密级：非密

复杂环境小型固定翼无人机规避防撞研究

结题报告

1 概述

针对小型固定翼无人机低空飞行安全问题，构建了复杂环境小型固定翼无人机规避防撞方案。系统总体方案分硬件平台集成方案和功能算法集成方案。系统集成方案基于低空固定翼平台飞行平台的特性和约束（载荷空间要求、感知范围和平台挂载方式等）进行感知设备的平台选型与安装集成，包括确立满足需求的捷联感知方案，依据飞行特性和场景制定传感器选型布置方案，并提供具体的数据分析；功能算法方面，给出了基于可用于感知算法的障碍物表征方式和运动规划算法设计方案，并分析规划方法的可行性。

1.1 应用背景

小型固定翼无人机在低空飞行过程中面临低空线缆、建筑等复杂障碍物和地形等多种碰撞威胁。传统的基于视觉、激光等感知方案在探测范围、响应时间等方面无法满足高动态无人机系统动力学约束下的碰撞规避需求；同时，传统的低空防撞算法受制于机载算力、感知范围等限制，无法高效的生成高可靠性的规避轨迹与规避机动。两者结合到一起，使无人机在低空复杂场景下的障碍规避面临极大挑战。为解决这一问题，本项目提出了面向复杂低空环境的避障方案，基于面向运动规划约束的障碍物探测算法以及实时避障运动规划算法实现高效可靠的障碍感知与规定位避机动，保障无人机在复杂低空环境的飞行安全。

1.2 算法功能

整体算法功能通过优化配置的传感器实现对复杂低空空域进行建模，并基于该建模结果进行碰撞检测与障碍规避。其中，主要算法功能分为：

障碍物探测定位算法：通过集成机载毫米波雷达数据，实现飞行空域的实时障碍物殿宇获取，在此基础上，通过融合机载导航数据与毫米波雷达点云数据，实现飞行空域的实时建图，并基于点云地图结果生成障碍物建模地图，用于后续的碰撞规避功能。

碰撞规避算法：基于障碍物探测定位算法获得的障碍物地图进行全局路径规划，并基于实时点云障碍物地图，采用基于运动基元的实时避撞功能实现障碍物的有效规避。

1.3 创新点

1. 在障碍物建模过程中，融合机载毫米波雷达数据与实时导航数据实现障碍物地图的有效构建；

2. 面向低空方砖过程，基于感知约束构建了高效的运动基于建模与选取方法，实现高可靠、高效率的障碍规避功能。

1.4 适用条件

算法输入约束：整体算法功能以无人机机载毫米波雷达数据、机载导航数据为主要输入；

使用条件：适用于无人机在低空稠密障碍场景中的障碍物探测与规避功能。

性能边界：适用于低空固定翼无人机，有效载荷能力>1.5kg，满足 100-250 米有限范围的检测与规避能力。

2 国内外研究现状

2.1 研究现状

美国、欧洲等国家在无人机具备作战能力之后就开始着手其安全防撞技术的研究。通过防撞能力的建立，使其无人机系统具备在任何空域飞行、快速自主到达全球战场的能力。以美国为例，美国将安全防撞能力的形成分为短期目标和中长期目标两种。即短期内基于现有的地基空中交通监测系统构建无人机的安全防撞系统，形成初步安全防撞能力；中长期目标是研制机载的安全防撞系统。美军在 2013 年已经完成地基安全防撞技术开发和演示验证，并于 2014 年在多个空军基地进行列装。近十年间，林肯实验室、美国空军实验室等机构在 RQ-9、全球鹰、虎鲨等多种侦查、打击无人机平台上对视觉、雷达、ADS-B 等多种安全防撞系统进行了功能验证和测试。欧盟在 2009 年批准执行空中防撞计划(MIDCAS)，计划用 10-15 年时间，使无人机系统具备自主安全防撞能力。在 2015 年，欧洲航空局在 MIDCAS 计划支持下，首次成功测试其非合作安全防撞系统。美国海军在 2020 立项进行 MQ-4 无人机安全防撞系统的研制。测试包括雷达、ADS-B 等综合安全防撞能力，并对其安全性进行验证。综上，美国、欧洲等经过技术积累与测试验证，已经使无人机系统具备安全防撞能力，但该能力仅能够支持单一无人机系统在稀疏空域下的安全防撞保障。近十年国内包括科技部、军科委等对安全防撞技术开展立项，已经逐步形成了无人机安全防撞的技术体系。在空域飞行环境感知，规避路径规划与机动控制等方面的研究均取得了一定的进展。西工大、国防科大等开展了简单场景下的单一传感器的空域感知与障碍规避能力的测试。但目前国内面向无人机的安全防撞的系统集成与验证仍处于起步阶段，与国外发展相比，目前仍然缺乏复杂应用场

景下系统的验证和测试，系统的可靠性、精确性无法支撑无人机更加复杂的低空任务环境下的飞行安全保障。近年来，随着美国海军、空军等在无人机集群、有-无人机混合等开展研究，未来无人机的应用样式由单机、稀疏空域、简单任务场景转向多机协同、密集低空空域、复杂任务场景，这就要求无人机的安全防撞具备更加精准完备的协同感知能力，具备复杂协作的安全防撞策略以及更加敏捷的安全防护控制执行能力。 这些将给现有的无人机安全防撞技术体系带来颠覆性的挑战。

实现无人机安全避撞的关键技术包括：实现空间环境感知，利用各种传感器对空间环境和各类障碍物进行检测，获取可能存在碰撞威胁的动态或静态目标的检测环节；通过有效的估计、跟踪算法实现飞行空间目标的飞行状态估计，位置、速度的跟踪环节；根据目标的运动状态和空间分离规则对碰撞进行威胁程度的计算；对存在多个碰撞威胁目标时，对目标进行基于威胁程度的等级划分和先后排序等威胁评估等环节；根据威胁评估结果判定给出飞行管理决策，根据最小分离点(Closest Point of Approach, CPA)、碰撞时间(Time To Collision, TTC)等计算规避路径等规避决策与路径规划环节；通过机动输出，执行规避决策和规避路径跟踪的规避机动环节等。

在面向物理安全的感知规避技术发展中，规避是重要的响应阶段，目前发展了大量感知规避相关方法，主要包括基于数学规划的方法、基于路标图的方法、基于空间搜索方法、遗传算法等。传统的动态规划法是应用较为广泛的一种方法，通过将规划问题等效为多级决策问题。Radmanesh M 等提出一种基于有限视野的动态 MILP 航路规划算法，从而有效地克服了传统 MILP 在规划航路时计算量较大的缺点。Turnbull O 等将 MILP 与 MPC 相结合，作为参考避障航路规划器，用于离线训练改进的语言决策树（Linguistic Decision Trees，LDTs），然后再利用训练好的具有高实时性的 LDTs 进行在线航路规划。Wu J 等[57] 将城市环境下无人机持续跟踪目标航路规划问题建模为一个分布式 MPC 问题，并利用自适应草蜢优化算法在线解算无人机的最优控制输入，从而使规划出的航路能够兼顾避障和目标跟踪。Luo G 等[58] 将 RHC 与人工势场法相结合，利用 RHC 优化人工势场中的附加控制力，从而实现无人机的在线避障。宁芊等[59, 60]将 Markov 生存模型引入航路规划算法中，得到一个可用来评估路径点生存概率的航路规划问题模型，从而实现对最优航路的动态搜索。基于路标图的方法主要包括可视图法（VisibilityGraph）[61]和 Voronoi 法[62]等，如朱杰等[63]提出了一种改进型的 Voronoi 图构造模型，该模型通过引入威胁源的不可穿越区域边界，利用折中原理，在 Delaunay 三角网的基础上构建航迹拓扑空间。基于空间搜索方法是另一个重要的规避方式，常见的搜索方法包括 A*算法和 D*算法。如占伟伟等[64]将二维 A*算法扩展到三维空间

中，分别在二维平面内和垂直方向上规划航路。由于传统 A*仅能用于静态环境规划，因此许多学者对其进行了改进，其中最为典型的为 D*算法[63]。D*算法与 A*算法的区别在于，当环境改变时，或无人机探测到的周围环境信息变化时，对路径代价值进行相应更新。另外一种空间搜索方法是快速随机搜索树(RRT) 方法,能够实现快速、有效的高维空间路径搜索。如 Lin 等[65]通过 Close-Loop RRT, 将闭环路机动控制与 RRT 路径搜索相结合, 通过闭环机动控制飞行轨迹预测实现更小的预测误差和碰撞风险, 路径规划考虑目标机动约束。随着强化学习的发展，很多学者将其成功的运用在了无人机的感知与规避过程中, La 等[66]提出了一种基于强化学习的障碍物/捕食者规避方法, 通过将高层的 Q-learning 决策过程与低级的无人机集群控制结合, 实现了对障碍物的有效规避。在其后续的成果中[67] , 完全基于 Q-learning 方法实现了障碍规避功能, 其中碰撞规避功能被建模为一项激励函数。在 Long 等[68]基于深度强化学习方法（DeepReinforcement Learning，DRL）, 构建深度神经网络模仿基于几何关系的障碍规避方法 ORCA 从而实现在多智能体设置下的碰撞规避。在文献[69]提出的类似的深度强化学习框架中, 通过对学习策略和激励函数的改进, 实现了比 ORCA 更好的规避性能。Lyu 等[70]基于集中式的强化学习框架和分布式的控制策略大大提高了强化学习的效率。除此之外，由于 Q 学习算法的状态空间和动作空间均为离散的，因此其规划航路的可飞性较差，且难以应付动态威胁。针对此缺陷，研究者提出将深度学习和强化学习相结合，组成深度强化学习算法，以满足状态空间或动作空间连续化的需求。DRL 的开端是深度 Q 网络（Deep Q Network，DQN），由 DeepMind 公司于 2015 年提出[71]。Yan 等人[72]在路径规划中引入了不同改进型的 DQN，取得了较好的效果，但由于 DQN 的动作空间仍然是离散形式的,因此规划的路径质量仍有进一步提升的空间。将深度学习和强化学习相结合，组成深度强化学习算法，以满足状态空间或动作空间连续化的需求。DRL 的开端是深度 Q 网络（Deep Q Network，DQN），由 DeepMind 公司于 2015 年提出[71]。Yan 等人[72]在路径规划中引入了不同改进型的 DQN，取得了较好的效果，但由于 DQN 的动作空间仍然是离散形式的，因此规划的路径质量仍有进一步提升的空间。通过以上分析，合作式感知方式的前提是本机与空域其他飞行器能够建立基于应答机制或广播的合作式信息交互通道，难以保证在真实飞行环境中非合作目标存在情况下的感知与规避，因此，单一的合作式感知与规避技术难以真正保证无人机的空域飞行安全。为了应对以上问题，诸多研究者将目光转向非合作式是感知。在非合作环境感知过程中, 无人机搭载的常用非合作目标感知设备包括雷达、激光雷达、超声波等主动感知设备以及光电、红外等被动感知设备。Accardo[83]描述了一种基于雷达的 SAA 系统, 通过 Kalman 滤波方法实现对目标飞行器的状态估计,并通过有人机的

