基于仿生动态视觉的低空智能感知关键技术

Key Technologies for Low-altitude Intelligent Perception with Bio-inspired Dynamic Vision

(申请清华大学工学博士学位论文)

培 养 单 位： 深圳国际研究生院

学　　科： 数据科学和信息技术

研 究 生： 王 灏 洋

指 导 教 师： 陈 鑫 磊　副教授

二〇二六年七月

Key Technologies for Low-altitude Intelligent Perception with Bio-inspired Dynamic Vision

Dissertation submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Data Science and Information Technology

by

Haoyang Wang

Dissertation Supervisor: Xinlei Chen, Associate Professor

July, 2026

摘 要

随着低空经济的发展，无人机应用场景由开阔空域向结构复杂的城市环境延伸，高精度、低延迟位姿估计、精准环境感知与智能多机协同，已成为确保其安全运行的基石。然而，现有定位与导航范式正面临严重的感知瓶颈：在单机个体维度，传统帧相机的离散采样机制在应对无人机高频飞控需求时，易产生运动模糊与感知滞后，表现出适应性不足的局限性；在多机系统维度，现有方案依赖外源信号，且局限于单体感知闭环，缺乏物理层面的信息共生，导致系统弱耦合。

为突破个体适应性不足与系统弱耦合带来的瓶颈，本研究结合具备微秒级时间分辨率的仿生动态视觉（事件相机），重塑无人机底层感知与交互逻辑，将其转化为具备“具身感知（Enacted）”与“交互协同（Engaged）”能力的智能体。然而，全面赋能无人机，使其成为低空智能体，仍面临三大挑战：*(i)* 构建外源具身感知方法时，事件流语义缺失，导致无人机相关特征难提取；*(ii)* 塑造内源具身感知方法时，事件流的时空稀疏性使其与环境特征难对齐；*(iii)* 迈向群体交互协同时，多影响因素导致多机系统间高效交互难建立。针对上述挑战，本文主要研究内容如下：

(1) 外源信息驱动的具身感知（Enacted）方法：针对起降等高风险阶段，本文提出融合事件相机与雷达的地面主动感知方法（mmE-Loc）。该方法深度挖掘跨模态数据的时空互补性，将地面设施转化为智能体的外部感官。实验表明，该方法定位误差为 0.083 米，延迟约 5.15 毫秒，并在商业机场完成约 10 小时实地验证。

(2) 内源信息驱动的具身感知（Enacted）方法：针对无人机自主生存需求，本文提出事件视觉增强的机载位置服务（EV-Pose），通过事件流与先验环境模型高精度对齐进行位姿估计，为无人机建立内源感知-动作环路。实验表明，方法旋转和平移精度均比基线方法高出 40%，延迟约 10.08 毫秒，能够支持无人机精确着陆。

(3) 面向多智能体的交互协同（Engaged）方法：为打破单体感知的物理局限，本文提出基于动态基础设施的异构协同方法（TransformLoc）。该方法利用高级无人机作为动态移动节点，为基础无人机提供间歇性精准校正。实验表明，该方法使得多基础无人机的感知精度提升 68%，自主导航成功率提高 60%。

总结而言，本研究构建了从个体具身感知到多机交互协同的“感知-交互”框架，破解了无人机面临的感知适应性不足与系统弱耦合等瓶颈，为低空经济提供了高响应速度的智能感知底座，也为无人机在复杂环境的自进化能力奠定了基础。

关键词：智能感知；事件相机；无人机；多模态融合；位姿估计

ABSTRACT

With the rapid development of the low-altitude economy, drone application scenarios are extending from open airspaces into structurally complex urban environments, making high-precision, low-latency pose estimation, accurate environmental perception, and intelligent multi-drone collaboration the foundational pillars for ensuring safe operation. However, existing localization and navigation paradigms are confronting severe perception bottlenecks: At the individual drone level, the discrete sampling mechanism of traditional frame-based cameras struggles to meet the high-frequency demands of flight control, exhibiting the inherent limitation of inadequate dynamic adaptability due to their susceptibility to motion blur and perception latency. At the multi-drone system level, current solutions rely heavily on exogenous signals and remain confined to isolated perception closed-loops. The lack of information symbiosis at the physical layer ultimately leads to a weakly coupled system.

To overcome the dual bottlenecks of inadequate individual adaptability and weak system coupling, this research integrates bio-inspired dynamic vision (event cameras) with microsecond-level temporal resolution to reshape the foundational perception and interaction logic of drones. This paradigm shift transforms drones into intelligent agents equipped with "Enacted" perception and "Engaged" collaborative capabilities. However, comprehensively empowering drones to become low-altitude intelligent agents still faces three major challenges: *(i)* In constructing external enacted perception, the absence of semantics in event streams makes the extraction of drone-relevant features highly difficult; *(ii)* In shaping endogenous enacted perception, the spatiotemporal sparsity of event streams complicates their alignment with environmental features; *(iii)* In advancing towards multi-drone interactive collaboration, multiple complex factors hinder the establishment of efficient interaction across multi-drone systems.

