}_l]^\wedge) \mathbf{R}_l^T (\mathbf{p} - \mathbf{p}_l) \\ \Delta \boldsymbol{\theta} &= -[\boldsymbol{\theta}_l]^\wedge \mathbf{R}_l^T \mathbf{R} \exp(-[\boldsymbol{\theta}]^\wedge) \end{aligned} \quad (3-51)$$

因此, 视觉里程计的实际观测与预测增量的差值如式(3-52)所示, 视觉里程计的观测方程雅克比矩阵如式(3-53)所示。

$$\mathbf{r} = \begin{pmatrix} \Delta \mathbf{p}_{vo} - \Delta \mathbf{p} \\ \Delta \boldsymbol{\theta}_{vo} - \Delta \boldsymbol{\theta} \end{pmatrix} \quad (3-52)$$

$$\mathbf{H}_{vo} = \begin{pmatrix} \mathbf{R}_l^T & \mathbf{0} & -\mathbf{R}_l^T & [\mathbf{R}_l^T (\mathbf{p} - \mathbf{p}_l)]^\wedge \\ \mathbf{0} & \mathbf{R}_l^T \mathbf{R} & \mathbf{0} & -\mathbf{I} \end{pmatrix} \quad (3-53)$$

根据 EKF 状态更新公式, 首先利用视觉里程计信息对预测的状态量进行更

新，如公式(3-54)所示，其中 EKF 增益为 $\mathbf{K}$ ，状态量更新为 $\tilde{\mathbf{X}}(k+1)$ ，协方差更新为 $\tilde{\mathbf{P}}(k+1)$ 。

$$\begin{aligned}\mathbf{K} &= \tilde{\mathbf{P}}(k+1|k)\mathbf{H}_{vo}^T ((\tilde{\mathbf{P}}(k+1|k)\mathbf{H}_{vo}^T + \mathbf{R})^{-1}) \\ \tilde{\mathbf{X}}(k+1) &= \tilde{\mathbf{X}}(k) + \mathbf{K}\mathbf{r} \\ \tilde{\mathbf{P}}(k+1) &= (\mathbf{I} - \mathbf{K}\mathbf{H}_{vo})\tilde{\mathbf{P}}(k+1|k)(\mathbf{I} - \mathbf{K}\mathbf{H}_{vo})^T + \mathbf{K}\mathbf{R}\mathbf{K}^T\end{aligned}\quad (3-54)$$

在本文的无人机系统中，IMU 传感器的更新频率为 200Hz，基于双目视觉的状态估计输出的视觉里程计帧率为 10Hz，而 GPS 信号的更新频率为 5Hz。因此根据视觉里程计对 IMU 预测的状态量进行状态更新之后，还需要利用 GPS 信息对状态量再次进行更新，GPS 测量值可以观测到 NED 坐标系下的无人机位置和速度信息，因此可以对状态量进行全局一致性的修正。公式(3-55)和(3-56)则为 GPS 测量的观测值和观测方程的雅克比矩阵。

$$\mathbf{Z}_{gps} = \begin{pmatrix} \mathbf{p} + \mathbf{R} \exp([\delta\boldsymbol{\theta}]^\wedge) \mathbf{t}_{bg} \\ \mathbf{R} \exp([\delta\boldsymbol{\theta}]^\wedge) (\mathbf{v}_b + [\mathbf{w} - \delta\mathbf{b}_g]^\wedge \mathbf{t}_{bg}) \end{pmatrix} \quad (3-55)$$

$$\mathbf{H}_{gps} = \begin{pmatrix} \mathbf{I} & -\mathbf{R}[\mathbf{t}_{bg}]^\wedge & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}[\mathbf{v}_b]^\wedge & \mathbf{R} & [\mathbf{t}_{bg}]^\wedge & \mathbf{0} \end{pmatrix} \quad (3-56)$$

GPS 状态更新如公式(3-57)所示，其中 $\mathbf{y}$ 为 GPS 观测数据与当前预测的状态量的误差， $\mathbf{K}$ 为 GPS 更新的卡尔曼增益， $\mathbf{X}(k+1)$ 为 GPS 更新后的状态量， $\mathbf{P}(k+1)$ 为更新状态量之后的协方差。

$$\begin{aligned}\mathbf{y} &= \mathbf{Z}_{gps} - \mathbf{H}_{gps}\mathbf{X} \\ \mathbf{K} &= \mathbf{P}(k+1|k)\mathbf{H}_{gps}^T ((\mathbf{P}(k+1|k)\mathbf{H}_{gps}^T + \mathbf{R})^{-1}) \\ \mathbf{X}(k+1) &= \mathbf{X}(k) + \mathbf{K}\mathbf{y} \\ \mathbf{P}(k+1) &= (\mathbf{I} - \mathbf{K}\mathbf{H}_{gps})\mathbf{P}(k+1|k)(\mathbf{I} - \mathbf{K}\mathbf{H}_{gps})^T + \mathbf{K}\mathbf{R}\mathbf{K}^T\end{aligned}\quad (3-57)$$

在本文的多传感融合框架中，以第一帧接收到的 GPS 信号作为 Navigation frame 的参考坐标，以视觉里程计的初始化结果作为系统状态量的初值。在系统运行的过程中，以 IMU 测量值积分作为系统的状态预测值，并且以视觉里程计和 GPS 测量对预测值进行局部和全局的状态更新。在利用 GPS 进行状态量更新时，首先会根据当前环境下 GPS 信号质量来评估 GPS 测量数据的置信度，当搜索的卫星数量过少时，系统会拒绝接收此次 GPS 测量数据，舍弃此次 GPS 状态更新。此外，将更新后的 EKF 状态量作为视觉里程计的估计初值，来提高视觉里程计的优化精度，从而提升整个无人机系统的状态估计鲁棒性。

3.3.障碍物地图的构建

3.3.1.双目立体匹配

基于双目视觉的深度估计是利用双目立体匹配算法获取左、右两幅图像相同视点的视差，进而获取图像的深度图。如图 3-13 所示，双目视觉的深度估计的流程分四个阶段：准备阶段、立体校正、立体匹配、深度映射。

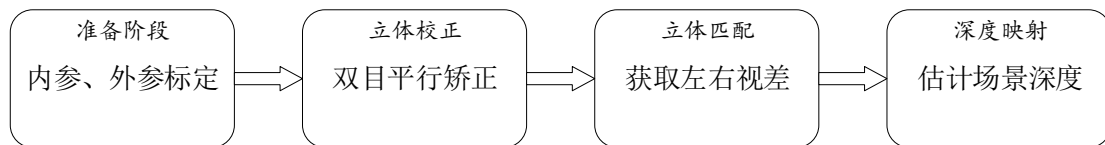


图 3-13 双目视觉深度估计的流程

准备阶段：首先进行相机内参标定和双目的相对外参标定。内参决定了每个视点在像素平面和相机归一化平面的映射关系，而双目摄像头的相对外参决定了同一时刻在左相机平面和右相机平面中相同视点的映射关系，所以标定相机的内参和相对外参是后序双目视觉深度估计的基础。相机内参标定和外参标定常用张氏棋盘格标定法^[54]。

立体校正：将同一时刻的左、右两幅图像进行立体图像对的极线校正，使得左、右图像行对准。通过这种方式可以将二维匹配搜索简化为一维搜索，提高匹配搜索的效率。

立体匹配：在立体校正之后，对左、右两幅图像进行立体匹配，并计算左、右图中相同视点的视差。常用的立体匹配算法有 SGBM(Semi-Global Block Matching)算法^[55]、BM(Block Matching)算法、SAD 算法^[56]。

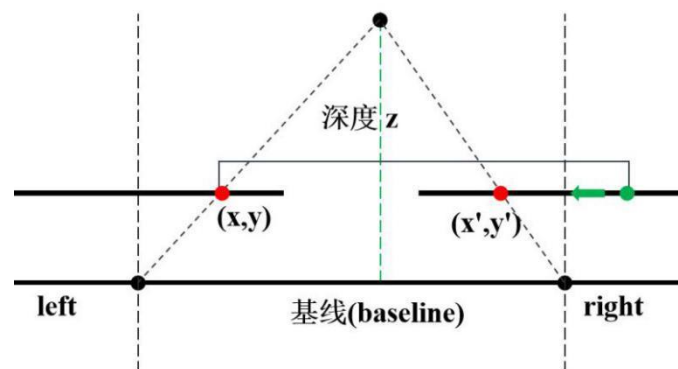


图 3-14 双目立体匹配与深度计算

深度映射：在视差图中，场景深度越近的物体视差越大，深度越远的物体视

差越小。如图 3-14 所示, 假设双目相机基线长度为 b , 焦距为 f , 空间中一点 P 在双目图像中成像 $P_L(x, y)$ 、 $P_R(x', y')$, 则 P 点在左右图中的视差 $d = x - x'$ 。根据几何关系可计算 P 点相对于相机平面的深度 z , 计算公式如式(3-58)、(3-59)所示。

$$\frac{z - f}{z} = \frac{b - (x - x')}{b} \quad (3-58)$$

$$z = \frac{fb}{d} \quad (3-59)$$

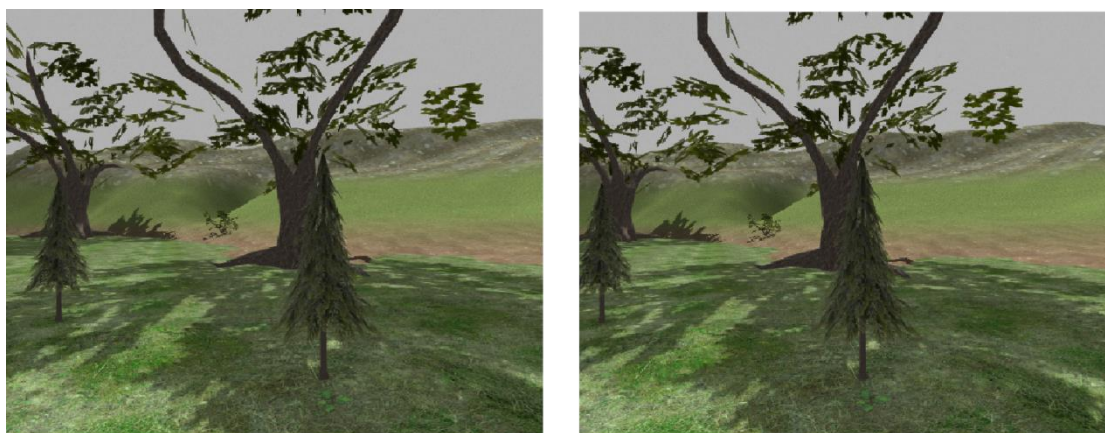


图 3-15 同一时刻左右双目图像

通过双目立体匹配能够计算出场景中相同视点在左右图中的视差, 进而估计出视点距离相机平面的深度。如图 3-16 所示, 即根据图 3-15 中左右图像所计算出的视差图。根据公式(3-59), 能够恢复出场景中视点的深度, 从而构建用来描述场景信息的障碍物点云地图, 如图 3-17 所示。

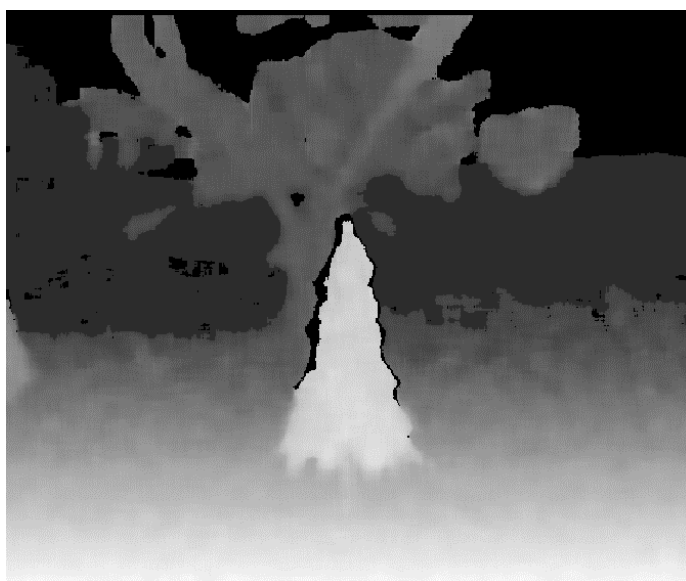


图 3-16 双目视差图

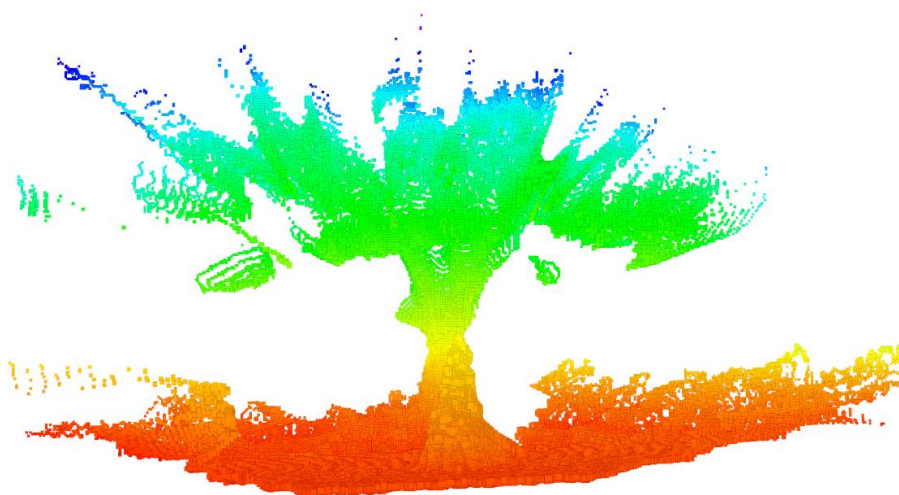


图 3-17 根据双目视觉映射的点云图

3.3.2.全局地图拼接

根据同一时刻的左右双目图像能够估计当前图像视野下的障碍物的深度信息。而在无人机自主感知与导航系统中，需要记录飞行器运行过程中所有扫描到的障碍物信息，所以用于无人机自主导航的障碍物地图需要满足以下性质：