飞行测试证明了其算法的有效性。Owen 等[84]设计了一种适用于大、中型无人机感知与规避任务的雷达系统，并通过提取信号特征和误差精度分析验证了该雷达的空域目标感知能力. Newmeyer 等[85]设计了一种适用于小型无人机的毫米波雷达空域感知系统，通过一系列的信号处理方法获得目标的距离和方位角度后，用 RANSAC 方法实现对目标的跟踪。在检测中通常使用单静态脉冲调制雷达有时也会用到连续波(CW)技术[86, 87]。针对大中型无人机系统，美国率先启动了地基感知与规避系统的研究,并在 2014 年前完成了地基感知与规避系统的验证与列装[88]。2018 年 Sahawneh 等人[89]提出了一个完整的、概念证明的小型无人驾驶飞机系统的感知和回避解决方案，包括一个小型低成本的地面雷达系统、多目标跟踪和估计、碰撞检测和回避计划。2019 年 Meer 等人[90]研究了无人机通过蜂窝技术将其位置信息传输到云，而 GBSAA 基站通过 ADS-B 技术访问聚合信息并将其发送到启用 ADS-B 的飞行器。并进行了分析和仿真研究，研究了不同网络参数(如无人机数量和 ADS-B 启用飞行器)下 ADS-B 信息碰撞概率。Fasano 等人[91]研究了一种基于雷达/光电数据融合的多传感器障碍物检测与跟踪系统的硬件/软件实现和飞行结果。Chen 等人[92]研究了一种新的基于主动感知的飞行机器人在动态环境中的避障模式。他们没有融合多个传感器来扩大视场(FOV)，而是引入了一种替代方法，利用具有独立旋转自由度的立体摄像机来主动感知障碍物。

针对小微型无人机，Alvarez 等[93] 提出了一种基于单目视觉的四旋翼无人机感知与规避系统，通过融合单目传感器感知信息和自身运动测量基于机载实时处理系统构建地图的方式，实现在室内场景中反应式的碰撞规避。Lyu 等[94]设计了一种单目视觉感知与规避系统，包括单目相机以及机载处理系统，不依赖深度信息,仅基于角度感知信息实现了小型无人机对空中目标的有效障碍规避。为了解决单目相机在深度估计方面的不足，基于双目相机[95]、深度相机[96]、激光雷达[97] 等感知方法和系统得到了广泛的研究。Gageik 等[98]提出了一种低成本多传感器障碍规避系统，通过搭载红外、视觉、激光等多种类型传感器和机载处理系统，能实现在复杂火灾场景中的搜救任务并保证障碍规避。

2.2 发展趋势

近年来，随着美国海军、空军等在无人机集群、有-无人机混合等开展研究，未来无人机的应用样式由单机、稀疏空域、简单任务场景转向多机协同、密集低空空域、复杂任务场景，这就要求无人机的安全防撞具备更加精准完备的协同感知能力，具备复杂协作的安全防撞策略以及更加敏捷的安全防护控制执行能力。 这些将给现有的无人机安全防撞技术体系带来颠覆性的挑战。

2.3 国内外对比分析

目前国内面向无人机的安全防撞的系统集成与验证仍处于起步阶段，与国外发展相比，目前仍然缺乏复杂应用场景下系统的验证和测试，系统的可靠性、精确性无法支撑无人机更加复杂的低空任务环境下的飞行安全保障。

2.4 必要性分析

随着巡飞器、小型固定翼无人机等在军、民多种领域的逐步推广应用，无人机系统面临的低空威胁逐渐增大。特别是面向军事行动中的超低空突防等任务需求，亟需要形成无人机低空感知与避撞能力。

3 超低空无人机障碍规避问题描述

3.1 感知建模

传感机制应保证操作空间的可靠覆盖

$$F_u = \left(\bigcup_{S_p} F_p\right) \cup \left(\left(\bigcup_{S_b} F_b\right) \cap \left(\bigcup_{S_r} F_r\right)\right) \cup F_g$$

其中 F 代表角度、距离和全局感知信息，S 为传感器配置。

感知信息足够精确，具备作为控制决策参考依据

$$\min_{\mathbb{S}} \mathbb{E}\left(\|\tilde{\mathbf{x}}_{o,k}\|^2 | \mathbf{x}_{i,k}, \mathbf{y}_{1:k}, \mathbb{S}\right)$$

在以协方差为不确定性建模的随机过程中，可建模为

$$\begin{aligned}
&\min_{\mathbb{S}} \mathbf{tr}\left(\mathbf{P}_{o,k}\right) \\
&\mathbf{P}_{o,k} = \left(\sum_{j \in \mathbb{S}} \mathbf{P}_{j,k}^{-1}\right)^{-1}, \\
&\hat{\mathbf{x}}(k) = \mathbf{P}_{o,k} \sum_{j \in \mathbb{S}} \mathbf{P}_{oj,k}^{-1} \hat{\mathbf{x}}_j(k).
\end{aligned}$$

3.2 规避建模

满足各类碰撞分离指标，如最小分离距离

$$\|\mathbf{x}_{o,k} - \mathbf{x}_{i,k}\| \geq d, k \in \mathbb{Z}^+,$$

在随机空间中，可进一步定义碰撞概率：

$$p_{o,k} \triangleq p(\mathbf{x}_{o,k} \in \mathcal{S}_{i,k}^d | \mathbf{y}_{1:k}) = \int_{\mathcal{S}_{i,k}^d} p(\mathbf{x} | \mathbf{y}_{1:k}) d\mathbf{x}$$

有效的碰撞检测机制和可靠的决策结论可建模为

$$p_{o,k} \approx \frac{1}{N}\sum_{l=1}^{N}\delta_{r_c}\left(\mathbf{x}_{o,k}^{(l)}\right) \qquad \delta_d\left(\mathbf{x}_{oi}^{j}\right)=\begin{cases}1 & \|\mathbf{x}_{i,k}-\mathbf{x}_{o,k}^{(l)}\|\le d\\ 0 & \|\mathbf{x}_{i,k}-\mathbf{x}_{o,k}^{(l)}\|> d\end{cases}$$

充分考虑任务可行效率和规避碰撞

$$\begin{aligned}\min_{\mathbf{x}_{i,k}} \quad & \sum_{t=k}^{k+h}\|\mathbf{x}_{i,t}-\mathbf{x}_{i,t}^{r}\|^2\\ \text{s.t.} \quad & \|\mathbf{x}_{i,t}-\mathbf{x}_{o,t}\|>d, \quad k\in[t,t+h]\end{aligned}$$

3.3 规避建模

应根据无人机型号和约束条件下达控制命令。 给出动态模型为

$$\mathbf{x}_{i,k+1}=f(\mathbf{x}_{i,k},\mathbf{u}_{i,k})$$

对于任意 $k\in\mathbb{Z}^{+}$，

$$\mathbf{x}_{i,k}-\mathbf{x}_{i,k}^{p}\in\mathcal{C} \qquad ,\ \mathbf{x}_{i,k}\in\mathcal{X} \qquad \mathbf{u}_{i,k}\in\mathcal{U}$$

应优化实现任务和避碰

$$\begin{aligned}\min \quad & \sum_{t=k}^{k+h}\|\mathbf{u}_{i,k}\|^2\\ \text{s.t.} \quad & \mathbf{x}_{i,k+1}^{p}=f(x_{i,k},\mathbf{u}_{i,k})\\ & \mathbf{x}_{i,k+1}\in\mathcal{X}\\ & \mathbf{u}_{i\,k}\in\mathcal{U}\end{aligned}$$

4 超低空无人机障碍规避算法整体框架

算法整体有两个个模块组成，分别是障碍物构建模块和障碍规避模块，其整体流程图如下图所示

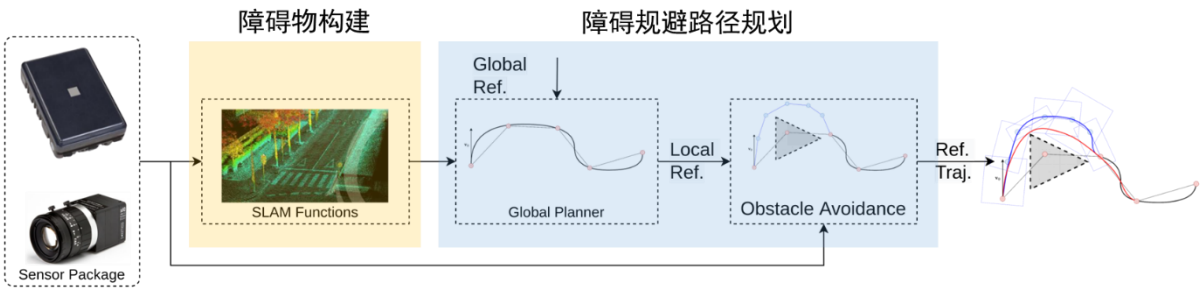


图 1 低空固定翼无人机障碍规避流程图

4.1 算法输入输出

输入：雷达、IMU 的量测数据。

输出：路径规划控制输出速度，航向等。

4.2 算法模块组成

障碍物构建模块：功能是基于雷达感知数据生成障碍物点云地图，输入是毫米波雷达点云与 IMU 数据，来自雷达与 IMU 传感器，输出是障碍物占据栅格地图，发送给障碍规避路径规划模块；

障碍规避模块：功能是基于点云地图与实时点云，完成障碍路径跟踪与障碍规避，输入是点云地图与实时点云，来自毫米波雷达数据与点云地图数据，输出是速度和偏航角度控制，发送给飞控系统；

