

课题编号：2022YFC3801102

密级：公开

国家重点研发计划项目
“建筑与市政公用设施智慧运维理论与方法”

课题二
“建筑与市政公用设施智慧运维性态全息
快速感知与多源异构数据融合”

专题一
“多源运维数据的全息智能采集
与多维实时感知方法”

开题报告

承担单位：哈尔滨工业大学

专题负责人：马琳

二零二三年一月

1 前言

1.1 研究背景

我国建筑与市政公用设施的运维管理尚处于十分低下的水平，传统运维模式由于存在技术瓶颈，不能满足量大面广的基础设施运维管理的需求。随着大量基础设施陆续进入服役的中后期，安全事故频发，我国亟待发展面向全息全时全域的智慧运维技术体系。运维数据感知与融合系统构成了智慧运维的底层数据基础。现阶段，我国缺少运维领域的数据标准化体系建设，同时面临数据监测系统覆盖范围小、精度低、协同性差，信息传输通道容量、质量、效率低下，多源异构数据集成度低等难题，难以实现支撑智能运维系统的数据服务。为此，本课题面向建筑与市政公用设施智慧运维信息的全面、快速、精准、智能感知的重大需求，研究全域全时全息空天地一体化智能监测技术，构建集成多端采集、边缘计算和动态调度的“云边端”多源运维性态实时感知方法，实现多源运维数据的近实时感知与传输；研究智慧运维多源异构数据的关联、融合与共享的理论方法，实现面向运维数据的融合与智能管理；研发汇集万级以上终端信息，能够上下级联合、横向贯通、逻辑一体化的数据资源支撑服务系统。本课题的实施将有助于全面提升我国智慧运维领域的数据感知与融合技术，实现服务于智能运维体系发展目标的数据支撑体系建设。

1.2 目标和意义

本项目针对我国建筑与市政公用设施智慧运维感知不够全面及时、性态识别评价预警不够精准智能、决策管控不够科学高效等突出问题，聚焦建筑与市政公用设施智慧运维的基础理论及方法的关键科学及技术问题进行攻关。研究建立我国建筑与市政公用设施智慧运维的理论框架与技术标准体系，研发端-边-云全息快速感知与海量多源异构数据融合技术，突破性态指标智能精准识别与性能自主评价的技术瓶颈，创新智能决策与网联管控新技术，研发智慧运维智能体系统，为我国建筑与市政公用设施运维智能化水平提升提供坚实的理论基础与技术支持，助力我国“智慧城市”建设。

2 国内外研究综述

2.1 文献检索概述

图像匹配技术是计算机视觉技术的一个关键技术之一，在图像拼接、视觉定位以及图像检索等领域有着广泛的应用。经典的匹配方式是 SSD (Sum of Squared Differences)、SAD (Sum of Absolute Differences)、以及 NCC (Normalized Cross Correlation) 等匹配方法，后续方法大多是基于它们的基础上的改进。这些经典的模板匹配方法都采用了滑动窗匹配的思想，即待匹配的模板图像逐像素点在搜索图像上滑动，并根据所采用的匹配算法计算每次滑动窗口内的搜索图像子图与模板图像之间的相似性得分，最后选择相似性得分最高窗口区域作为匹配结果输出。这种匹配方式要求模板图像与搜索图像在同一尺度空间下，并且对于图像的形变没有适应能力。

相较于单架无人机而言，由多架无人机组成的飞行自组网 FANET (Flying Ad Hoc Network) 不依赖于地面基站，它是由传统的移动自组网 MANET (Mobile Ad Hoc Network) 发展而来的，灵活性更好、覆盖面积更广、信息搜集能力更强、执行任务的效率更高。相较于发展较成熟的单无人机系统，FANET 相关研究才刚起步，通过组网内无人机节点数量的冗余来提高鲁棒性和可靠性。目前对于飞行自组网的研究主要侧重于在多架无人机构建的动态网相关协议以及算法的设计优化等方面，在理论方面取得了显著成就，然而相关研究均是以理论方向为主，想要更好地应用于实际生活中还需要对飞行自组网进行更加深入的研究和性能优化。由于相邻无人机之间的通信距离有限，当两架无人机不在彼此通信范围时，就需要寻找其他无人机充当中继节点，寻找中继节点的过程需要选择合适的路由协议，通过计算路由表以完成路由寻路，建立通信链接以及维护路由表，从而完成无人机之间信息的转发。综合分析上述飞行自组网的特点，无人机作业场景大都需要保持实时性，还需要路由协议运行时尽可能少产生路由控制开销。

2.2 相关研究综述

为了克服经典模板图像对图像尺度和形变没有适应性的问题，Korman 等人提出可以适应任意图像仿射变换的 FAST-Match (FAST affine template Matching)

模板匹配算法,该方法根据图像仿射变换矩阵的 6 个自由度构建从模板图像到搜索图像的所有仿射变换集合,然后遍历所有仿射变换矩阵计算模板图像变换到搜索图像对应区域的像素灰度绝对差值的和 SAD,最后选取 SAD 最小的仿射变换矩阵作为最佳匹配矩阵得到匹配结果。该方法能够较好的匹配到全局最优的匹配位置,但是由于在计算 SAD 时只采用图像的灰度信息,这对于彩色图像的匹配来说损失了图像的彩色空间信息,使得图像对于光照变化不均匀的匹配准确率较低。为了解决这一问题,Jia 等人在 FAST-Match 的基础上提出能够适应彩色图像匹配的 CFAST-Match (Color FAST Match) 模板匹配算法,该方法将灰度空间的 SAD 拓展到彩色 RGB 空间形成 CSAD (Color SAD) 相似性衡量因子。CFAST-Match 通过基于 RGB 矢量密度的聚类算法计算每个像素点所属类别,然后统计每类像素点的个数和 RGB 各通道的累计值,以此求解每个 RGB 类别的矢量质心,将其作为 CSAD 的判定条件,每类像素个数的倒数作为类别区分权重系数,以此建立新的相似性度量机制。该方法匹配存在明显的颜色差异的图像时具有较高匹配精度,但部分参数的设置完全依据经验来确定,且对于大尺寸的图像没有很好的适应性。

模板和搜索图像子窗口之间的相似性度量是模板匹配的核心部分,在大多数的匹配方法中都是采用逐像素比较的方式,如上述方法提到的 SSD、SAD、CSAD 以及 SV-NCC 等都是通过逐像素点计算的方式来计算相似度。这种依靠所有图像像素特性的方法在图像背景存在杂波干扰或者图像发生复杂形变的情况下难以得到准确的匹配结果。Oron 等人基于模板与搜索图像间的 NN(Nearest Neighbor) 匹配属性提出一种新的 BBS (Best-Buddies Similarity) 相似性度量方法,该方法首先将图像进行分块,然后将每个图像块的特征转化为图像点,图像的特征可以是颜色特征或者卷积特征等,并通过滑动窗口的方式统计模板图像点与搜索图像点互为 NN 匹配点的数量,最后将 NN 匹配数量最多的滑动子窗口作为最终匹配位置。BBS 算法通过分块统计的匹配方式,减少了图像噪声和微小形变对匹配结果的影响,但是该算法在图像发生剧烈非刚性形变、光照不均匀或者遮挡面积较大等环境下匹配结果的鲁棒性较差。为此,王刚等人提出利用曼哈顿距离代替 BBS 算法中的欧氏距离,并对生成的匹配概率得分图进行滤波和阈值筛选,这些改进能够较好地解决非均匀光照、模板图像中杂乱外点较多以及图像发生旋转变换等多种复杂条件下的匹配问题。

BBS 算法通过双向 NN 匹配的方式统计图像点的最近邻点,该方式的计算时间较长,导致图像匹配效率较低。为此,Talmi 等人提出基于单向 NN 匹配的 DDS (Deformable Diversity Similarity) 方法。该方法首先用主成分分析技术对图像点进行降维,并通过 TreeCANN 算法计算搜索图像窗口点在模板中的 NN 匹配点,然后统计匹配点的数量来计算像素点的置信度。其次采用欧氏距离计算搜索图像点和对应的 NN 匹配点之间的距离,最后结合度量模板和搜索图像窗口间的相似性获得匹配结果。

为了解决 BBS、DDIS 等匹配算法计算复杂度较高,图像处理时间长的问题,Talker 等人提出了一种高效的 DIWU (Deformable Image Weighted Unpopularity) 匹配方法。该方法基于单向 NN 匹配实现,通过计算整幅搜索图像点在模板中的最近邻匹配点,其计算复杂度只与搜索图像大小相关,极大的提高了匹配效率,使得基于 NN 的模板匹配更具备实际应用的价值。

BBS 和 DDIS 匹配方法在解决图像形变和遮挡问题上都是通过计算图像子块间的相似性来实现的,该方式处理遮挡程度的大小受到滑动窗口的限制。Korman 等人提出一种适用于存在高度遮挡的 OATM (Occlusion Aware Template Matching) 模板匹配算法。该方法提出基于共识集最大化的哈希方案,较好地解决了图像遮挡问题。此外,OATM 通过降低搜索维度极大提高了算法的处理速度。

目标定位为计算机视觉领域的重要技术,随着无人机在军事和民用领域应用的扩大,基于图像的目标定位已成为无人机应用的一项重要能力。在美国陆军对无人机 (UAV) 的军事需求排名中,目标定位 (获取目标大地测量位置信息) 的能力在侦察之后占据第二重要的位置。多导弹多机协同攻击、机载光电侦察等任务也对目标定位提出了强烈要求。

针对基于计算机视觉的航拍图像目标定位算法,国内外专家进行了大量的研究。基于相机成像模型的目标定位,主要通过对航拍图像标校,或对变焦镜头标定了不同焦距的最佳畸变,依据无人机的位置信息,相机等内部参数,运用齐次坐标转换和几何求解,采用递归最小二乘法或蒙特卡洛方法,或利用激光测距的深度测量信息,减弱因倾斜角导致的大范围偏移效应,获得目标的经纬度信息,实现了目标精确定位。

石红岩等建立坐标转换模型,将目标从航拍图像坐标转换为地球坐标。Wang

等人提出了一种基于无人机电光稳定成像系统的实时多目标定位方案。首先,研究目标位置模型。然后,使用齐次坐标变换获得多目标的大地坐标,并采用实时变焦镜头畸变校正方法和基于无人机航位推算的递推最小二乘(RLS)滤波方法去减少不同帧的静止定位误差。Han 等消除了对地形数据和云台相机控制的要求,开发出鲁棒性的相机校准程序,能同时对多个目标进行地理定位。董巍通过多个相机进行标定和同步采集图像,运用齐次坐标转换求解目标的大地坐标,并运用递归最小二乘法,减少定位误差。Wang 等提出了一种改进的蒙特卡洛方法,使用统计方法估计每次测量的指向偏差的偏差,然后在方差优化过程中减去估计的偏差,消除视线影响的目标指向偏差的影响。Johnston 等在不考虑相机畸变的前提下,基于八大理想假设,利用基本旋转矩阵和摄影机系统几何矩阵,计算出现航拍图像像素位置在 WGS84 坐标系的位置。武汉大学结合视频影像与机关测距信息,利用激光测距的深度测量信息,减弱因大倾角导致的大范围偏移效应。乔川等采用载机位置等信息,并运用齐次坐标变换和用地球椭球模型等,直接对地面目标进行定位。Guo 等提出在无需地形数据和高度计数据的情况下,通过 Harris 角点提取、光流法和逆速度法,利用飞行器信息,就能在 MAV 平台上对运动目标进行地理定位的方法。Cai 等针对倾斜视角下建筑物目标的定位,通过使用成像角度估算建筑物的高度,并结合空间坐标变换矩阵来获得高精度地理位置,减少在倾斜视角下因目标高度而带来的定位误差。

王亚丽等以历史遥感图像作为先验信息数据库,采用视觉特征提取和配准技术进行视觉场景匹配。Liu 等引入历史遥感图像以生成稳定且独特的先验信息数据库,并将改进的基于 SIFT 的鲁棒性视觉特征提取和配准技术应用于视觉场景匹配,具有目标定位精确度高、可多点同时定位等优点。Han 等在已知相机固有参数的情况下,不需要地理参考的地形数据库和高度计来估算无人机和目标的高度,采用了微分光流法和 KLT 特征跟踪器对多个目标进行检测跟踪,并实现目标地理定位;并通过 SIFT 特征匹配和重建特征点三维坐标,获得特征高度信息。朱慧民等结合地图对目标匹配,以便标定参照点,将无人机仅作为目标与参照点的中继,消除了无人机飞行过程中不确定因素造成的目标定位误差。于越在摄像机垂直向下时研究低空俯拍图像中目标的定位,通过结合传感器信息和 BP 神经网络特性,获得可用于变焦相机的航拍目标的位置信息。郭博雷等对于单帧的航

拍图像，具体说明飞行器高度和传感器信息给定位目标位置带来的影响。Barbe 等针对单帧航拍图像的目标定位，提出了递归最小二乘自适应滤波、偏差估计、飞行路径选择和风力估计等 4 种方法，将定位误差控制在五米以内。

飞行自组网的业务模型有多种模式：无人机与地面的任务、多无人机协同、无人机与车联网、传统自组网的协调配合等。本项目主要涉及到的飞行自组网的业务是 UAV 对地面环境的信息采集。通过调研文献可知，适应于此类场景的飞行自组网结构主要有三种。

(1) 第一种通信结构是集中式结构，此类结构中无人机节点只能与同一个无人机单线通信，其他无人机节点之间不能通信。这种结构无人机通信链路稳定但网络覆盖范围小，常用于执行单一、简单的任务，各无人机间不能互相协同。

(2) 第二种通信结构是分布式结构，集中式结构的基础上进行了拓展，某些无人机可以在距离中心无人机较远的地方工作，并利用周围的无人机充当中继节点，通过多跳通信的方式，完成无人机之间的信息传输，信息获取较为全面，活动区域非常灵活并且覆盖范围得到了极大的提升。

(3) 第三种通信结构是分簇式结构，分簇式结构在前两种通信结构的基础上引入了簇的概念。若干架无人机会构成一个簇，在通信时在簇内的无人机中通过选举算法选出本簇的簇首作为长机负责与簇内其余节点以及其他簇的簇首之间的通信，通信能力强，覆盖面广。

飞行自组网与传统移动自组网有内在联系，但也有一定的区别。飞行自组网具有移动自组网的一些共性特点比如没有中心，自行组网，多点中继通信等，但由于其硬件设备的独特性，飞行自组网也存在一些传统移动自组网不具备的特点：网络拓扑连接变化快、对 QoS 指标要求严格、可扩展性强等。单架无人机的覆盖范围有限，且和飞行高度有关。但飞行自组网多架无人机构成的这种网络通过增加无人机数量可以拓宽覆盖范围。且大部分无人机都是小型无人机其价格和维修成本相对较小，非常符合无人机与地面信息采集的场景。

为解决对地面任务中的多无人机间的通信问题，一些研究人员们将 MANET 中的特征拓展到飞行自组网中，利用无人机快速构建通信网，并且感知邻居节点以处理和转发指令，最终传输无人机搜集的信息。通过调研相关文献可知，目前国内外对飞行自组网方面的研究已经取得一定的成就。

Wu 等开发了一种基于软件定义网络的异构架构，用于 FANETs 中的可靠通

信。这种体系结构应用扩展卡尔曼滤波器对无人机进行精确的机动性估计和预测,将基于软件定义网络的异构 FANETs 中的路由问题公式化为图论决策问题,进一步提出了一种定向粒子群优化方法。仿真结果表明,所提出的优化方法路由在提高吞吐量、分组投递率和时延方面表现出优越的性能。Yang 等提出了一种分布式时分多址接入调度协议。该协议可以通过监控时隙预留请求包来动态分配时隙。提出的分布式时分多址接入调度协议具有根据业务意图进行自我分析和自我配置的能力,通过典型的汇聚和并行流量类型的流量意图感知仿真场景进行仿真实验,相关结果表明所提出的协议减少了节点移动性的影响,提高了时隙利用率。Aissa 等针对 FANET 中无人机节点高速移动导致网络稳定性和通信效率降低的问题,提出了一种基于能量和移动感知的安全分簇算法,该算法适应无人机的快速移动,并提供安全的无人机之间的距离、稳定的通信和网络链路寿命时间,仿真结果表明,与现有的 BICSF 和 EALC 算法相比,所提出的算法具有更好的簇稳定性、更好的安全性、更高的分组投递率、更好的节能性和更低的时延。为了在有限的无线信道资源下实现飞行自组网中多种业务类型的并行传输,并满足最高优先级业务的高可靠性和超低时延要求,Zheng 等针对基于信道阈值的多址接入协议提出了一种新的多优先级排队机制,将该机制建模为一个带有服务员概率休假的多优先级 M/G/1 排队系统并针对不同的优先级类型提出两种优先级排队方案。仿真结果表明,方案二中针对最高优先级服务的抢占恢复机制牺牲其他优先级服务的少量性能为代价,获得了最高优先级服务的较大性能提升。

FANET 中各无人机节点正常通信离不开路由协议的支持,因此为了能够使得飞行自组网应用于更多场景,需要设计更贴切于飞行自组网的路由协议,在设计 FANET 路由协议时,要重点结合上文中所提到的无人机业务模型以及飞行自组网网络特点的影响。在上述背景下,国内外研究学者对相关的 MANET 路由协议进行改进,根据 FANET 应用场景的不同,将不同类别的路由协议进行优化,并将改进后的路由协议应用于飞行自组网中。

借鉴 MANETs 中按路由协议实现的原理,FANET 中的路由协议在空间和结构上分为两种。空间上的协议是基于地理位置的路由协议,结构上的协议为基于拓扑的路由协议,需要根据无人机网络的结构特点选择合适的路由协议,从而提高网络的整体性能。下面对飞行自组网中的路由协议进行具体的分析。

（1）基于地理位置的路由协议

随着定位技术更新，国内外学者开始研究基于地理位置的路由协议。此类路由协议中的节点通过定位系统以获得自身的实时地理位置信息，并向全网广播路由消息获得其他节点的实时位置信息，并结合相应路由算法以确定下一跳目的节点的地址。所以该类型的路由协议不用建立路由表，省去了路由发现和维护的过程，降低了网络中的路由控制开销，一定程度上可以优化飞行自组织网络的路由性能，但不维持路由表和路由发现过程导致时延增大、定位不准确等问题，能使用的应用场景十分受限。其中 LAR（Location Aided Routing）和 GPSR（Greedy Perimeter Stateless Routing）路由协议是比较常见的基于地理位置的路由协议，这两种都是利用地理位置信息来控制路由查找范围。

（2）基于拓扑的平面路由协议

平面路由协议具体包含静态路由、主动（表驱动）路由和被动（按需）路由三种，每种路由协议都有优缺点，因此需要选择适用于具体实际应用场景的路由协议，下面分别对三类路由协议进行简述。

第一类为静态路由。此类路由协议的路由表是静态的，不需要更新，在无人机作业之前网络就已经确定好拓扑结构和邻居关系。无人机在此类网络中运行时只需计算路由表并全网广播。缺点是一旦节点故障连接不上网络，就要重新记录拓扑信息并计算路由。因此此类路由协议的网络容错率低，只适用于静态网络场景。主要的静态路由协议有数据中心路由（Data Centric Routing, DCR）和负载携带和传递路由（Load Carry and Deliver Routing, LCAD）等。

第二类为主动路由。主动路由是通过路由表维持网络拓扑的路由协议，需要定期地对节点路由表计算和更新以实时查看节点的位置信息。主动路由的节点会定期向网络发送路由分组和拓扑信息，以此维护节点储存的路由表。此类路由的优点是节点可以根据路由表提供的信息迅速建立起通信链路从而进行信息传输，通信端到端时延较小；缺点是由于网络中的每个节点都需要周期性地转发控制消息，维持路由表，会在一定程度上增加网络开销。其中比较常见的主动式路由协议有优化链路状态路由（Optimized Link State Routing, OLSR）、目的序列距离矢量路由（Destination Sequenced Distanced Vector Routing, DSDV）等。

第三类为被动路由。被动式路由协议又称为按需路由协议，此类路由协议只有当节点有通信需求才会进行路由发现和查找，建立通信链路进行数据传输。网

络中节点不需要实时更新和维护路由表,不需要周期性广播消息和保存路由信息,因此在一定程度上降低了网络开销,优化网络资源。但缺点是网络节点较多时需要等待路由发现,这会导致时延过高。常见被动式路由协议有按需距离矢量路由 AODV (Ad-hoc On-demand Distance Vector Routing)、动态源路由 DSR (Dynamic Source Routing)、基于关联性路由 ABR (Associativity-Based Routing) 等。