- (1) 地图的稠密性；
- (2) 地图场景的多样性；
- (3) 地图的全局一致性。

首先，地图的稠密性是为了确保无人机进行路径搜索和轨迹优化的安全。因为后续的路径规划部分都是基于感知算法输出的状态估计和障碍物地图，稀疏的地图仅能保证规划的路径远离特征点而不能保证无人机不会碰撞到空间中其他障碍物，因此需要构建稠密的障碍物地图。本文采用 MYNT-EYE 双目立体深度相机，利用深度图构建稠密障碍物地图。在室内场景中使用双目摄像头和红外结构光传感器，利用相机传感器芯片实时计算场景的深度。在室外场景中，由于自然光的干扰，仅使用双目摄像头，利用双目立体匹配估计场景深度。

其次，构建的障碍物地图需要满足场景的多样性。自主无人机系统根据不同的需求通常应用在室内、室外等多种场景，在室内场景中，地图的尺度范围只有米级，而在室外场景中，地图的尺度范围通常能够达到公里级。为了满足地图场景尺度的多样性，同时提高地图的构建效率，本文采用构建障碍物表面地图的方式，利用场景的深度图和灰度图，提取障碍物表面深度信息来构建稠密地图。不仅满足场景的多样性，而且也提高了建图的效率。

最后，构建的障碍物地图需要保证全局一致性。在无人机运动的过程中，飞行器的位姿随着视觉 SLAM 的后端非线性优化不断调优，如果每次深度映射仅使用预测的飞行器位姿进行全局地图拼接，不仅无法保证地图的精度，而且无法保证地图的全局一致性。本文所采取的策略是将障碍物地图划分成多个子图(Submap)的形式，每一个子图依附其最近的视觉关键帧(Keyframe)，这样不仅障碍物点云会随着关键帧位姿的优化不断调优，而且当关键帧检测到回环时，能够实现地图的复用。采用这种策略不仅提高了地图拼接的精度，而且保证了全局地图的一致性。下图 3-19 为使用本文地图构建方法在苏黎世理工 ASL Cow and Lady 数据集^[57]上构建的全局障碍物地图。图 3-18 为场景的图片，图 3-19 为基于图像数据构建的障碍物点云图。



图 3-18 ASL Cow and Lady 数据集场景示意图

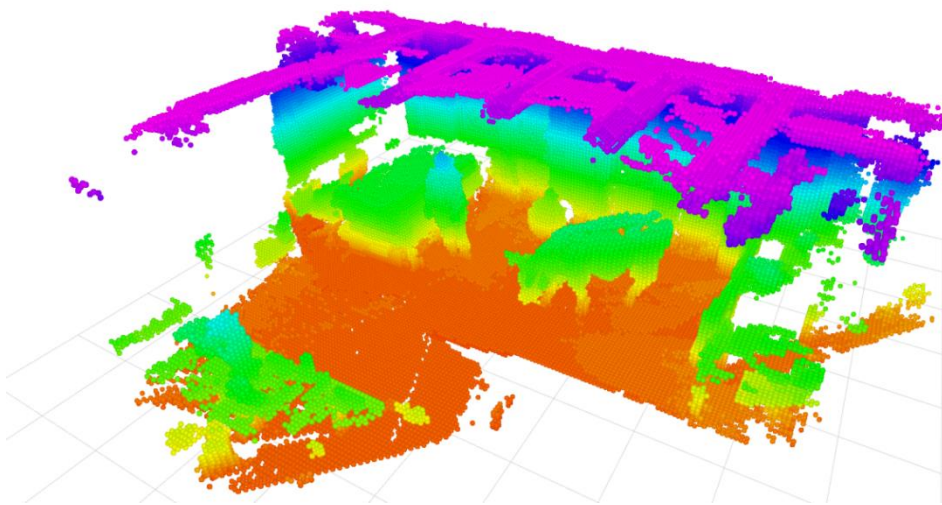


图 3-19 全局障碍物点云地图构建

3.3.3.ESDF 地图的转化

在本文的无人机自主感知与导航系统中,障碍物地图的主要作用是记录无人机在飞行过程中所探测到的三维空间障碍物点的位置,并服务于路径规划算法,从而实现无人机自主避障的功能。在三维空间中描述障碍物位置的地图有多种不同的表现形式,如图 3-20 所示,障碍物点云地图(Point Cloud Map)直接将检测到的障碍物点投影到三维空间内,而数据的组织形式为一个个离散的三维坐标点。三维空间栅格地图(Voxel Grids Map)在点云地图的基础之上,将三维空间划分为一个个单位大小的立体网格,当点云映射到对应网格则呈现占有状态(Occupied),否则为空闲状态(Free)。通过设置单位立体像素的分辨率可以控制栅格地图的精度(如 $10*10*10\text{cm}$ 每个 Voxel),并且通过下标索引的方式可以在 $O(1)$ 的时间复杂度下查询一个栅格的状态。为了更高效地检索地图中被占有栅格的位置,八叉树地图(Octo-map)将地图存储为八叉树的数据结构形式。

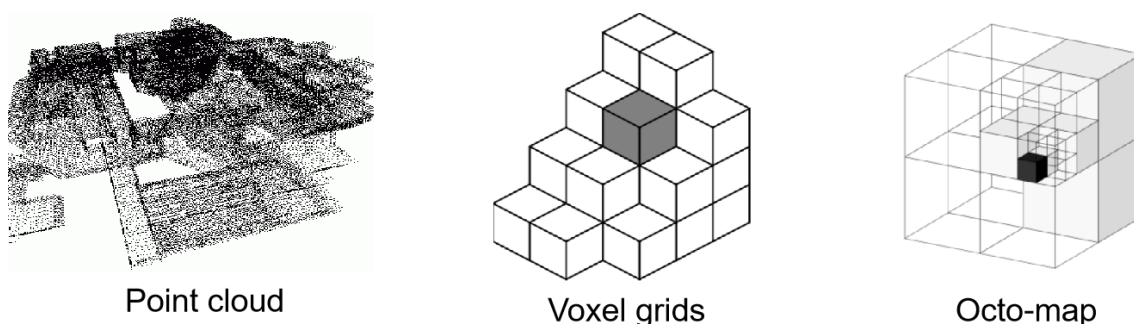


图 3-20 障碍物地图的多种表现形式

在无人机进行轨迹搜索和路径规划时,算法更加关注地图中没有被占用的空闲区域,这样可在空闲区域中搜索出无人机可通行的飞行路径。如果能够进一步表示空间中每一个立体栅格距离它最近的障碍物的距离,那么就可以利用距离的梯度信息,在轨迹搜索之上进一步优化出远离障碍物的安全轨迹,保证无人机更加安全地飞行。因此在三维栅格地图的基础上,可将地图转化为 ESDF(Euclidean Signed Distance Fields)地图形式,ESDF 称之为有符号欧式距离场,每一个栅格中不再表示被占用或者空闲的状态而是存储距离其最近的障碍物的欧式距离,如图 3-21 所示,表示将一块三维栅格地图转化为 ESDF 地图的示意图,图中灰色的栅格表示为被占用,说明有障碍物填充,白色的方格表示为空闲,而转化为对应的 ESDF 地图后,红色栅格表示距离障碍物最近的空间、蓝色栅格表示距离障碍物距离稍微远的空间。通过这种方式,当检索每个栅格时,便可以知道当前栅格距离其最近的障碍物有多远,为路径规划算法提供了重要的信息。

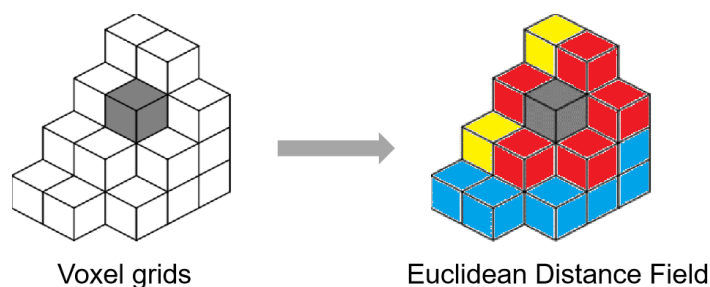


图 3-21 三维空间栅格地图与 ESDF 地图的对应示意图

图 3-22 描述如何将一个栅格地图转换为 ESDF 地图的过程，以二维栅格为例，首先在 ESDF 地图中将障碍物标记为被占用状态，并且距离最近的障碍物距离为零，然后障碍物附近的栅格距离最近障碍物的距离沿梯度方向依次向外递增。距离障碍物距离越近，栅格的值越小，距离越远，栅格的值越大。

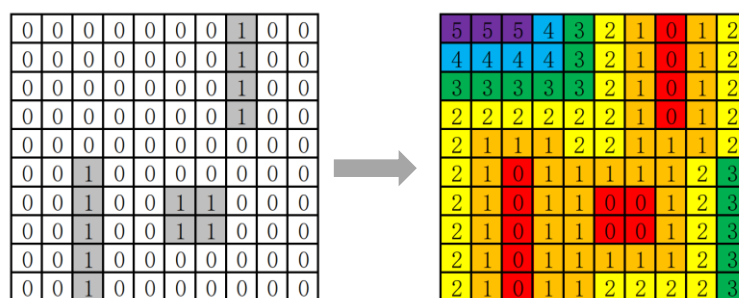


图 3-22 二维空间栅格地图与 ESDF 地图的转换示意图

本文的障碍物地图的构建分为全局点云地图和局部 ESDF 地图，全局点云地图由连续相机的双目立体匹配得到，并且将全局点云地图与栅格地图相对应，主要用于路径规划中的轨迹搜索，在无人机飞行过程中，根据全局栅格地图和当前无人机状态，在无人机当前位姿周围构建一个 ESDF 局部地图，主要用于无人机的轨迹优化，计算出远离障碍物的更加安全的无人机飞行轨迹。

3.4. 本章小结

无人机的自主感知包含对于无人机自身的状态估计和构建周围的环境地图两个方面。本章针对于无人机多场景感知，提出了一种基于双目立体视觉与 IMU 和 GPS 相结合的多传感器融合无人机状态估计算法，主要包含双目视觉里程计、视觉与 GPS 融合两个部分。在视觉里程计中采用双目图像与 IMU 数据紧耦合和非线性优化的方法求解相机位姿，视觉与 GPS 融合采用 EKF 松耦合框架，来修正视觉里程计的累积误差。最后，基于无人机状态估计的位姿，采用双目立体匹配，来构建环境地图。从而实现无人机对于自身位姿的估计和周围环境的感知。

第4章 无人机自主导航避障算法的设计

4.1. 引言

本章主要阐述了无人机自主导航避障算法的设计方法。飞行器在运行的过程中利用机载传感器信息实时完成对于自身的状态估计，并且构建周围环境的障碍物地图。无人机自主导航系统所实现的功能是当无人机收到飞行目的地信息后，根据感知的状态信息和环境地图，合理地规划出一条从当前状态到达目标状态的最优轨迹，然后驱动无人机飞行控制器沿着轨迹航线到达目标地。如下图 4-1 所示，本文将无人机自主导航系统的设计主要分成三个部分：前端路径搜索、后端轨迹优化和轨迹服务器。

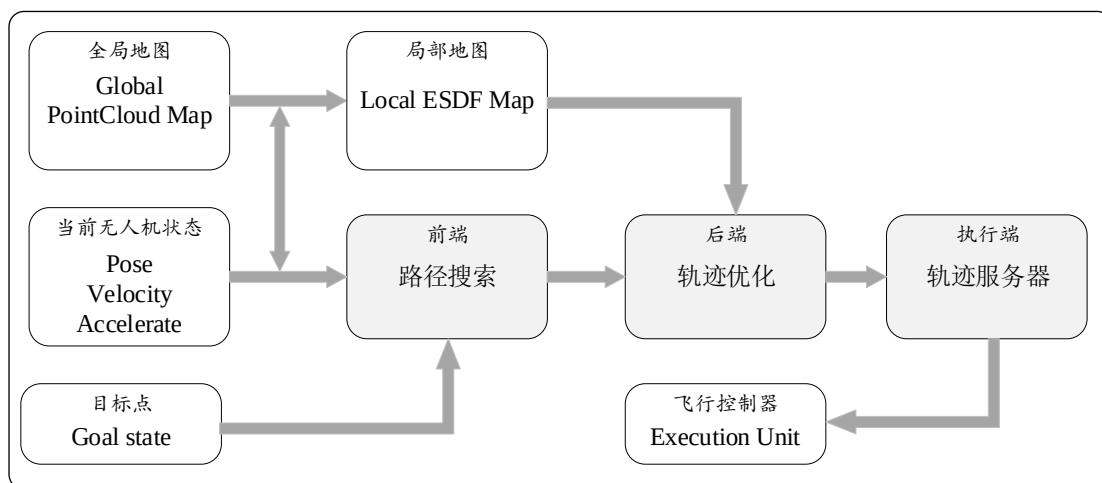


图 4-1 无人机自主导航算法执行流程图

前端路径搜索：根据感知系统产生的当前无人机位姿信息，以及全局的障碍物环境地图，使用路径搜索算法计算出从当前位置到目标点的可通行飞行路径，常用的路径搜索算法有 A*算法^[58]、RRT 算法^[59]等，本文在 A*算法的基础之上，根据四旋翼无人机飞行器动力学模型，对 A*算法进行优化，设计了一种混合状态 A*算法，使得在前端路径搜索中计算出安全的、接近最短路径长度的、光滑的、满足无人机动力学约束的轨迹。

后端估计优化：轨迹优化主要目的是对前端路径搜索所产生的飞行路径进行调优，使得轨迹更加光滑、满足无人机飞行的动力学模型、尽可能远离障碍物、尽可能减少能量消耗和飞行时间。本文后端轨迹优化的设计方法为基于梯度的轨

迹优化方法，根据当前无人机状态和全局障碍物点云地图生成局部的 ESDF 地图，基于 ESDF 地图对于前端路径搜索的结果进行优化。最后通过轨迹服务器将优化后的轨迹信息转化成无人机的控制命令并下发给无人机执行。