5 超低空无人机障碍规避算法关键技术

5.1 障碍物构建技术

障碍物构建整体算法流程如图所示。

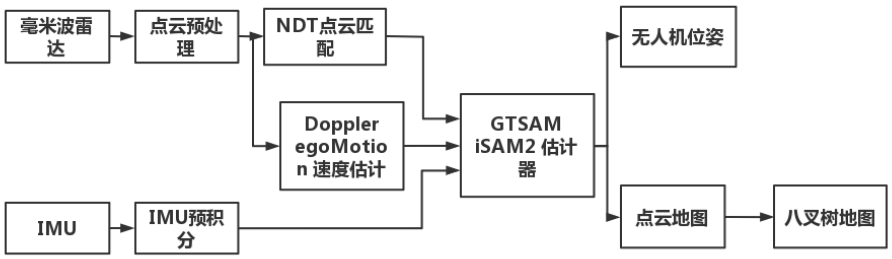


图 2 低空障碍物构建流程图

5.1.1 点云预处理技术：

毫米波雷达由于数据获取方式和数据结构的原因，获得的点云经常收到严重异常值和噪声的污染，本文利用一种针对室外场景低质量点云数据的鲁棒整合算法，算法主要包括两个步骤①异常值滤波②噪声平滑。首先设计了一个基于连通性的方案来来评估离群值，从而检测稀疏离群值。同时，采用聚类方法进一步去除小的密集异常值。这两种离群值去除方法对邻域大小的选择和离群值的水平都不敏感。随后，提出了一种基于鲁棒部分排序的噪声点法向量估计方法，这是噪声平滑的基础。因此，此方法能够快速地平滑噪声，同时可以保留点云数据中的较为明显的特征点，并且在各种户外场景的点云上评估了所提方法的有效性。

5.1.2 雷达自运动估计技术：

通过毫米波雷达点云中的探测物径向速度信息，构建最小二乘问题求解估计雷达自运动速度。鉴于点云数据中有很多噪点和来自运动物体的点，用所有的测量方法求解方程都容易出错，环境不能假设是静态的，由于噪声、反射或鬼影而产生的异常值必须被删除。随机样本和一致性(RANSAC)应用于最小二乘(LSQ)技术，是一种常见的离群值抑制技术。RANSAC 是一种迭代方法，它从一组数据点中去除异常值。在每次迭代中，抽取一组随机的数据点来生成一个假设。接下来，用剩下的数据点对这个假设进行检验。所有通过这一检验的数据点都被认为是假设的内点。本文提出的方法并不一定依赖于毫米波雷达。只要给出目标的方位角、仰角和速度，每次扫描检测到几个目标，就可以应用。

自我速度估计基于单次雷达扫描，每次扫描由雷达帧中的 3D 点和从多普勒频移获得的相应速度组成。自我速度的估计可以基于最小平方解进行。

$$\mathrm{r}^r = \frac{p^r}{\left\| p^r \right\|}$$

给定目标位置 p^r，通过归一化得到方向 r^r，测得的多普勒速度 v_d^r 是方向 r^r 与目标速度 v^r 的标量积:

$$v_d^r = \mathrm{r}^r \cdot v^r = r_x^r v_x^r + r_y^r v_y^r + r_z^r v_z^r$$

给定一组 N 个测量值，并应用矩阵表示法得到:

$$\begin{bmatrix} v_{d,1}^r \\ v_{d,2}^r \\ \vdots \\ v_{d,N}^r \end{bmatrix} = \begin{bmatrix} r_{x,1} & r_{y,1} & r_{z,1} \\ r_{x,1} & r_{y,1} & r_{z,1} \\ \vdots & \vdots & \vdots \\ r_{x,N} & r_{y,N} & r_{z,N} \end{bmatrix} \begin{bmatrix} v_x^r \\ v_y^r \\ v_z^r \end{bmatrix}$$

利用线性最小二乘(LSQ)解可得到如下解：

$$\tilde{v}^r = (H^T H)^{-1} H^T y^r$$

用所有的测量方法求解此最小二乘问题都容易出错。环境不能假设是静态的，由于噪声、反射或重影而产生的异常值必须被删除，因此在前述的点云预处理中已经将此问题解决。

5.1.3 多传感器融合 SLAM 技术：

在基于点云匹配的 SLAM 系统中，对初值的选取尤为关键，传统的基于点云配准的 SLAM

算法中位姿是初始化后迭代更新的，缺乏其他观测信息的融合，本文中使用 IMU 为点云配准提供初始先验位姿信息，并通过 IMU 预积分的方式提高位姿输出频率，并通过在 gtsam 因子图模型中加入雷达速度因子来进一步优化位姿结果，实现复杂环境下 4D 毫米波雷达与 IMU 融合的 SLAM 系统。

在多传感器融合的 SLAM 框架中，引入一个新的雷达速度因子。首先进行雷达量测模型的建模，

$$\tilde{v}^r(t) = v^r(t) + b^r(t) + \eta^r(t)$$

雷达实测线速度 $\tilde{v}^r$ 中，主要包括时变白噪声 η^r ，真实线速度 v^r 和雷达偏置 b^r 。前文所设计的模块测量了在世界坐标系下表示的三维瞬时线性速度。定义一个因子，考虑到两个时刻之间的速度和位置变化量，并且需要初始化旋转，才能使用雷达因子，因为采用的是 IMU 融合方案，因此采用 IMU 的旋转数据。

$$_W\mathrm{v}(\mathrm{t}+\Delta\mathrm{t})=\mathrm{R}(\mathrm{t}+\Delta\mathrm{t})_B v_{WB}(\mathrm{t}+\Delta\mathrm{t})$$
$$_W\mathrm{p}(\mathrm{t}+\Delta\mathrm{t})=_W\mathrm{p}(\mathrm{t})+_W\mathrm{v}(\mathrm{t})\Delta\mathrm{t}+1/2\{_W\mathrm{v}(\mathrm{t}+\Delta\mathrm{t})-_W\mathrm{v}(\mathrm{t})\}\Delta\mathrm{t}$$

由于被测速度是只影响单个节点的绝对因子，而位置变化则是由相邻节点上的两个速度计算出的相对测量值，因此瞬时速度和相对位置的测量模型推导如下。

$$\tilde{v}_j = v_j + \delta v_j = R_j(\tilde{v}_j^r - b_j^r)$$

$$\Delta\tilde{p}_{ij} = \Delta p_{ij} + \delta p_{ij} = 1/2R_i^T R_j(\tilde{v}_j^r - b_j^r)\Delta t - 1/2(\tilde{v}_j^r - b_j^r)\Delta t$$

另外当获得新估计值时，雷达的偏置需要更新，因此也需要建模分析。最后对雷达因子中的残差和雅克比进行计算，得到位置速度和偏置的残差。

5.1.4 障碍物建模与地图转换

点云图有几个缺点：①规模大。提供很多不必要的细节，放在地图里浪费空间。②处理重叠的方式不够好。在构建点云时，直接按照估计位姿拼在了一起。在位姿存在误差时，会导致地图出现明显的重叠。③难以用于导航。点云地图无法得知哪些地方可通过，哪些地方不可通过。OctoMap 可以解决上面的问题的，把点云地图转化为八叉树存储的地图。即该算法的输入为点云地图，输出为八叉树地图。它可以压缩、更新地图，并且分辨率可调，相比点云，能够省下大把的空间。

5.2 障碍规避技术

整体路径规划算法如图 1 所示。全局运动规划器负责寻找到目标的无碰撞轨迹。局部运动规划负责避免与新感知到的障碍物发生碰撞，以下称为避障算法。 全局规划器利用 SLAM 获得的环境地图，给避障算法一个局部目标，即沿着全局路径的位置。 避障的主要目标是避开障碍物，达到局部目标是次要的，因为全局规划器最终会更新这个局部目标。 此外，不能保证避障会达到局部目标，因为它只对瞬时点云数据进行操作。

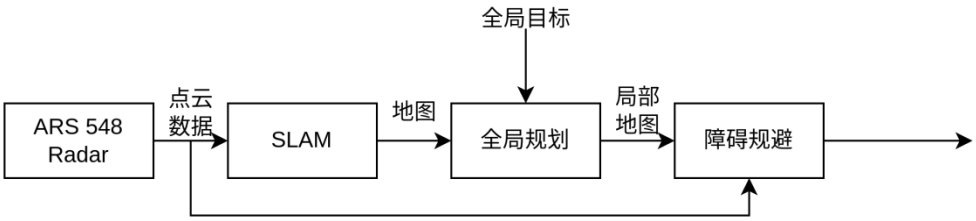


图 3 规避过程流程图

由于其动力学的性质和复杂性，使用固定翼飞机在未知的杂乱环境中自主飞行比旋翼飞行器更加复杂。 本项目采用基于反应式的规避策略，直接通过获得的点云数据，并基于运动基元生成规避轨迹。

5.2.1 无障碍场景的全局轨迹规划

给定需要经过的路径点，采用多项式轨迹规划来生成全局轨迹。轨迹要满足一系列的约束条件：经过设定的起点、终点和某些路径点；相邻轨迹段连接处平滑（位置连续、速度连续等）；最大速度、最大加速度约束。 通常满足约束条件的轨迹有无数条，而实际问题中，往往需要一条特定的轨迹，所以又需要构建一个最优的函数，在可行的轨迹中找出“最优”的那条特定的轨迹。用 n 阶多项式表示空间中两点的轨迹，即：

$$p(t) = p_n t^N + p_{n-1} t^{N-1} + \cdots + p_0 = \sum_{n=0}^{N} p_n t^n$$

对于任意时刻 t，可以根据参数计算出轨迹的位置、速度、加速度等。

$$\begin{cases} v(t) = p'(t) \\ a(t) = p''(t) \\ \text{jerk}(t) = p^{(3)}(t) \\ snap(t) = p^{(4)}(t) \end{cases}$$

轨迹规划的目的：求轨迹的多项式参数 $p_0, p_1, \ldots, p_n$ 。

多项式的优化代价函数为:

$$J = \int_0^T c_0 p(t)^2 + c_1 p^{'}(t)^2 + c_2 p^{''}(t)^2 + \cdots + c_N p^{(N)}(t)^2 dt$$