(3) 基于拓扑的混合路由协议

混合式路由协议是综合了主动式路由协议和被动式路由协议的特点而形成的一种分级路由协议。分级的方式分为物理分级和逻辑分级两种。此类协议在小范围内使用主动路由建立路由表并寻路,范围之外采用被动路由的算法按需进行路由发现,优点是在一定程度上能够减小端到端时延并且降低网络路由开销,优化网络带宽资源,但网络结构复杂且管理不易。其中现阶段被广泛研究比较常见的混合式路由协议有空间域路由 (Zone Routing Protocol, ZRP)、混合无线 Mesh 协议 (Hybrid Wireless Mesh Protocol, HWMP) 等。

2.3 综述小结

本项目对于 UAV 航拍图像与卫星影像匹配的研究主要分为图像配准和图像匹配两大模块。在图像配准模块中,以卫星影像为基准,UAV 航拍图像通过获取机载相机成像时刻的方位角大小来确定它与卫星影像的方向偏移,然后将航拍图像按方位角相反方向旋转实现图像的方向矫正配准。配准方向后的图像再根据机载相机成像的焦距和成像高度等信息计算出航拍图像的实际地面覆盖范围大小,然后通过图像采样调整像素尺度到卫星影像像素尺度的大小,从而实现两类图像尺度空间的配准。在图像匹配模块中,由于卫星影像表示的地面覆盖区域通常比较广阔,而航拍图像覆盖区域较小,如果直接将航拍图像与整个卫星影像进行匹配,特征提取与特征匹配的计算量是非常大的,而这也是没有必要的匹配。因为在 UAV 航拍图像过程中已经记录下了成像点的经纬高等位置信息,通过这些信息可以大致确定航拍图像所在区域,因此在匹配阶段首先根据 UAV 航拍图像的位置信息及其定位精度进行粗匹配得到一个卫星影像子图,然后将卫星影像子图与配准后的航拍图像输入到特征提取网络提取对应的特征向量,并将配准后的航拍图像特征向量与卫星影像子图的特征向量作余弦相乘,然后根据余弦相乘结果计算向量相似性得分,最终选择相似性得分最高的区域最为 UAV 航拍图像

与卫星影像精匹配结果完成匹配过程。

本项目对 OLSR 协议展开调研,通过深入调研其工作流程,针对其应用于飞行自组网所产生的路由开销较大、路由选择标准单一等弊端,提出两种创新方法对传统的 OLSR 路由协议进行优化,主要研究分为两部分:

(1) 针对 OLSR 协议仅靠节点可达度选择中继节点转发 TC 分组应用于节点移动速度快、网络拓扑结构变化频繁的飞行自组网时出现路由失效、丢包严重等问题,提出了一种基于链路感知和链路持久性的路由选择策略,通过对链路质量、链路预测生存时间以及节点可达度这三个指标进行加权计算,得到新的 MPR 选择标准,将链路最大保持时间加入到路由表,选择链路生存时间最长的一条作为最优传输路径。

(2) 针对 OLSR 采用固定的控制分组发送间隔应用于飞行自组网中出现路由断裂失效或产生较多路由控制消息以及优化路由选择策略在控制消息中添加新内容导致网络中路由控制开销变大,吞吐量下降的问题,提出了一种基于拓扑变化的控制消息自适应算法。通过对 OLSR 协议中链路信息表中元素的增删情况自适应调整 HELLO 分组发送间隔,对 MPR Selector 集中元素的变化情况预测 TC 消息的发送频率,并根据 TC 分组的自适应发送间隔优化 TTL 值以达到对 TC 分组在时间和空间上的优化。

3 研究内容与创新点

3.1 研究内容

研究基于天基卫星快速解译、空基多智能体协同巡检以及地基 IOT 群体传感的空天地一体化运维性态信息的实时智能感知方法,研发基于网联资源调度的“端-边”数据校验与保障算法,研究基于多智能体无人系统的多源运维数据感知方法。

3.2 拟解决的关键科学和技术问题

- (1) 面向设施群的空天地一体化协同快速监测技术。
- (2) 多无人机智能体编队协同智能巡检技术。
- (3) 超低延时的“云-边-端”分布式多源运维数据智能感知技术。

3.3 创新点

创新点 1：超低延时的“云-边-端”分布式多源运维数据智能数据感知技术

基于新型智能运维感知智能体，设计智能体分簇策略和基于链路感知的路由选择策略，建立簇首智能体无人自组网方法；研究自组网多点广播中继机制，提出近“端-边”侧数据校验与保障算法，突破基于负载均衡的网联资源调度分配和智能自组网技术，实现“端-边”数据有效、高效并网并提高通信网络吞吐量。面向运维过程中准实时性、高准确性和强鲁棒性的需求，搭建分层分布式的多智能体协同处理架构，提升多智能体间、多智能体与运维设施间的信息协同、状态感知与联动控制能力。该技术预期开发领先的分布式自组网和感知智能体群组网技术，降低智能感知系统的延时性，预计获得专利、学术论文等形式的知识产权。

创新点 2：面向设施群的空天地一体化协同监测技术

面向区域建筑群与市政公用设施集群全寿命期运维情景，研究空天地一体化广域运维监测与多源数据协同采集方法，结合卫星、5G、物联网、光纤等技术，研发空天地一体化全息协同感知技术，实现多源运维数据的全息采集与实时感知，提升广区域、多层次、立体化的空间协同监测能力。该技术预期实现覆盖范围不小于 2 平方公里级的城市建筑群、桥梁隧道边坡群的区域集群空天地协同监测，获得专利、学术论文等形式的知识产权。

4 研究方法与技术路线

4.1 研究方法

4.1.1 基于航拍图像与卫星影响的图像匹配算法

1) 航拍图像与卫星影像匹配算法概述

本项目所提出的 UAV 航拍图像与卫星影像的匹配算法技术路线如图 1 所示。由图 1 可见，本项目对于 UAV 航拍图像与卫星影像匹配的研究主要分为图像配准和图像匹配两大模块。在图像配准模块中，以卫星影像为基准，UAV 航拍图像通过获取机载相机成像时刻的方位角大小来确定它与卫星影像的方向偏移，然后将航拍图像按方位角相反方向旋转实现图像的方向矫正配准。配准方向后的图像再根据机载相机成像的焦距和成像高度等信息计算出航拍图像的实际地面覆

盖范围大小，然后通过图像采样调整像素尺度到卫星影像像素尺度的大小，从而实现两类图像尺度空间的配准。在图像匹配模块中，由于卫星影像表示的地面覆盖区域通常比较广阔，而航拍图像覆盖区域较小，如果直接将航拍图像与整个卫星影像进行匹配，特征提取与特征匹配的计算量是非常大的，而这也是没有必要的匹配。

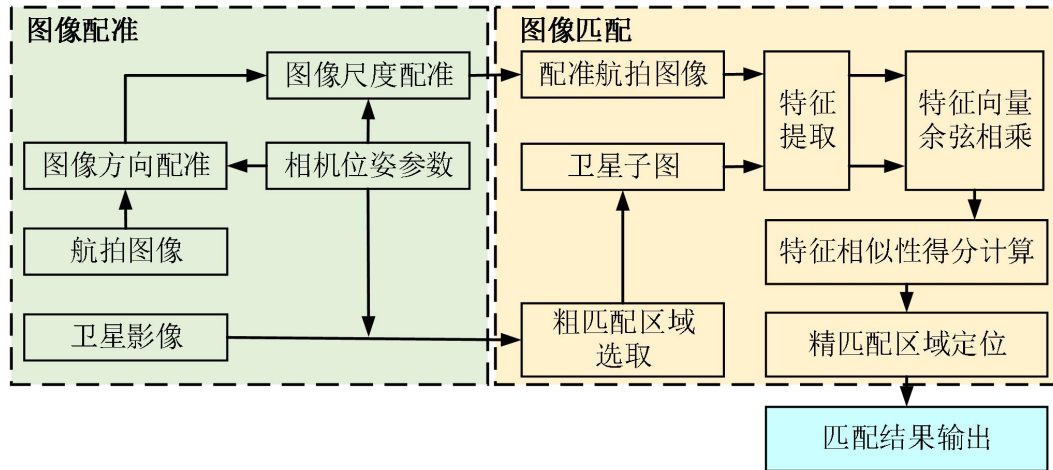


图 1 图像匹配研究技术路线图

因为在 UAV 航拍图像过程中已经记录下了成像点的经纬高等位置信息，通过这些信息可以大致确定航拍图像所在区域，因此在匹配阶段首先根据 UAV 航拍图像的位置信息及其定位精度进行粗匹配得到一个卫星影像子图，然后将卫星影像子图与配准后的航拍图像输入到特征提取网络提取对应的特征向量，并将配准后的航拍图像特征向量与卫星影像子图的特征向量作余弦相乘，然后根据余弦相乘结果计算向量相似性得分，最终选择相似性得分最高的区域最为 UAV 航拍图像与卫星影像精匹配结果完成匹配过程。

2) 基于相机位姿参数的图像配准

1 实验数据采集

为了完成课题的研究，需要选取合适试验地点拍摄 UAV 航拍图像以及对应区域的卫星影像。目前课题选择航拍地点为哈尔滨工业大学一校区，选取的航拍 UAV 为大疆 MATRICE 200 V2，搭载航拍相机为大疆禅思 X5S 云台相机，UAV 实物如图 2 所示。



图2 图像采集 UAV 设备

为满足课题研究需要，目前已经在哈尔滨工业大学校内田径场采集到多张 UAV 航拍图像，图像采集飞行高度均为 100m，并且记录了图像采集过程时刻对应的位置以及相机姿态等参数信息，采集到的航拍图像实例如图 3 所示。

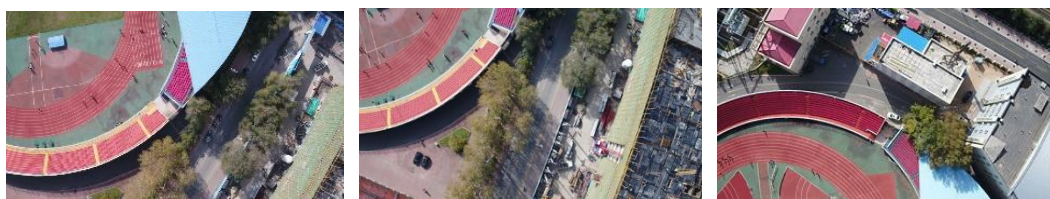


图3 UAV 航拍图像

实验采用的卫星影像从大地图卫星图像查看软件下载，所下载卫星影像为 18 级影像，其比例尺大小为 1:1575。图 4 为实际下载的哈尔滨工业大学田径场卫星影像图像，为方便其与 UAV 航拍图像尺度的配准，下载的同时记录下影像地面覆盖区域所对应的经度范围为 $126.624550^{\circ}\text{E}$ - $126.632227^{\circ}\text{E}$ ，纬度范围为 $45.734169^{\circ}\text{N}$ - $45.738873^{\circ}\text{N}$ 。



图4 卫星影像

2 空间坐标系建立

在基于位置信息的图像尺度配准过程中，为了精确描述相机拍摄到的地面覆盖区域大小以及如何根据卫星影像区域经纬度信息计算其实际覆盖地面区域，需要将它们放到一个统一的坐标下衡量，因此涉及到一些坐标转换关系的计算，主要用到的坐标系有地心空间直角坐标系，机载相机坐标系，以及局部空间直角坐标系等，具体的定义如下：

(1) 地心空间直角坐标系 $O_w X_w Y_w Z_w$

地心空间直角坐标系是以地球质心为原点，其中， $X_w O_w Y_w$ 平面与地球赤道平面重合， $O_w X_w$ 轴与首子午面和赤道面的交线重合， $O_w Z_w$ 轴指向北极星且垂直于赤道面，坐标系符合右手定则，利用地心空间直角坐标系可以描述 UAV 航拍图像和卫星影像的空间绝对坐标。

(2) 机载相机坐标系 $O_c X_c Y_c Z_c$

机载相机坐标系将 UAV 及其所载相机作为一个整体，坐标系原点位于整体质心上，其坐标轴指向通常与 UAV 的姿态有关，通常可以选择 UAV 机头正前方为 $O_c X_c$ 方向，UAV 机顶方向为 $O_c Z_c$ 方向，坐标系符合右手定则。利用机载相机坐标系可以描述相机拍摄到的地面覆盖区域坐标。

(3) 局部空间直角坐标系 $O_L X_L Y_L Z_L$

局部空间直角坐标系以 UAV 瞬时空间位置为原点，正北方向为 Y 轴，背离地心方向为 Z 轴，建立右手坐标系。局部空间直角坐标系用于描述当前时刻机载相机拍摄到的图像与卫星影像之间的坐标关系。

在上述坐标系的基础之上，可以利用局部空间直角坐标系计算机载相机航拍到的图像覆盖地面区域大小。假设 UAV 飞行高为 H ，航向角为 a ，俯仰角为 b ，滚转角为 c 。机载相机方位角为 Az ， Az 为 $O_c X_c Y_c Z_c$ 坐标系下与 $O_c X_c$ 轴的夹角，逆时针方向为正，俯仰角为 EI ，视场角为 $\theta_x \times \theta_y$ ，地面覆盖区域中心点在相机视轴所在方向，在以机载相机坐标系为中心的坐标系中，相机视轴中心向量 $\mathbf{A}$ 如图 5 所示。

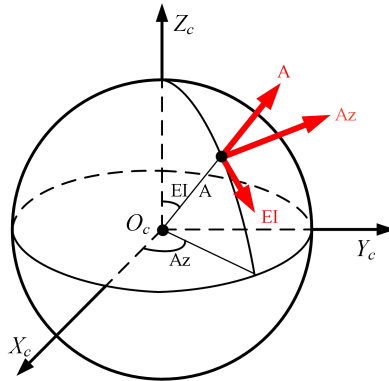


图 5 球坐标转直角坐标

由图 5 可得相机视轴中心向量 $\mathbf{A}$ 计算公式如式所示。

$$\begin{cases} x_a = \sin(EI) \cdot \cos(Az) \\ y_a = \sin(EI) \cdot \sin(Az) \\ z_a = \cos(EI) \end{cases} \quad (1)$$

根据 UAV 的飞行姿态角度等信息，可以计算出从机载相机坐标系 $O_cX_cY_cZ_c$ 到局部空间直角坐标系 $O_LX_LY_LZ_L$ 的变换矩阵为：

$$R = R_x \cdot R_y \cdot R_z \quad (2)$$

其中，

$$\begin{aligned} R_x &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos c & \sin c \\ 0 & -\sin c & \cos c \end{bmatrix} \\ R_y &= \begin{bmatrix} \cos b & 0 & -\sin b \\ 0 & 1 & 0 \\ \sin b & 0 & \cos b \end{bmatrix} \\ R_z &= \begin{bmatrix} \cos a & \sin a & 0 \\ -\sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3)$$

其中， R_x 、 R_y 、 R_z 分别为机载相机坐标系 $O_cX_cY_cZ_c$ 相对于局部空间直角坐标系， $O_LX_LY_LZ_L$ 绕 X、Y、Z 轴旋转的旋转矩阵。

3 图像方向配准方法

无人机航拍图像过程中，机载相机方位角 Az 的改变会导致航拍图像与卫星影像之间存在旋转变换关系，因此需要对航拍图像进行旋转方向的配准。在本项目中，通过解码图像成像时机载相机的方位角 Az ，然后将图像按方位角相反方向进行旋转实现图像的方向矫正配准。

图像旋转的本质是像素点坐标位置绕图像中心的旋转变换，这一变化过程可以通过变换矩阵来实现。但是由于图像的像素坐标通常以图像左上角为原点，因此图像变换前首先需要实现图像的旋转坐标系变换将图像由像素坐标系变换到图像中心坐标系，具体变化关系如图 6 所示。

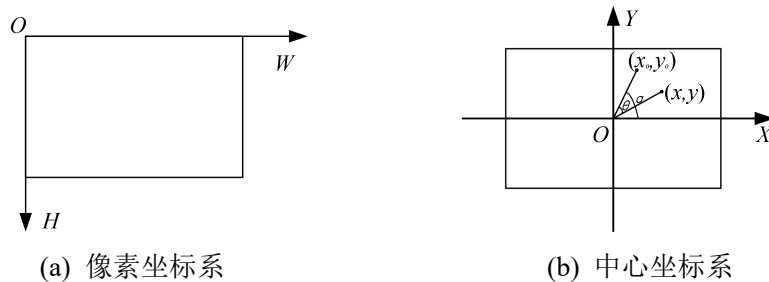


图 6 图像坐标关系

根据图 6 所示的图像像素坐标系和中心坐标系的关系可得，当假设图像高度为 H ，宽度为 W 时，像素坐标系的一点 (X_0, Y_0) 变换到中心坐标系下的坐标为 (X, Y) ，其具体变换公式为：

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -0.5W \\ 0 & -1 & 0.5H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix} \quad (4)$$

由于图像的显示是基于像素坐标系的，因此在图像中心坐标系下实现图像旋转后需要将中心坐标系下的像素坐标进行逆变换到像素坐标系下，根据矩阵计算关系，逆变换矩阵为正变换矩阵的逆矩阵，即由图像中心坐标系变换到像素坐标系的变换公式：

$$\begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.5W \\ 0 & -1 & 0.5H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (5)$$

以上式和式描述了图像像素坐标系与图像中心坐标系之间的转换关系，在实际操作中，图像的旋转是在图像中心坐标系下实现的，因此对图像进行旋转需要先将图像从像素坐标系变换到中心坐标系，在中心坐标系下实现图像绕原点的旋转，然后将旋转后的中心坐标系下的图像逆变换回到图像像素坐标系下显示。图像在中心坐标系下实现旋转的数学理论基础如下。图 6(b)展示了图像在中心坐标系下的旋转关系，假设图中点 (x_0, y_0) 到坐标原点距离为 R ，与 X 轴的夹角为 α ，将该点按顺时针方向旋转 θ 度后的坐标为 (x, y) ，因此根据几何关系可以写出坐标关系如下：

$$\begin{aligned} x_0 &= R \cos \alpha \\ y_0 &= R \sin \alpha \end{aligned} \quad (6)$$

$$\begin{aligned} x &= R \cos(\alpha - \theta) \\ y &= R \sin(\alpha - \theta) \end{aligned} \quad (7)$$

联合式与，结合三角函数加法定理展开并写成矩阵的形式，坐标 (x_0, y_0) 旋转 θ 度后的坐标计算公式如式所示。

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (8)$$

综合上述推导，图像像素坐标系下的一点 (X_0, Y_0) 绕图像中心顺时针旋转 θ 度后的坐标 (X'_0, Y'_0) 可以由计算。

$$\begin{bmatrix} X'_0 \\ Y'_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.5W \\ 0 & -1 & 0.5H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.5W \\ 0 & -1 & 0.5H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix} \quad (9)$$

根据以上图像旋转的推导公式，编程实现的图像方向配准如图 7 所示，图中原图成像机载相机的方位角为 156.9° ，因此在实际配准过程中，将图像按相反方向旋转 156.9° 实现图像的方向配准，配准后的图像如图 7(b) 所示。



图 7 图像方向配准示例

由图 7(b) 可见，旋转过后的图像存在边角的空白区域，这些空白区域的存在对于采用模板匹配方式的图像匹配来说是一种严重的噪声干扰，因此需要对旋转后的图像进行剪裁，裁去图像四周空白区域并最大限度的保留图像区域。在本项目实现中，通过裁剪旋转图像的最大内接矩形的方式来实现，最大内接矩形区域的选取计算如下。