4.2.前端路径搜索

感知系统估计出当前无人机状态信息，以及全局的障碍物环境地图，在前端路径搜索过程中，首先将障碍物点云地图转换成三维空间中的占用栅格地图 (Occupied GridMap)，通过控制单位栅格的分辨率可调节地图的转换精度。然后路径搜索算法计算出无人机当前状态在栅格地图的位置到目标位置的可通行路径。A*算法是一种典型的路径搜索算法，它通过启发式的策略能够快速搜寻到从起点到终点的最短路径。但是 A*算法不考虑机器人的动力学模型和轨迹的光滑性，通常需要进行二次规划(Re-planning)。2010 年 Sebastian Thrun 等人提出了用于自动驾驶汽车的混合状态 A*搜索算法(Hybrid-state A* Search)^[60]，极大优化了 A*算法的搜索性能。本文在 Hybrid-state A* Search 算法的基础之上，重新设计了启发函数并在状态空间中考虑四旋翼无人机的动力学模型，使得前端搜索的路径安全、光滑、满足无人机动力学约束并且长度接近最短路径。如下图 4-2 所示，即为 A*算法和混合状态 A*算法搜索结果比较，其中蓝色的 A*算法搜索路径、红色的光滑路径为混合状态 A*算法搜寻的路径。

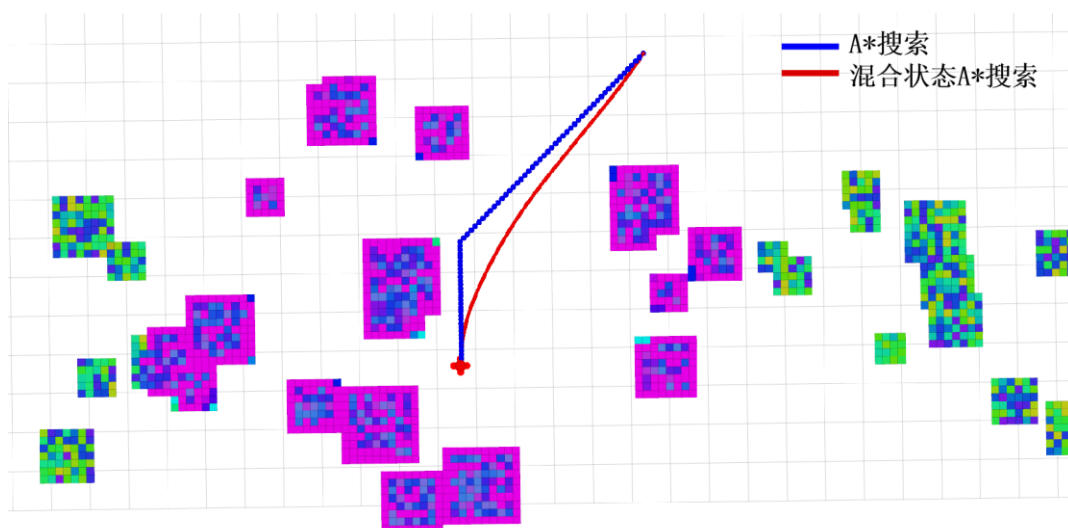


图 4-2 A*与混合状态 A*算法结果的比较

A*算法是一种典型的基于栅格地图的搜索算法。在算法中，每一个栅格节点被分配了两种代价(Cost)，第一种被称为路径代价(Path Cost)，记为 $g(n_i)$ ，表示

从起点走到当前节点花费的代价，一般指从起点开始到达当前节点的最短路径的距离。另外一种被称为启发代价(Heuristic Cost)，记为 $h(n_i)$ ，表示从当前节点到达目标点预计所需要花费的代价，一般指忽略障碍物后从当前节点到达终点的距离，不同的距离表达形式对应着不同启发函数，如图 4-3 所示，常用的形式包括 Manhattan Heuristic、Euclidean Heuristic、Diagonal Heuristic。两种代价的加权和即为 $f(n_i)$ 。A*算法中包含 Open List 和 Close List 两个重要的集合，Open List 表示需要搜索的格点，Close List 表示已经搜索完成的格点。

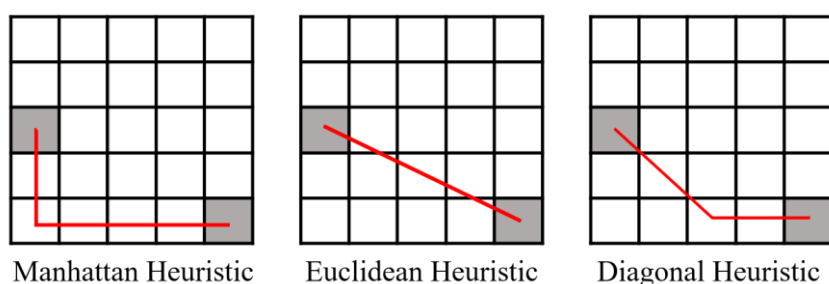


图 4-3 不同启发函数的距离形式

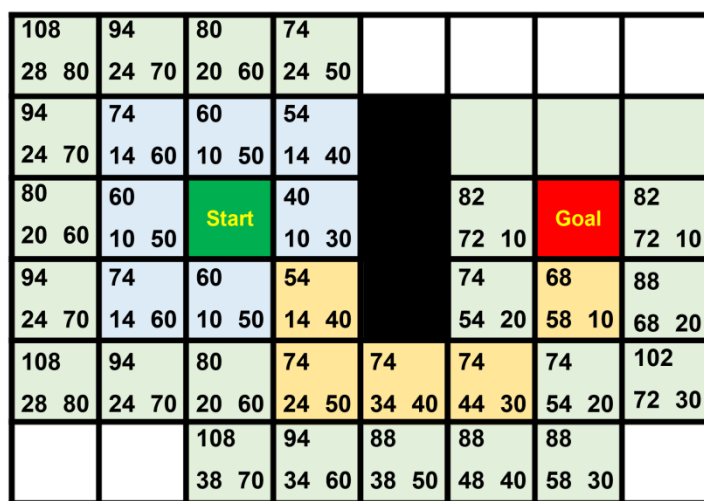


图 4-4 A*算法搜索示意图

图 4-4 展示了 A*算法的搜索流程，首先将每一个栅格表示为一个 Node 节点，每个节点的状态包含其位置 (x, y, z) 、代价 $f(n_i)$ 以及其父节点的 id。初始化时先将起点放入到 Open List 中。后序过程中，只要 Open List 非空，则每次从中取出代价最小的节点并将其添加到 Close List 中，然后遍历其周围的邻接节点，(二维栅格地图中，每一个节点周围有 8 个邻接节点，三维栅格地图中，每一个节点周围有 26 个邻接节点)。当一个邻接节点超出地图范围、被障碍物填充、或者已经存在于 Close List 中，则继续遍历下一个邻接节点。如果当前邻接节点没

有包含在 Open List 中，则将其添加到 Open List 中。如果当前邻接节点已经包含在 Open List 中，则比较当前计算的代价值与 Open List 中该节点的代价值，判断是否需要更新，只有当邻接节点探索到目标点的位置，即找到一条从起点到达目标点的最短路径。下图 4-5 则为本文在仿真系统中利用 A*算法搜索的无人机到达目标点的最短路径。

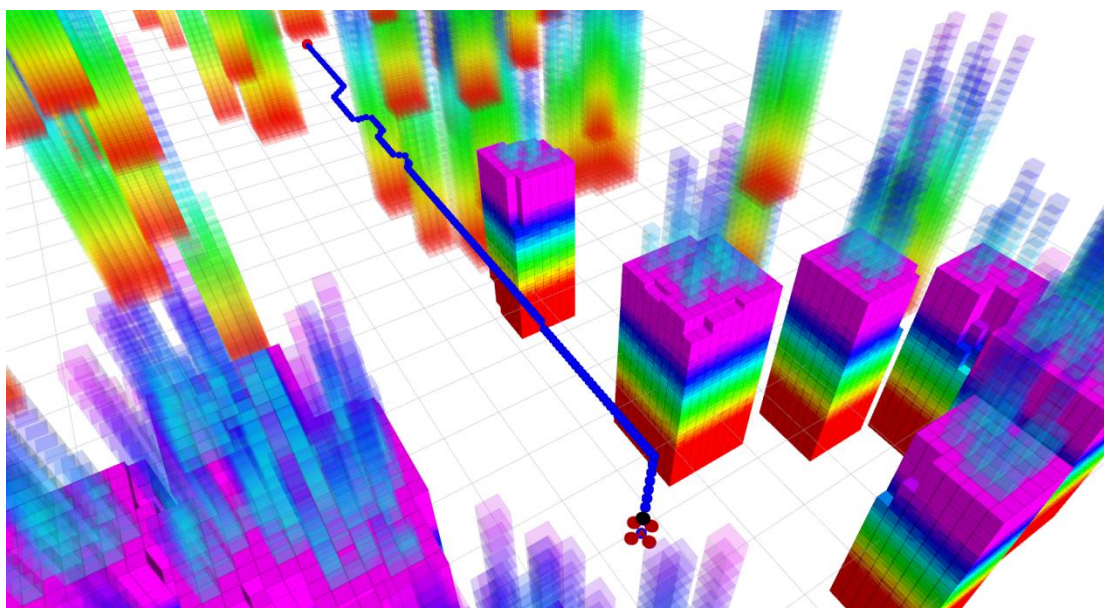


图 4-5 A*算法搜索的最短路径

在使用 A*算法进行路径搜索时，虽然总是能够找到从起点到达目标点的最短路径，但是使用 A*路径搜索存在着明显的缺陷。首先轨迹不够平滑，而且生成的路径总贴近障碍物表面，同时在搜索的过程中不会考虑飞行器的运动方向和飞行器的动力学模型，总是从一个节点转移到其相邻节点。此外 A*算法中的启发函数仅仅估算了当前节点移动到目标点的距离代价，使得节点向目标点的移动速度比较慢。针对这些问题，本文针对四旋翼无人机系统在混合状态 A*算法中做出相应的改进。首先将每一个节点的状态量由原来的位置 (x, y, z) ，提升为位置、速度、加速度 $(\text{pos}, \text{vel}, \text{acc})^T$ ，每次节点转移不再直接搜索其各个方向的相邻节点，而是根据四旋翼飞行器的动力学模型，根据当前节点的位置、速度和加速度进行搜索，如下图 4-6 所示。启发函数的设计不再仅仅考虑距离代价，而是添加时间代价和控制成本的约束，从而保证了混合状态 A*算法的搜索轨迹平滑且满足四旋翼无人机的动力学约束，并且以最小化的时间间隔和控制成本逼近最短路径。

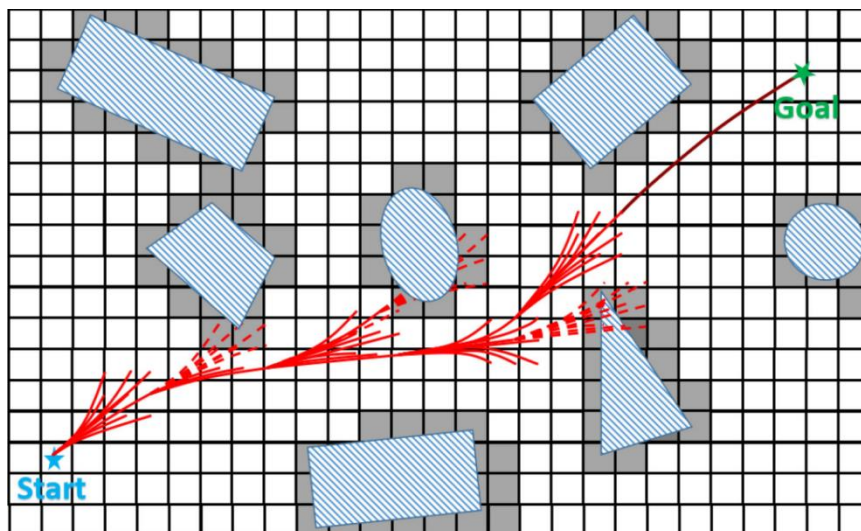


图 4-6 混合状态 A*搜索示意图

图 4-7 即为利用仿真器构建虚拟的障碍物环境地图，并且地图中设置无人机的起点和目标点，在三维空间中分别使用 A*算法和混合状态 A*算法进行路径搜索。其中蓝色轨迹为 A*算法搜索结果，路径贴近障碍物而且不够平滑。红色轨迹为使用混合状态 A*算法搜索的结果，轨迹光滑、满足无人机动力学约束，并且能够最小化从起点到达目标点的时间间隔和控制成本。

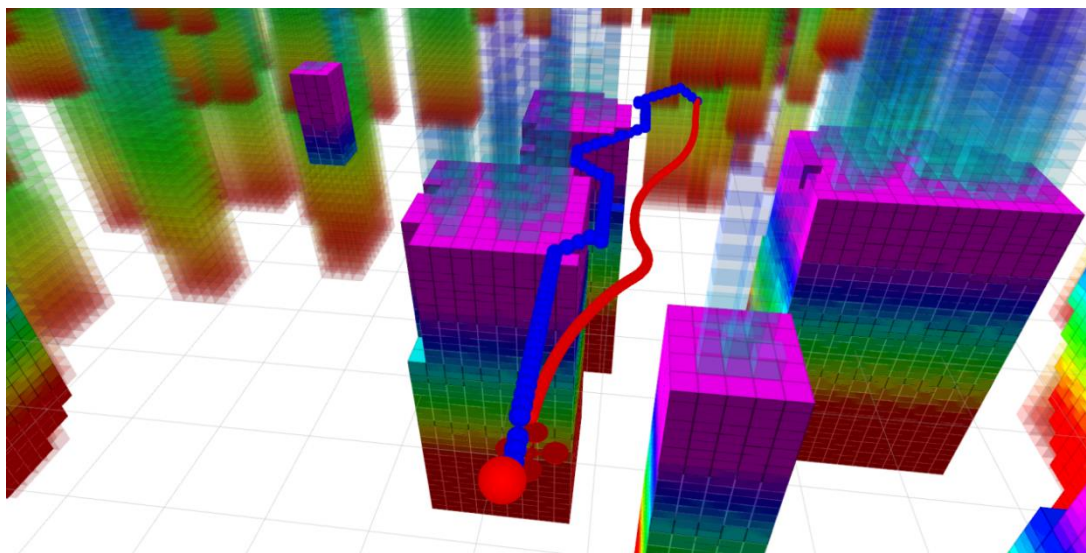


图 4-7 A*与混合状态 A*路径搜索结果比较

4.3.后端轨迹优化

在仿真系统中，混合状态 A*算法的搜索结果可以直接用于无人机运动控制，但是在实际应用系统中由于地图构建的精度、状态估计的误差以及无人机的控制

误差,会导致无人机过于靠近障碍物甚至存在碰撞障碍物的危险。因此在运动规划的后端需要对混合状态 A*算法的搜索结果进一步优化,使得轨迹尽可能满足无人机的动力学约束并且远离地图中的障碍物。因此本文在运动规划的后端采用 B 样条轨迹优化算法,对前端混合状态 A*算法的搜索路径进一步优化。如下图 4-8 所示,靠近障碍物的红色轨迹为前端轨迹搜索的结果,利用均匀 B 样条的凸包性质以及 ESDF 地图的障碍物梯度信息,优化求解得到远离障碍物的绿色轨迹。