其中 T 是轨迹段的遍历时间。这个函数可以写成矩阵形式，其中 p 是一个多项式系数向量，Q 是一个代价矩阵。

$$J = p^T Q p$$

本课题以最小化 snap 为目标函数，令 c_4 不为 0，其余系数均为 0。则目标函数为，其中决策变量是多项式轨迹的系数。

$$\min_p p^T Q p = \min\left(p^{(4)}(t) \right)^2$$

5.2.2 基于地图的实时避障算法

由于无人机实时构建的地图是局部的，且该地图随着无人机的飞行实时的更新，因此无人机需要基于局部地图实时的重规划轨迹以避开原始轨迹上的障碍。无碰撞轨迹生成策略可以细分为三种：基于搜索和平滑相结合的方法、基于优化的方法和基于运动基元的方法。将局部地图与无碰撞轨迹生成策略相结合，提出了两种实时局部轨迹重规划方法：一是采用 RRT*算法在局部地图中搜索安全路径，然后利用 B 样条对路径进行平滑得到局部安全轨迹；二是直接采用 B 样条优化的方法实时重规划得到平滑且安全的局部轨迹。将局部轨迹重规划问题解耦为前端路径搜索和后端轨迹拟合。基于机载传感器构建的局部地图，首先用基于 RRT*算法在地图中寻找路径点。然后采用均匀 b 样条拟合路径点，将由线段组成的路径转换为平滑且安全的轨迹。该方法随着四旋翼的运动实时的更新局部地图，并以一定的频率不断的进行局部轨迹重规划，直至到达目标点。

（1）基于快速探索随机树（RRT*）算法的路径点生成

首先初始化树产生第一个节点 x_{init}，在每一次循环过程中，产生一个随机点 x_{rand}。随机点的生成是任意的，即可以在整个状态空间内。在产生随机点后，遍历随机树中的每一个节点，计算每一个节点与随机点之间的距离，找出距离此随机点最近的节点，记为 $x_{nearest}$。在 $x_{nearest}$ 与 x_{rand} 连线方向扩展产生新的节点 x_{new}。快速探索随机树算法在新节点 x_{new} 附近的规定范围 r 内寻找相邻节点，作为替换 x_{new} 父节点的备选。最终选择使得路径代价最小的点最为 x_{new} 的父节点。然后，快速探索随机树算法对树进行重新布线使所有节点的路径代价最小。

（2） 基于均匀 b 样条的轨迹平滑

基于均匀 b 样条的轨迹平滑的部分采用五次均匀 b 样条来拟合快速探索随机树生成的路径控制点，得到平滑的轨迹。

$k-1$次 b 样条可由如下公式表示：

$$p(t)=\sum_{i=0}^{n} p_i B_{i,k}(t)$$

$p_i \in R^3$ 是 t_i 时刻 b 样条曲线的控制点，$B_{i,k}(t)$ 是基函数。为了使轨迹能够被无人机跟随，轨迹必须连续到位置的四阶导数。因此使用五次均匀 b 样条曲线来表示轨迹。均匀 b 样条控制点之间有固定的时间间隔 Δt 。对于 5 次贝兹曲线样条，在时间 $t \in [t_i, t_{i+1})$ ， $p(t)$ 的值仅取决于 6 个连续的控制点 $[p_i, p_{i+1}, \cdots, p_{i+5}]$ 。令:

$$u(t)=\frac{t-t_i}{t_{i+1}-t_i}=\frac{t-t_i}{\Delta t}, t \in [t_i, t_{i+1}]$$

则轨迹可以表示为：

$$p(u(t))=\begin{pmatrix}1\\ u\\ u^2\\ \vdots\\ u^5\end{pmatrix}^T M^6 \begin{pmatrix}p_i\\ p_{i+1}\\ p_{i+2}\\ \vdots\\ p_{i+5}\end{pmatrix}$$

其中 M^k 是一个大小为 $k \times k$ 的矩阵

$$M_{i,j}^{k}=\frac{1}{(k-1)!}C_{k-1}^{k-1-i}\sum_{s=j}^{k-1}(-1)^{s-j}\times C_k^{s-j}(k-s-1)^{k-s-i}$$

$$C_n^i=\frac{n!}{i!(n-i)!}$$

令 $P=[p_i, p_{i+1}, \cdots, p_{i+5}]^T$ ，可以计算轨迹上任意时刻的速度和加速度。

$$v(t)=p'(u(t))=\frac{1}{\Delta t}\begin{bmatrix}0 & 1 & u & u^2 & \cdots & u^{k-2}\end{bmatrix}M^k P$$

$$a(t)=p''(u(t))=\frac{1}{\Delta t^2}\begin{bmatrix}0 & 0 & 1 & u & \cdots & u^{k-3}\end{bmatrix}M^k P$$

我们在构建的八叉树地图上使用 RRT*算法搜索初始路径点，然后对其进行 b 样条拟合得到平滑的轨迹，如图 4 所示。

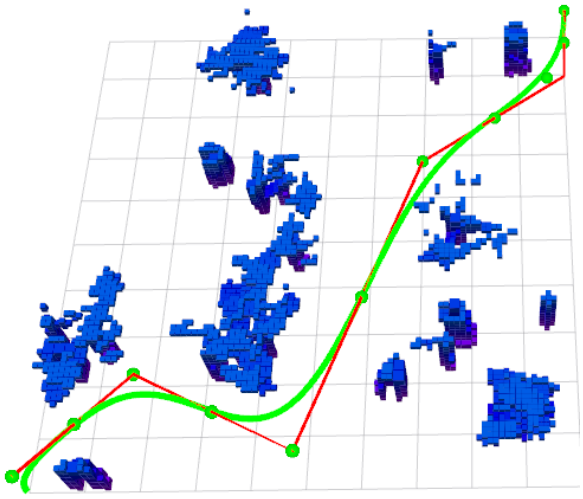


图 4 轨迹规划策略

由于相机的感知范围有限，我们设计了一个局部轨迹规划策略。在飞行开始前，设置目标点，规划到达目标的全局轨迹，此时认为环境是无障碍的。在飞行过程中，实时构建局部地图，在线重规划局部轨迹。利用一个定时器，每隔 t 秒进行一次局部轨迹重规划，即每隔 t 秒在全局轨迹上选择当前时间对应的位置点作为局部重规划的目标点，以确保局部规划的轨迹不会偏离全局轨迹。在局部重规划中，首先基于构建的局部地图，使用 RRT* 算法得到路径点，然后将路径点作为控制点插入 b 样条中。如果局部规划器的目标点被占用（如有障碍）或不在局部地图内，则放弃此次规划，并复制最后一个路径点插入 b 样条中。这是为了在局部规划没有生成新的路径点的时候无人机能够在最后的控制点处保持悬停。结果如图 5 所示。

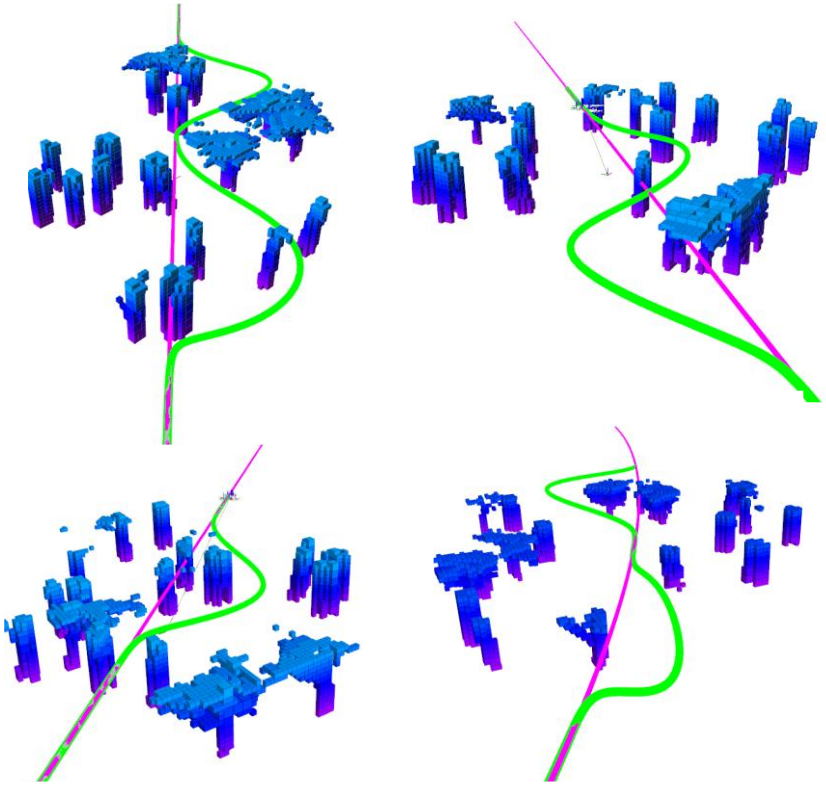


图 5 实时轨迹重规划结果的示意图

5.2.3 基于运动基元的应急规避控制

（1）规避基元生成

许多运动规划算法在获得参考轨迹之前会多次预测机器人的动力学模型，但这在实时中是不可行的，计算能力有限且动力学模型复杂。最常见的方法是降低动力学模型的复杂性，使其可以实时执行，或者离线运行计算量大的动力学模型，并将其存储在库中以供实时使用。前一种方法使运动规划器能够根据需要传播动力学，但代价是由于近似而导致建模不准确，而后一种方法使运动规划器能够使用复杂的动力学模型，但将运动计划限制于运动基元库中。

本项目避障方法使用预先计算的轨迹库。离线计算的运动原语库使该策略能够利用无人机的大部分飞行包线，而无需实时求解复杂的运动方程。运动基元被约束为恒定的速度、滚动、俯仰和控制输入。在假设无侧滑影响的情况下，优化目标是最小化控制量。 每个运动基元由其速度、偏航率和爬升率定义，这些在整个轨迹中都是恒定的。轨迹库中包含的运动基元包括：

- 直线和水平飞行
- 爬升/下降
- 倾斜转弯
- 螺旋倾斜转弯

所有的运动基元都通过求解轨迹优化问题

$$\min\ J \triangleq \int_0^{t_f} \left(\delta_a^2 + \delta_e^2 + \delta_r^2 + \left(\frac{\omega_T}{8000} \right)^2 \right) dt$$

$$\begin{aligned}
&V = V_d, \quad \dot{\phi} = \dot{\theta} = 0, \quad \dot{\psi} = \dot{\psi}_d, \quad \dot{z} = \dot{z}_d \\
&\delta_a \in [-\delta_{a_{max}}, \delta_{a_{max}}], \qquad \dot{\delta_a} = 0^\circ\ \mathrm{s}^{-1} \\
&\delta_e \in [-\delta_{e_{max}}, \delta_{e_{max}}], \qquad \dot{\delta_e} = 0^\circ\ \mathrm{s}^{-1} \\
&\delta_r \in [-\delta_{r_{max}}, \delta_{r_{max}}], \qquad \dot{\delta_r} = 0^\circ\ \mathrm{s}^{-1} \\
&\omega_T \in [\omega_{T_{min}}, \omega_{T_{max}}], \qquad \dot{\omega}_T = 0\ \mathrm{rpm/s}
\end{aligned}$$