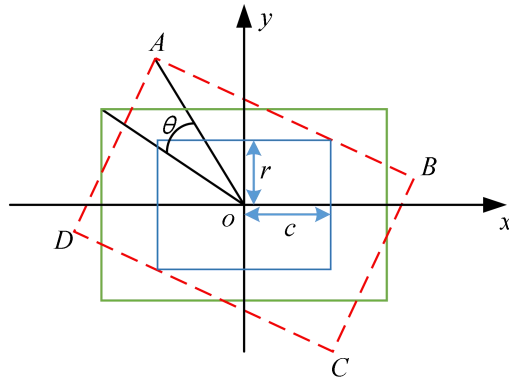


图 8 最大内接矩形选取

在图 8 中，红色虚线矩形 $ABCD$ 是绿色矩形绕图像中心旋转一定角度后的

结果,蓝色矩形为需要选取的最大内接矩形区域。假设只考虑旋转角 θ 范围为 $(0,90)$ 的情况, $A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)$, 根据 ABC 三点的坐标可以写出直线 $\overline{AB}$ 和直线 $\overline{BC}$ 的数学方程表示如式、所示。

$$l_{AB} : \frac{y-y_2}{y_1-y_2} = \frac{x-x_2}{x_1-x_2} \quad (10)$$

$$l_{BC} : \frac{y-y_3}{y_2-y_3} = \frac{x-x_3}{x_2-x_3} \quad (11)$$

令 $\Delta x_1 = x_1 - x_2, \Delta x_2 = x_2 - x_3, \Delta y_1 = y_1 - y_2, \Delta y_2 = y_2 - y_3$, 将式、展开并归一化表示可得式所示公式。

$$\begin{aligned} l_{AB} : a_1 y + b_1 x + s_1 &= 0 \\ l_{BC} : a_2 y + b_2 x + s_2 &= 0 \end{aligned} \quad (12)$$

其中,

$$\begin{aligned} a_1 &= \frac{\Delta x_1}{|\Delta x_1^2 + \Delta y_1^2|^{\frac{1}{2}}}, & a_2 &= \frac{\Delta x_2}{|\Delta x_2^2 + \Delta y_2^2|^{\frac{1}{2}}} \\ b_1 &= \frac{-\Delta y_1}{|\Delta x_1^2 + \Delta y_1^2|^{\frac{1}{2}}}, & b_2 &= \frac{-\Delta y_2}{|\Delta x_2^2 + \Delta y_2^2|^{\frac{1}{2}}} \\ s_1 &= \frac{-\Delta x_1 \cdot y_2 + \Delta y_1 \cdot x_2}{|\Delta x_1^2 + \Delta y_1^2|^{\frac{1}{2}}}, & s_2 &= \frac{-\Delta x_2 \cdot y_3 + \Delta y_2 \cdot x_3}{|\Delta x_2^2 + \Delta y_2^2|^{\frac{1}{2}}} \end{aligned} \quad (13)$$

根据图 8 关系,蓝色矩形框的面积可以表示为式所示。

$$S = 4cr = -\frac{4b_1}{a_1}c^2 - \frac{4s_1}{a_1}c \quad (14)$$

由式可见,蓝色矩形最大面积是关于 c 的二次函数,且当旋转角 θ 在 $(0,90)$ 之间时总有 $a_1 b_1 > 0$ 。因此,随着 c 的值从 0 开始增大,蓝色矩形框的面积先增大后减小,极大值的边界约束条件是蓝色矩形框右下角坐标始终位于直线 $\overline{BC}$ 左侧,故可以求解当蓝色矩形取得极大值 $S_{\max}$ 时, c 的取值如式所示。

$$c = \min \left\{ \left| -\frac{a_1 s_2 + a_2 s_1}{a_2 b_1 + a_1 b_2} \right|, \left| -\frac{s_1}{2b_1} \right| \right\} \quad (15)$$

至此完成了顺时针旋转角 θ 在 $(0,90)$ 之间的推导,当旋转角度为 90° 或其倍数时,旋转图像不存在空白区域,因此不需要进行图像剪裁,当旋转角 θ 在 $(90,180)$

之间时，等价于 $180-\theta$ 的情况，其他范围内的角度也是以 180° 为周期的，因此对于旋转图像求取最大内接矩形的理论推导分析结束。最终通过编程实现截取结果如图 9 所示，实验中图像的旋转角度为逆时针方向 156.9° 。



图 9 最大内接矩形裁剪结果

4 图像尺度空间配准方法

在本项目实践中，图像采集系统为大疆禅思 X5S 相机云台，采集到的照片存储格式为 JPEG（Joint Photographic Experts Group），该格式图像中嵌入了相机拍摄图像时的经纬高以及相机姿态角度等位姿信息，本项目通过图像中记录的成像参数来计算图像实际覆盖地面区域的大小，然后通过图像采样调整像素尺度到卫星影像像素尺度的大小，从而实现两类图像尺度空间的配准。

首先假设 UAV 飞行高度为 H ，地面覆盖区域中心点在 $O_L X_L Y_L Z_L$ 中的坐标为 $B = (X, Y, -H)$ ，则根据坐标建立关系， B 具有如式所示关系式。

$$B = R \cdot L \cdot A \quad (16)$$

其中， L 为机载相机坐标系原点到地面覆盖区域中心的距离。

根据 L 以及机载相机的视场角大小，可以得到机载相机能够拍摄到的地面覆盖区域范围大小如图 10 所示。

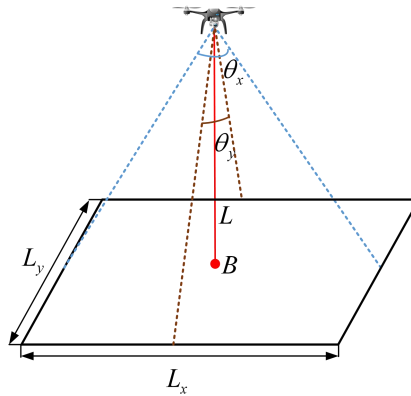


图 10 UAV 扫描区域示意

由图 10 可以得出地面覆盖区域长 L_x 与宽 L_y 的计算公式如式所示。

$$\begin{cases} L_x = 2L \cdot \tan \frac{\theta_x}{2} \\ L_y = 2L \cdot \tan \frac{\theta_y}{2} \end{cases} \quad (17)$$

通过以上步骤即可计算出 UAV 航拍图像实际的地面覆盖区域大小。

假设当前选取的卫星影像覆盖区域长为 L_{ax} ，宽为 L_{ay} ，其图像大小为 $H_a \times W_a$ ， H_a 为图像行像素点数， W_a 为列像素点数。相应地，UAV 航拍图像大小为 $h \times w$ 。因此对于卫星影像可以计算出其 x, y 方向上单个像素点实际占据地面范围大小，具体计算方式如式所示。

$$\begin{aligned} dU &= \frac{L_{ax}}{W_a} \\ dV &= \frac{L_{ay}}{H_a} \end{aligned} \quad (18)$$

同理，根据 UAV 航拍图像地面覆盖区域长 L_x 和宽 L_y 以及图像的大小 $h \times w$ 可以求得航拍图像在 x, y 方向上单位像素大小分别为 du ， dv 。在此，航拍图像与卫星影像之间的尺度关系也即它们之间单个像素点表示实际区域大小的关系，可以根据公式分别计算出 x, y 方向上的尺度大小对应关系。

$$\begin{aligned} \sigma_x &= \frac{du}{dU} \\ \sigma_y &= \frac{dv}{dV} \end{aligned} \quad (19)$$

至此即完成了 UAV 航拍图像与卫星影像之间的图像尺度关系计算过程。最后可以通过图像上采样的方式对 UAV 图像进行尺度调整实现航拍图像与卫星影像之间的尺度配准。假设 UAV 航拍图像上采样调整后的图像大小为 $h' \times w'$ ，其数值可以根据以式计算得到。

$$\begin{aligned} w' &= \lfloor \sigma_x \cdot w \rfloor \\ h' &= \lfloor \sigma_y \cdot h \rfloor \end{aligned} \quad (20)$$

其中， $\lfloor \cdot \rfloor$ 表示对数据向下取整。

3) 基于分步的航拍图像与卫星影像匹配算法研究

1 基于位置信息的匹配区域选取方法

图像匹配中，由于卫星影像表示的地面覆盖区域通常比较广阔，而航拍图像覆盖区域较小，如果直接将航拍图像与整个卫星影像进行匹配，特征提取与特征匹配的计算量是非常大的，而这也是没有必要的匹配。因为在 UAV 航拍图像过程中已经记录下了成像点的经纬高等位置信息，通过这些信息可以大致确定航拍图像所在区域，因此在匹配阶段可以根据航拍图像的位置信息及其定位精度确定航拍图像在卫星影像可能出现的区域，然后选取该区域作为一个卫星影像的子图与航拍图像进行匹配。该方法可以有效减少不相关的卫星影像区域与航拍图像的匹配计算量，提高匹配速率。此外，由于去除掉了无关匹配区域，避免了其它与真实匹配区域具有相似纹理特征区域对匹配结果的干扰，有效减小误匹配出现的概率。

在图像尺度空间配置当中，UAV 航拍图像与卫星影像的尺度配准是根据两种图像之间实际覆盖地面大小来实现的。UAV 航拍图像通过相机参数和成像位置等信息来计算航拍图像实际覆盖地面区域的大小，而卫星影像中通常包含着经纬度等位置信息，因此可以根据卫星影像的位置参数来计算其地面覆盖区域的大小，具体计算方式如下。

在卫星影像中，假设影像左上角 A 的经纬坐标为 $(lonA, latA)$ ，右下角 B 的经纬坐标为 $(lonB, latB)$ ，由于所取的卫星影像地面覆盖区域大小相对于地球来说非常的小，因此可以忽略地球球面的影响，将卫星影像覆盖的地面区域视为平面区域计算其实际距离，具体如图 11 所示。

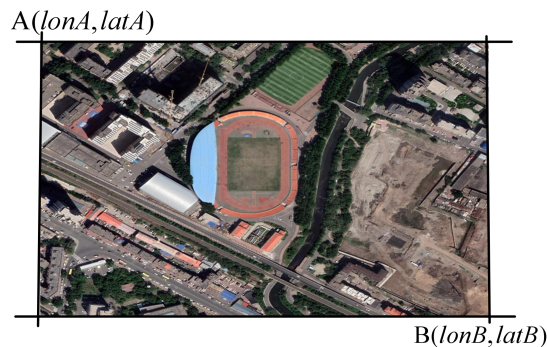


图 11 卫星影像经纬坐标示意

由图 11 可见，若要计算出已知经纬坐标信息的卫星影像所覆盖的地面距离，需要先计算出经纬度平均每度所对应的地面距离大小。在本项目的计算中，主要

考虑北半球的地理范围，在 WGS-84 坐标系下，当纬度一定时，经度间隔一度对应的距离可以根据式计算得到。

$$S = \frac{\cos(B) \times (R - lat \times (R - r) / 90) \times \pi}{180} \quad (21)$$

其中， B 为纬度大小，单位为度， $R=6378137\text{m}$ 为地球赤道半径大小， $r=6356752.314\text{m}$ 为地球极轴半径大小。

根据卫星影像坐标位置及式，可以计算出卫星影像在水平方向上的地面覆盖距离大小如式所示。

$$L_x = |lonB - lonA| \cdot \frac{\cos(lat) \times (R - lat \times (R - r) / 90) \times \pi}{180} \quad (22)$$

其中， lat 为 A、B 两点纬度的均值，取均值的目的是为了尽量缩小地球曲面对实际地面距离大小计算的影响。

同理可以根据式计算出卫星影像在竖直方向上的地面覆盖距离大小。

$$L_y = |latB - latA| \cdot \frac{(R - lat \times (R - r) / 90) \times \pi}{180} \quad (23)$$

本项目采取的基于位置信息的匹配区域选取根据 UAV 航拍图像成像时刻记录的相机经纬位置信息 $(lonT, latT)$ 作为航拍图像中心点的经纬位置，然后根据相机定位精度设定一个阈值 Rt ，在卫星影像中以航拍图像经纬度为中心，以阈值 Rt 为半径在全局卫星影像中截取卫星子图与 UAV 航拍图像进行匹配，从而达到提升匹配效率和匹配精度的效果，卫星影像子图的具体选取示意如图 12 所示，其中红色虚线框区域为根据上述地面覆盖区域距离计算以及阈值 Rt 大小选取的子图区域。

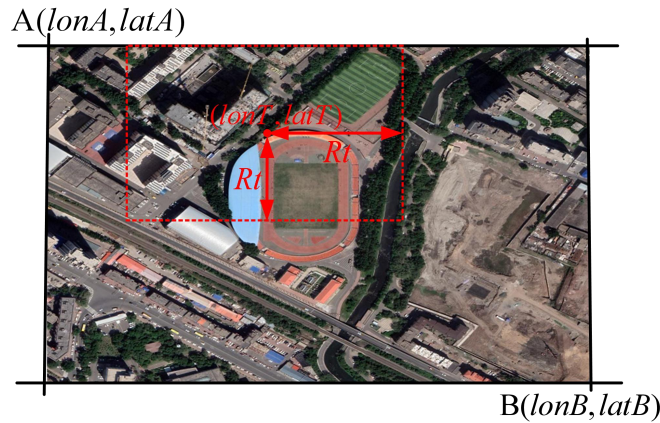


图 12 卫星影像子图截取示意

2 基于 CNN 特征的图片匹配方法

本项目通过相机位置参数和卫星影像的经纬度信息实现了 UAV 航拍图像和卫星影像的尺度和方向的配准，图像的配准是通过对 UAV 航拍图像的重采样来完成的，这个过程中会对图像造成一些细节信息的丢失，因此即使实现了图像的配准，也无法通过传统的特征点匹配或者模板匹配的方式来实现两者的精确匹配。为此，本项目通过深度学习网络提取图像特征向量，然后用模板匹配的思想对图像特征向量进行匹配，从而实现航拍图像与卫星影像的匹配，具体匹配流程如图 13 所示。

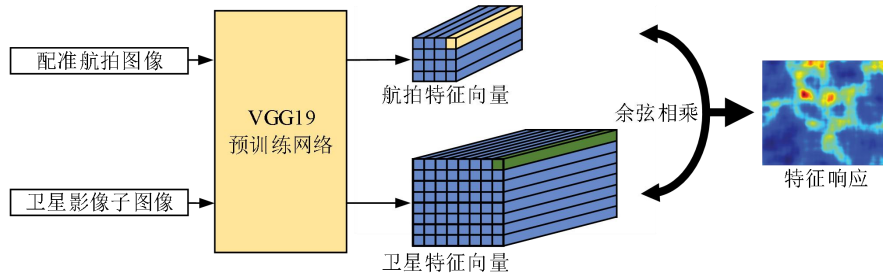


图 13 图像匹配流程

由图 13 可见，本项目图像匹配流程采用无人航拍图像配准后的图像和根据航拍图像位置信息在卫星影像截取的子图像作为匹配网络的输入，然后利用在 ImageNet 数据集上预先训练好的 VGG19 网络来分别提取待匹配图像的特征得到特征向量矩阵。将得到的航拍特征向量和卫星特征向量做余弦计算得到特征响应图像，然后定义一个特征匹配质量感知函数来求解匹配质量最高的区域作为匹配结果输出。

本项目以 UAV 航拍图像作为模板图像 T ，卫星影像的子图像定义为搜索图像 S ，匹配的目标是在搜索图像中找到模板图像的最佳匹配区域。图像匹配质量感知函数定义为 $Q(s, t)$ ，其中 t 和 s 分别是模板图像 T 和搜索图像 S 中的一小块区域，它们对应的特征向量表示为 f_t 和 f_s ，维度大小为 320。

给定一个搜索图像 S 中的块区域 s ，然后定义模板图像 T 中的块区域 t 与之相匹配的概率为 $P(t|s)$ ，其具体计算方式如式所示。

$$P(t|s) = \frac{\exp\{\alpha \cdot \rho(f_t, f_s)\}}{\sum_{t' \in T} \exp\{\alpha \cdot \rho(f_{t'}, f_s)\}} \quad (24)$$

其中, α 是正权重。就匹配质量而言, 该概率可以被解释为当前块区域 t 相对于模板图像 T 中所有其它块区域与搜索图像 S 中块区域 s 的匹配程度的排序。 $P(t|s)$ 的取值应该在 $[0, 1]$ 的范围内, 其值越大, 意味着块区域 t 和块区域 s 匹配的相似度越高。

根据定义可以得到概率 $P(t|s)$ 的本质是搜索图像 S 的一个块区域与模板图像 T 的匹配相似度得分, 同理可以定义模板图像 T 的一个块区域与搜索图像 S 的匹配相似度得分为 $P(s|t)$, 因此图像匹配质量感知函数 $Q(s, t)$ 可以根据式得出。

$$Q(s, t) = P(t|s) \cdot P(s|t) \quad (25)$$

得到匹配质量感知函数 $Q(s, t)$ 后, 根据整体匹配质量最优化原则, 在匹配质量感知函数中找出匹配质量响应极大值区域即为最佳匹配区域, 寻找最佳匹配区域的具体计算方式如式所示。

$$R = \arg \max_S \left\{ \sum_{s \in S} \max \{Q(s, t) | t \in T\} \right\} \quad (26)$$

其中, R 即为在搜索图像 S 中找到的模板图像 T 的最佳匹配区域。

3 图像匹配结果分析

为了验证本项目所提方法的可行性, 需要选取一个实验场景进行 UAV 航拍, 并获取该场景的卫星影像, 将航拍到的图像与卫星影像进行匹配分析。本项目选取的场地为哈尔滨工业大学田径场, UAV 航拍高度为 100m。实验采用的卫星影像从大地图卫星图像查看软件下载, 所下载卫星影像为 18 级影像, 其比例尺大小为 1:1575。图 4 所示为实际下载的哈尔滨工业大学田径场卫星影像图像, 所下载的影像地面覆盖区域对应的经度范围坐标为 126.624550°E - 126.632227°E , 纬度坐标范围为 45.734169°N - 45.738873°N 。

为了对实验结果进行准确的描述, 需要对航拍图像与卫星影像的匹配结果有一个定量分析比较。在实际分析过程中, 本项目利用匹配结果与真实匹配位置的交并比 (Intersection over Union, IOU) 来评价图像匹配质量的好坏。IOU 是一种测量在特定数据集中检测相应物体准确度的一个标准, 它的具体计算方式是交叠区域的交集面积与并集面积大小的比值, 具体如图 14 所示。

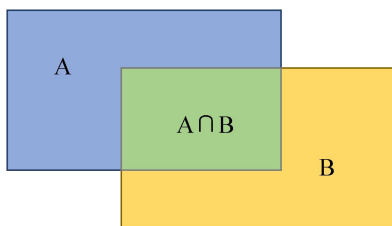


图 14 交并比示意图

在图 14 所示的交并比示意图中，假设蓝色长方形 A 的面积表示为 S_A ，黄色长方形面积表示为 S_B ，它们的交叠部分绿色长方形面积表示为 S_{AB} ，则根据定义 IOU 的计算公式如式所示。

$$IOU = \frac{S_{AB}}{S_A + S_B - S_{AB}} \quad (27)$$

在本项目的实验结果分析中，UAV 航拍图像和卫星影像匹配的真实匹配结果是通过手工选取航拍图像与卫星影像的特征匹配点对，然后根据匹配点对将航拍图像映射到卫星影像来获取，最终通过编程实现的匹配结果如图 15 所示。

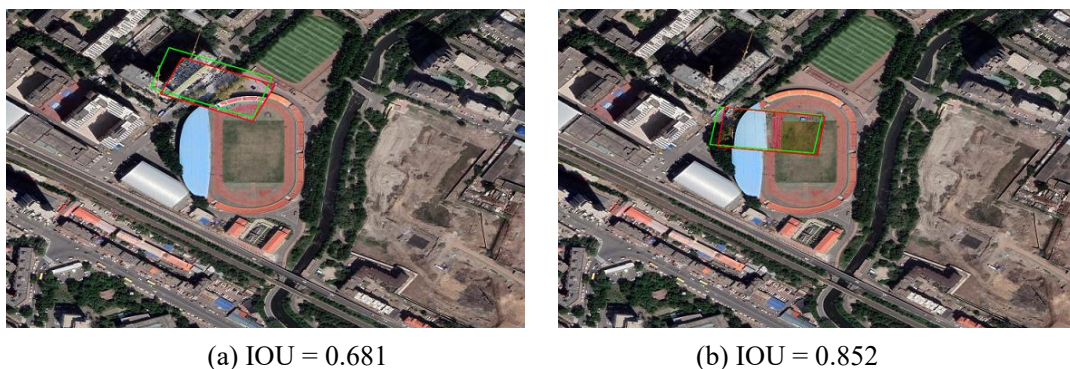


图 15 匹配结果

在实验结果分析中，本项目实验采用了 UAV 在哈尔滨工业大学田径场航拍到的 10 张图像与卫星影像进行匹配结果分析，并计算了匹配结果与真实匹配框之间的 IOU 大小，图 15 中红色框为匹配结果，绿色框为真实匹配位置。图 15(a) 匹配结果 IOU 为 0.681，为本次实验结果分析中匹配最坏的情况，图 15(b) 匹配结果 IOU 为 0.852，为本次实验结果分析中匹配结果最坏的情况。本次实验的全部匹配结果 IOU 具体如表 1 所示，匹配结果的平均 IOU 为 0.7823。可见，在最坏匹配结果情况下，UAV 航拍图像与卫星影像的匹配结果与真实匹配框都具有较大的交叠区域，达到比较好的匹配效果。但是在图 15(a) 中的红色匹配框和绿色真值框的形状大小以及方向差异较大，而图 15(b) 中的红色匹配框与绿色真值