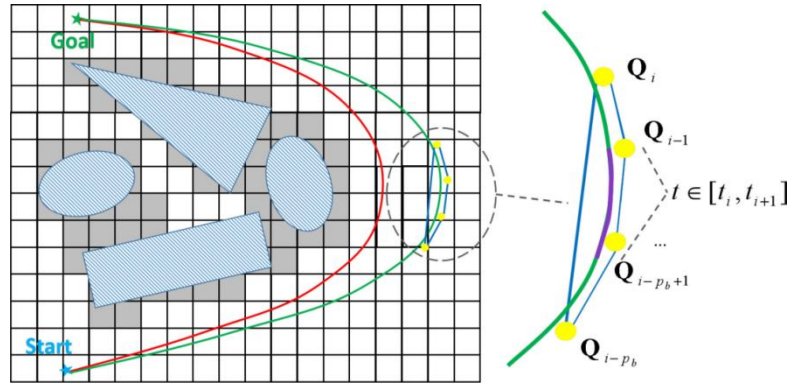


图 4-8 B 样条轨迹优化示意图^[61]

首先将待优化轨迹表示为 B 样条曲线,即一个分段多项式函数。它由 B 样条曲线的度 p_b 、控制点 $\{Q_0, Q_1, \dots, Q_N\}$ 以及结向量 $[t_0, t_1, \dots, t_M]$ 唯一确定。通常取 $p_b=3$, 表示三次样条插值, 每一个控制点 $Q_i \in \mathbf{R}^3$, 表示无人机在三维空间中的位置, $t_m \in \mathbf{R}$, 表示控制时间。在均匀 B 样条曲线中, 每个分段多项式的时间间隔均相等, 即 $\Delta t = \Delta t_m = t_{m+1} - t_m$, 对于任意位置的时间点 $t \in [t_m, t_{m+1}) \subset [t_{p_b}, t_{M-p_b}]$, 取其归一化时间 $s(t) = (t - t_m) / \Delta t$, 那么对于任意一个分段多项式 t 时刻的位置则可以表示为式(4-1), 其中 M_{p_b+1} 为关于 p_b 的常数矩阵。

$$\mathbf{p}(s(t)) = \mathbf{s}(t)^T \mathbf{M}_{p_b+1} \mathbf{q}_m \quad (4-1)$$

$$\mathbf{s}(t) = [1 \quad s(t) \quad s^2(t) \quad \dots \quad s^{p_b}(t)]^T \quad (4-2)$$

$$\mathbf{q}_m = [Q_{m-p_b} \quad Q_{m-p_b+1} \quad Q_{m-p_b+2} \quad \dots \quad Q_m]^T \quad (4-3)$$

对于一个由 $N+1$ 个控制点 $\{Q_0, Q_1, \dots, Q_N\}$ 定义的度为 p_b 的 B 样条轨迹曲线, 通过优化中间的 $N+1-2p_b$ 个控制点 $\{Q_{p_b}, Q_{p_b+1}, \dots, Q_{N-p_b}\}$ 来优化整个轨迹曲线。利用 B 样条曲线的凸包性质可以给轨迹曲线添加多种约束条件, 如式(4-4)所示, 为了使得轨迹光滑、远离障碍物、并且满足无人机速度和加速度的约束, 分别对样条曲线添加了光滑度代价函数(smoothness cost) f_s 、避免碰撞的代价函数(collision cost) f_c 以及无人机速度和加速度的代价函数 f_v 、 f_a 。通过调节权重

λ_1 、 λ_2 、 λ_3 分配来调整轨迹的光滑性、安全性和无人机动态可行性。

$$f = \lambda_1 f_s + \lambda_2 f_c + \lambda_3 (f_v + f_a) \quad (4-4)$$

在设计光滑度代价函数 f_s 时, 需要利用样条曲线中轨迹点的几何信息, 本文采用了 Christoph 等人提出的使用橡皮筋代价函数(Elastic Band Cost Function)进行轨迹优化的方法^[62]。对于任意三个连续的控制点 $\mathbf{Q}_{i-1}$ 、 $\mathbf{Q}_i$ 、 $\mathbf{Q}_{i+1}$ 可以将其看成两段相连的橡皮筋 $F_{i-1,i} = \mathbf{Q}_{i-1} - \mathbf{Q}_i$ 、 $F_{i+1,i} = \mathbf{Q}_{i+1} - \mathbf{Q}_i$, 则两段橡皮筋的合力为 $F_{i-1,i} + F_{i+1,i}$ 。如果两个控制点距离较远, 表示橡皮筋被拉伸的较长, 受力较大。理想情况下, 如果每一项约束值均为零, 则表示所有的控制点均匀地分布在一条直线上, 表示轨迹绝对光滑。所以光滑度代价函数 f_s 的表达式如公式(4-5)所示。

$$f_s = \sum_{i=p_b-1}^{N-p_b+1} \left\| \underbrace{(\mathbf{Q}_{i+1} - \mathbf{Q}_i)}_{F_{i+1,i}} + \underbrace{(\mathbf{Q}_{i-1} - \mathbf{Q}_i)}_{F_{i-1,i}} \right\|^2 \quad (4-5)$$

在设计避免碰撞的代价函数 f_c 时, 需要考虑每一个控制点到地图中最近障碍物点的距离。而 ESDF 地图中的每一个栅格存储的信息即为当前栅格距离最近障碍物点的欧氏距离, 所以利用 ESDF 地图可快速查询当前控制点 $\mathbf{Q}_i$ 到地图中最近障碍物点的距离, 记为 $d(\mathbf{Q}_i)$ 。为了使得每一个轨迹控制点尽量远离地图障碍物, 可以设定一个虚拟的排斥力 F_c , 如公式(4-6)所示, 当 $d(\mathbf{Q}_i)$ 小于安全阈值 d_{th} 时, 说明当前控制点距离障碍物过近, 此时产生一个排斥力 $F_c(d(\mathbf{Q}_i))$ 对当前控制点进行惩罚, 使其远离障碍物。因此避免碰撞的代价函数 f_c 的表达式如公式(4-7)所示。

$$F_c(d(\mathbf{Q}_i)) = \begin{cases} (d(\mathbf{Q}_i) - d_{th})^2 & d(\mathbf{Q}_i) \leq d_{th} \\ 0 & d(\mathbf{Q}_i) > d_{th} \end{cases} \quad (4-6)$$

$$f_c = \sum_{i=p_b}^{N-p_b} F_c(d(\mathbf{Q}_i)) \quad (4-7)$$

在设计无人机的动力约束代价函数 f_v 、 f_a 时, 需要限定无人机飞行的最大速度 $v_{\max}$ 和最大加速度 $a_{\max}$ 。如公式(4-8)所示, 根据相邻控制点之间的距离和时间间隔可以计算出无人机移动的理论速度 $\mathbf{V}_i$ 和加速度 $\mathbf{A}_i$ 。在真实的无人机系统中, 根据飞行器的动力学约束, 飞行的速度和加速度受到一定的约束, 如公式(4-9)所示, 当计算的速度和加速度超过限定阈值时, 将产生速度和加速度的惩罚来约束控制点的位置。因此无人机的速度和加速度的动力约束代价函数 f_v 、 f_a 如公式(4-10)所示。

$$\mathbf{V}_i = \frac{1}{\Delta t}(\mathbf{Q}_{i+1} - \mathbf{Q}_i), \quad \mathbf{A}_i = \frac{1}{\Delta t}(\mathbf{V}_{i+1} - \mathbf{V}_i) \quad (4-8)$$

$$F_v(v_\mu) = \begin{cases} (v_\mu^2 - v_{\max}^2)^2 & v_\mu^2 > v_{\max}^2 \\ 0 & v_\mu^2 \leq v_{\max}^2 \end{cases} \quad F_a(a_\mu) = \begin{cases} (a_\mu^2 - a_{\max}^2)^2 & a_\mu^2 > a_{\max}^2 \\ 0 & a_\mu^2 \leq a_{\max}^2 \end{cases} \quad (4-9)$$

$$f_v = \sum_{\mu \in \{x,y,z\}} \sum_{i=p_b-1}^{N-p_b} F_v(V_{i\mu}) \quad f_a = \sum_{\mu \in \{x,y,z\}} \sum_{i=p_b-1}^{N-p_b} F_a(A_{i\mu}) \quad (4-10)$$

如图 4-9 所示, 在仿真环境中随机生成障碍物点云并构建障碍物环境地图, 然后分别使用三维空间的 A*算法、混合状态 A*算法, 以及基于前端的路径搜索结果再进行 B 样条轨迹优化算法, 产生了 3 条从无人机起点位置到达目标点的路径, 其中蓝色轨迹为 A*算法的搜索结果, 红色轨迹为混合状态 A*算法的搜索结果, 相比于 A*算法, 结果有很大提升, 通过添加无人机动力学约束以及考虑了时间代价和控制成本的启发式函数, 使得轨迹较为平滑, 但是没有考虑控制点到空间中障碍物的几何信息, 因此仍贴近障碍物表面。黄色轨迹为采用均匀 B 样条轨迹优化算法的结果, 它是以混合状态 A*算法的搜索结果为基础, 对红色轨迹进行均匀 B 样条插值, 并遵循光滑度代价函数、避免碰撞的代价函数和无人机的动力约束代价函数对轨迹进行优化调整, 从而产生光滑的、安全的、满足无人机动力约束的可通行轨迹。

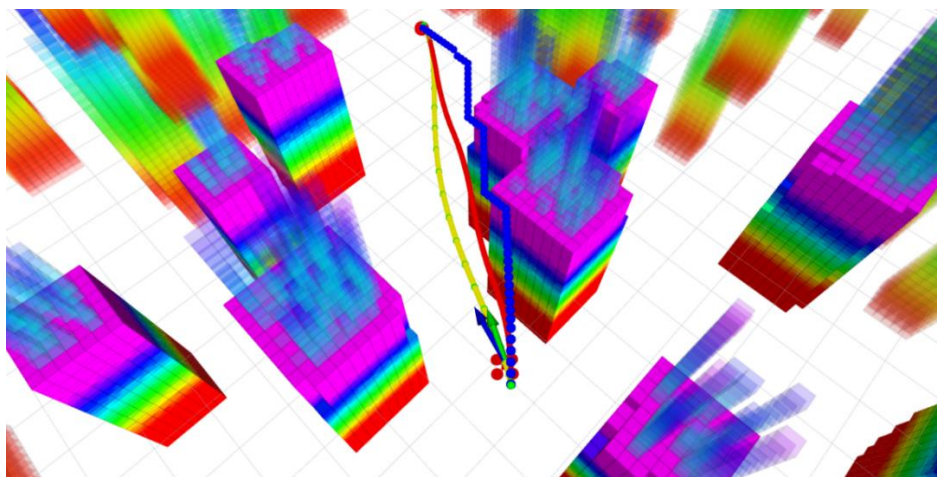


图 4-9 无人机路径规划方法的结果比较

4.4. 本章小结

本章分析了无人机自主导航避障的设计方法, 在前端路径搜索中, 介绍了 3D A*搜索算法流程, 分析了算法的优劣, 并基于四旋翼无人机的动力学模型设计了混合状态路径搜索算法, 不仅搜索的路径更加平滑, 而且路径搜索的解更接近于后端优化的结果, 因此可以加速整体的无人机导航规划算法。后端轨迹优化采用 B 样条轨迹优化算法, 并设计了光滑度代价函数、避免碰撞的代价函数以及无人机最大速度和加速度的代价函数对轨迹进行约束。通过前端路径搜索和后端估计优化相结合, 求解出光滑、安全、并满足无人机动力约束的可通行轨迹。

第5章 无人机系统实验分析

本章通过实验对无人机的感知算法、导航算法以及整个无人机系统进行了详细的分析。在感知算法部分，针对于本文提出的基于多传感器融合的无人机状态估计算法，分别在 KITTI 和 EuRoC 数据集进行评估测试，并且以若干种同类算法为基准，分析比较了本文视觉里程计算法的性能。然后，基于无人机状态估计的位姿，并结合双目视觉立体匹配，分别构建了基于 KITTI 数据集的室外大场景和 EuRoC 数据集的室内小场景等不同尺度的障碍物环境地图；在运动规划算法部分，首先通过设置不同分辨率的栅格地图，测试运动规划算法中路径搜索和轨迹优化的实时性和准确性。然后分别在仿真场景中和真实的数据场景中进行前端路径搜索和后端轨迹优化的测试；最后，对于整个无人机自主感知与导航避障系统，分别在本文搭建的 Gazebo 与 AirSim 的无人机仿真平台、以及基于 NVIDIA TX2 与 Pixhawk 的实际四旋翼飞行器上进行完整的系统测试。