成本函数最后一项的分母中的权重使得推力的惩罚在幅度上与控制面输入上的惩罚大致相似，因此它不会主导成本。权重与控制输入单元、控制面的弧度和推力的转速成正比。控制输入副翼 δa，升降舵 δe，方向舵 δr，和推力 ωT，可以在它们的物理极限内取任何值，但是对它们的导数施加约束，以便它们保持不变。 所需速度 Vd 设置为恒定的 21 ms-1，这是这架飞机的正常巡航速度。 滚动和俯仰率 $\dot{\phi}$ 和 $\dot{\theta}$ 设置为零，以便机动稳定。

每个运动基元都是针对所需偏航率 $\dot{\psi}_d$ 和爬升/下降率 $\dot{z}_d$ 的不同组合获得的。 例如，直线和水平飞行将这两个值都设置为零。 为了保持机动空间的大小紧凑，以增量值求解基元。偏航以 10° 为增量，爬升/下降率 以 1 m/s 为增量从 -2 到 2 m/s，总共生成 115 个运动基元。直线平飞、4 次爬升/下降、22 次倾斜转弯、88 次螺旋倾斜转弯。 定义每个操作的状态和控制输入存储在控制系统使用的查找表中。 查找表还包括悬停原语的状态和控制输入，它们是通过解决飞机运动方程来确定的。

（2）感知约束的规避生成

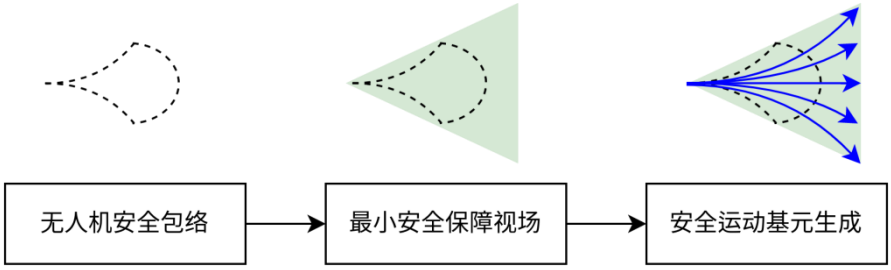


图 6 感知约束生成过程

在飞机巡航时，我们使用库中的修剪原语来避开障碍物并引导飞机朝向目标。 使用来自感知数据的点云，我们提出了飞机视野 (FOV) 内的运动计划。首先，我们计算一组从当前飞机位置到 FOV 边缘不同位置的运动基元。 FOV 边缘的目标位置由图中的红点表示，在这些位置结束的修剪轨迹由紫色箭头表示，其中箭头的方向是沿轨迹的偏航 . 如图所示，并非所有最终目标位置都可以使用动态可行的修剪原语到达。 最终目标位置固定在 FOV 的边缘，因此它们的位置是飞机姿态（和位置）的函数。

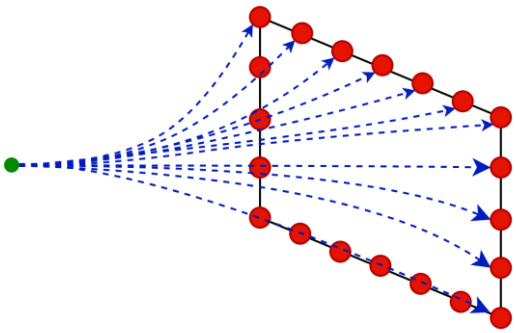


图 7 视觉边界约束的无人机运动基元

基于如图所示的

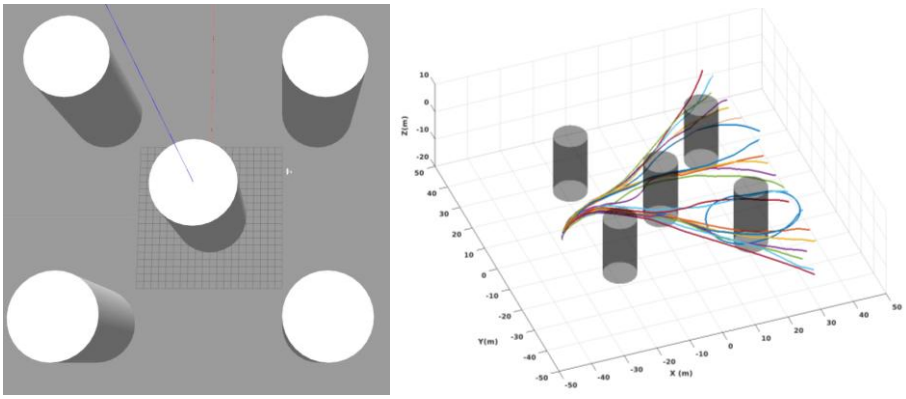
不同尺度(0.01 ~ 25m)

不同密度（5% ~ 15%）

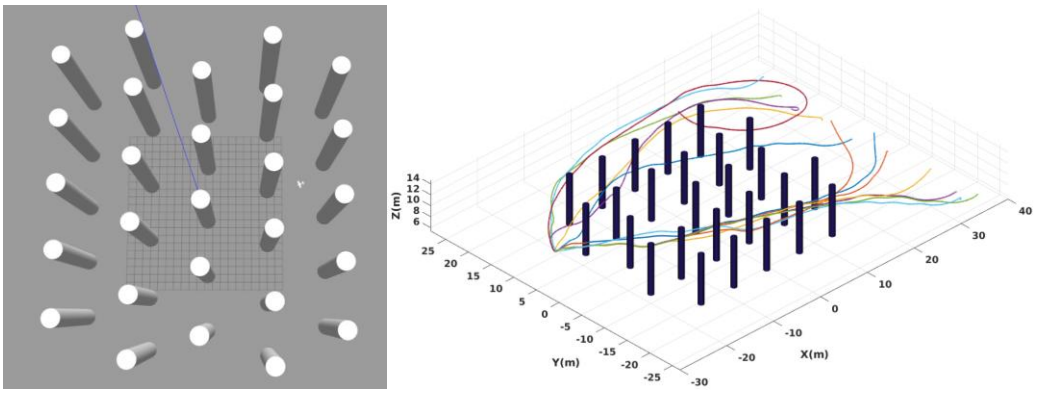
不同速度 （15m/s ~ 25m/s）

不同传感器配置（视场 70°~90°，探测距离 50~ 150m）

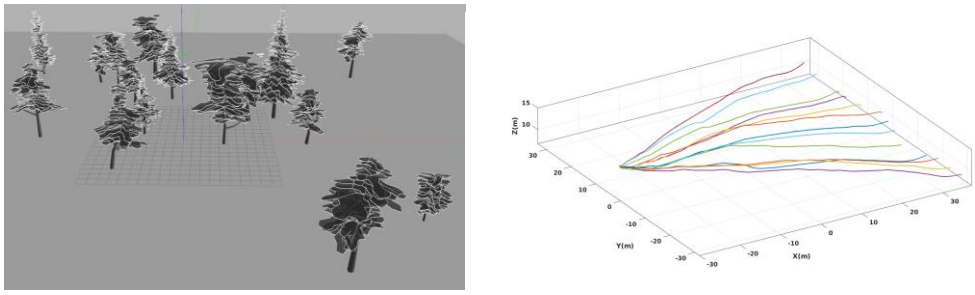
证明在飞机巡航速度<23m/s，基于探测视场 70 度，探测距离>60m 的情况下，实现障碍物密度≤10% ，尺度>0.1m 场景的 100% 碰撞规避。如图



场景 1



场景 2



场景 3

图 8 多场景应急机动示意图

6 超低空无人机障碍规避算法试验验证

6.1 试验大纲

整体测试流程如下图所示

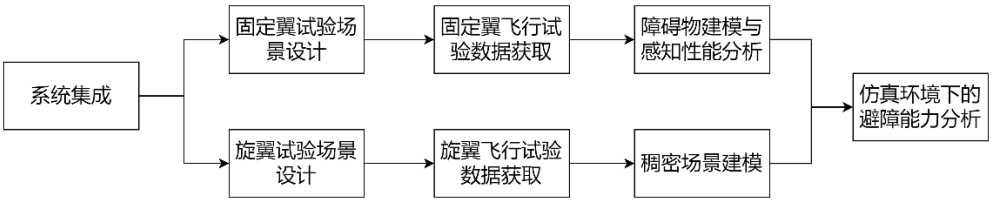


图 9 整体试验流程

首先，基于无人机低空防撞系统总体设计方案完成系统的软、硬件集成；根据固定翼无人机和旋翼无人机的飞行特点，分别针对稀疏空域下的线、塔、地形、建筑等目标以及稠密超低空飞行中的建筑群、峡谷等环境进行飞行试验设计；通过飞行试验获取关于多种典型飞行场景下的障碍目标的感知数据并构建障碍物数据库；基于感知数据进行障碍物飞行场景与障碍物建模；最终讲障碍物场景与目标导入仿真环境中进行感知算法性能与避障算法能力的分析。

以下按照系统整体功能测试和技术指标测试两部分介绍测试方法。

6.1.1 无人机试验平台

考虑无人机低空飞行场景特点，以及平台在低空飞行的可行性，本项目基于两类平台，即固定翼无人机平台以及旋翼无人机平台分别针对稀疏空域下的典型目标以及稠密超低空空域的场景进行数据获取、感知算法验证与规避策略验证。

(1) 固定翼飞行试验平台

固定翼飞行平台采用 Inno 3000 平台搭载 ARS548 毫米波雷达，XiQ 可见光相机，通过 NUC 计算机进行飞行数据的有效获取。系统连接如图 1 所示。其中相机采用 USB3.0 高速数据接口连接。ARS548 通过以太网转换器以 RJ45 连接。机载计算机通过串口订阅无人机状态数据。平台通过地面站系统进行航路数据上传和任务执行管理，遥控器完成平台的应急控制。