框的形状大小和方向基本一致，其匹配精度也更高。这说明了通过位置参数实现的图像配准与真实配准图像之间是存在着一定误差的，而且这个误差对匹配精度存在较大的影响，这是因为 UAV 在航拍图像过程中 UAV 存在飞行漂移以及相机存在抖动等现象导致相机记录的位姿参数具有一定的偏差造成的，因此在后期的研究中需要对匹配算法进行改进，尽量减小相机位姿参数对匹配精度的影响。

表 1 匹配结果 IOU

序号	IOU	序号	IOU
1	0.681	6	0.793
2	0.819	7	0.807
3	0.810	8	0.715
4	0.852	9	0.745
5	0.802	10	0.799

4.1.2 基于图像匹配映射关系的定位方法

1) 基于图像匹配映射关系的定位原理

随着 UAV 航拍技术的普及，利用 UAV 航拍图像进行目标检测、目标跟踪以及场景识别等研究成为近年来的研究热点。在目标检测和目标跟踪的应用中，通过深度学习或者特征检测匹配等技术能够方便快捷的确定感兴趣目标在一幅图像中的像素位置，但是它们的实际地理坐标的求取通常较为复杂，而在大多数基于 UAV 航拍图像的目标检测和跟踪的应用中，如军事基地监查、罪犯追踪等，得知感兴趣目标的地理坐标位置是比较重要的。因此，本小节旨在利用 UAV 航拍图像与卫星影像匹配的先决条件，研究求解 UAV 航拍图像上特定目标的地理坐标位置的方法。

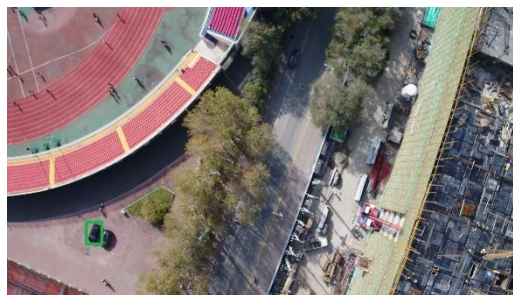


图 16 航拍图像目标检测结果

由于本项目研究的重点在于通过 UAV 航拍图像与卫星影像的匹配来解算航拍图像中感兴趣目标的地理位置，因此不再做目标检测的研究而是直接假设已知感兴趣目标在航拍图像中的具体像素位置。如图 16 所示绿色框检测到的小车，

假设绿色框中心点在像素坐标系下的坐标为 (w_0, h_0) ，当 UAV 航拍图像匹配到卫星影像上时，由于卫星影像的像素尺寸大小以及对应的经纬度坐标范围是已知的，因此只要计算出坐标 (w_0, h_0) 匹配到卫星影像上对应点像素坐标 (w_1, h_1) ，然后根据坐标 (w_1, h_1) 与卫星影像像素尺寸大小和经纬度范围关系便可求得目标小车的地理坐标位置。

回顾 UAV 航拍图像到卫星影像的匹配整体流程，UAV 航拍图像首先根据航拍时刻机载相机方位角的大小进行图像旋转实现航拍图像与卫星影像的方向配准；然后根据图像尺度空间配准算法对图像进行重采样实现航拍图像与卫星影像的尺度大小统一；最后通过分级匹配算法在卫星影像中找到与 UAV 航拍图像相似性最高的区域并将航拍图像变换到该区域中，从而实现航拍图像与卫星影像的匹配。

根据 UAV 航拍图像与卫星影像的匹配过程可见，求解目标小车在卫星影像上的像素坐标 (w_1, h_1) 需要先求解出原坐标 (w_0, h_0) 经过方向和尺度配准后的坐标，在此假设坐标 (w_0, h_0) 经过方向配准后的坐标为 (w_θ, h_θ) ，具体如图 17 所示。

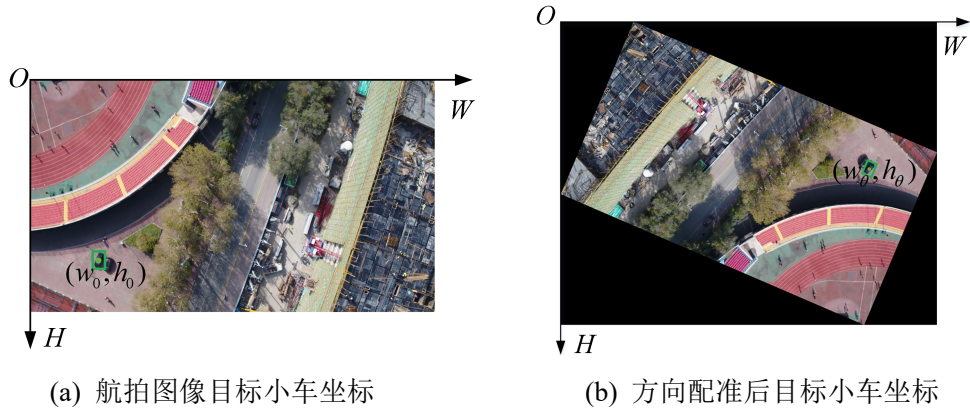


图 17 目标小车像素坐标

假设当前包含目标小车的航拍图像成像时刻相机的方位角为 φ ，则进行方向配准后的目标小车像素坐标 (w_θ, h_θ) 可以根据图像方向配准小节推导的图像旋转变换公式计算，具体如式所示。

$$\begin{bmatrix} w_\theta \\ h_\theta \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.5W \\ 0 & -1 & 0.5H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.5W \\ 0 & -1 & 0.5H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ h_0 \\ 1 \end{bmatrix} \quad (28)$$

其中， H 为航拍图像像素高度， W 为航拍图像像素宽度。

经过方向配准后的图像，再通过图像重采样进行尺度空间配准，将目标小车经过尺度空间配准后的像素坐标标记为 (w'_θ, h'_θ) ，尺度空间配准后的目标小车像素坐标 (w'_θ, h'_θ) 可以根据式求解。

$$\begin{bmatrix} w'_\theta \\ h'_\theta \end{bmatrix} = \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \begin{bmatrix} w_\theta \\ h_\theta \end{bmatrix} \quad (29)$$

其中， σ_x 、 σ_y 为图像像素尺度大小比例因子。

由于航拍图像已经经过方向和尺度上的配准，因此 UAV 航拍图像变换到卫星影像匹配区域的过程实际上是航拍图像像素坐标偏移的过程，即尺度空间配准后的目标小车像素坐标 (w'_θ, h'_θ) 经过偏移即可变换到其在卫星影像上对应位置的像素坐标 (w_1, h_1) ，其坐标的偏移量大小为卫星影像与航拍图像匹配区域的中心像素坐标与配准后的航拍图像中心像素坐标的差值，具体如图 18 所示。

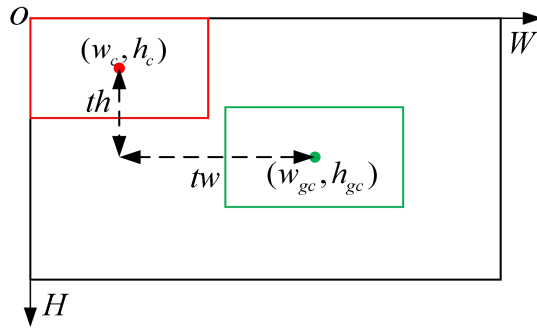


图 18 坐标偏移量计算示意

在图 18 中，黑色矩形框表示卫星影像，红色矩形框表示配准后的 UAV 航拍图像，红色点表示图像中心在像素坐标系下的坐标，绿色矩形框表示 UAV 航拍图像与卫星影像的匹配区域，绿色点表示匹配区域中心在卫星影像像素坐标系下的坐标。根据图示关系，配准后的航拍图像变换到卫星影像匹配区域的偏移量可以通过式计算。

$$\begin{aligned} tw &= w_{gc} - w_c \\ th &= h_{gc} - h_c \end{aligned} \quad (30)$$

其中, tw 表示像素坐标系下 W 方向的偏移量, th 表示像素坐标系下 H 方向的偏移量。

综合上述推导, UAV 航拍图像上的目标小车像素坐标 (w_0, h_0) 经过匹配后变换到卫星影像像素坐标系下的坐标 (w_1, h_1) 的最终求解公式如式所示。

$$\begin{bmatrix} w_1 \\ h_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \sigma_x & 0 & tw \\ 0 & \sigma_y & th \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0.5W \\ 0 & -1 & 0.5H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.5W \\ 0 & -1 & 0.5H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ h_0 \\ 1 \end{bmatrix} \quad (31)$$

根据卫星影像经纬度坐标关系, 当卫星影像像素大小为 $H_a \times W_a$, 左上角经纬度坐标为 $(lonA, latA)$, 右下角经纬度坐标为 $(lonB, latB)$ 时, 将目标小车的经纬度坐标记为 $(lonC, latC)$, 则根据像素坐标系的定义, 可以将目标小车的像素坐标与卫星影像像素大小的比值作为其位置偏移的大小, 由此目标小车的经纬坐标 $(lonC, latC)$ 可以通过公式计算得到。至此, 完成了利用 UAV 航拍图像与卫星影像的匹配关系解算航拍图像中特定目标的地理坐标推导过程。

$$\begin{aligned} lonC &= lonA + (lonB - lonA) \cdot \frac{w_1}{W_a} \\ latC &= latA + (latB - latA) \cdot \frac{h_1}{H_a} \end{aligned} \quad (32)$$

2) 目标定位实验结果分析

为了验证本项目提出的基于航拍图像与卫星影像匹配的目标定位方法的可行性, 本节将对定位实验结果的定位误差进行实验仿真分析。假设航拍图像目标的真实经纬坐标为 $P(lon_0, lat_0)$, 该目标利用本项目所提出的定位方法解算出的经纬坐标为 $P'(lon_1, lat_1)$, 则目标的定位误差为真实位置与定位结果之间的距离大小, 具体计算方式可以根据式进行计算。

$$e = \pi \bar{R} \cdot \arccos(\cos(lat_1) \cos(lat_0) \cos(lon_1 - lon_0) + \sin(lat_1) \sin(lat_0)) / 180 \quad (33)$$

其中, $\bar{R} = 6371000 m$, 表示地球平均半径。

为了验证定位精度, 本项目在匹配的 30 张 UAV 航拍图像中共选取了 53 个地面固定点如旗杆位置、田径场草坪角点等进行定位, 并按公式进行定位误差计算, 最终统计定位误差的累计分布函数 CDF 曲线如图 19 所示。

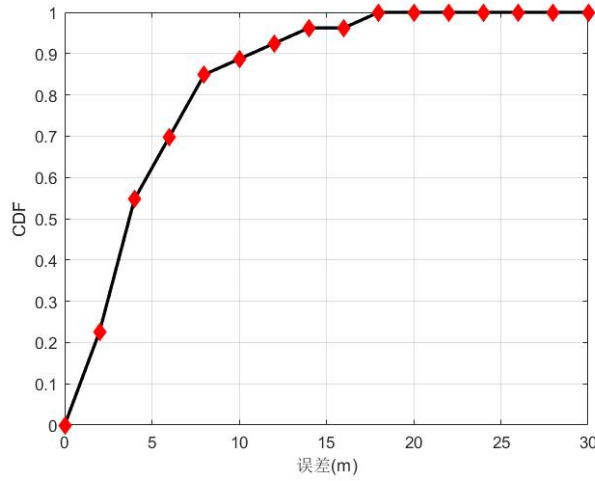


图 19 定位结果误差 CDF 曲线

通过图 19 定位误差曲线可见, 本项目所提基于航拍图像与卫星影像匹配的航拍图像目标定位算法的最大定位误差为 18m, 定位误差在 10m 以内的累计概率大约为 90%。在定位误差分析中, 定位点的真实位置是通过手动在卫星影像上选取确定的, 由于卫星影像的分辨率较低, 因此不可避免的存在一定的偏差。本项目所提航拍图像目标定位精度是在卫星影像尺度上进行度量的, 而所采用的卫星影像尺度比例为 1:1575, 因此真实位置选取的偏差相对于卫星影像的尺度大小来说是微小的, 可以忽略不计。而本项目所提定位方法 18m 的定位误差结果相对卫星影像尺度来说是一个比较高的精度, 满足一般高空航拍图像目标跟踪定位的要求, 具有一定的实际应用价值。

4.1.3 基于链路感知和链路持久性的路由选择策略

1) 路由选择策略理论分析

1 标准 MPR 选择算法

文档 RC3626 提出了 OLSR 路由协议的多点中继技术, 即 MPR 选择算法。核心思想是节点向周围节点发送 HELLO 分组获取一跳邻居节点, 通过一跳邻居

表中的信息获取与一跳邻居节点具有对称链路的严格二跳邻居节点的节点可达度。从具有最高节点可达度的一跳邻居节点中选择 MPR 节点。由于 OLSR 规定只有被选择为 MPR 的节点才有转发 TC 分组的资格，因此通过多点中继的方法极大降低了主动路由协议由于全网洪泛的机制不断发送控制分组所产生的路由开销。MPR 节点示意图如图 20 所示。

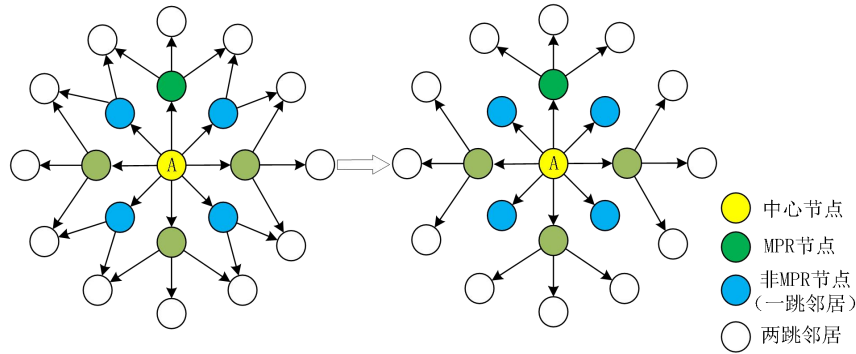


图20 MPR节点示意图

对标准 MPR 选择算法进行阐述之前，需要声明 OLSR 中的一些符号的含义：

- (1) M ：本节点 i 的 MPR 集合；
- (2) N ：本节点 i 的所有一跳邻居节点集合；
- (3) $N2$ ：本节点 i 的 2 跳邻居节点集合， $N2$ 中不包括以下特征的节点：
 - (i) 转发意愿为 0 的节点；
 - (ii) 节点 i 本身；
 - (iii) 与节点 i 存在对称链路，可以一跳到达。
- (4) $N_willingness$ ：转发意愿，一共分为 5 个级别，如表 2 所示。

表2 MPR参数转发意愿

参数	值	含义
WILL_NEVER	0	不转发
WILL_LOW	1	转发意愿较低
WILL_DEFAULT	3	默认
WILL_HIGH	6	转发意愿较高
WILL_ALWAYS	7	总是同意转发

- (5) $D(y)$ ：节点可达度，即通过本节点 i 的一跳邻居集 N 中的某一邻居节点 y 所能到达的二跳邻居节点的个数，但不包括以下情况：
 - (i) 该二跳邻居节点位于一跳邻居集 N 中；
 - (ii) 本节点 i 自身。

(6) **Reachability**: 节点可达性, 用于计算 MPR 集。此概念表示撤去 MPR 集节点以及它们连接到的二跳邻居节点后, 一跳邻居集 N 中的某个节点所覆盖的二跳邻居节点的个数, 这些二跳邻居节点中不包含 N 中的成员以及本节点 i 。具体算法流程如表3所示。

表3 MPR节点选择过程

Step1.	一跳邻居集 N 中节点的转发意愿是否都为 WILL_ALWAYS, 若是, 执行step2 若不是, 丢弃不符合的节点后继续执行step1
Step2.	计算 N 集中节点 y 的可达度 $D(y)$
Step3.	判断 N 集中是否存在到达 $N2$ 集中的某一节点的唯一节点, 若是, 则将其加入到MPR集中, 并移除 $N2$ 集中被覆盖的节点 若否, 则执行step4.
Step4.	判断 $N2$ 集中是否存在未被移除的节点, 且 N 集中还存在节点 若是, 则执行step5 若否, 则算法结束
Step5	计算 N 集中所有一跳节点的可达性
Step6	以转发意愿为准则按从高到底的顺序将节点加入MPR集, 若存在相同转发意愿强度的节点那么挑选可达性最高的, 若可达性相同那么挑选可达度最大的节点, 算法结束

通过上述过程可以筛选出最适合广播 TC 分组的 MPR 节点, 从而快速建立起路由表和拓扑表进行信息传输, 降低由于网络洪泛广播所带来的路由开销。传统 MPR 算法在设计时并没有考虑到节点的高速移动带来的拓扑频繁变化的情况, 拓扑的频繁变动会进一步加剧组网中无人机之间的通信链路的断裂或失效等状况。除此之外, 影响无人机通信的因素还有链路的传输质量、链路的有效时间等等。因此, 传统 MPR 选择算法仅使用转发意愿下的单一可达性作为选择 MPR 节点的判别标准并不能很好地适用于当前飞行自组网这类复杂的网络环境。

2 优化 MPR 集算法

2.1 相邻无人机节点链路质量计算

本节对链路传输质量、链路生存时间进行预测, 并结合节点的可达性, 综合考虑上述三种指标, 提出了一种基于链路感知的 MPR 选择策略对传统的 OLSR 协议进行优化, 选择质量更好的 MPR 节点作为中继路由, 以稳定网络结构, 更好地适用于飞行自组网的网络环境。

无人机节点之间的链路传输质量可以通过传输数期望 ETX (Expected Transmission Count) 来表示, 无人机节点 i 和 j 之间的链路质量公式表达式为:

$$ETX_{ij} = \frac{1}{R_f \times R_r} \quad (34)$$

其中, R_f 是转发接收比, 即数据包成功到达接收方的概率; R_r 为反向接收比, 即 ACK 报文被成功接收的概率。换句话说, ETX 会对数据包的预期传输 (包括重传输) 进行估计, 以便在目的地不会错误地接收到它。

R_f 和 R_r 是通过 HELLO 分组进行链路探测来衡量的, 在 OLSR 协议中, 接受比 R_f 使用指数移动平均来计算, 其表达式为:

$$\begin{cases} R_f^l = \alpha h_l + (1-\alpha)R_f^{l-1} \\ R_f^0 = 0 \end{cases} \quad 0 \leq \alpha \leq 1 \quad (35)$$

其中, 若接收到第 l 个 HELLO 消息分组, h_l 为 1, 若没有接收到, 则 h_l 为 0。 α 是 OLSR 参数, 称为链路质量(Link-Quality Aging)。需要注意, α 的取值关系到链路质量估计的准确性。 α 值越大, 接收比就越平均, 从而得到更稳定可靠的估计; α 越小, 接收比变化越大, 会促使节点发送更多的 HELLO 分组以更新链路质量。

为了计算方便, 将 R_f 和 R_r 用两个节点一定周期范围内收到它的邻居节点所发送 HELLO 消息的比例来表示, 计算公式如下:

$$R_f = \frac{R_{ij}}{S_{ij}} \quad (36)$$

$$R_r = \frac{R_{ji}}{S_{ji}} \quad (37)$$

其中, R_{ij} 表示节点 i 从节点 j 接收到的 HELLO 分组数, S_{ij} 表示节点 i 发送给节点 j 的 HELLO 分组数, R_{ji} 表示节点 j 从节点 i 接收到的 HELLO 分组数, S_{ji} 表示节点 j 发送给节点 i 的 HELLO 分组数。