5.1. 无人机感知算法的实验分析

5.1.1. 视觉里程计与多传感器融合实验分析

针对于双目视觉里程计，本文首先在 KITTI 数据集^[63]上进行了实验分析。KITTI 采用自动驾驶平台 Annieway，通过配备高分辨率的彩色相机、灰度相机、激光雷达和 GPS 等传感器，制作了立体视觉、光流、视觉里程计、三维物体检测、三维物体跟踪等多个不同类型的数据集。图 5-1 即为 KITTI Odometry 数据集的图像，与无人机室外低空飞行的场景十分类似。



图 5-1 KITTI Odometry 数据集图片

本文使用 KITTI Odometry 的原始数据作为实验数据，主要利用数据集中同步的双目图像和 GPS 测量数据，以及提供的参考真值，对本文的双目视觉里程计性能进行评估。KITTI Odometry 数据集的前 11 个序列提供了 Odometry 的真实值，因此可以使用前 11 个双目图像序列测试本文双目视觉里程计的精度，同时还比较了目前学术主流的双目视觉里程计 ORB_SLAM2 和 VINS_Fusion 和本文的视觉里程计的性能差异。此外还利用 KITTI Odometry 原始数据中的 GPS 传感器数据，测试了本文多传感器融合的无人机状态估计算法的精度。表 5-1 和图 5-2 和 5-3 即为在 KITTI 数据集上的实验结果。

表 5-1 双目视觉定位算法在 KITTI 不同序列的位移均方根误差(RMSE %m)

Sequence	ORB_SLAM2	VINS_Fusion	Proposed	Proposed(with_gps)
03	2.42	1.58	2.40	0.75
04	2.93	4.21	2.95	0.43
05	1.91	3.29	2.75	0.41
06	2.12	3.07	2.82	0.50
07	1.91	2.71	2.19	0.50
10	2.43	3.21	2.50	0.61

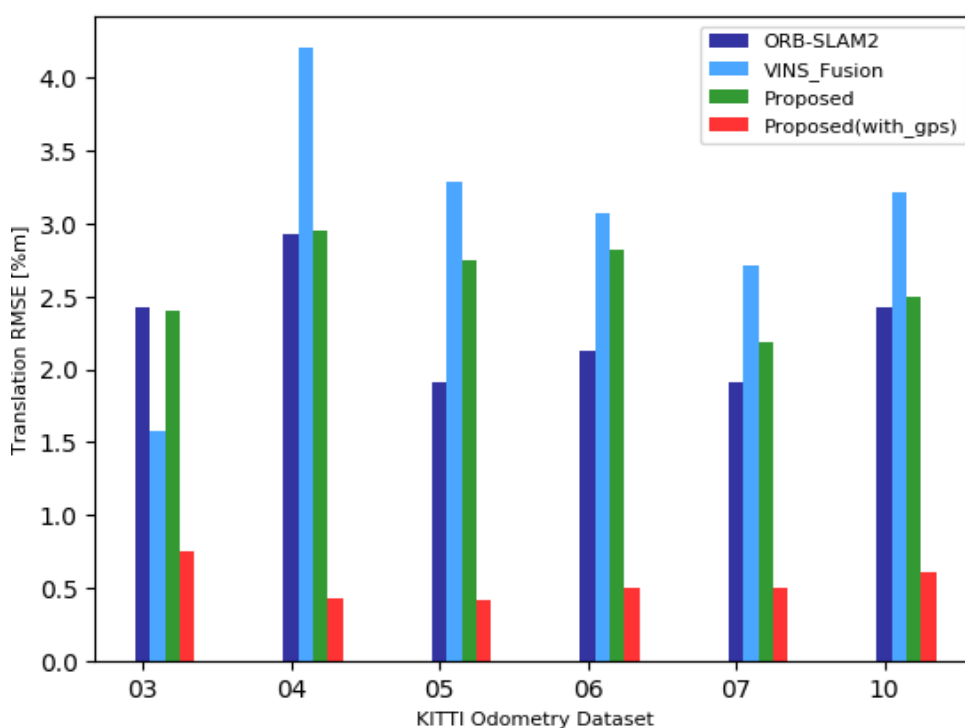


图 5-2 不同算法在 KITTI 数据集上的位移均方根误差比较

在表 5-1 中对比了 ORB_SLAM2、VINS_Fusion 和本文所提出双目视觉里程计算法以及本文双目视觉与 GPS 相融合的算法，在 KITTI 不同的序列上定量分析的结果，评价的指标为位移均方根误差(RMSE)，因为在室外大场景下，所以设置的单位为(%m)，即表示每行驶 100 米位移的偏移量。从表 5-1 和图 5-2 可以

看出, 如果仅使用双目图像进行位姿估计, ORB_SLAM2、VINS_Fusion 和本文的视觉里程计的 RMSE(%m)均在 1.5 到 4.5 米之间, 即行驶 100 米, 最大误差 4.5 米。从 KITTI 数据集上的实验数据来看, 本文的双目视觉里程计算法精度与 ORB_SLAM2 相当, 并且略微优于 VINS_Fusion。误差存在的主要原因旋转的偏差和位移的累积偏差。因此, 当进一步将双目视觉里程计与 GPS 信号相融合, 提供全局的位姿纠正, 使得双目视觉里程计的精度显著提升。从图 5-3 可直观地看出, ORB_SLAM2、VINS_Fusion 以及本文的双目视觉里程计均会产生累积漂移, 并且随着误差的累积, 估计的轨迹会越来越偏移真实轨迹, 当融合 GPS 数据之后可以明显减小误差, 从图中可以看出, 多传感器融合的轨迹几乎与真实轨迹重合。这也充分验证了本文多传感器融合的双目视觉里程计的精度和有效性。

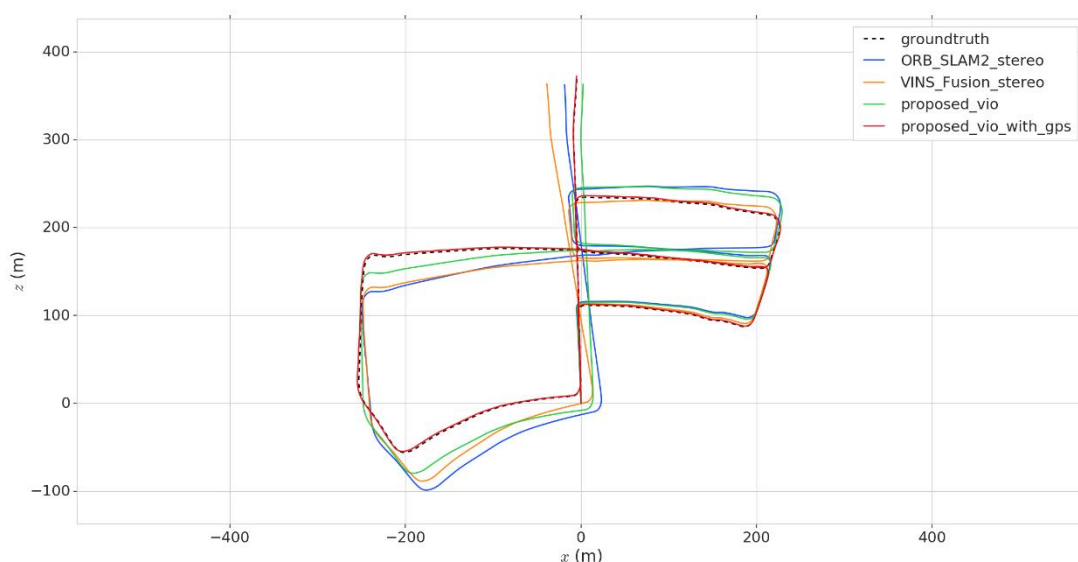


图 5-3 在 KITTI 序列上不同算法估计的轨迹图比较

为了进一步分析实际飞行过程中无人机状态估计的精度, 本文在 EuRoC 数据集^[64]上继续进行了相关实验分析。EuRoC 数据集是苏黎世联邦理工学院 ASL 实验室发布的专门针对微型飞行系统的视觉、惯导数据集, 它利用微型四旋翼飞行器搭载双目摄像头和 MEMS IMU 传感器, 实时记录无人机飞行过程中的双目图像序列和 IMU 传感器数据, 并利用 Vicon 运动追踪设备和 Leica MS50 雷达追踪设备对无人机姿态进行实时追踪, 为数据集提供真实标签值。EuRoC 数据集中共包含了三个不同场景(Machine Hall、Vicon Room 1、Vicon Room 2), 在每个场景中分别录制了多个序列(MH-01~05、V1-01~03、V2-01~03)。在每个场景中根据无人机飞行的速度、转动的幅度等参数把序列分成简单、中等、困难三种等级。如图 5-4 即为 EuRoC 数据集的三个场景图。



图 5-4 EuRoC 数据集场景图 (a) Machine Hall (b)Vicon Room 1 (c)Vicon Room 2

本文针对 EuRoC 数据集的每一个序列都进行测试，并将本文的双目视觉里程计与 ROVIO、VINS_Fusion 两种主流的双目视觉、惯导相融合的视觉 SLAM 比较。此外，因为室内场景没有 GPS 信号，本文采用 Vicon 运动追踪设备提供的全局位姿，为视觉里程计提供全局参考，从而进行多传感器融合的双目视觉里程计测试。表 5-2 和图 5-6、5-7 为各种双目视觉里程计在 EuRoC 数据集上的测试结果。表中的数值表示不同算法在每一个 EuRoC 序列上估计的相机运动位姿与数据集提供的真实的均方根误差。因为 EuRoC 数据均在室内小场景下，所以表中的数值单位均为米。而图 5-6、5-7 为了更直观地显示，将 RMSE 单位设置为 %m。

表 5-2 双目视觉定位算法在 EuRoC 不同序列的位移均方根误差(RMSE)

Sequence	ROVIO	VINS Fusion	Proposed	Proposed(with_gps)
MH_01_easy	0.04078	0.00829	0.02796	0.00303
MH_02_easy	0.04436	0.02895	0.02798	0.00254
MH_03_medium	0.06656	0.00996	0.0714	0.00731
MH_04_difficult	0.0628	0.01268	0.06992	0.00687
MH_05_difficult	0.05897	0.01158	0.0618	0.01194
V1_01_easy	0.02401	0.02024	0.02596	0.00542
V1_02_medium	0.05778	0.03877	0.05181	0.00699
V1_03_difficult	0.05031	0.12866	0.04937	0.00804
V2_01_easy	0.02291	0.01706	0.01905	0.01254
V2_02_medium	0.05413	0.03446	0.0424	0.03446
V2_03_difficult	0.07306	0.1813	0.12347	0.01793

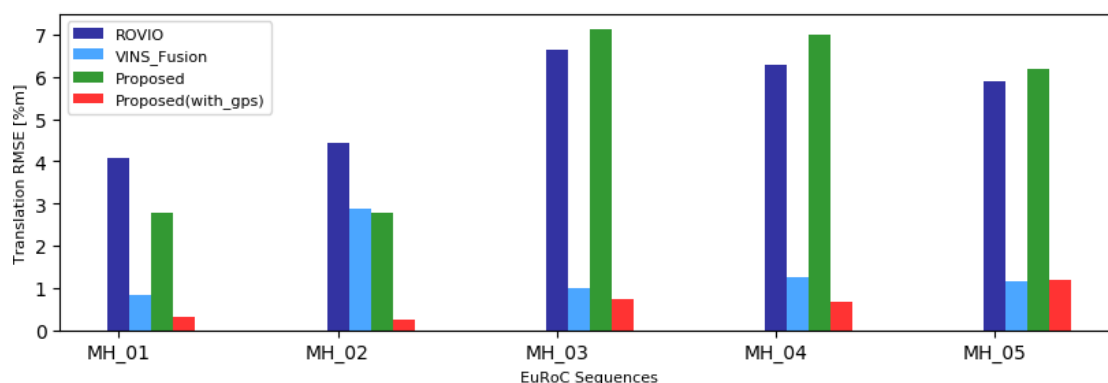


图 5-5 不同算法在 EuRoC MH_01~MH_05 数据集上的位移均方根误差比较

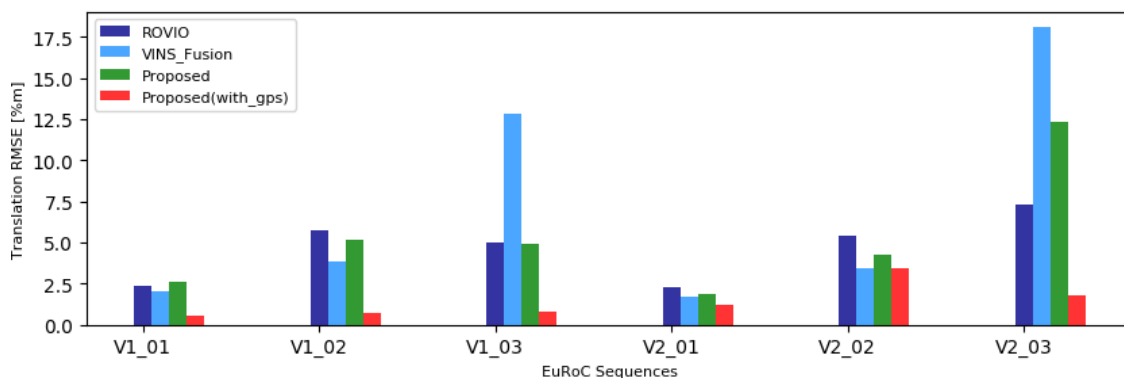


图 5-6 不同算法在 EuRoC V1 V2 数据集上的位移均方根误差比较

通过分析表 5-2 和图 5-6、5-7，首先，可以发现视觉里程计算法在 EuRoC 数据集上的误差分布不均匀，MH_03、MH_04、MH_05 明显高于 MH_01、MH_02，V1_03 明显高于 V1_01，V2_03 明显高于 V2_01，主要因为每个场景中无人机飞行的速度、转动的幅度不同，这与实际情况相吻合。当无人机运动速度越快、转动越剧烈，则视觉里程计的位姿估计精度越差。其次，在 EuRoC 数据集上，本文的视觉里程计算法精度与 ROVIO 和 VINS_Fusion 相当，位姿估计误差基本均在 10 厘米以内，(除去 VINS_Fusion 在 V1_03 和 V2_03 上表现较差)。最后从图 5-6 和 5-7 发现，红色柱状图即本文的多传感器融合算法的位移均方根误差明显降低，在每个序列上误差均在 5 厘米以内，充分说明了多传感器融合对于无人机状态估计的精度有所提升。