图 10 固定翼试验系统连接图

NUC 加载 XiQ 相机驱动，ARS548 驱动程序，飞行器数据驱动程序，并发布相应数据。其中

- ARS_548 节点发布数据结构包括/ars50x/detectionlist_array、/ars50x/pointclouds，其中前阵为完整点云数据，后者转换为 ros pc2 格式，用于显示功能。
- Ximea_cam 节点发布数据为/ximeacam/image_raw 为原始图像数据。
- Collision_sensing 节点接收/ars50x/detectionlist_array 数据并发布/collision_cones 数据。
- Data_map 节点接收 Inno_tele 节点的/inno_tele 数据和 ARX_50x 节点点云数据，生成并发布/data_map（ESDF map）数据。
- Global/Local Path planning Nodes 节点接收 /collision_cones 和/global_map 数据并生成/vel_cmd 节点发送至无人机。当前设置，无人机不做响应。
- Inno_tele 节点接收并发布无人机状态数据。

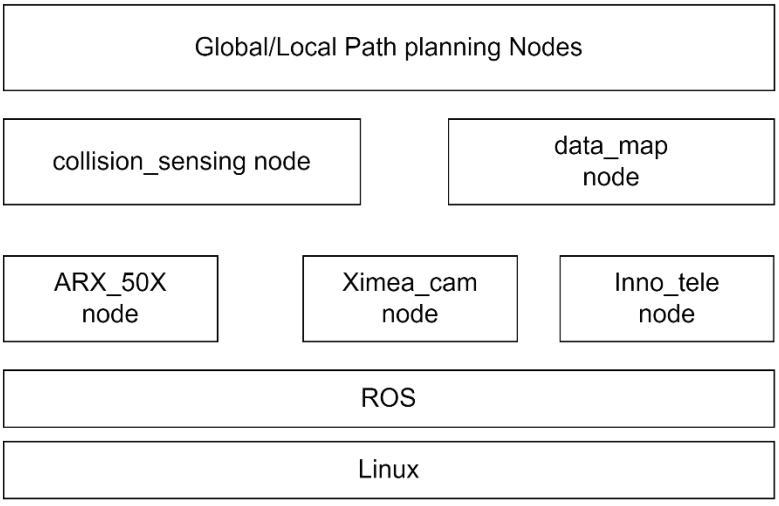


图 11 系统软件结构

(2) DJI M300 飞行试验平台

固定翼飞行平台采用 DJI M300 平台搭载 ARS548 毫米波雷达，事件相机、可见光相机等，通过 manifold 2C 计算机进行飞行数据的有效获取。系统连接如图 1 所示。其中可见光相机、事件相机采用 USB3.0 高速数据接口连接。ARS548 通过以太网转换器以 RJ45 连接。机载计算机通过串口和 DJI OSDK 进行数据和指令交互。DJI 遥控器完成平台的任务给定和应急控制。

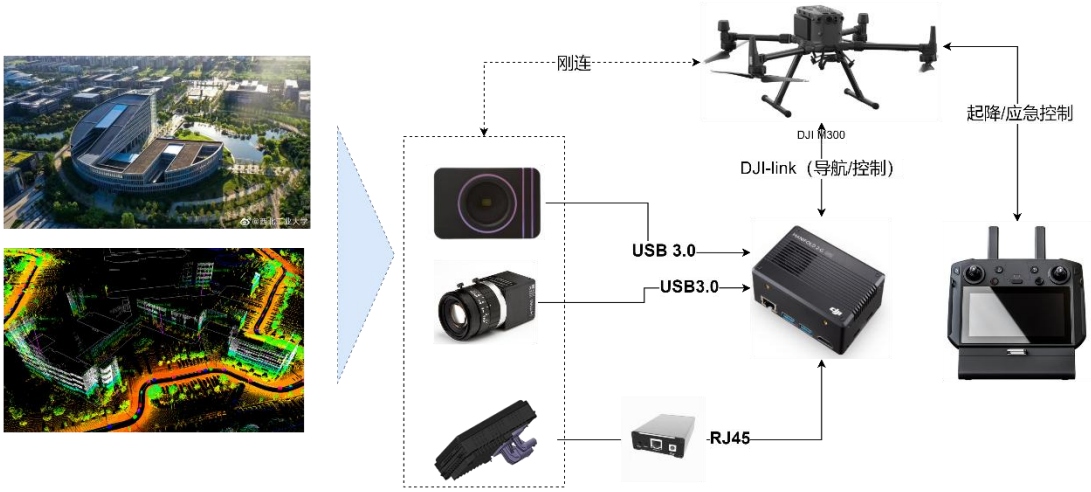


图 12 旋翼无人机试验系统连接图

Manifold 2C 加载可见光相机、事件相机驱动、ARS548 雷达驱动程序、DJI OSDK 数据驱动程序，并发布相应数据。其中

- ARS_548 节点发布数据结构包括 /ars50x/detectionlist_array 、/ars50x/pointclouds，其中前阵为完整点云数据，后者转换为 ros pc2 格式，用于显示功能。
- Ximea_cam 节点发布数据为/ximeacam/image_raw 为原始图像数据。
- Collision_sensing 节点接收 /ars50x/detectionlist_array 数据并发布 /collision_cones 数据。
- Data_map 节点接收/dji_osdk/telemetry_node 节点的 telementry 数据和 ARX_50x 节点点云数据，生成并发布/data_map（ESDF map）数据。
- Global/Local Path planning Nodes 节点接收 /collision_cones 和/global_map 数据并发布/mission_node 节点 mission waypoint 数据至无人机。

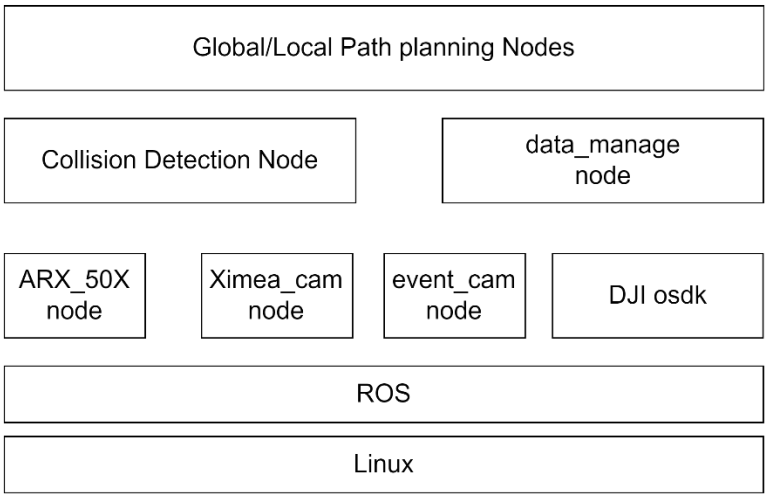


图 13 系统软件结构

6.1.2 场景设计

(1) 低空稀疏场景设计

低空稀疏场景中主要完成低空的高压线、塔，地形、建筑等数据。

针对固定翼无人机从不同入场角度、巡航飞行速度和飞行高度穿越观测，如下图。其中，为保证无人机的飞行安全，飞行观测路径与目标保持大于 20 米的安全距离。

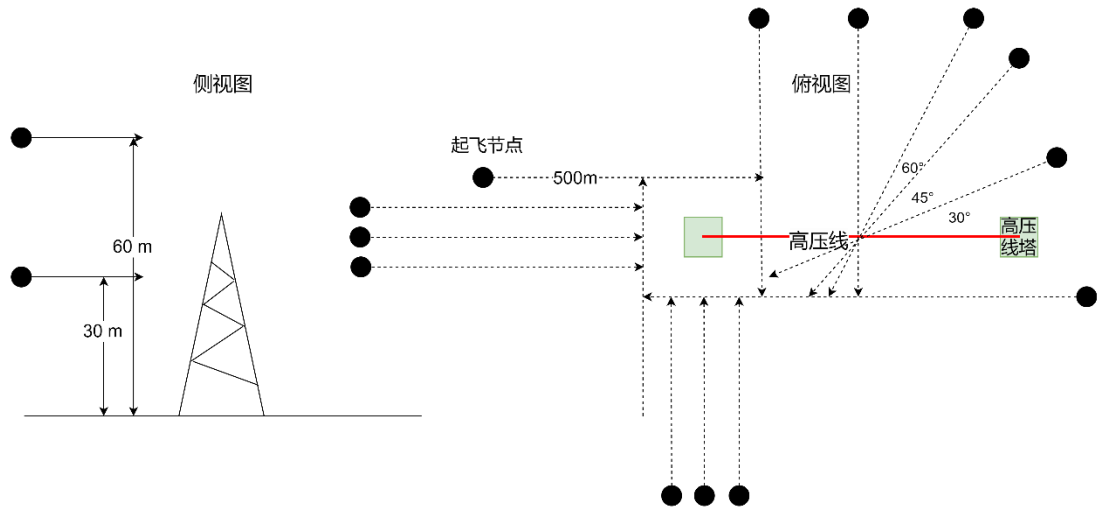


图 14 无人机从不同角度对潜在障碍物目标(如高压线塔)进行观测的路径

(2) 低空稠密场景设计

采用旋翼无人机平台实现对超低空稠密场景数据的有效获取。具体飞行路径如下图所示。在场景中，重点针对连续的建筑群、建筑走廊、树林等稠密场景进行数据获取。图中红线主要针对稠密建筑群进行有效感知，绿线主要针对建筑走廊和树林进行数据获取。