然而 OLSR 协议中本节点只可以获得自身发送给邻居节点以及邻居节点发送给自身的 HELLO 分组, 即 R_{ij} 和 R_{ji} , 仅通过这两项数据是无法计算转发接受比和反向接受比的, 进而不能得到链路质量值。因此, 将 R_f 和 R_r 的分子互换, 得到 R_{f_i} 和 R_{r_i} 同时不影响 ETX 的计算, 如公式和公式所示。

$$R_{f_i} = \frac{R_{ji}}{S_{ij}} \quad (38)$$

$$R_{r_i} = \frac{R_{ij}}{S_{ji}} \quad (39)$$

ETX的表达式为：

$$ETX_{ij} = \frac{1}{R_{f_i} \times R_{r_i}} \quad (40)$$

路由R的ETX为组成该路由的各链路的ETX度量之和，公式如下：

$$ETX_R = \sum_{(i,j) \in R} ETX_{ij} \quad (41)$$

2.2 相邻无人机节点链路保持时间计算

本节中所使用的每一架无人机都装备了全球定位系统（Global Positioning System, GPS），因此每个无人机节点都能感知自身的位置和移动速度，可以通过节点之间相对运动的情况以预测节点之间的链路保持时间。定义链接保持时间 LET（Link Expiration Time），节点相对运动的示意图如图 21 所示。

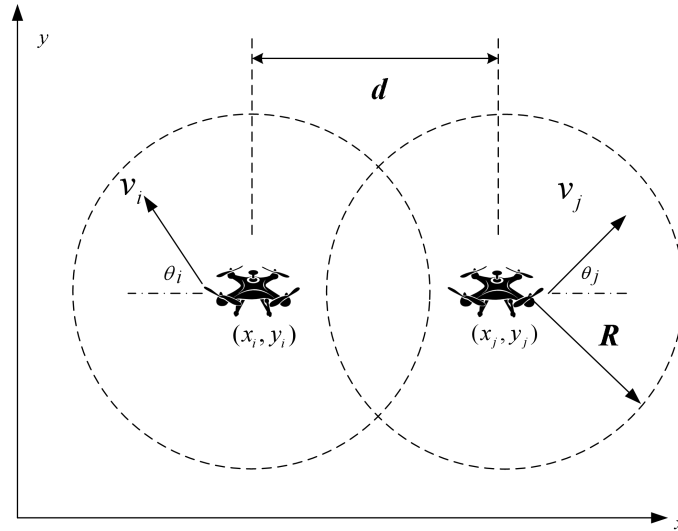


图21 无人机节点相对运动示意图

图中的各符号的含义如表 4 所示。

表4 节点预测过程各符号参数解释

参数	含义
v_i	节点 <i>i</i> 移动速度
θ_i	节点 <i>i</i> 移动方向
(x_i, y_i)	节点 <i>i</i> 坐标
d	节点 <i>ij</i> 的距离

$$\underline{\quad\quad\quad R \quad\quad\quad} \quad \text{节点通信范围}$$

要保证相邻无人机节点之间的通信链路有效，需要两架无人机之间的距离不能超出其最大通信范围，也就是：

$$d < R \quad (42)$$

其中，无人机之间的实时距离可以通过 GPS 设备中的位置信息得到，即：

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (43)$$

经过一段时间之后，两架无人机都各自产生了一段位移，此时两节点的水平方向距离的为 D_x ，垂直方向的方向距离 D_y ，则有

$$\begin{aligned} D_x &= x_i - x_j + [v_i \cos(\theta_i) - v_j \cos(\theta_j)]t \\ D_y &= y_i - y_j + [v_i \sin(\theta_i) - v_j \sin(\theta_j)]t \end{aligned} \quad (44)$$

因此，带入公式，则有

$$D_x^2 + D_y^2 < R^2 \quad (45)$$

通过解方程可以得到链路保持时间的表达式如公式所示：

$$LET = \frac{-(ab + cd) + \left[(a^2 + c^2)r^2 - (ad - bc)^2 \right]^{\frac{1}{2}}}{a^2 + c^2} \quad (46)$$

其中，

$$\begin{aligned} a &= v_i \cos(\theta_i) - v_j \cos(\theta_j), \quad b = x_i - x_j \\ c &= v_i \sin(\theta_i) - v_j \sin(\theta_j), \quad d = y_i - y_j \end{aligned} \quad (47)$$

2.3 中继节点选择策略

N1 代表无人机源节点 i 的一跳邻居集，在选择节点 i 的 MPR 节点集合时，需要综合链路质量、链路保持时间以及节点的可达性对于 MPR 选择的影响，因此，选择 MPR 节点的指标描述如下：

(1) 强连通性：该指标用于衡量无人机相邻节点之间的通信链路质量水平，使用 L 表示。节点之间的 ETX 值越大，说明通信链路质量越好，优化算法将选择 L 值最大的节点作为 MPR 节点。

$$L = \min_{j \in N1} ETX_{ij} \quad (48)$$

(2) 高持久性：该指标用于衡量无人机相邻节点之间通信时间的稳定性，使用 ΔT 来表示。节点之间的 LET 越大说明通信链路保持时间越久，性能越好，

优化算法将选择 LET 值最大的节点充当 MPR 节点。

$$\Delta T = \max_{j \in N1} LET_{ij} \quad (49)$$

(3) 可达性好：该指标用于衡量无人机本节点可以覆盖的二跳邻居节点的数量，令 Re_j 代表节点 j 的可达性，可达性越强说明节点 j 的邻居数量越多，优化算法将选择 Re_j 最大的节点作为 MPR 节点，以减少网络洪泛的次数，从而降低开销，提升网络性能。

$$Re = \max_{j \in N1} Re_j \quad (50)$$

通过进一步分析发现，链路保持时间和节点可达性的值越大说明链路稳定性越好，而 ETX 的值域为 $[1, +\infty]$ ，链路传输数期望越小说明链路传输质量越高，变化趋势和其他指标相反，不便于对整体的 MPR 选择算法的指标进行加权平均处理运算，因此将链路传输期望的表达式取倒数，定义了一个新的链路传输质量指 $RETX$ ，其值域为 $[0, 1]$ ，计算公式如所示

$$RETX_{ij} = R_f \times R_r \quad (51)$$

使用公式将三者进行加权计算，选择数值最大的节点作为 MPR 节点。

$$N_w = \begin{cases} \varepsilon \times RETX_{ij} + \beta \times LET_{ij} + \gamma \times Re_j & d < R \\ 0 & d \geq R \end{cases} \quad (52)$$

其中， ε 、 β 、 γ 分别是三个变量的加权因子，且满足 $\varepsilon + \beta + \gamma = 1$ ，可以根据飞行自组网的侧重方向进行权值的调整。 d 是无人机 i 与 j 的距离， R 代表无人机 i 的通信范围，如果经过一个周期之后无人机 i 发送 HELLO 分组时，预测无人机 j 仍在无人机 i 的有效通信距离内，那么无人机 j 就是无人机 i 待选的 MPR 节点；若过一个周期之后无人机 j 已经不在无人机 i 的通信范围内，那么无人机 j 就不在待选范围内。 N_w 越大表明该无人机节点越容易被选择为 MPR 节点。通过上述过程可以得到一个新的 MPR 节点选择指标，优化 MPR 选择算法的流程如图 22 所示。

通过对图 22 的分析，可以得到优化 MPR 集的评判准则：

(1) 检测 N 集中所有节点的转发意愿是否都为 WILL_ALWAYS，将不是的节点移除，然后执行步骤 (2)；

(2) 计算节点可达度 $D(v)$ ；

(3) 判断 N 集中是否存在到达 $N2$ 集某一节点的唯一节点，若存在，则将该节点加入到 MPR 集中并移除 $N2$ 中所被覆盖的节点，继续执行步骤 (3)；若不存在，则执行步骤 (4)；

(4) 将 N 集中 N_w 值最大的节点加入到准 MPR 集中；

(5) 判断准 MPR 集是否将 2 跳邻居节点集 $N2$ 覆盖，若不是，则返回步骤 (4)；若存在，将准 MPR 集变为 MPR 集，算法结束。

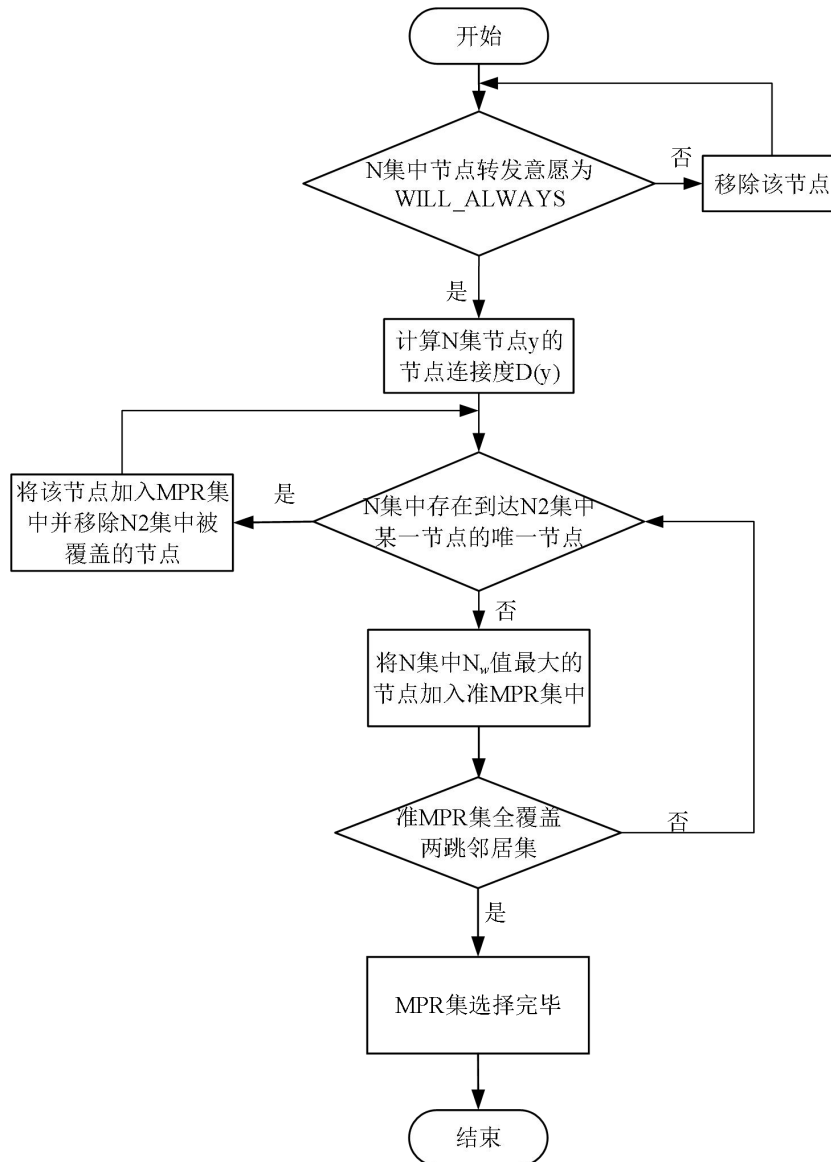


图22 优化MPR集算法过程

3 基于链路持久性的路由算法设计

通过对链路预测时间的估计，提出一种基于链路保持时间的最大链路持久性的路由算法。该算法用于在计算路由时会选择一跳最优路径。如存在多条路径时，

选取路径跳数较短的一条，若跳数相同则选择相邻节点链路保持时间最长的一条路径。

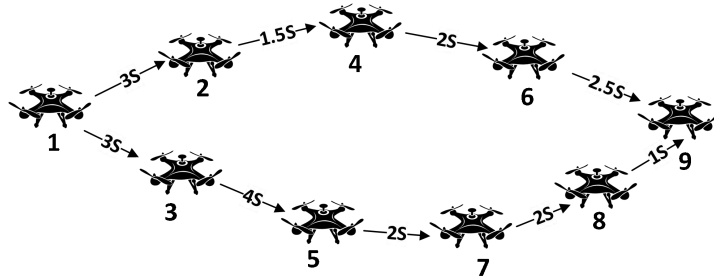


图23 链路持久性路由示意图

图 23 表明，无人机节点从 1 到 9 有 2 条备选路径，那么 1→2→4→6→9 这条路径最大保持时间是 1.5S，而 1→3→5→7→8→9 这条路径最大的保持时间是取 1S，因此无人机 1 到无人机 9 的路径最大保持时间即 1→2→4→6→9 对应的 1.5S。

算法实现过程如表 5 中的伪代码所示。在改进算法过程中，在 OLSR 中的 TC 分组中新加入一个 Keetime 字段，Keetime 是指每个 MPR Selector Address 字段和 MPR 节点预测的链路保持时间，占 8bytes，取值范围是 0-255，其计算公式如下所示：

$$t_{\text{Keep_time}} = \begin{cases} \left\lfloor \frac{t}{\Delta T} \times 255 \right\rfloor & t < \Delta T \\ 255 & t \geq \Delta T \end{cases} \quad (53)$$

公式中的 t 代表是预测的两架无人机节点之间的链路保持时间，单位是毫秒； ΔT 代表 TC 分组的发送间隔， t 的值在 0 到 255 之间，方便在 TC 分组中储存。

表5 链路持久性路由选择算法

1	BEGIN
2	While(路径<N)
3	{
4	IF ((路径1跳数<路径2跳数) (路径1跳数=路径2跳数&&路径1时间>路径2时间))
5	返回路径1;
6	ELSE 返回路径2;
7	}
8	END

改进后的路由表计算步骤如下：

- 1) 清空路由表；
- 2) 对于中心节点，从一跳邻居表中将与之具有对称链路的节点添加到路由表中，根据表项记录链路的有效时间。
- 3) 将二跳邻居集中的节点添加至路由表中，若网络中的路径有跳数相同的情况，则选择路径上中链路保持时间最长的一跳路径作为传输路由，更新跳数与链路保持时间。
- 4) 在MPR节点选择后，MPR节点携带TC分组进行全网广播以建立拓扑表。扫描表中的所有项，若路由表中不存在该拓扑表（记为L1）的目的接口地址，但路由表中存在与L1的目的接口地址前一跳地址相同的表项时，将L1的目的接口地址添加至路由表中，设L1中的链路保持时间为 T_1 ，拓扑表中的链路保持时间为 T_2 ，链路保持时间更新公式如下所示：

$$T_{R_keeping_time} = \begin{cases} T_1 & T_1 < T_2 \\ T_2 & T_1 \geq T_2 \end{cases} \quad (54)$$

若存在可用路由时，会优先选跳数最短的路由，若跳数相同时则选择邻节点间链路保持最久的路由加入到表项中并更新路由表。

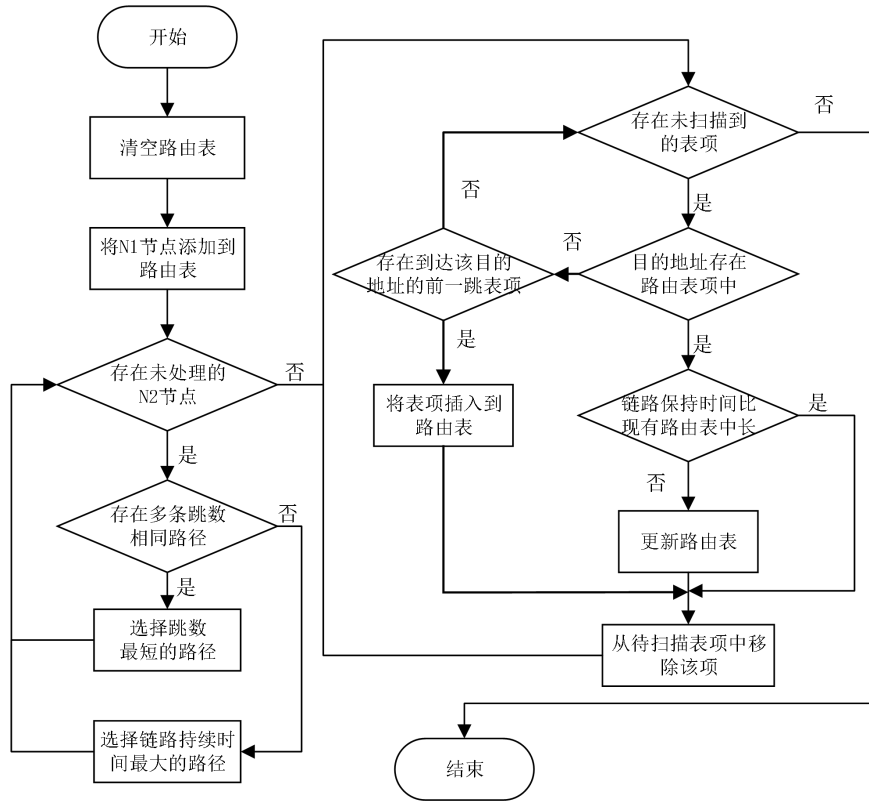


图24 路由计算流程

以上，在对 MPR 节点选择算法的优化和链路持久性进行选择之后，得到计算全局的路由表和拓扑表。将改进后的基于链路感知和链路保持时间的路由策略加入原始 OLSR 协议中，得到改进的 LK-OLSR (Link awareness and Keeping time based OLSR) 协议，并对改进协议的各种路由表项以及控制分组进行详细的修改。

2) 路由选择策略算法实现

通过上一节的理论分析可知，优化后的 MPR 选择指标需要得知链路传输质量和链路生存时间的信息，但原 OLSR 协议中的信息表分组中并不携带相关数据，因此需要对协议中分组消息的表格式进行修改，以加入相关信息。对 HELLO 分组的修改如图 25 所示。

0	8	16	24	32
Reserved		Htime	Willingness	
X-speed	Y-speed	X-position	Y-position	
Link Code	Reserved	Link Message Size		
ETX				
LET				
Neighbor Interface Address				
...				
Link Code	Reserved	Link Message Size		
Neighbor Interface Address				
ETX				
LET				
Neighbor Interface Address				
...				

图25 修改后HELLO分组表格式

(1) 虽然无人机携带 GPS 设备可以感知自身位置和速度信息，但原 OLSR 协议并不能通过无人机获取。因此需要将无人机节点二维平面上的位置坐标和实时速度信息加入到 HELLO 分组中。

(2) 计算 MPR 选择指标 N_w 中的 ETX 时，需要知道转发接受比和反向接受比，但可以通过收发 HELLO 数据包的数量得知，因此只需要将计算后的 ETX 指标加入到 HELLO 分组的新字段中。

(3) 通过公式可知， LET 的计算需要知道无人机节点的位置信息和速度信息，在步骤(1)中已经添加了速度和位置信息，连同 LET 的值一同加入到 HELLO 数据中，并将速度和位置信息共用 32 比特以尽可能不增加额外的网络开销。

除了对 HELLO 分组进行修改之外，无人机节点还需要将 ETX 和 LET 的值导入到本地链路信息表、一跳邻居集、以及二跳邻居集中，从而通过表中的信息计算新的 MPR 选择指标。

修改后的本地链路表如图 26 所示。

L_local_iface_addr	L_neighbor_iface_addr	L_SYM_time	L_ASYM_time	L_time	ETX	LET
--------------------	-----------------------	------------	-------------	--------	-----	-----

图26 修改后的本地链路表

修改后的一跳邻居表如图 27 所示。

N_neighbor_main_addr	N_status	N_willingness	ETX	LET
----------------------	----------	---------------	-----	-----

图27 修改后的一跳邻居表

修改后的二跳邻居表如图 28 所示。

N_neighbor_main_addr	N_2hop_addr	N_time	ETX	LET
----------------------	-------------	--------	-----	-----

图28 修改后的二跳邻居表

为了获得路径上的最大链路保持时间，需要完善 TC 分组的表格式，将 Keptime（链路保持时间）字段添加到 TC 表中，通过 Keptime 字段可以预测每个 MPR Selector Address 字段和 MPR 节点的链路保持时间，占 8bytes。修改后的 TC 分组表格式如图 29 所示。

ANSN		Reserved	
MPR Selector Address			
MPR Selector Address			
Keetime	Keetime	Keetime	Keetime
...			