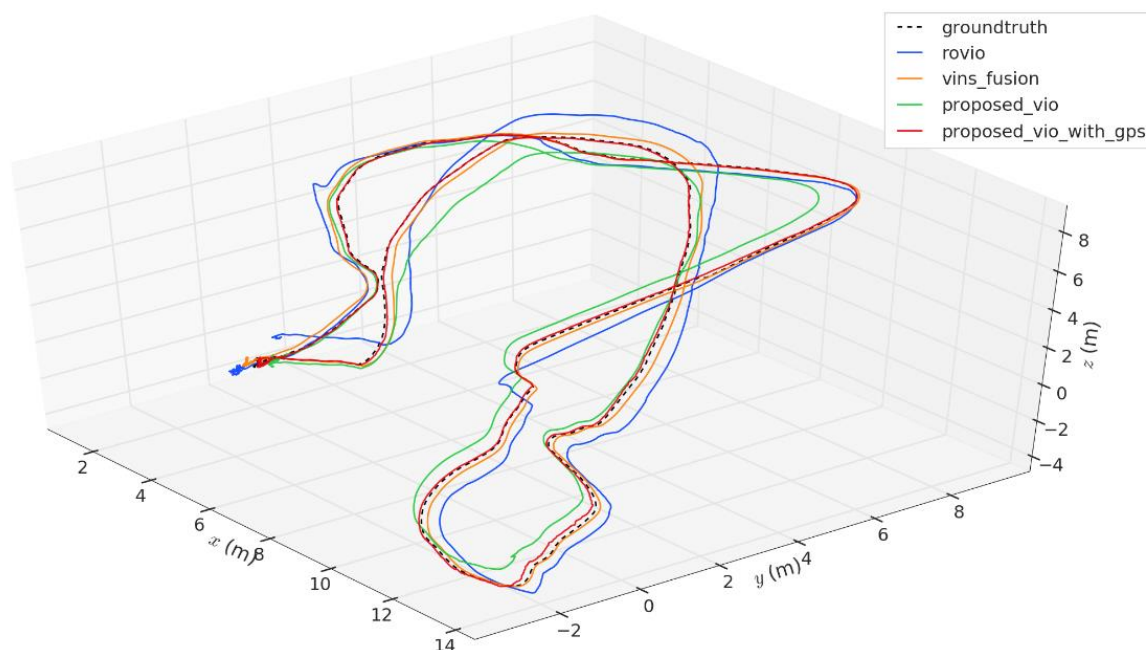


图 5-7 在 EuRoC 的序列 MH_04 上不同算法测试的轨迹图

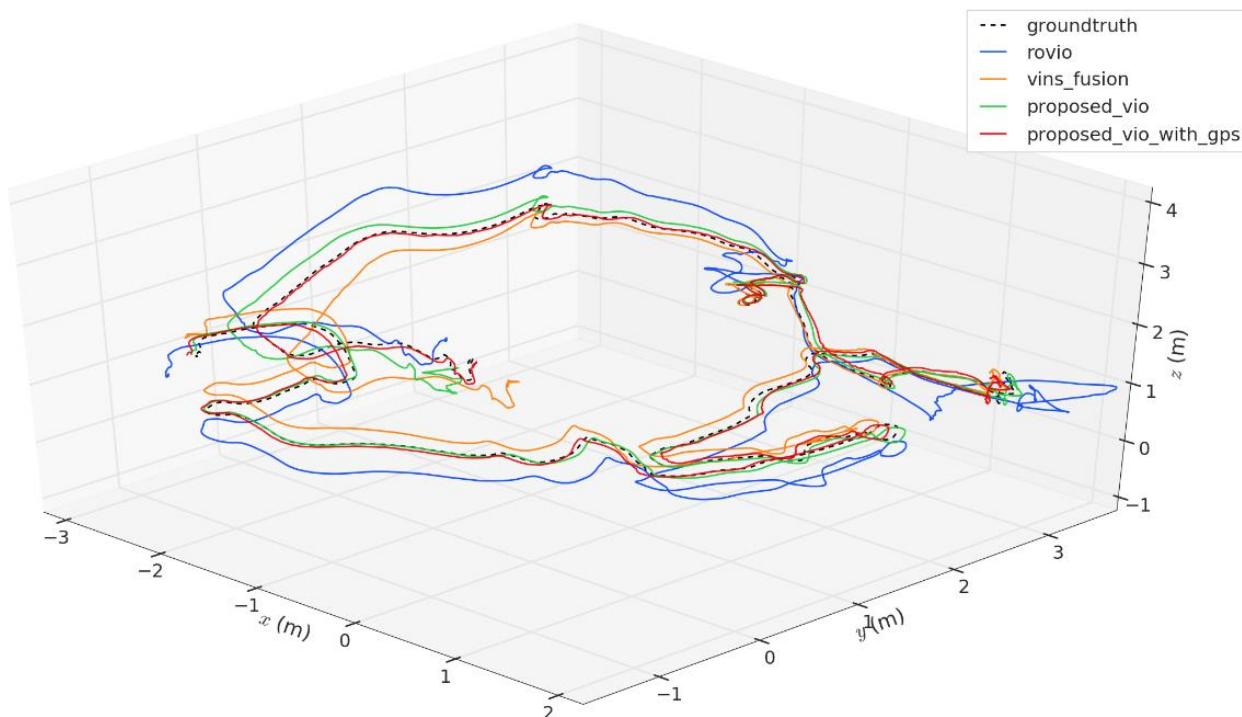


图 5-8 在 EuRoC V2_01 序列上不同算法测试的轨迹图比较

图 5-7、5-8 为 ROVIO、VINS_Fusion、本文的双目视觉里程计以及本文的多传感器融合算法在 EuRoC 数据集上估计的相机运动位姿轨迹与数据集参考的真实轨迹的比较。从图中可以看出。本文的多传感器融合轨迹(红色)与轨迹真实值最为接近，其他三种形式的双目视觉里程计与真实轨迹相比，均有 10 厘米左右的误差，也充分说明了多传感器融合对于无人机状态估计的有效性。

5.1.2. 地图构建实验分析

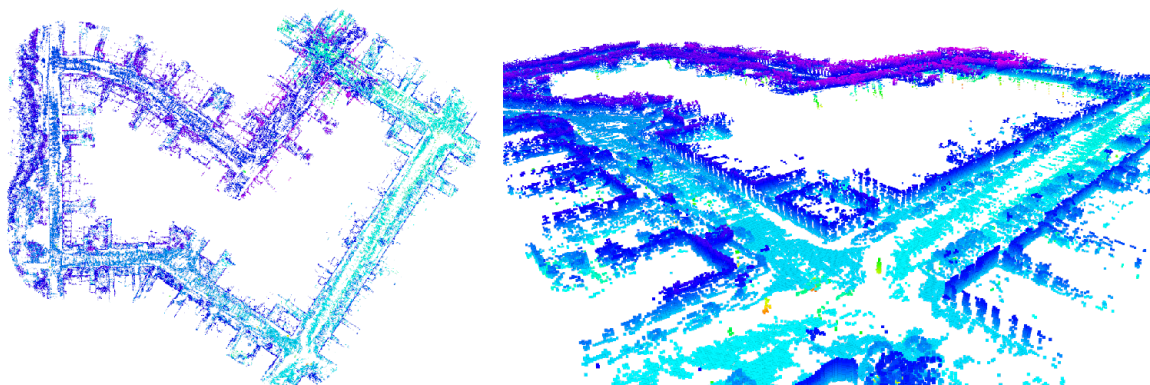


图 5-9 基于 KITTI 数据的室外大场景建图测试 (a)全局图 (b)局部细节图

本文地图构建的方法采用无人机状态估计算法输出的位姿，并结合双目立体匹配算法，将每一个相机位姿所观测的障碍物投影到三维空间，从而构建无人机

的障碍物环境地图。基于 KITTI 数据集和 EuRoC 数据集，分别构建了室外大场景和室内小场景的障碍物环境地图。验证地图构建算法是否满足地图稠密性、场景多样性和全局一致性的需求。如图 5-9 所示，表示在 KITTI 数据集上，采用本文的多传感器融合的视觉里程计估计出每一帧相机的位姿，并结合每一帧图像的深度立体匹配，构建全局环境地图。图 5-10 表示在 EuRoC 数据集上构建的室内小场景障碍物环境地图，从图中可以清晰看出墙壁、梯子和障碍物的轮廓。为了进一步满足后序的无人机路径规划算法的需求，将障碍物点云地图转化为 TSDF 和 ESDF 的栅格地图。如图 5-11 所示，为了更直观的显示，本文只显示了栅格地图中高度为 1 米的一层。图中灰色部分为原本的障碍物点云地图，在 TSDF 栅格地图上能清晰分辨出障碍物的边缘轮廓，在 ESKF 地图中能快速查到距离障碍物较远的空闲栅格。

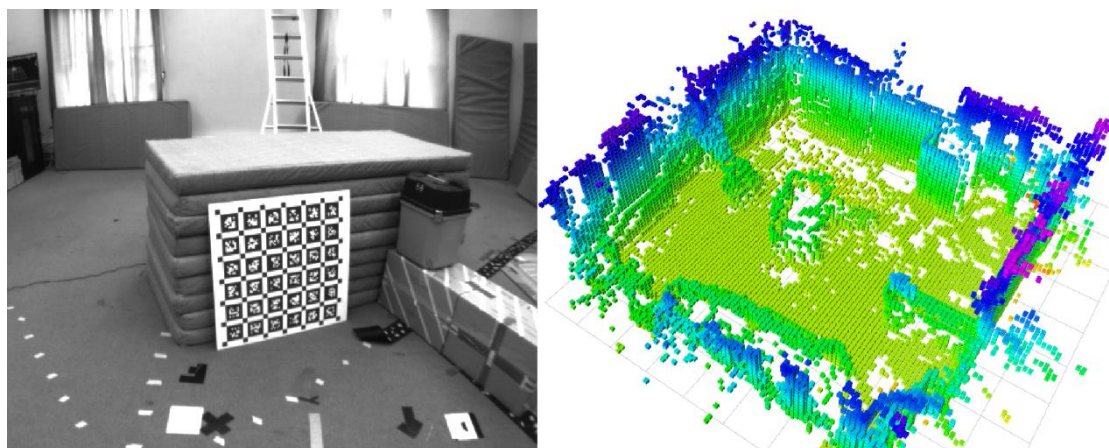


图 5-10 基于 EuRoC 数据的室内小场景建图测试 (a)场景图 (b)障碍物点云图

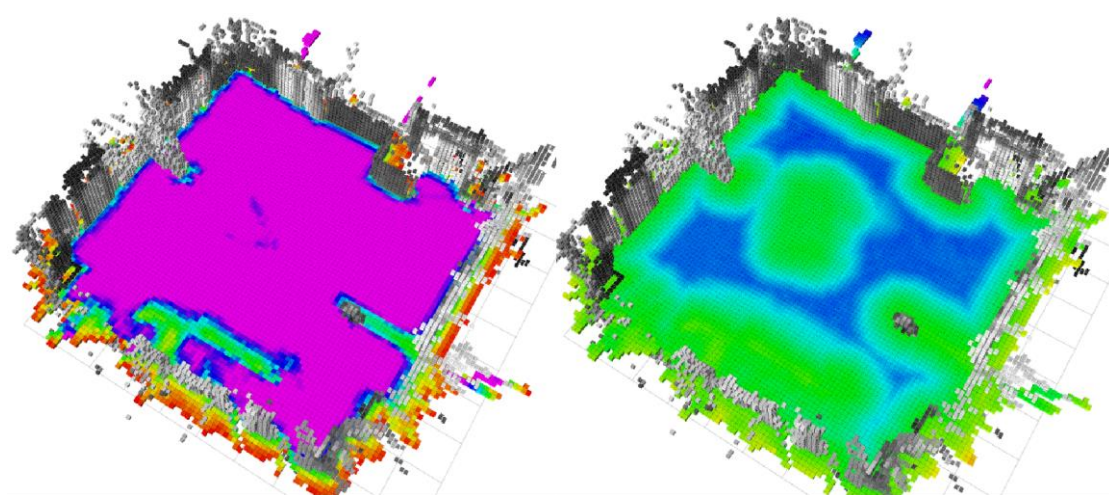


图 5-11 基于 EuRoC 数据的室内小场景建图测试 (c)TSDF 栅格地图 (d)ESKF 栅格地图

5.2. 路径规划算法的实验分析

为了验证路径规划算法的实时性和准确性，本文设计了多组实验。首先利用仿真系统生成一个长宽高为 $50\times 50\times 5\text{m}$ 的障碍物地图，并随机生成 150 个障碍物。每次从无人机的当前位置随机选定一个 20 米远的目标点，同时设定无人机的最大飞行速度 3m/s ,最大加速度 2m/s ，然后运行无人机运动规划算法，记录每次运行过程中路径搜索与轨迹优化的时间、从起点飞行到目标点的时间，以及规划并成功到达目标点的成功率。如图 5-12 所示，为路径规划算法仿真实验环境。