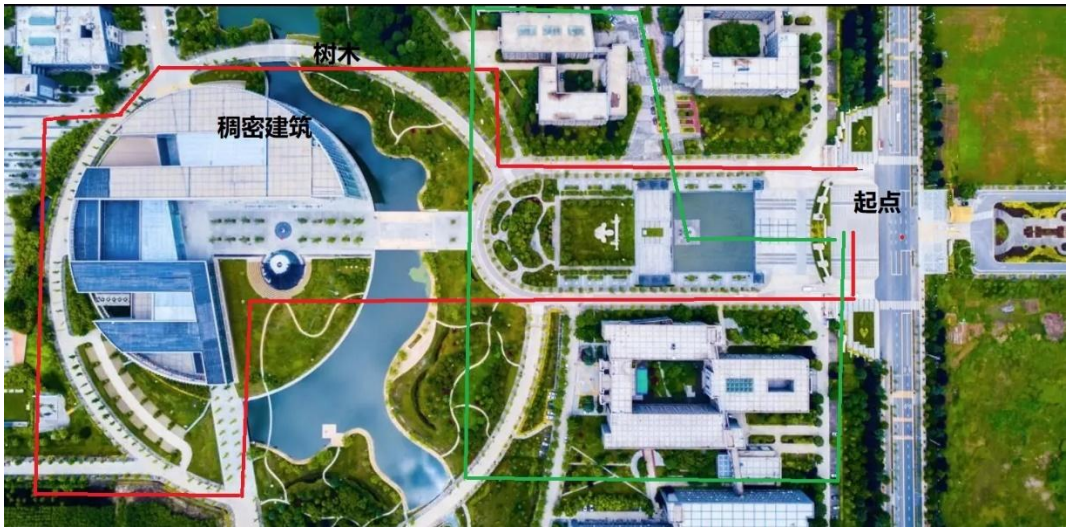


图 15 无人机超低空场景数据采集路径

6.1.3 飞行数据获取

6.1.3.1 固定翼无人机飞行试验数据获取

A. 飞前检查

（1）固定翼无人机及机载实验设备安装与电气检查

首先，将固定翼无人机平台上电，通过地面站系统检测无人机状态是否正常。然后，将机载计算机系统与传感器系统通电，观察相机、雷达灯系统是否上电正常。

（2）连接便携显示器，运行 sensor_management 脚本启动传感器数据通讯，具体命令包括：

```
sudo vconfig add eth0 19 (建立 vlan ID 19)
ifconfig eth0.19 10.13.1.100 netmask 255.255.255.0 up (启动节点)
ping 10.13.1.113 （确认 vlan 是否建立）

echo "0" | sudo tee /sys/module/usbcore/parameters/usbfs_memory_mb (修改 USB buffer)
sudo ./camTools 打开 camTools 确定 camera 数据是否正常.
```

（3）启动传感器驱动节点 /Ximea_cam node, /ARS_50X node，/Inno_tele 节点，通过 echo 相关数据确定节点是否运行正常。检测正常后关闭所有节点。

B. 飞行操作

（1） 运行 Start_all 脚本，执行雷达、相机和导航数据驱动节点，运行 collision_detection 节点。

```
source /opt/ros/kinetic/setup.bash
```

```
xterm -geometry 80x36+0+0 -e “roslaunch arx_50x arx_50x.launch“ && sleep 5
xterm -geometry 80x36+0+0 -e “roslaunch ximea_cam ximea_cam.launch“ && sleep 5
xterm -geometry 80x36+0+0 -e “roslaunch inno_uav inno_tele.launch“ && sleep 5
xterm -geometry 80x36+0+0 -e “ roslaunch collision_detection collision_local.launch“
```

（2） 运行 Save_all 脚本，通过 rosbag record 实现对各个传感器节点数据以及障碍物检测结果进行保存。

（3） 根据飞行路径在地面站设置飞行航路点。

（4） 起飞，切换至固定翼飞行模式，执行航点跟踪

（5） 到达航点，降落。

C．数据检查与保存

将数据拷贝至固态硬盘保存。

6.1.3.2 旋翼无人机数据获取

A. 飞前检查

（1）固定翼无人机及机载实验设备安装与电气检查

首先，将 M300 无人机平台上电，通过遥控器检测无人机状态是否正常。然后，将机载计算机系统与传感器系统通电，观察相机、雷达灯系统是否上电正常。

（2）连接便携显示器，运行 sensor_management 脚本启动传感器数据通讯，具体命令包括：

```
sudo vconfig add enp20s 19 (建立 vlan ID 19)
ifconfig enp20s.19 10.13.1.100 netmask 255.255.255.0 up (启动节点)
ping 10.13.1.113 （确认 vlan 是否建立）

echo "0" | sudo tee /sys/module/usbcore/parameters/usbfs_memory_mb (修改 USB buffer)
sudo ./camTools 打开 camTools 确定 camera 数据是否正常.
```

（3）启动传感器驱动节点 /Ximea_cam node, /ARS_50X node，/dji_osdk_ros 节点，通过 echo 相关数据确定节点是否运行正常。检测正常后关闭所有节点。

B. 飞行操作

（1） 运行 Start_all 脚本，执行雷达、相机和导航数据驱动节点，运行 radar_slam 节点。

```
source /opt/ros/kinetic/setup.bash
xterm -geometry 80x36+0+0 -e “roslaunch arx_50x arx_50x.launch“ && sleep 5
xterm -geometry 80x36+0+0 -e “roslaunch ximea_cam ximea_cam.launch“ && sleep 5
xterm -geometry 80x36+0+0 -e “roslaunch dji_osdk_ros dji_tele.launch“ && sleep 5
xterm -geometry 80x36+0+0 -e “roslaunch radar_slam arx50_slam.launch“ && sleep 5
xterm -geometry 80x36+0+0 -e “ roslaunch collision_detection collision_local.launch“
```

（2） 运行 Save_all 脚本，实现对各个传感器节点数据以及障碍物检测结果进行保存。

（3） 根据飞行路径在遥控端设置飞行航路点。

（4） 起飞，切换至航点飞行模式，执行航点跟踪。

（5） 到达航点，降落。

C．数据检查与保存

将数据拷贝至固态硬盘保存。

6.1.4 感知结果分析

（1）障碍物建模与仿真分析

运行 rosbag_read.m 进行数据的读取。将目标真实坐标输入至 genrate_ground_truth.m 进行目标检测的真值生成。运行 read_collision_detection.m 计算得到最远检测距离和有效检测概率P_{CT}，其中

- 最远检测距离定义为 collision_detection 节点连续三次生成并发布 collision_cone 数据。

  有效检测概率定义为

$$P_{CT} = \frac{N(S_{det} \cap S_{col})}{N(S_{col})}$$

其中S_{det}为一个飞行序列中生成碰撞告警的数据帧集合，S_{col}为碰撞告警真值。

- **漏检概率**和**虚警概率**分别定义为：$\frac{N(S_{col}) \backslash N(S_{det})}{N(S_{col})}$，$\frac{N(S_{det}) \backslash N(S_{col})}{N(S_{col})}$。

- **告警延迟**定义为 $\Delta T = T_{det} - T_{gt}$，其中$T_{det}$为最远检测距离定义的时刻， T_{gt}为真实的无人机进入碰撞告警区域。
- **定位误差**定义为：$\min_{i \in S_{det}} \|P_{pc}^{i} - P_{gt}\|$，其中$P_{gt}$为真实的目标中心位置，$P_{pc}^{i}$为检测到的碰撞点云。

(2) 稠密场景建模

稠密场景建模通过 Radar_slam 节点的 save_map service 保存生成的.ply 格式的 slam 地图数据。

（3） 避障能力分析

- 将地图导入 gazebo 构建 gazebo world。
- 修改 config.yaml 中的无人机初始状态，运行低空固定翼仿真器 roslaunch fix_wing_auto fix_wing_env。
- 运行运动基元避障方法 roslaunch mp_controller mp_contro.launch。
- rosbag record /trajectory 数据。

（4） 仿真分析

通过设置不同的无人机起始条件和不同巡航飞行速度条件下通过蒙特卡洛试验验证实时避障运动规划算法的表现。

导入西工大高清 3D Map 数据库，计算 trajectory 与 3D Map 点云的最小距离，即为**最小分离距离**：

$$d_{sep} = \min \|p_{traj}^{i} - p_{map}^{j}\|$$

规划响应时间定义为$\Delta T = T_{ca} - T_{det}$为生成避撞指令的时间与生成碰撞告警时间的插值。**避撞成功率**为$P_{CA} = \frac{N_{suc}}{N_{trials}}$。