图29 修改后的TC分组表格式

为了选取链路持久性最大的路由，将提出的基于链路持久性的路由算法应用于路由选择中，需要修改路由表项，将 R_keeping_time 字段（路由保持时间）添加到路由表中，如存在多条路径时，选取路径跳数较短的一条，若跳数相同则选择相邻节点链路保持时间最长的一条路径。

修改后的路由表格式如图 30 所示。

R_dest_addr	R_next_addr	R_dist	R_iface_addr	R_keeping_time
-------------	-------------	--------	--------------	----------------

图30 修改后的路由表格式

4.1.4 基于拓扑变化的控制消息自适应算法

1) 控制消息的自适应调整

1 HELLO 分组自适应优化

OLSR 协议周期性地向网络中广播控制消息建立路由表以及拓扑结构，并在路由断裂时进行路由更新和维护，但飞行自组网中无人机之间的相对移动速度很

快,使得整体拓扑变化频繁而导致路由断裂,不稳定的拓扑结构会使得网络的整体性能下降。另外由于无人机的移动模型通常是随机航路点模型,节点移动速度是随机的,因此,OLSR 协议周期性地广播控制消息不能适应飞行自组网的拓扑变化情况,造成网络控制消息的多余或者网络拓扑更新不及时的情况。

针对上述分析情况,本节通过侦听节点的一跳邻居集获得本节点和一跳节点的相关信息,通过链路信息的变化情况改变 HELLO 消息的发包间隔;由于 TC 分组只能由 MPR 节点发送,因此通过侦听 MPR Selector 集的变化情况来自适应调整 HELLO 信息的发送间隔。总体上在拓扑变化频繁时降低控制分组的发送间隔,在拓扑变化缓慢时增大路由控制分组的发送间隔,在不影响时延和分组投递率的前提下对 OLSR 协议的网络路由开销进行正向优化,尽量降低控制分组转发,降低路由控制开销,提高网络的性能。

OLSR 协议的性能与控制消息的发送频率有关,发送间隔通常在某在一个区间中变化,若发送频率过大,网络中会充斥大量未被处理的控制消息,导致网络带宽被大量占用,能耗增加,进而导致丢包率上升;若控制消息发送频率过小,特别是在飞行自组网这种拓扑变化频繁的网络中,控制消息少会导致路由来不及更新,链路失效,进而影响网络的整体性能。OLSR 协议在 RFC3626 文档中规定了控制包的发送间隔限制以及最长有效时间,统一存放在 OLSR 包的消息头部,有效时间的表达式如下:

$$V_{time} = c \times (1 + \frac{a}{16}) \times 2^b \quad (55)$$

其中, a 和 b 分别代表控制消息间隔和有效时间, C 为常数,默认值为 0.0625s, a 和 b 最大值为 15,最小值为 0,可根据实际情况设置。

OLSR 协议中的节点通过向邻居节点发送 HELLO 分组进行建立双向链路,获取一跳邻居节点到二跳邻居节点的可达度,在优化 MPR 算法的基础上对 MPR Selector 集进行挑选。然而原始 OLSR 协议中 HELLO 分组的发送间隔是固定值,不能很好地适用于飞行自组网这类高动态环境,因此需要对其进行调整。由于 HELLO 分组发送的目标是原始节点的一跳邻居,只需链路信息集中的节点链路变化情况来自适应调整 HELLO 包的发送时间间隔。

OLSR 协议本地链路信息表中储存了本地和邻居节点的接口地址、链路状态

以及链路的有效时间等字段，一跳邻居表中储存了一跳邻居节点的接口地址和链路状态以及链路的有效时间等等。拓扑变化比如新节点入网或者就节点出网造成的链路状态改变都会记录在相应的信息表中。基于上述内容，对 HELLO 分组的发送间隔进行优化处理的核心思想就是根据链路信息集中链路变化情况对发送间隔自适应处理。

首先需要计算两次 HELLO 分组发送间隔 ΔH 内的链路变化情况，定义一个参数 M 用以统计一个 ΔH 内的链路变化数量。为了增加可靠性，需要对三个 ΔH 内的 M 值进行加权处理，得到节点的链路稳定度 D ，通过 D 值可以计算下一个 HELLO 包消息的发送时间。

由于 OLSR 中 HELLO 包的 ΔH 是固定值， ΔH 默认为 2s。在调整发送周期之后， ΔH 的值应该在一个范围之内变化。设置 ΔH 的最大值为 ΔH_H ，最小值为 ΔH_L ，其中 $\Delta H_L \leq \Delta H \leq \Delta H_H$ 。根据仿真结果分析，当 ΔH_L 较小时，网络性能有一定程度改善，但占用过多的网络带宽，得不偿失；当 ΔH_H 较大时，路由更新速度慢，导致链路断裂，丢包率上升，造成网络性能的严重下降。因此 ΔH 取值范围应该满足以下条件：

$$\begin{cases} \Delta H_L \geq 0.5\Delta H \\ \Delta H_H \leq 0.5\Delta H \end{cases} \quad (56)$$

根据上述理论可知，满足以下三个条件都会使得 M 值发生变化：

(1) 当节点信息库有新的链路出现时， M 的值加 3，新链路增加的数量记 L_1 ，原因是此时节点附近新增加了一个一跳邻居节点，此节点想要与新的邻居节点建立对称链路需要进行邻居侦听，该过程要经过 3 次握手机制。若想要链路建立的过程更快，则要增加该条件的权重值。因此当节点的信息库中有新的节点加入时，链路变化数量 M 的值加 3。

(2) 当节点的信息库中某条链路状态由其他状态变为 SYM_LINK， M 加 1，链路状态改变的数量记为 L_2 。原因是 SYM_LINK 代表链路状态为对称链路，说明此时本地节点已经和新增加的节点建立了对称链路，需要将此变化进行统计，

所以 M 值加 1。

(3) 当节点的信息库中某条链路由对称状态变为非对称状态时, M 值加 2, 此时的链路状态改变的次数记为 L_3 。对称链路状态为 **SYM_LINK**, 非对称链路状态为 **ASYM_LINK**, 说明节点此时已经不能跟另一个节点之间产生双向通路, 想要通信需要重新连接, 针对此状况应该及时处理, 减小 HELLO 分组消息的发送间隔, 此时 M 值加 2。

一个特殊情况是当节点信息库中的某条链路被删除或者状态变为 **LOST_LINK**, 说明此时某个节点不再是本地节点的一跳邻居节点, 这种情况不会对本地节点的邻居探测过程产生影响, 因此 M 值不发生变化。

经过上述分析, 可以得到 ΔH 内的 M 值计算公式:

$$M = 3L_1 + L_2 + 2L_3 \quad (57)$$

但是 M 值只能代表当前 HELLO 分组发送间隔之内链路变化的情况, 该方法并没有考虑到之前的 ΔH 对于链路变化情况的影响, 若邻居节点的变化较大时, M 值在两个相邻的 ΔH 内会有很大的变化, 会对当前 ΔH 内的 M 值的计算产生误差, 使得下一个 ΔH 设置不准确, 从而造成网络性能的下降。为了避免 ΔH 出现大幅度变动, 本章在 ΔH 的计算中引入缓冲机制, 结合之前的 ΔH 内计算出的 M 值, 对三个 ΔH 内的数据进行加权得到节点的稳定度 D , 作为 ΔH 设置的判据, D 的计算公式如所示:

$$D = 0.6M_1 + 0.3M_2 + 0.1M_3 \quad (58)$$

其中 M_1 、 M_2 、 M_3 分别代表当前 HELLO 发送间隔、前一个 HELLO 发送间隔以及前两个 HELLO 发送间隔的 M 值, 权值由大到小是因为当前时刻和上一时刻的 HELLO 分组发送间隔对结果影响较大, 因此取权重较大。

公式是为了在本地信息库有链路变化时, ΔH 的值可以在一个区间 $[\Delta H_L, \Delta H_H]$ 内变化。定义优化后的 HELLO 分组发送间隔用 H_{hello} 表示, 规定当 D 为 0 时, $H_{\text{hello}} = \Delta H_H$; 当 $D = 1$ 时, $H_{\text{hello}} = \Delta H$; 当 $D = 2$ 时, $H_{\text{hello}} = \Delta H_L$ 。HELLO 发送间隔与节点稳定度关系如图 31 所示。

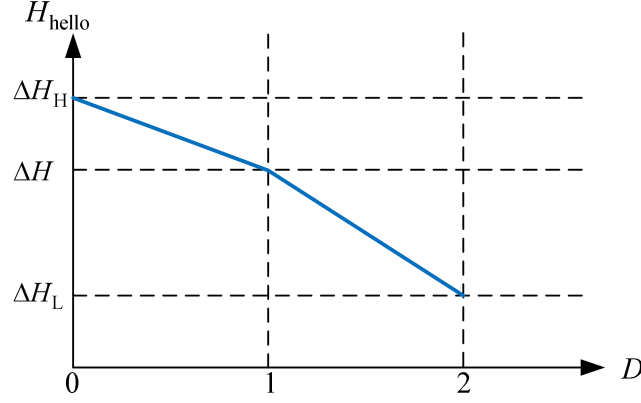


图31 HELLO间隔与节点稳定度的关系

综上，HELLO 分组发送间隔 H_{hello} 自适应调整公式如式所示：

$$H_{\text{hello}} = \begin{cases} \Delta H_H - (\Delta H_H - \Delta H)D & 0 \leq D < 1 \\ 2\Delta H + (\Delta H_L - \Delta H)D - \Delta H_L & 1 \leq D < 2 \\ \Delta H_L & D \geq 2 \end{cases} \quad (59)$$

原始 OLSR 协议中 ΔH 的默认值 2s，HELLO 控制包的保持时间 $Vtime_H$ 也为默认值，二者关系为：

$$Vtime_H = \lambda \Delta H \quad (60)$$

其中， λ 为 OLSR 协议中固定值，默认为 3s。

在优化完 HELLO 消息的发送间隔之后，其保持时间也会发生改变。下一个 HELLO 消息预发送时间为当前时间加上 H_{hello} 。当 H_{hello} 为区间下限 ΔH_L 的时候，HELLO 分组的有效保持时间会变久，导致网络中存在大量过期节点；当 H_{hello} 为区间上限 ΔH_H 时，HELLO 分组的保持时间会变短，来不及更新路由导致链路断裂。

为了不对自适应的 HELLO 分组发送间隔后的 HELLO 信息保持时间产生影响，采用如下公式，对 HELLO 包的保持时间优化：

$$Vtime_{\text{hello}} = \begin{cases} H_{\text{hello}} + (\lambda - 1)\Delta H_L & \Delta H_L \leq H_{\text{hello}} < \Delta H \\ H_{\text{hello}} + (\lambda - 1)\Delta H_H & \Delta H \leq H_{\text{hello}} < \Delta H_H \end{cases} \quad (61)$$

通过上述分析，根据节点链路信息集中元素变化情况来间接反应飞行自组网中无人机节点之间相对运动的快慢，并自适应地调整 HELLO 包的发送时间间隔，可以一定程度上降低网络中的控制消息开销。

2 TC 分组自适应优化

2.1 TC 分组发送间隔自适应优化

TC 分组中携带网络拓扑消息，在经过 MPR 节点的选择之后，由 MPR 节点向全网广播，并向网络中的节点声明自己会被那些节点选为 MPR 节点，选择自身成为 MPR 节点的节点集合就是 MPR Selector 集。MPR 节点集合中会记录到达该集合的全部链路信息，一旦 MPR 节点的 Selector 集合出现变动，MPR 节点会向全网广播 TC 分组以进行路由更新和维护。因此 TC 分组的消息发送与 MPR Selector 集的变化有关，若 MPR Selector 集在两次 TC 分组发送间隔之内的变化不显著，降低 TC 分组的发送间隔；若 MPR Selector 集在两次 TC 分组发送间隔之内的变化较为明显时，提高 TC 分组的发送间隔。

对 TC 分组的发送间隔进行优化与 HELLO 分组有不同之处，原始 OLSR 协议中 TC 分组的发送时间间隔为固定值 ΔT ，上一个 TC 分组的发送间隔为 $\Delta T'$ ，设定发送间隔最大值为 $\Delta T_{\max}$ ，最小值为 $\Delta T_{\min}$ ，则 $\Delta T \in [\Delta T_{\min}, \Delta T_{\max}]$ 。规定 MPR Selector 集中节点变化的数量记为 K ，一个 TC 分组发送间隔内 MPR Selector 集的平均变化的数量为 A_K ，计算公式如下：

$$A_K = K \frac{\Delta T}{\Delta T'} \quad (62)$$

根据 A_K 的值就能计算出 TC 分组的自适应发送间隔 ΔT_{tc} ，当 $A_K = 0$ 时，MPR Selector 集中节点相对稳定，此时可以提高 ΔT_{tc} 到最大值 $\Delta T_{\max}$ ；当 $A_K \neq 0$ 时，MPR Selector 集中的节点不稳定，此时可以降低 ΔT_{tc} 到最小值 $\Delta T_{\min}$ ；当定时器过期时，需要重新计算 ΔT_{tc} 。但因为标准 OLSR 协议中的 ΔT 默认值比 HELLO 包中 ΔH 的大，因此 MPR Selector 集中节点的变化频率相对较小，在优化 ΔT_{tc} 时可以小幅度调整。 ΔT_{tc} 的计算公式如下：

$$\Delta T_{tc} = \begin{cases} \max(\Delta T_{\max}, \Delta T \times 2^{\lfloor i/3 \rfloor}) & A_K = 0 \\ \Delta T_{\min} & A_K \neq 0 \end{cases} \quad (63)$$

其中， i 代表计数器，该计数器记录 MPR Selector 集的稳定状态期数，默认为 0。

当 $A_K = 0$ 时 i 加 1；当 $A_K \neq 0$ 时， i 值回归 0。上式表示，若三个发送间隔内 MPR Selector 集的节点都处于稳定状态即 $A_K = 0$ 时，则按照指数规律提高 ΔT_{tc} ，到达最大值 ΔT_{max} 位置；若 MPR Selector 集存在状态不稳定节点时即，不稳定状态即 $A_K \neq 0$ 时，将本次 ΔT_{tc} 设置为最小值 ΔT_{min} 。

同样地，原始OLSR协议来中的TC分组保持时间 $Vtime_T$ 也是一固定值，与TC分组发送间隔的关系表达式如下：

$$Vtime_T = \mu \Delta T \quad (64)$$

其中， μ 为一固定值，默认为 3s。当 TC 分组发送间隔自适应处理时，要对 TC 分组的保持时间 $Vtime_{tc}$ 同样自适应优化，公式如下：

$$Vtime_{tc} = \begin{cases} \Delta T_{tc} + (\mu - 1) \Delta T & \Delta T_{tc} = \Delta T_{min} \\ \Delta T_{tc} + (\mu - 1) \Delta T_{max} & \Delta T_{tc} = \Delta T \\ (\mu - 1) \Delta T_{tc} & \Delta T_{tc} = \Delta T_{max} \end{cases} \quad (65)$$

2.2 TC 分组的 TTL 优化

优化后的 TC 分组机制会根据网络拓扑变化的快慢来自适应调整 TC 分组的发送间隔。但节点在移动速度较快时，为了能够及时反应网络状况，会频繁发送 TC 分组到全网，虽然对比原始 OLSR 协议有所改善，但网络中仍存在大量数据包冗余，造成网络堵塞。与本地节点相距跳数较大的节点，在本地节点选择下一跳节点时，该节点的链路状态是否变动对本地节点影响越小，因此远方节点的拓扑更新间隔较长；但相对于本地节点跳数较少的节点，拓扑更新的周期会很短，因此可以针对 TC 分组限制其转发，在空间上进行优化。

TC 分组的记录了 MPR Selector 集中的链路信息、TC 消息的生存时间 (Time To Live, TTL)，TC 分组经过 MPR 节点转发后，TTL 字段的值就减 1，直至该字段值减到 0 就清除该分组，TTL 反应了 TC 分组可以被转发的最大范围。因此可以针对与本地节点距离不同的节点，设置不同的 TC 发送间隔。TC 分组在空间中传播的区域如图 32 所示，Q 越大表示距离中心节点越远，TC 发送间隔越大。

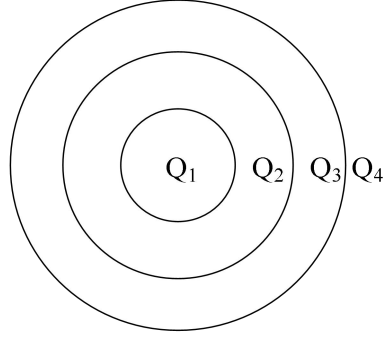


图32 TC分组在空间中传播的范围

本节结合模糊视觉链路传播的原理对 TC 分组的 TTL 进行限制转发。TTL 值越小，转发范围越小，TC 分组发送间隔越小；TTL 值越大，说明转发范围大 TTL，TC 分组发送间隔就越大。其传播规则为：节点在 $2^{i-1}\Delta T$ 后，设置 TC 分组的 TTL 值为 Q_i ，之后在 $(2k+1)\times 2^{i-1}\Delta T$ 后重新发送 TTL 值为 Q_i 的 TC 消息(i, k 均为整数且大于 1)，其中 $\{Q_i\}$ 是递增的。定义本节点到网络中位置最远的节点相距的跳数为 Q ，如果满足：

$$Q_{i-1} < Q < Q_i \quad (66)$$

则设置 TC 分组的 TTL 值为全网最大值 Q_i ，之后重新设置定时器，清空其历史值并令 $i=1$ ，开始循环发送 TC 控制包。

当 $\{Q_i\}$ 为 $\{2^i\}$ 时，性能优化最好，此时网络开销最低。此时 TC 分组的 TTL 值随时间变化的趋势如图 33 所示。

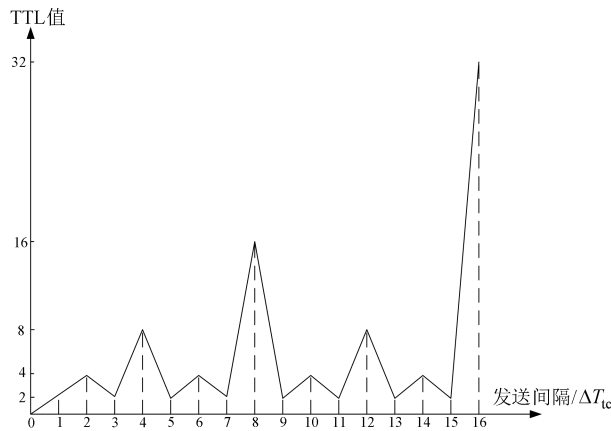


图33 TTL与TC分组发送间隔的关系

结合自适应 TC 分组发送时间间隔其分发规则为：

节点在 $2^{i-1}\Delta T_{tc}$ 后, 设置 TC 分组的 TTL 值为 2^i , 之后在 $(2k+1)\times 2^{i-1}\Delta T_{tc}$ 后再次发送 TTL 为 2^i 的 TC 消息(i, k 均为整数且大于 1)。定义此时网络中最远的节点距离本节点的跳数为 Q , 若满足:

$$2^{i-1} < Q < 2^i \quad (67)$$

则设置 TTL 的最大值为 2^i , 包含该 TTL 值的 TC 分组将被转发至全网, 然后重置定时器令 $i=1$ 循环发送 TC 分组消息。

因此结合上述模糊视觉链路传播的原理和 TC 分组自适应发送机制, 根据 MPR Selector 集的变化情况确定其下次 ΔT_{tc} 的值, 通过自适应的发送间隔 ΔT_{tc} 可以预测 TC 分组中的 TTL 应该设置的大小, 对于距离本地节点较近的节点提高发送频率, 对于距离本地节点较远的节点降低发送频率。这样就在时间和空间上都对 TC 分组实现了优化。

2) 控制消息自适应调整算法实现

1 HELLO 分组发送间隔优化算法实现

自适应 HELLO 分组发送间隔的算法运行流程如图 34 所示。仿真前设置 ΔH_L 的值为 1s, ΔH 为 2s, ΔH_H 为 3s。算法描述如下:

- (1) 将各个参数初始化, 进行链路感知获取本地节点一跳邻居信息;
- (2) 根据一跳邻居表中的信息判断本节点与邻居节点是否存在对称链路, 若不存在, 则更新链路状态, 重新进行链路感知;
- (3) 判断链路信息库中的元素改动是否满足 M 的触发条件, 若不满足, 则进行邻居侦听过程则更新链路状态, 重新进行链路感知;
- (4) 根据公式计算链路变化的数量 M , 根据公式由对三个发送间隔内的变化量进行加权处理, 得到节点稳定度 D , 根据公式由节点稳定度计算自适应 HELLO 分组发送间隔 H_{hello} , 根据公式由 H_{hello} 计算 HELLO 分组的保持时间 $Vtime_{hello}$ 。
- (5) 设置 HELLO 分组的定时器, 根据自适应发送间隔定期发送 HELLO 信息。