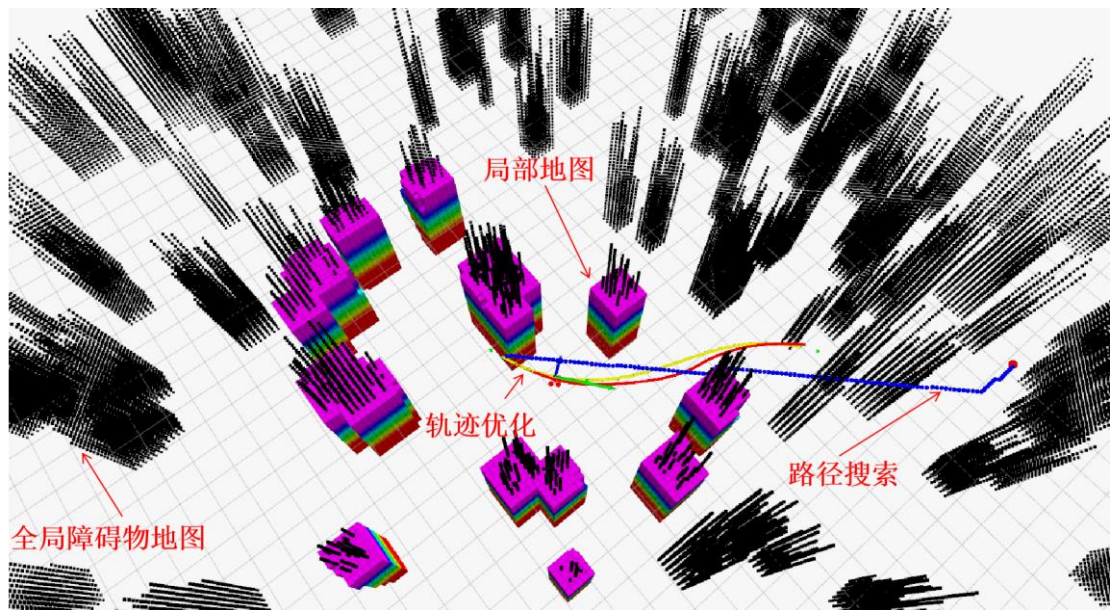


图 5-12 路径规划算法的仿真实验环境

表 5-3 无人机路径规划实验结果

地图分辨率		路径规划时间	控制执行时间	成功率
0.04m/pixel	均值	0.0287	7.843	100%
	最大值	0.2077	10.624	
	标准差	0.0153	1.033	
0.2m/pixel	均值	0.0258	7.82	94%
	最大值	0.0527	10.624	
	标准差	0.0127	1.041	
1m/pixel	均值	0.0257	7.769	78%
	最大值	0.0299	15.624	
	标准差	0.0017	1.416	

从表 5-3 中可以看出,本文的运动规划算法在不同的地图分辨率下均能有效地规划出无人机飞行路径。从实验结果分析,不同的地图分辨率下路径规划的效率和成功率存在差异,当栅格地图分辨率为 $1\text{m}/\text{pixel}$ 时,路径规划算法运行速度最快,但是准确率较差,当分辨率为 $0.04\text{m}/\text{pixel}$ 时,由于搜索空间更加精细,所以算法的运行时间更长,同时路径规划的成功率也最高。而在分辨率为 $0.2\text{m}/\text{pixel}$ 的地图上,不仅路径规划运行速度快,而且准确率高,因此在实际系统中为了满足路径规划算法的实时性,同时也要尽可能提高路径规划的成功率,采用分辨率为 $0.2\text{m}/\text{pixel}$ 的地图精度配置。

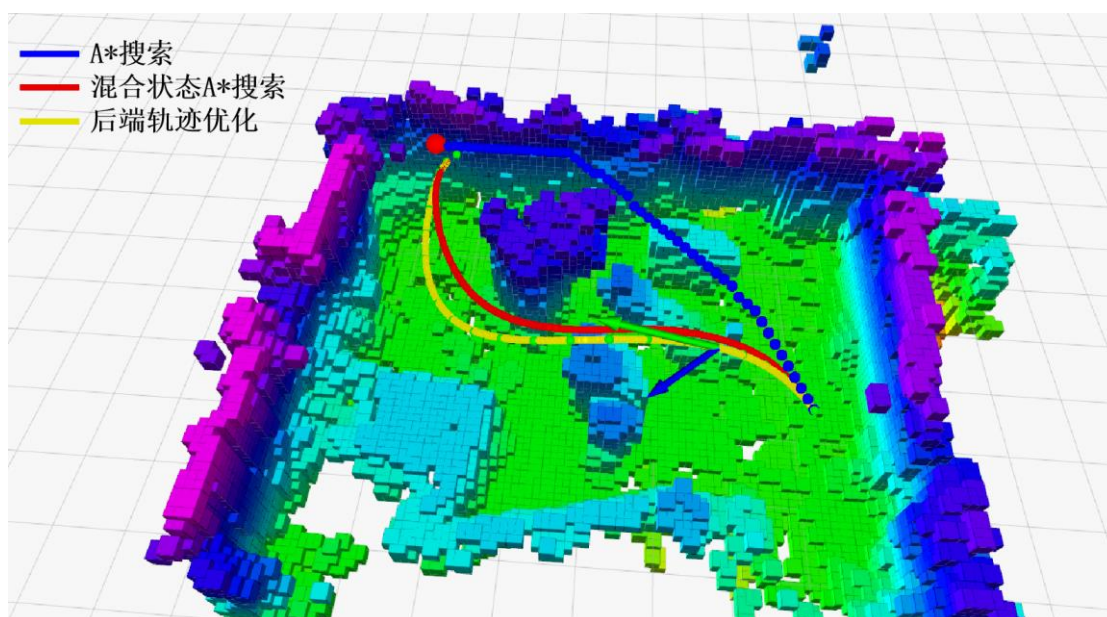


图 5-13 基于 EuRoC Vicin Room 2 室内场景的无人机路径规划测试

为了进一步验证路径规划算法在真实无人机系统上的有效性,本文基于 EuRoC Vicin Room 2 无人机室内飞行的数据设计了相应的实验。先使用双目图像和 IMU 数据进行无人机位姿估计和障碍物环境地图的构建,然后基于环境地图和仿真的无人机模型进行路径规划算法测试。如图 5-13 所示,为在 Vicin Room 2 场景内进行 A* 路径搜索、混合状态 A* 路径搜索以及后端轨迹优化算法所计算的无人机轨迹。与 A* 路径搜索相比较,混合状态 A* 路径搜索满足无人机动力约束,更加平滑,且更加接近后端轨迹优化的解,因此能够提高整体路径规划算法的效率。

5.3. 无人机自主导航避障测试

为了对整体的无人机自主感知与导航避障系统进行验证, 本文分别设计了基于 Gazebo 和 AirSim 的自主无人机仿真平台, 以及基于 NVIDIA TX2、Pixhawk 和 MYNTEYE 立体相机的四旋翼飞行器。在仿真场景中和真实环境下对无人机系统进行了自主导航避障测试。

如图 5-14 所示, 首先在 Gazebo 仿真器中构建虚拟的仿真场景, 图中设计了一个简单室内建筑场景并创建了搭载双目视觉的四旋翼无人机模型, 通过在仿真环境中运行自主感知与路径规划算法, 构建了稠密的障碍物环境地图并生成了自主导航避障的轨迹。

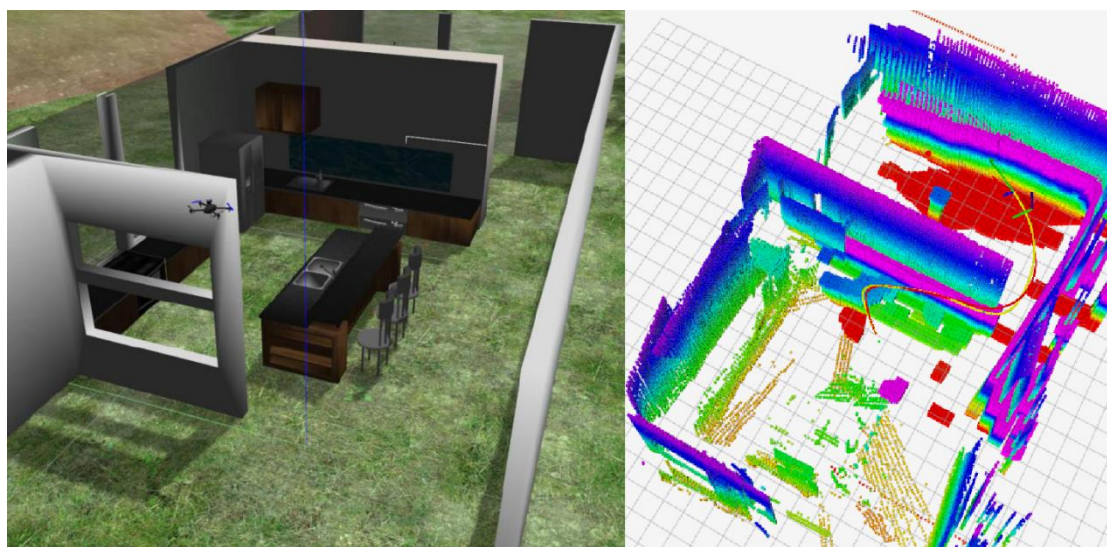


图 5-14 基于 Gazebo 仿真的无人机自主导航避障系统测试
(a) Gazebo 场景与无人机模型 (b) 可视化障碍物地图与导航轨迹



图 5-15 真实场景下无人机自主导航避障系统测试
(a) 真实场景与无人机硬件系统 (b) 可视化障碍物地图与导航轨迹

如图 5-15 所示,在本文所搭建的基于双目视觉的四旋翼无人机硬件系统上进行无人机自主导航避障的测试。无人机系统挂载了 MYNTEYE 立体相机用于无人机视觉定位和构建环境地图,搭载了 NVIDIA TX2 机载计算机用于感知算法和导航算法的实时在线运行,使用 Pixhawk 飞行控制器来执行规划的航迹和无人机控制。从图 5-15 中可以看出,无人机在一个行道树的障碍物场景下,能够估计出无人机的状态信息并构建障碍物环境地图,同时,规划出到达目标的安全、平滑轨迹。实现了无人机的自主感知与导航避障功能。

5.4. 本章小结

本章分别对无人机感知算法、路径规划算法和整个无人机自主导航避障系统都进行了充分的实验分析。在无人机感知算法实验分析部分,分别在 KITTI 和 EuRoC 数据集进行评估测试。根据 KITTI 数据集,利用双目图像数据和 GPS 数据分别比较分析了 ORB_SLAM2、VINS_Fusion 和本文的双目视觉里程计以及多传感器融合的状态估计算法。根据 EuRoC 数据集,利用双目图像、IMU 传感器数据以及 Vicon 追踪系统提供全局位姿,分别比较分析了 ROVIO、VINS_Fusion 以及本文的多传感器融合的无人机状态估计算法,验证了多传感器融合对于无人机状态估计精度的显著提升。同时,基于无人机状态估计的结果,分在构建了 KITTI 室外大场景和 EuRoC 室内小场景的障碍物环境地图,验证了障碍物地图满足稠密性、场景多样性和全局一致性。在无人机路径规划算法的实验分析部分,分别在路径规划算法仿真实验环境和 EuRoC 真实的无人机飞行场景中,分析了路径规划算法的实时性和准确性。在无人机自主导航避障测试部分,本文分别搭建了 Gazebo 与 AirSim 的无人机仿真平台、以及基于 NVIDIA TX2 与 Pixhawk 的实际四旋翼飞行器上进行完整的系统测试。分别在仿真场景中和真实环境下对无人机系统进行了自主导航避障测试,实现了无人机的自主感知与导航避障功能。

第6章 总结与展望

6.1. 本文的工作总结

本文深入研究了无人机自主感知与导航系统的设计理论,梳理了机器人领域自主感知技术和自主导航避障技术的发展历程、研究现状和最新进展。详细分析了无人机自主感知与导航系统的关键技术,针对于无人机的多场景感知问题,本文提出了一种基于双目立体视觉与 IMU 和 GPS 相结合的多传感器融合无人机状态估计算法,并构建了一套完整的无人机自主感知与导航避障系统。同时,搭建了挂载双目立体相机的无人机系统,并基于 Gazebo 仿真器和 AirSim 仿真器设计了自主无人机软件仿真系统,最后在公开数据集、仿真环境和真实场景中均设计了大量的实验,详细分析系统和算法的各个指标。实验结果充分论证了本文所设计的无人机自主感知与导航系统的有效性。

在无人机自主感知算法章节中,本文提出了一种基于双目立体视觉与 IMU 和 GPS 相结合的多传感器融合无人机状态估计算法,包含双目视觉里程计和基于松耦合的多传感器状态估计两个部分。同时,采用双目相机的立体匹配构建全局的障碍物点云地图并转换为全局的三维空间栅格地图和局部的 ESDF 地图。在双目立体视觉与惯导相融合的无人机状态估计部分,详细阐述了本文中多传感器融合的视觉 SLAM 算法的设计原理与设计思路,说明了 IMU 测量数据的建模方法和双目立体相机的工作原理,介绍了 IMU 连续时间离散数据的积分和预计分的计算方法以及 IMU 传感器测量误差的构建方式。在双目图像处理部分阐明了稀疏特征点的提取、光流追踪、利用多视图几何求解相邻帧间相对位姿、3D 点的深度信息以及视觉重投影误差约束的计算,并且阐述了利用 IMU 测量误差约束和视觉重投影约束构建非线性优化,对估计的位姿和路标点进行优化调整的计算方法。在多传感器融合部分,阐述了基于 EKF 框架的视觉里程计与 IMU 和 GPS 数据融合的无人机位姿估计的设计方法。在构建障碍物地图部分,介绍了双目立体匹配的工作原理,以及如何利用立体匹配估计深度并构建障碍物点云地图,最后将点云地图对应到三维空间栅格地图并最终转化为 ESDF 地图用于轨迹优化。

在无人机自主导航避障算法设计章节中,阐述了本文中前端路径搜索与后端轨迹优化相结合的无人机路径规划算法,在前端路径搜索中,分析了 A*路径搜

索算法的工作原理,并且针对于 A*算法的缺点将混合状态 A*算法应用于无人机路径搜索,从而保证前端能搜索出路径平滑、满足四旋翼无人机的动力学约束,并最小化时间间隔和控制成本的路径。在后端轨迹中,本文采用了 B 样条轨迹优化算法,对前端混合状态 A*算法的搜索路径进一步调优,遵循光滑度代价函数、避免碰撞的代价函数和无人机的动力约束代价函数对前端路径调整,最终产生光滑的、安全的、满足无人机动力约束的可通行轨迹。