6.2 试验结果分析

（1）旋翼无人机建模场景

在如图所示的地空稠密场景进行地图构建试验：

图 16 低空稠密场景示意

SLAM 系统运行界面如下图所示

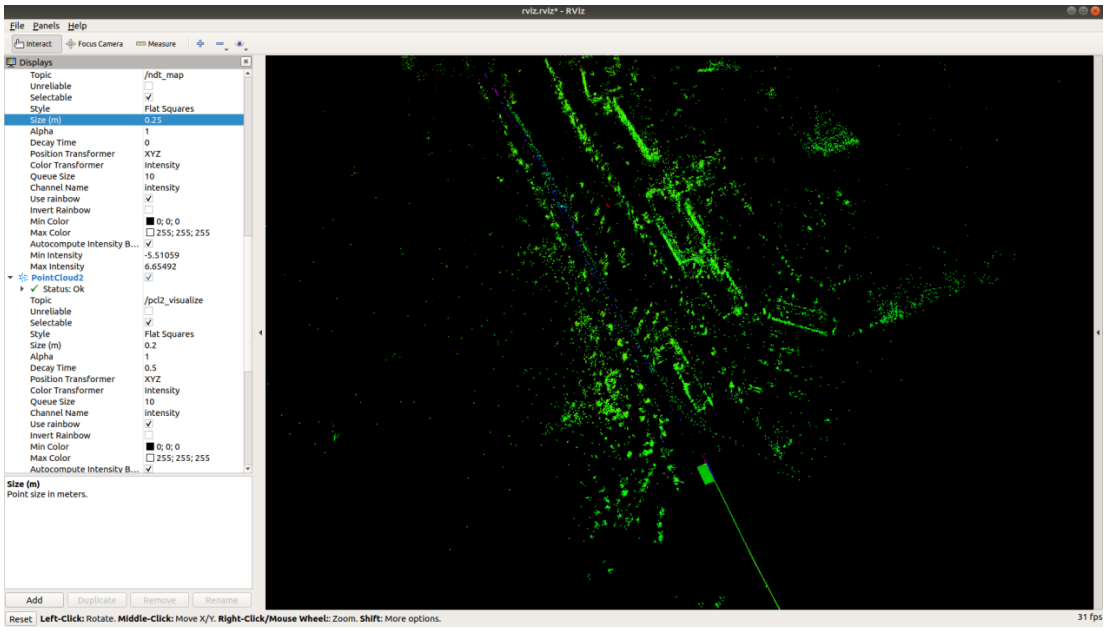


图 17 Rviz 中 SLAM 可视化

选取前述场景，进行大场景下存在回环的路径进行测试，运行过程中实时可视化结果如下图

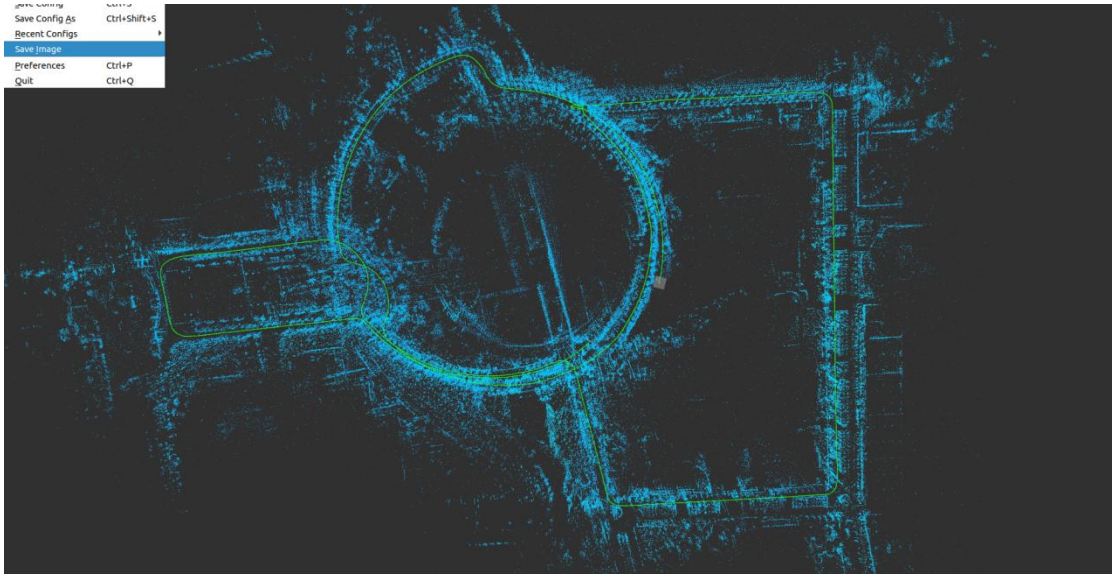


图 18 大场景下建图结果

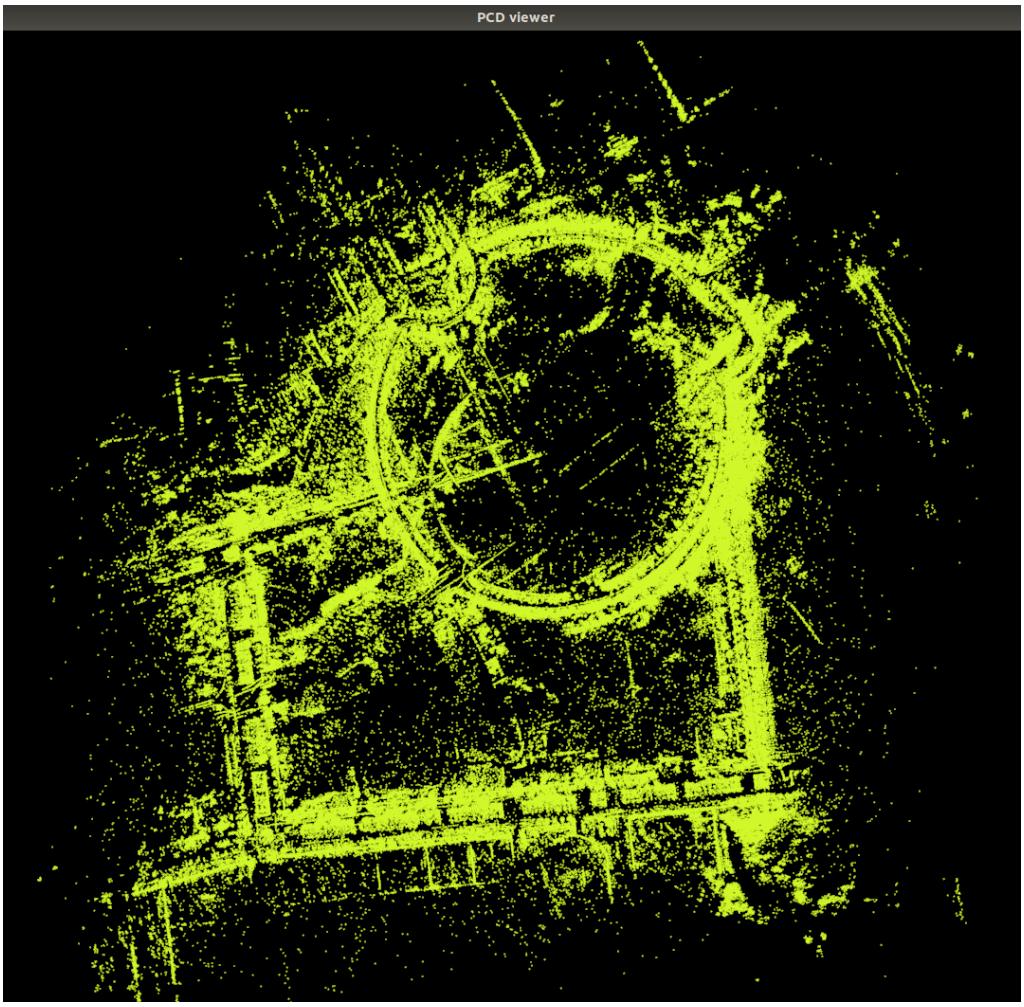


图 19 保存为 pcd 格式的地图

将结果保存为 pcd 文件格式，可视化效果如下图

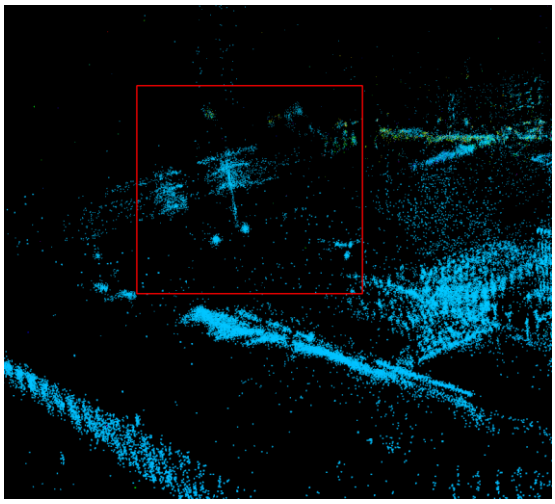


图 20 毫米波雷达对显著特征目标的感知效果

为测试实现点云、目标级别的障碍物有效检测，选取 200 米内有高压线的场景，进行数据采集，从结果可看出，雷达针对较远但特征明显的障碍物能够有效识别并建图。

为后续加入避障运动规划模块，考虑飞行器的动力学特性和障碍物表征形式，将点云地图转化为八叉树地图，在线和离线可视化效果如下图所示

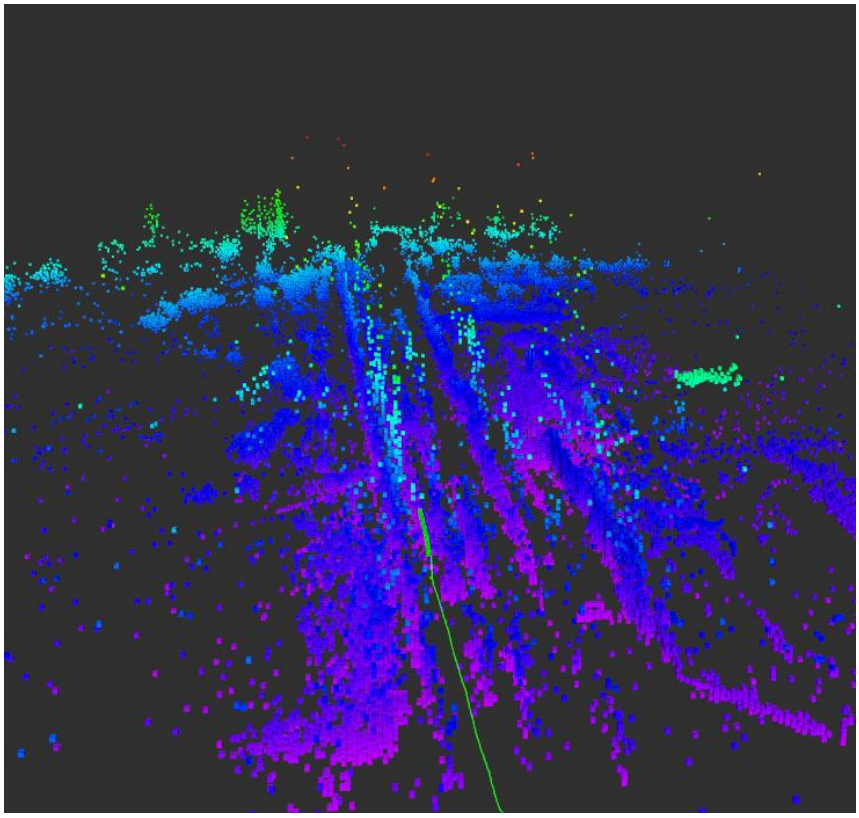


图 21 定位与栅格地图的构建

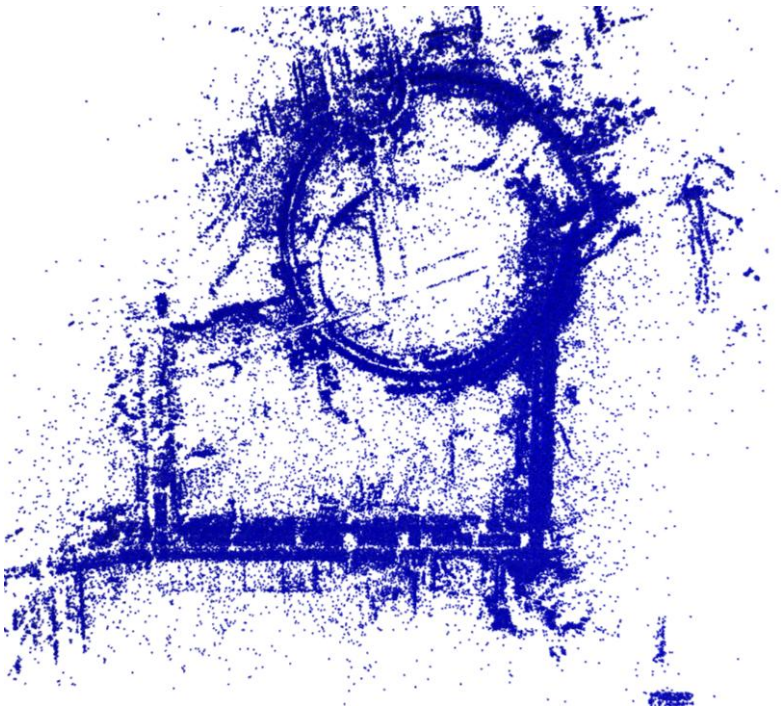


图 22 保存的栅格地图