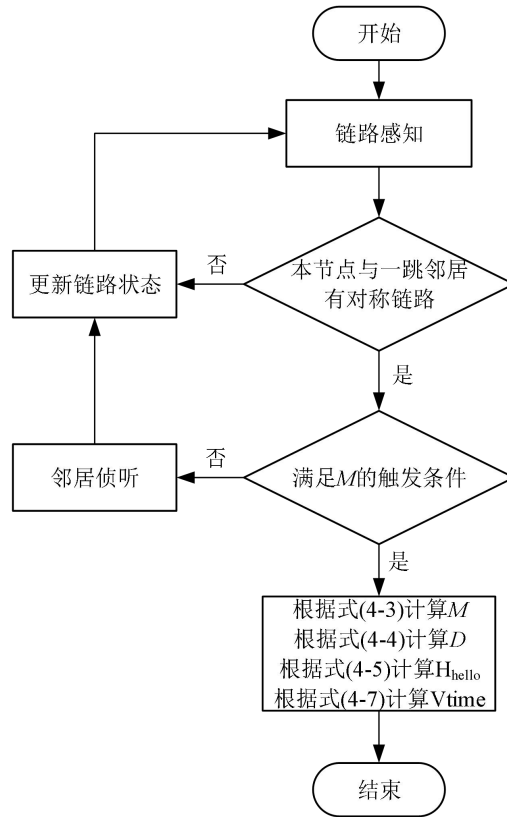


图34 HELLO分组发送间隔自适应流程

2 TC 分组的优化算法实现

TC 分组优化算法的流程如下：

- (1) 初始化 OLSR 协议后, 设置 TC 分组的定时器, 默认值为发送间隔为 ΔT 。
- (2) 判断定时器是否超时, 若超时则计算当前 ΔT 内的 MPR Selector 集中元素的变化情况, 根据公式计算变化的数量 A_K ；
- (3) 根据公式由 A_K 计算并自适应调整 TC 分组的发送间隔 ΔT_{tc} ；
- (4) 根据公式由 ΔT_{tc} 计算 TC 分组的自适应保持时间 $Vtime_{tc}$ ；
- (5) 对 TC 分组的 TTL 字段值进行处理, 如表 6 所示, 当 ΔT_{tc} 在 15 个以内时, 每发送一次 TC 分组, 累加一个 ΔT_{tc} ；当超过 15 个 ΔT_{tc} 后, 计数器清零；
- (6) 对 TC 分组定时器的过期时间为 ΔT_{tc} 进行处理, 通过 MPR 节点向全网广播洪泛 TC 分组, 继续判断定时器消息是否超时。

表6 TTL随第*i*个分组发送间隔变化关系

ΔT_{tc}	0	1	2	3	4	5	6	7
-----------------	---	---	---	---	---	---	---	---

TTL	32	2	4	2	8	2	4	2
ΔT_{tc}	8	9	10	11	12	13	14	15
TTL	16	2	4	2	8	2	4	2

综上所述，TC 分组自适应发送机制的算法流程如图 35 所示。

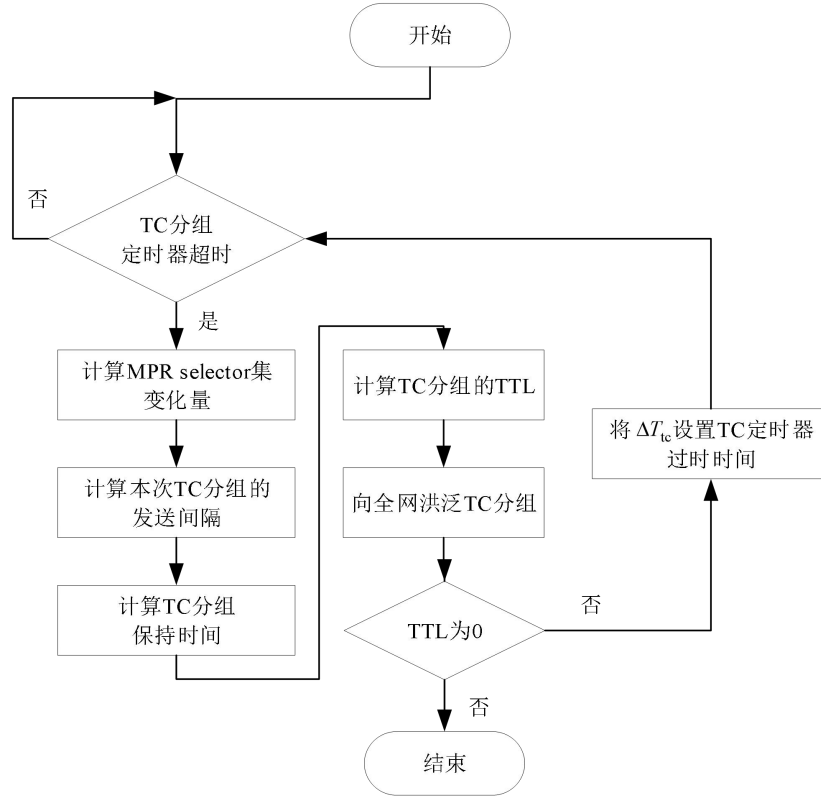
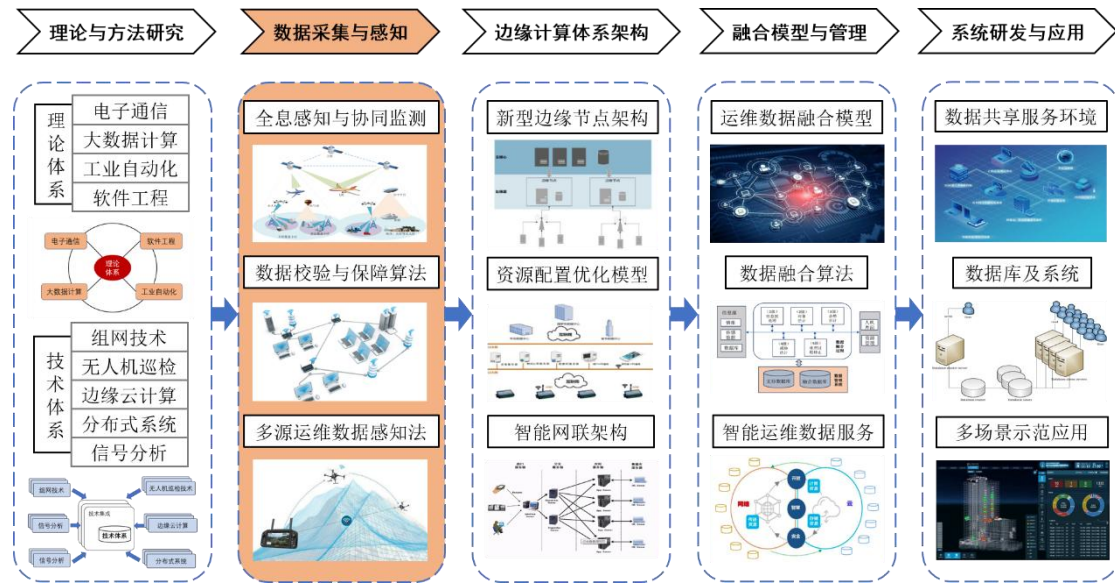


图35 TC分组自适应优化流程

在基于链路感知和链路持久性的路由策略改进基础上，将本节所提出的基于拓扑变化的控制消息自适应算法添加到 LK-OLSR 协议中，将二次改进后的协议命名为 TC-OLSR（Topology Changing Based OLSR）路由协议。

4.2 技术路线

本课题采用系统开发、试验测试与原型验证思路开展研究。首先构建空天地一体化区域集群协同监测技术，开发传感器组网协同感知技术，开发多智能体无人机群巡检最优规划与任务分配算法，实现多源数据的全息采集与感知。整体研究技术路线如下图橙色区域所示。



5 基础研究与研发进展

5.1 基础研究

课题负责人及其所在课题组在未知环境探测等相关领域如定位导航、计算机视觉分析领域进行了多年的基础研究和应用开发, 获得了丰富的研究经验并积累一定的研究成果。同时, 在自组织网络和无线信号处理方面近几年也已经做了很多研究。这些工作基础为本项目的顺利进行提供了有力保障。与本项目有关的工作基础可以归结为以下几个方面:

(1) 多层次固定翼无人机群立体组网技术: 为降低无人机之间的互干扰, 根据无人机工作区域进行立体分簇, 同时为均衡无人机的能量消耗、避免热点无人机被过度消耗, 提出了相关动态簇首选举方法。针对不同业务对服务质量的需求, 提出了业务驱动的无人机群负载均衡立体路由算法。并根据不同的服务质量指标设置权重参数, 通过非线性加权实现立体路由选择, 在保证不同业务的不同服务质量要求的同时, 均衡网络中的业务负载, 降低网络拥塞概率。(国家自然科学基金-国际合作项目: 4181101180)

(2) 自由空间最短路径寻路技术: 面对复杂未知的室内环境, 任意的两个节点所处的自由空间中可能会有障碍物存在。课题组采用启发式搜索算法, 在状态空间中对每一个搜索的位置进行评估, 得到最好的位置, 再从这个位置进行搜索直到目标。这样, 通过对障碍物进行规避, 实时规划出一条合理、较优的路径,

并能以最小的代价沿此路径移动至目标位置。节省大量路径搜索，提高了效率。

（国家自然科学基金项目：61571162）

5.2 研究团队

研究团队成员来自于哈尔滨工业大学通信技术研究所，现有 1 个国家级重点学科，1 个一级学科博士点，2 个二级学科博士点。主要从事通信与信息系统学科及相关领域的研究工作。目前开展了定位导航、数据通信、移动通信、卫星通信、程控交换等方面的教学、科研、开发和高级人才的培养工作。截止至 2020 年初，哈尔滨工业大学通信技术研究所所有教师和专职研究人员 43 人，其中教授 16 人，副教授 22 人，讲师 4 人，国内外客座教授（研究员）6 名，有 15 人在省一级以上学术机构中任职，是一支结构合理、具有奉献和合作精神的高素质教学科研队伍。近年来，与美国佐治亚理工学院、德国慕尼黑工业大学、加拿大多伦多大学、加拿大麦吉尔大学、新加坡南洋理工大学等国外著名高校建立了长期友好合作关系，互派学者进行交流。哈尔滨工业大学通信技术研究所拥有实验室面积 5000 多平方米，万元以上设备 200 余台，固定资产超过 3000 余万元，是航天工业总公司专用通信系统重点实验室。近年来，共获国家级科技进步二等奖 1 项、三等奖 3 项，委、部级科技进步一等奖 5 项、二等奖 9 项、三等奖 8 项，正式出版统编教材、科技著作 15 本，在学术刊物及学术会议上发表论文 800 余篇，其中 600 余篇被 SCI 和 EI 检索。现已培养博士 100 余名。哈尔滨工业大学通信技术研究所拥有原航天部通信技术重点实验室，目前建立宽带无线通信与网络黑龙江省重点实验室，2010 年又被教育部批准为专用通信系统教育部工程中心。985 二期建设得到了国家重点投资建设，通信学科获得 1000 万元的实验条件建设经费。

5.3 研究进展

根据本课题研究内容，提出四旋翼无人机硬件初步设计方案，如图 36 所示。

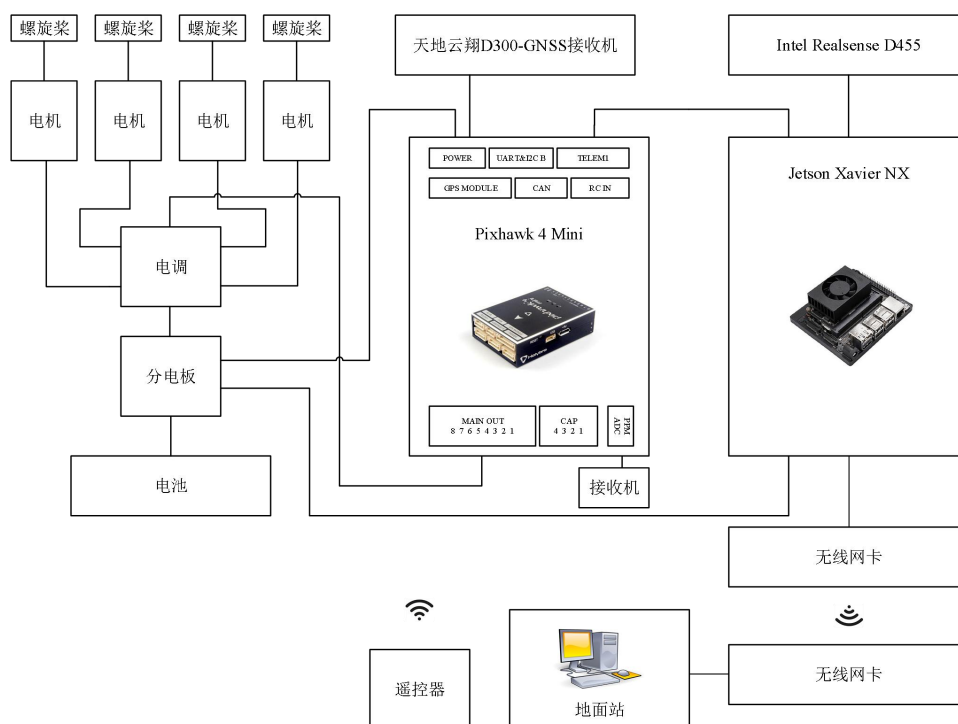


图 36 无人机硬件结构

无人机硬件设备主要包括 Pixhawk 4 Mini 飞行控制器、Jetson Xavier NX 开发板、Intel Realsense D455 相机、天地云翔 D300-GNSS 接收机、RT3070 无线网卡、遥控器、接收机、螺旋桨、电机、电调、分电板、电池和机架。GNSS 接收机 COM 接口与飞控 GPS MODULE 接口连接，向飞控发送 RTK 定位数据。遥控器接收机 PPM 接口与飞控 PPM 接口连接，用于接收遥控信号，实现无人机的解锁、上锁和飞行模式切换等操作。电调板与飞控 MAIN OUT 接口连接，通过接收飞控发送的 PWM 信号来将输入电源转化为不同的电压，并输出到电机，从而达到使电机产生不同转速的目的。飞控 TELEMI 接口通过 USB 转 TTL 串口模块与开发板 USB 接口连接，用于开发板获取无人机状态信息，并向飞控发送控制信息，飞控与开发板之间通过 MAVLink 协议进行通信。开发板通过 USB 分别与相机和无线网卡进行连接。无人机与地面站之间通过 WiFi 网络进行信息传输，无人机之间通过无线网卡进行 Ad-Hoc 组网进行数据传输。目前实验室所具备的两种型号的无人机如图 37 所示。



图 37 实验室所具备的无人机

无人机选用的机架是对角线轴距为 860mm 的全碳纤维机架，并配备了 21 英寸碳纤维折叠桨、F60 PRO IV KV1750 电机、BLHeli-S Dshot600 电子调速器、GS-4000mAh-25C-3S1P 电池、富斯 i6S 遥控器和富斯 iA6B 接收机。

无人机选用的飞行控制器是 Pixhawk 4 Mini。Pixhawk 是目前十分流行且非常成熟的一款开源无人机自驾仪，其来源于瑞士苏黎世联邦理工大学计算机视觉与几何实验室、自主系统实验室和自动控制实验室联合研发的软硬件开源项目。目前，Pixhawk 自驾仪的使用人数较多，具有较高的集成度，代码的各个模块之间的结构化清晰，资料齐全，有利于进行开发。

Pixhawk 自驾仪由 FMU 和 IO 两个部分组成。FMU 负责传感器数据的采集处理、导航算法、控制算法等；IO 负责遥控器信号输入和电机控制信号 PWM 的输出。Pixhawk 4 Mini 是 Pixhawk 系列飞行控制器的其中一个版本，基于 FMUv5 架构进行设计，采用了先进的处理器和传感器技术，以及 NuttX 实时操作系统，在无人机的飞行控制上性能优异，具有较高的灵活性和可靠性。微控制器配备了 2MB 的闪存和 512KB 的 RAM，有利于实现较为复杂的算法和模型。

IMU 具有较低的噪声，能够使无人机获得可靠的飞行性能。Pixhawk 4 Mini 还具有非常丰富的硬件接口，它拥有 8 个 PWM 伺服输出接口、4 个专用 PWM/Capture 输出接口、CPPM 专用 R/C 输入接口、Spektrum/DSM 专用 R/C 输入接口、3 个通用串行接口、2 个 I2C 接口、3 条 SPI 总线、1 条用于 CAN 电调的 CAN 总线、电压/电流模拟输入接口和 1 个额外的模拟输入接口。Pixhawk 4 Mini 的具体技术参数如表 7 所示。

表 7 Pixhawk 4 Mini 技术参数

参数名称	参数内容
FMU 处理器	32 位 Arm Cortex-M7, 216MHz, 2MB 闪存, 512KB RAM
加速度计/陀螺仪	ICM-20689、BMI055
磁力计	IST8310
气压计	MS5611
尺寸	38×55×15.5mm
重量	37.2g

在 Pixhawk 硬件平台上,运行的是 PX4 飞行控制程序。PX4 是一款著名的开源飞控程序,支持常见的多种无人机机型,如四旋翼、六旋翼、八旋翼、直升机和固定翼等。PX4 飞控程序结构复杂、性能优异、配置灵活,其主要由两个主要部分组成:第一个部分是飞行控制栈,该部分主要包括状态估计系统和飞行控制系统;第二个部分是中间层,该部分是一个通用的机器人应用层,可支持任意类型的自主机器人,主要负责机器人内部与外部的通信和硬件整合。飞行控制栈主要包括多旋翼控制器和位置、姿态估计器,其工作过程如图 38 所示。

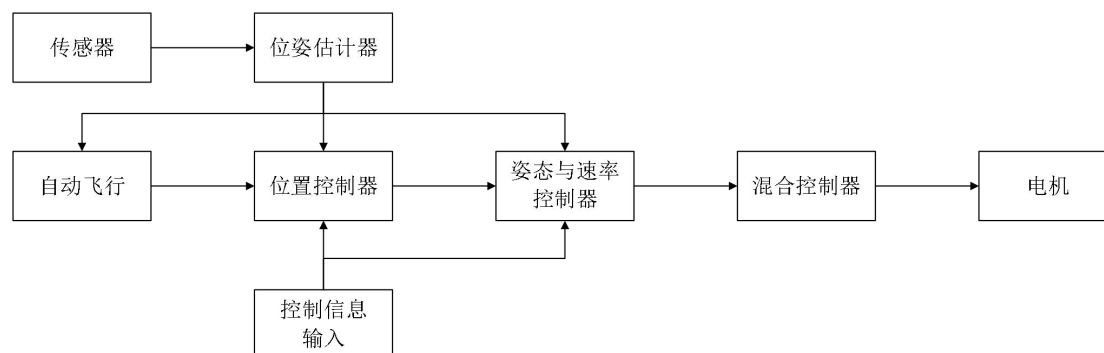


图 38 飞行控制栈工作过程

其中,传感器包括加速度计、陀螺仪、磁力计、气压计等,对数据进行初步过滤与处理。位姿估计器接收一个或结合多个传感器的输入,采用扩展卡尔曼滤波对其作进一步的处理,计算出无人机的位姿状态。自动飞行模块能够根据存储的飞行任务信息进行自动飞行功能,将位置期望点传递给飞行控制模块,并以指定高度和航向飞行。控制器接收期望点和位姿估计状态,通过调整飞行过程中相关变量的值,使其与期望值相匹配。例如,位置控制器外环 PID 将当前期望位置输出为期望速度,内环 PID 根据期望速度计算出期望姿态,输出给姿态控制器;姿态控制器外环 PID 将当前期望状态输出为期望角速度,内环 PID 根据期望角速度计算出期望控制量,输出给混合控制器。混合控制器接收期望控制量,并将

其转换为多路控制 PWM 信号，控制无人机进行飞行。

中间层主要由嵌入式传感器设备驱动程序、机载电脑和地面站通信程序和 uORB 消息订阅-发布总线组成。嵌入式传感器设备驱动程序主要有 GPS 驱动、遥控器输入驱动、数传驱动、IMU 驱动和相机驱动等，在编译程序时可以根据使用需要进行选择。机载电脑和地面站通信程序使用的是 MAVLink 通信协议，它是一种轻量级、简洁、跨平台的通信协议，其数据包格式如图 39 所示。