在实验分析部分,本文搭建了完全软件在环仿真的无人机仿真平台和搭载多传感器的真实无人机硬件平台。分别在仿真平台和硬件平台以及大型公开数据集上设计了丰富的实验,验证了多传感器融合的无人机状态估计算法的精度、环境地图构建的准确度以及路径规划算法的实时性和成功率,最后分别在仿真场景中进行无人机的飞行实验和真实场景中无人机的自主避障,均取得良好的效果,充分验证了本文所设计的无人机自主感知与导航系统的有效性。

6.2. 下一步的工作方向

在机器人应用领域,相比于轮式移动机器人、无人车,空中无人机拥有更高维度的状态空间和更复杂的控制模型,所以本文的无人机自主感知与导航系统可以方便地应用于轮式移动机器人和自动驾驶汽车领域,但是当把算法真正部署到面向应用的移动机器人和无人车系统时,仍然面临许多挑战,比如,如何保证 SLAM 定位系统在多动态场景全天候长时间鲁棒地运行,如何处理地图的动态更新,以及如何保障机器人在真实环境中总能成功地规划出安全、合理的最优通行路径。所以后序的研究工作将从以下几个方面进行:

(1) 进一步优化多传感器融合的 SLAM 系统,研究能够覆盖更多场景更加鲁棒的状态估计,同时考虑添加地图复用和地图更新的机制,从而设计出通用的 Life-long SLAM 系统。

(2) 本文针对无人机自主避障功能设计了前端路径搜索与后端轨迹优化相结合的无人机路径规划算法,这种方式利用路径搜索和轨迹优化,在无人机的状态空间内计算出局部优化的解,采用这种方法不能够保证每次都能规划出最优的路径,所以后序将探索采用深度强化学习的方法来解决无人机路径规划问题。

参考文献

- [1] 裴信彪. 新型多旋翼无人机的低成本组合导航与稳定控制[D]. 中国科学院大学(中国科学院长春光学精密机械与物理研究所), 2018.
- [2] Kayton M, Fried W R. Avionics navigation systems[M]. John Wiley & Sons, 1997.
- [3] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-Time Single Camera SLAM[J]. IEEE Trans Pattern Anal Mach Intell, 2007, 29(6):1052-1067.
- [4] Klein G, Murray D. Improving the agility of keyframe-based SLAM[C]. European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2008: 802-815.
- [5] Li M, Hong B, Cai Z, et al. Novel Rao-Blackwellized particle filter for mobile robot SLAM using monocular vision[J]. International Journal of Intelligent Technology, 2006, 1(1): 63-69.
- [6] Zach C. Robust bundle adjustment revisited[C]. European Conference on Computer Vision. Springer, Cham, 2014: 772-787.
- [7] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]. Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2007: 1-10.
- [8] 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.
- [9] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]. 2014 IEEE international conference on robotics and automation (ICRA). IEEE, 2014: 15-22.
- [10] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]. European conference on computer vision. Springer, Cham, 2014: 834-849.
- [11] Engel J, Koltun V, Cremers D. Direct sparse odometry[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(3): 611-625.
- [12] Mur-Artal R, Tardos J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.
- [13] Wang R, Schworer M, Cremers D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras[C]. Proceedings of the IEEE International Conference on Computer Vision. 2017: 3903-3911.

- [14] Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]. Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007: 3565-3572.
- [15] Delmerico J, Scaramuzza D. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots[C]. 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 2502-2509.
- [16] Bloesch M, Omari S, Hutter M, et al. Robust visual inertial odometry using a direct EKF-based approach[C]. 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2015: 298-304.
- [17] Qin T, Li P, Shen S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.
- [18] Ling Y, Wang K, Shen S. Probabilistic dense reconstruction from a moving camera[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 6364-6371.
- [19] Gao F, Wu W, Gao W, et al. Flying on point clouds: Online trajectory generation and autonomous navigation for quadrotors in cluttered environments[J]. Journal of Field Robotics, 2019, 36(4): 710-733.
- [20] Wurm K M, Hornung A, Bennewitz M, et al. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems[C]. Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation. 2010, 2.
- [21] Newcombe R A, Izadi S, Hilliges O, et al. Kinectfusion: Real-time dense surface mapping and tracking[C]//ISMAR. 2011, 11(2011): 127-136.
- [22] Oleynikova H, Millane A, Taylor Z, et al. Signed distance fields: A natural representation for both mapping and planning[C]//RSS 2016 Workshop: Geometry and Beyond-Representations, Physics, and Scene Understanding for Robotics. University of Michigan, 2016.
- [23] Oleynikova H, Taylor Z, Fehr M, et al. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 1366-1373.
- [24] Han L, Gao F, Zhou B, et al. FIESTA: Fast Incremental Euclidean Distance Fields for Online Motion Planning of Aerial Robots[J]. arXiv preprint arXiv:1903.02144, 2019.
- [25] Meier L, Tanskanen P, Fraundorfer F, et al. Pixhawk: A system for autonomous flight using onboard computer vision[C]//2011 IEEE International Conference on Robotics and Automation. IEEE, 2011: 2992-2997.

- [26] Duchon F, Babinec A, Kajan M, et al. Path planning with modified a star algorithm for a mobile robot[J]. Procedia Engineering, 2014, 96: 59-69.
- [27] Cheng P, LaValle S M. Resolution complete rapidly-exploring random trees[C]. Proceedings 2002 IEEE international conference on robotics and automation (cat. no. 02CH37292). IEEE, 2002, 1: 267-272.
- [28] Mellinger D, Kumar V. Minimum snap trajectory generation and control for quadrotors[C]//2011 IEEE International Conference on Robotics and Automation. IEEE, 2011: 2520-2525.
- [29] Ratliff N, Zucker M, Bagnell J A, et al. CHOMP: gradient optimization techniques for efficient motion planning[C]//Proceedings of the 2009 IEEE international conference on Robotics and Automation. IEEE Press, 2009: 4030-4035.
- [30] Gao F, Wu W, Lin Y, et al. Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 344-351.
- [31] Quigley M, Conley K, Gerkey B, et al. ROS: an open-source Robot Operating System[C]//ICRA workshop on open source software. 2009, 3(3.2): 5.
- [32] 胡春旭. ROS 机器人开发实战[M]. 北京: 机械工业出版社, 2018: 31-36.
- [33] Culjak I, Abram D, Pribanic T, et al. A brief introduction to OpenCV[C]//2012 proceedings of the 35th international convention MIPRO. IEEE, 2012: 1725-1730.
- [34] Rusu R B, Cousins S. 3d is here: Point cloud library (pcl)[C]//2011 IEEE international conference on robotics and automation. IEEE, 2011: 1-4.
- [35] Kam H R, Lee S H, Park T, et al. Rviz: a toolkit for real domain data visualization[J]. Telecommunication Systems, 2015, 60(2): 337-345.
- [36] Koenig N, Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator[C]//2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). IEEE, 2004, 3: 2149-2154.
- [37] Sherman M A, Seth A, Delp S L. Simbody: multibody dynamics for biomedical research[J]. Procedia Iutam, 2011, 2: 241-261.
- [38] Ramirez-Atencia C, Camacho D. Extending QGroundControl for Automated Mission Planning of UAVs[J]. Sensors, 2018, 18(7): 2339.
- [39] Shah S, Dey D, Lovett C, et al. Airsim: High-fidelity visual and physical simulation for autonomous vehicles[C]//Field and service robotics. Springer, Cham, 2018: 621-635.

- [40] Qiu W, Yuille A. Connecting computer vision to unreal engine[C]//European Conference on Computer Vision. Springer, Cham, 2016: 909-916.
- [41] Wang S, Mao Z, Zeng C, et al. A new method of virtual reality based on Unity3D[C]. 2010 18th international conference on Geoinformatics. IEEE, 2010: 1-5.
- [42] Forster C, Carlone L, Dellaert F, et al. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry[J]. IEEE Transactions on Robotics, 2016, 33(1): 1-21.
- [43] 高翔. 视觉 SLAM 十四讲[M]. 北京: 电子工业出版社, 2018: 133-136.
- [44] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [45] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2006: 404-417.
- [46] Rosten E, Drummond T. Machine learning for high-speed corner detection[C]//European conference on computer vision. 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, 2012.
- [48] Lucas B D. Generalized image matching by the method of differences[M]. Generalized Image Matching by the Method of Differences. 1985.
- [49] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6):381-395.
- [50] 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.
- [51] Hartley H O. The modified Gauss-Newton method for the fitting of non-linear regression functions by least squares[J]. Technometrics, 1961, 3(2): 269-280.
- [52] Moré J. The Levenberg-Marquardt algorithm: implementation and theory[M]. Numerical analysis. Springer, Berlin, Heidelberg, 1978: 105-116.
- [53] Mourikis A I, Roumeliotis S I, Burdick J W. Sc-kf mobile robot localization: A stochastic cloning kalman filter for processing relative-state measurements[J]. IEEE Transactions on Robotics, 2007, 23(4): 717-730.
- [54] Zhang Z. A Flexible New Technique for Camera Calibration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11):1330-1334.

- [55] Hirschmuller H. Accurate and efficient stereo processing by semi-global matching and mutual information[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, 2: 807-814.
- [56] Vassiliadis S, Hakkennes E A, Wong J, et al. The sum-absolute-difference motion estimation accelerator[C]//Proceedings. 24th EUROMICRO Conference (Cat. No. 98EX204). IEEE, 1998, 2: 559-566.
- [57] Oleynikova H, Taylor Z, Fehr M, et al. Voxblox: Building 3d signed distance fields for planning[J]. arXiv, 2016: arXiv: 1611.03631.
- [58] Hart P E, Nilsson N J, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2):100-107.
- [59] Rodriguez, Tang X, Lien J M, et al. An obstacle-based rapidly-exploring random tree[C]// IEEE International Conference on Robotics & Automation. 2006.
- [60] Dolgov D, Thrun S, Montemerlo M, et al. Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments[J]. The International Journal of Robotics Research, 2010, 29(5):485-501.
- [61] Zhou B, Gao F, Wang L, et al. Robust and efficient quadrotor trajectory generation for fast autonomous flight[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 3529-3536.
- [62] Rosmann C, Feiten W, Wosch T, et al. Efficient trajectory optimization using a sparse model[C]// Mobile Robots (ECMR), 2013 European Conference on. IEEE, 2013.
- [63] Geiger A. Are we ready for autonomous driving? The KITTI vision benchmark suite[C]. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2012.
- [64] Burri M, Nikolic J, Gohl P, et al. The EuRoC micro aerial vehicle datasets[J]. The International Journal of Robotics Research, 2016, 35(10): 1157-1163.

发表论文和参加科研情况说明

发表论文:

- [1] 侯永宏, 刘艳, 吕华龙, 等. 一种基于双目视觉的无人机自主导航系统[J]. 天津大学学报(自然科学与工程技术版), 2019, 52(12): 1262-1269.

专利申请:

- [1] 侯永宏, 吕华龙, 吴琦, 等. 一种基于移动网络的无人机控制信令和数据传输方法. 2018.10, 中国, 申请号: 201811183392.4
- [2] 侯永宏, 吴琦, 吕华龙, 等. 多旋翼无人机在复杂环境下的快速实时避障路径选择方法. 2018.10, 中国, 申请号: 201811183406.2
- [3] 侯永宏, 许贤哲, 吕华龙, 等. 一种基于深度卷积网络的双目深度估计方法. 2018.9, 中国, 申请号: 201811151718.5

参与的科研项目:

- [1] 企业委托项目-基于双目立体视觉的无人机避障与互联网通信系统开发, 2017.8-2018.9.

致 谢

经过在天津大学三年的学习，终于顺利完成了研究生课题的研究和论文的写作，我的学生生涯也即将落下帷幕。回想起本科伊始，因为兴趣选择了从事无人机飞行器的研究与设计。在石樑、张涛学长的带领下，我们一起建设实验室、申请全国大学生创新训练项目、参与创客马拉松和大学生电子设计竞赛，让我的大学时光充实而丰富多彩，在此，对与我一起参加比赛的队友，罗凯欣、胡春荣、王立鹏、杨昕、马浩博、张威力、李前同学表示感谢。研究生阶段，同样因为喜爱，我继续选择从事无人机自主感知与导航系统的研究。在侯老师的悉心指导下，我与实验室同学叶秀峰、吕晓冬、刘春源、吴琦、许贤哲，一同参与无人机自主导航避障项目的研发，从最初对课题方向的陌生，通过查找相关资料和书籍、多次的仿真实验和系统调试，逐渐使我对课题和所研究的方向有了更深的理解。在这个过程中使我受益匪浅，锻炼了我精益求精和勇于创新的精神。



图1 2015年本科阶段506四旋翼无人机实验室成员合影

论文的完成离不开我敬爱的导师、同实验室的兄弟姐妹、同学室友以及家人给予的学习上的指导以及生活上的帮助。首先，由衷地感谢导师侯永宏老师对我的谆谆教导，当我在科研中遇到难题与不解时，当我在写会议论文语句还很生涩时，当我在书写毕业论文过程中懈怠或是表达不畅时，侯老师总能实时地与我讨论研究进度与算法实现中的问题，耐心指导我前行的方向，教导我不断提高认真负责、严谨治学、独立思考问题的能力。



图 2 2018 年研究生阶段 540 实验室成员合影

其次，感谢师兄师姐，叶秀峰、李传坤、王涛、吕晓冬、刘春源、王利伟、王丽君、张浩元，感谢实验室的兄弟姐妹，吴琦、李世超、肖仁义、许贤哲、刘艳、李翔宇，感谢他们对于科研学习上的指导和生活中的帮助。感谢同宿舍的兄弟，王炜杰、吴琦、张凯昱，对我生活中的帮助。

此外，要特别感谢天津大学给予的学习资源和营造的良好学习氛围，实事求是的教训让我在不断前行中无时无刻不反省自己的不足和缺陷，使我可以在不断的成长中变成更好的自己。

最后，感谢家人在生活和精神上的鼓励和支持，感谢他们给了我选择的自由，给了我勇气和信心在学习中遇到困难时披荆斩棘，激励着我一直前行。