STX	LEN	INC FLAGS	CMP FLAGS	SEQ	SYS ID	COMP ID	MSG ID (3 bytes)	PAYLOAD (0-255 bytes)	CHECKSUM (2 bytes)	SIGNATURE (13 bytes)
-----	-----	--------------	--------------	-----	-----------	------------	---------------------	--------------------------	-----------------------	-------------------------

图 39 MAVLink 协议数据包格式

用户可以在 PAYLOAD 部分获取无人机当前位置、姿态信息，或添加用户期望的位置、速度等信息。MAVLink 协议数据包中各个部分的名称、内容、含义如表 8 所示。

表 8 MAVLink 协议数据包各个部分的名称、内容、含义

名称	内容	含义
STX	起始标记	用于指示新数据包的起始位置
LEN	载荷长度	用于声明数据包中载荷的长度
INC FLAGS	不兼容标志	如果不兼容 MAVLink 协议，则丢弃数据包
CMP FLAGS	兼容标志	如果兼容 MAVLink 协议，则对数据包进行处理
SEQ	序列号码	数据包的序列号码，用于检测数据包的丢失情况
SYS ID	系统 ID	发送数据的无人机的 ID，用于区分不同的无人机
COMP ID	组件 ID	发送数据的组件的 ID，用于区分不同的组件
MSG ID	消息 ID	载荷中数据类型的 ID，用于数据的解码
PAYLOAD	载荷信息	发送的数据信息
CHECKSUM	校验码	用于数据信息的 CRC 校验
SIGNATURE	数据签名	用于检测发送的数据信息是否被篡改，为可选项

PX4 飞控程序中各个模块之间的内部消息传递采用的是 uORB 异步消息机制。uORB 在数据发布与接收过程中，并不保证发送者的所有数据都可以被接收者接收到，只保证接收者在想要接收时能够受到最新的数据。每个模块之间并不需要了解数据的来源或数据流向，只需关心当前数据内容，提高了各个模块之间的独立性。

无人机选用的机载计算机是 NVIDIA Jetson Xavier NX，它是一款用于嵌入式和边缘系统中高性能计算的开发板，具有较小的尺寸，在低功耗的条件下可以提供高达 21TOPS 的性能，并且可以同时运行多个神经网络并处理来自多个传感器

的数据，从而满足系统的需求。其支持官方提供的完整的 NVIDIA 软件堆栈，包括经过加速的软件开发工具包和适用于应用程序开发和优化的最新 NVIDIA 工具，可以大大提高深度学习、计算机视觉和视觉多媒体等领域的计算效率。Jetson Xavier NX 开发板能够满足无人机进行图像处理、自主导航、编队控制和自组网数据传输算法所需要的性能。其提供了 1 个 Micro SD 卡插槽、1 个 5V 电源接口、1 个 HDMI 输出接口、1 个 DisplayPort 接口、1 个千兆以太网接口、1 个 USB 2.0 Micro-B 接口、4 个 USB 3.1 接口、2 个 CSI 摄像头接口和 1 个 40PIN I/O 接口。Jetson Xavier NX 开发者套件的主要技术参数如表 9 所示。

表 9 Jetson Xavier NX 开发者套件技术参数

参数名称	参数内容
GPU	NVIDIA Volta 架构，搭载 384 个 NVIDIA CUDA 核心和 48 个 Tensor Core
CPU	6 核 NVIDIA Carmel ARM v8.2 64 位 CPU
DLA	2× NVDLA
视觉加速器	2× PVA
显存	8GB 128 位 LPDDR4x 59.7GB/s
尺寸	103×90.5×34mm

近年来，机器人行业发展迅速，相关的代码库越来越庞大。为了提高机器人的研发效率，便于实现不同研发人员之间相互的资源共享与协作，一款开源的机器人操作系统（Robot Operating System，ROS）应运而生。由于 ROS 具有模块化和复用性的特点，在 ROS 开源社区中移植、集成了大量开源项目中的代码，在已有开源代码的基础上进行研究可以有效地提高无人机开发过程中的效率，因此本课题选择基于 ROS 系统进行部分功能模块的开发。

ROS 是一个适用于机器人的开源元操作系统，它提供了操作系统应有的服务，包括硬件抽象、底层设备控制、常用函数的实现、进程间消息传递和包管理等功能，同时它也提供了用于获取、编译、编写和跨计算机运行代码所需的工具和库函数。ROS 的主要设计目的是为了提高在机器人技术研发过程中的代码复用性，它能够将实现不同功能的模块进行独立的封装。在使用功能包时不需要了解模块内部的具体实现过程，只需要了解功能包之间的接口就能够实现功能包的复用，实现模块之间的点对点松耦合连接。ROS 是一种分布式的机器人操作系统，其中的进程可以在不同的主机上运行，不同主机之间能够实现数据的共享，减轻了

主机计算的压力，提高了计算的效率。

除了便于实现代码复用的优点以外，ROS 系统还提供了通信机制和开发工具。ROS 的通信机制主要包括话题通信、服务通信和参数服务器，它们的实现模式分别为发布订阅模式、请求相应模式和参数共享模式，能够满足不同场景下进程之间通信的需求。ROS 的开发工具主要包括命令行工具、Rviz 和 rqt 等。命令行工具提供了一系列对节点、话题、服务、消息、参数等进行查看、修改、管理等操作的命令。Rviz 是 ROS 的可视化工具，能够将消息数据进行可视化表达，例如，显示激光雷达获取的点云图、双目相机的图像和无人机的飞行轨迹等信息。rqt 工具箱也是 ROS 系统中的一个可视化工具，主要用于显示计算图、绘制曲线图和显示日志信息等。

ROS 架构可以分为三层，分别是文件系统层、计算图层和开源社区层。ROS 文件系统层是指在硬盘上 ROS 源代码的组织形式，其结构如图 40 所示。

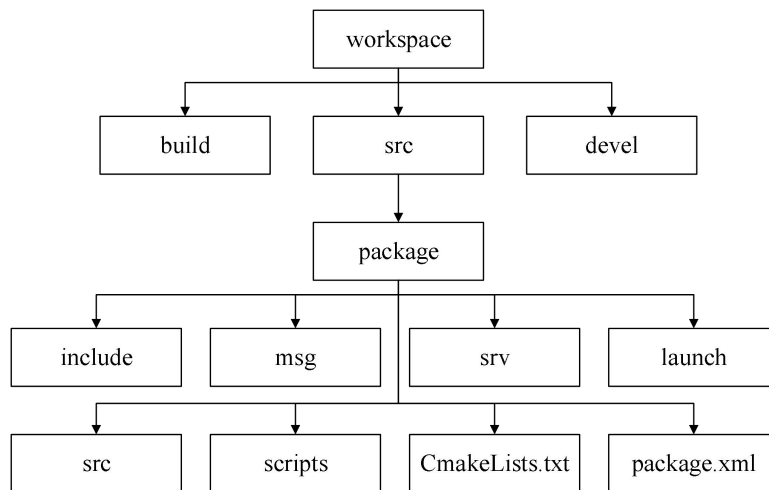


图 40 ROS 系统中的文件结构

其中，workspace 为工作空间，build 为编译空间，devel 为开发空间，src 为源代码存储文件夹。src 文件夹内一般包括多个 package，即多个功能包。每个 package 文件夹内包含多个节点、库和配置文件，include 存储头文件，msg 存储消息通信格式文件，srv 存储服务通信格式文件，launch 存储节点启动与参数设置文件，src 存储 C++源代码文件，scripts 存储 python 源代码文件，CmakeLists.txt 为功能包编译配置文件，package.xml 为功能包属性文件。

ROS 计算图层用于可视化 ROS 进程中所包含的数据处理过程，包括节点、消息、话题、服务和节点管理等概念。

(1)节点：节点是执行计算任务的进程，一个机器人系统通常包含许多节点，每个节点完成不同的计算任务。

(2)消息：节点之间通过传递消息进行相互通信，每一条消息都是一种数据结构，支持整型、浮点型、布尔型等标准数据类型，以及任意嵌套的结构体和数组。

(3)话题：消息通过发布/订阅的方式进行传输，节点可以在特定话题上进行消息的发布和订阅。一个话题可以具有多个发布者和订阅者，一个节点也可以发布和订阅多个主题。

(4)服务：服务是基于客户端/服务器模型的通信方式，由一对数据类型进行定义，分别用于请求和应答。客户端需要向节点发送请求服务消息，并等待节点的回复。

(5)节点管理器：节点管理器用于管理计算图层的其他部分，包括节点之间的名称注册、话题查找和建立连接等，还提供了参数服务器，便于参数的存储和共享。

一个计算图的例子如图 41 所示，图中展示出了 ROS 系统中运行的两个节点之间的数据交互过程。



图 41 ROS 系统中的计算图

ROS 开源社区层是 ROS 网络上进行代码发布的一种表现形式。发行版是指可以独立安装、带有版本号的一系列综合功能包。软件库是一个代码共享系统，不同的机构、组织和个人都可以在这里发布和分享自己开发的机器人软件库程序。ROS 维基用于记录系统信息的记录，用户可以贡献自己的文件、提供更正和编写教程等。

本课题设计的无人机中 Jetson Xavier NX 开发板内运行的是 Ubuntu 18.04 系统，并在 Ubuntu 系统中安装了 ROS 的 Melodic 版本。除此之外，还安装了一些代码编译过程中所需要的库，并编译运行了 Realsense-ros、MAVROS 和 Prometheus 等功能包。

Realsense-ros 功能包用于在 ROS 系统中使用 Intel Realsense D455 相机，它能

够将相机获取到的图像、深度信息发布在相应的话题上，并且还能够对分辨率、发布频率等参数进行修改，供其他功能包对数据进行读取与使用。Realsense-ros 功能包发布的话题名称、消息类型和内容如表 10 所示。

表 10 Realsense-ros 功能包发布的话题名称、消息类型和内容

话题名称	消息类型	内容
/camera/infra1/image_rect_raw	sensor_msgs/Image	左目图像
/camera/infra2/image_rect_raw	sensor_msgs/Image	右目图像
/camera/depth/image_rect_raw	sensor_msgs/Image	深度图像
/camera/imu	sensor_msgs/Imu	IMU 数据

MAVROS 功能包用于 ROS 系统与支持 MAVLink 协议的设备之间进行通信，包括机载计算机与飞控之间的通信，以及飞控与地面站之间的通信，该功能包的实现过程如图 42 所示。

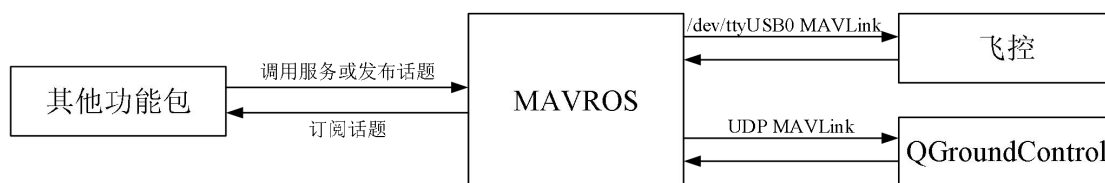


图 42 MAVROS 功能包的实现过程

其中，机载计算机其他功能包中的程序可以通过请求服务或发布话题的方式，向 MAVROS 节点发送状态控制信息或期望值数据，机载计算机与飞控之间通过 /dev/ttyUSB0 串口进行连接，MAVROS 节点将数据转换为 MAVLink 数据格式，发送到飞控，其他功能包中的程序同样可以通过订阅话题来获取无人机当前状态信息；MAVROS 接收到飞控发送的 MAVLink 消息后，通过机载计算机的无线网卡以 UDP 的形式将 MAVLink 消息发送给地面站软件 QGroundControl。MAVROS 功能包订阅和发布的话题名称、消息类型和内容如表 11 所示。

表 11 MAVROS 功能包发布的话题名称、消息类型和内容

话题名称	消息类型	内容
/mavros/set_mode	mavros_msgs/SetMode	设置飞行模式
/mavros/local_position/pose	geometry_msgs/PoseStamped	当前位置
/mavros/local_position/velocity	geometry_msgs/TwistStamped	当前线速度
/mavros/setpoint_position/local	geometry_msgs/PoseStamped	期望位置
/mavros/setpoint_raw/local	mavros_msgs/PositionTarget	期望速度

Prometheus 功能包是一个基于 PX4 开源飞控固件和 ROS 机器人操作系统的开源自主无人机软件系统平台，集成了 Gazebo 仿真、飞行控制、SLAM、目标

检测和路径规划等功能模块，为无人机的智能与自主飞行提供全套解决方案，开发者可以在该功能包的基础上进行二次开发，简化开发过程。

无人机选用的相机是 Intel Realsense D455，其具有 2 个红外相机、1 个红外线投射器和 1 个 RGB 相机，内部集成了 IMU。通过该相机能够获取红外图像、RGB 图像和利用双目视觉原理得到的深度图像，以及 IMU 的测量数据，以完成无人机的图像采集、视觉惯性里程计和位姿估计等功能。Intel Realsense D455 的主要技术参数如表 12 所示。

表 12 Intel Realsense D455 技术参数

参数名称	参数内容
使用环境	室内、室外
深度视场	87°×58°
深度分辨率	1280×720
深度帧率	90 帧/秒
RGB 视场	90°×65°
RGB 分辨率	1280×800
RGB 帧率	30 帧/秒
接口	USB-C 3.1 Gen1
尺寸	124×26×29mm

无人机选用的 RTK 定位设备是天地云翔 D300-GNSS 接收机，它是一款通过接收地面基准站差分数据链，实现实时载波相位差分定位的设备。接收机内置全星座全频高精度定位定向模组，支持 BDS、GPS、GLONASS 和 GALILEO 等卫星导航系统。采用快速载波相位模糊度解算技术和多径抑制算法，能够快速精确的解算出两个天线相对位置信息和两个天线相位中心连线与真北之间的夹角。通过该接收机和千寻 FindCM 服务可以为无人机提供动态厘米级，静态毫米级的高精度位置信息。接收机配备了 GNSS1、GNSS2、Radio/4G、COM1 和 COM2 接口。接收机使用方式如图 43 所示。

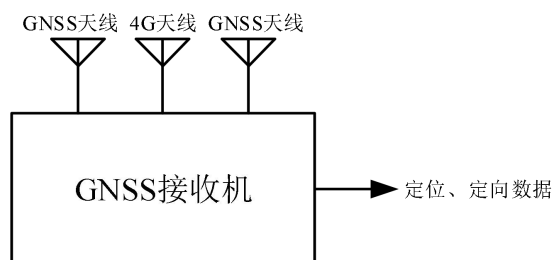


图 43 GNSS 接收机使用方式

其中，GNSS1 和 GNSS2 接口与 GNSS 天线连接，用于接收卫星导航信号，Radio/4G 接口与 4G 天线连接，用于接收基准站差分数据，解算获得的 RTK 定位和定向数据通过 COM1 和 COM2 接口传输。

接收机的主要技术参数如表 13 所示。

表 13 天地云翔 D300-GNSS 接收机技术参数

参数名称	参数内容
单点定位（RMS）	平面 1.5m，高程 2.5m
DGPS（RMS）	平面 0.4m，高程 0.8m
RTK（RMS）	平面 10mm+1ppm，高程 15mm+1ppm
定向精度（RMS）	0.2 度/1m 基线
时间精度（RMS）	20ns
速度精度（RMS）	0.03m/s
首次定位时间	冷启动：40s
数据更新率	20Hz
串口	UART、CAN/RS485
无线	数传电台、蓝牙 5.0、4G 全网通
尺寸	78×48×18mm
重量	116g

地面站运行的软件是 QGroundControl，通过该软件可以观察无人机状态信息，同时可以进行飞行控制与参数设置，其界面如图 44 所示。

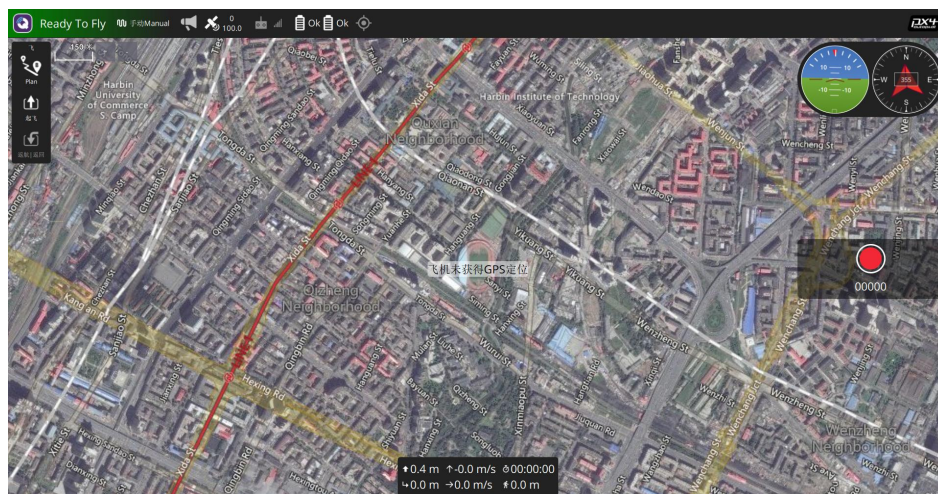


图 44 QGroundControl 软件界面

其中，主界面显示的是无人机当前所处的位置和飞行任务信息。界面上方显示了无人机当前的飞行模式、卫星数量和 HDOP、RC 信号强度和电池电量信息。界面左上角用于无人机的飞行控制，可以进行起飞、降落、添加航点和设置地理围栏等操作。界面右上角显示无人机的姿态和朝向信息。界面下方显示无人机的飞

行高度、飞行时间、飞行速度等信息。在车辆设置选项内，可以完成飞控固件烧写、机架种类选择、传感器校准、遥控器校准与飞行模式设置、电机设置和 PID 参数设置等功能。在分析工具选项内，可以观察到实时 MAVLink 消息，并能通过 MAVLink 控制台对相关参数进行控制。

6 考核指标与工作计划

6.1 考核指标

考核指标		考核方式（方法）及评价手段
指标名称	完成时指标值/状态	
1 面向设施群的空天地一体化协同快速监测技术	建立空天地协同检测技术	"a) 发表学术论文 2 篇； b) 申请或授权发明专利 2 项； c) 技术报告 1 项，专家评审证明； d) 培养 1 名博士研究生、1 名硕士研究生"
2 多无人机智能体编队协同智能巡检技术	a) 3 台以上无人机群协同编队的智能巡检（建筑玻璃幕墙、城市桥梁隧道外观状态巡检）； b) 低视场环境下无人机视角导航与巡检技术	
3 超低延时的“云-边-端”分布式多源运维数据智能感知技术	a) 分布式自组网数据传输跳数能力不低于 3 跳； b) 感知智能体群自主组网技术	

6.2 工作计划

(1) 2022 年 11 月—2023 年 04 月

- ✓ 任务：开展基于空天地一体化的运维数据全息智能采集理论研究；
- ✓ 考核指标：无人机硬件初步设计方案；
- ✓ 成果形式：阶段性研究报告 1 份

(2) 2023 年 05 月—2023 年 10 月

- ✓ 任务：研究资源配置优化模型和自适应网络动态调度理论；
- ✓ 考核指标：智能巡检无人机原型 1 架；
- ✓ 成果形式：投稿或发表论文 1 篇，申请或授权发明专利 1 项；阶段性研究报告 1 份。

(3) 2023 年 11 月—2024 年 04 月

- ✓ 任务：提出基于多智能体的无人巡检运维的理论方法；
- ✓ 成果形式：阶段性研究报告 1 份。

(4) 2024 年 05 月—2024 年 10 月

- ✓ 任务：完成基于多智能体的无人巡检运维技术研发；研究基于网联资源调度算法的智能自组网技术理论；
- ✓ 考核指标：多无人机智能体编队协同智能巡检技术，实现无人机 3 跳以上数据传输和交互控制；
- ✓ 成果形式：投稿或发表论文 1 篇，申请或发明专利 1 项；阶段性研究报告 1 份。

(5) 2024 年 11 月—2025 年 04 月

- ✓ 任务：研究分布式无线自组网路由技术；
- ✓ 成果形式：阶段性研究报告 1 份。

(6) 2025 年 04 月—2025 年 10 月

- ✓ 任务：实现多场景智能运维服务的示范应用。
- ✓ 成果形式：阶段性研究报告 1 份。