To address these challenges, the main research contents of this thesis are as follows:

(1) External Information-based Enacted Perception Method: Targeting high-risk flight phases such as takeoff and landing, this thesis proposes a ground-based active perception method (mmE-Loc) that fuses event cameras and radar. By thoroughly leveraging the spatiotemporal complementarity of cross-modal data, this method transforms ground infrastructure into the external sensory organ of the drone. Experimental results

demonstrate that the method achieves a localization error of 0.083 m and a latency of approximately 5.15 microseconds, having successfully completed roughly 10 hours of field validation at a commercial drone delivery hub.

(2) Internal Information-based Enacted Perception Method: To address the autonomous requirements of drones, this thesis proposes an event-camera-enhanced onboard visual positioning service method (EV-Pose). By achieving high-precision alignment between event streams and prior environmental models for pose estimation, the method establishes an endogenous perception-action loop for drones. Experimental results demonstrate that both the rotation and translation accuracies of the method outperform baseline methods by over 40%, with a latency of approximately 10.08 microseconds, effectively enabling precise drone landing.

(3) Multi-agent-oriented Engaged Collaboration Method: To break through the physical limitations of individual perception, this thesis proposes a heterogeneous collaborative method based on dynamic infrastructure (TransformLoc). This method leverages advanced drones as dynamic mobile nodes to provide intermittent, high-precision calibration for basic drones. Experimental results demonstrate that the method enhances the perception accuracy of multiple basic drones by 68% and increases the success rate of autonomous navigation by 60%.

In summary, this research constructs a "Perception-Interaction" dual-driven technical framework spanning from individual enacted perception to systemic multi-drone coordination. This paradigm shift effectively resolves the bottlenecks of perceptual rigidity and systemic isolation faced by drones in complex environments. It not only provides an intelligent perception foundation with high response speeds for the safe operation of the low-altitude economy, but also lays the groundwork for the future transition of drone swarms toward collective intelligence with evolutionary adaptation capabilities.

Keywords: Intelligent perception; drone; event camera; pose estimation; multi-modal fusion

TABLE OF CONTENTS

摘　要.....C
ABSTRACT .....D
TABLE OF CONTENTS ..... F
LIST OF FIGURES AND TABLES .....J
LIST OF SYMBOLS AND ACRONYMS ..... M
CHAPTER 1 INTRODUCTION..... 1
1.1 Research Background..... 1
1.2 Research Problems and Current Status 4
1.3 Research Content and Contribution..... 6
1.4 Thesis Structure.....10
CHAPTER 2 RELATED WORK.....12
2.1 External Information-based Enacted Perception Methods 12
2.1.1 Satellite-based Systems.....12
2.1.2 Optics-based Systems.....13
2.1.3 Wireless Signal-based Systems.....14
2.2 Internal Information-based Enacted Perception Methods.....15
2.2.1 Onboard Frame Camera-based Localization 16
2.2.2 Onboard LiDAR-based Localization.....17
2.2.3 Multi-Modal Sensor Fusion for Drone Localization.....17
2.3 Multi-drone System-oriented Engaged Coordination Methods 18
2.3.1 Multi-drone System-based Collaborative Sensing.....18
2.3.2 Collaborative SLAM-based Multi-drone System Localization 19
2.3.3 Infrastructure-based Multi-drone System Localization 20
2.4 Summary.....21
CHAPTER 3 EXTERNAL INFORMATION-BASED ENACTED PERCEPTION METHOD 22
3.1 Introduction 23
3.2 System Overview 27
3.2.1 Problem Formulation 27
3.2.2 Overview 28

3.3 Consistency-instructed Collaborative Tracking....29
3.3.1 Radar Tracking Model....30
3.3.2 Event Tracking Model....31
3.3.3 Consistency-instructed Measurements Filter....34
3.4 Graph-informed Adaptive Joint Optimization....35
3.4.1 Factor Graph-based Optimization....36
3.4.2 Probabilistic Representation....37
3.4.3 Fusion-based Tracking....37
3.4.4 Motion-aware Adaptive Optimization Scheme....39
3.5 Implementation....39
3.5.1 Incremental Optimization Method....39
3.5.2 Platform Implementation....40
3.6 Experimental Evaluation....40
3.6.1 Experimental Methodology....41
3.6.2 Overall Performance....43
3.6.3 System Robustness Evaluation....46
3.6.4 Ablation Study....49
3.6.5 System Efficiency Study....52
3.7 Case study: Real-world Drone Airport....53
3.8 Discussion....54
3.9 Summary....58
CHAPTER 4 INTERNAL INFORMATION-BASED ENACTED PERCEPTION METHOD....59
4.1 Introduction....60
4.2 Background and motivation....63
4.2.1 RGB Camera-based VPS....64
4.2.2 Event Camera-enhanced VPS....65
4.3 System Overview of EV-Pose....66
4.3.1 Problem Statement....66
4.3.2 Overview....67
4.4 Spatio-Temporal Feature-instructed Pose Estimation....68
4.4.1 Separated Polarity Time-Surface (SP-TS)....69
4.4.2 Novel Event-based Feature: Temporal Distance Field....70

4.4.3 Cross-Modality Pose Optimization via TDF Alignment....71
4.5 Motion-aware Hierarchical Fusion and Optimization....72
4.5.1 Early-stage Fusion: Motion-optical Flow-instructed Event Filtering....72
4.5.2 Later-stage Fusion: Graph-informed Joint Fusion and Optimization....75
4.6 System Implementation....79
4.6.1 Testbed Configuration....79
4.6.2 Map Optimization....79
4.6.3 Push Limit of Accuracy and Efficiency....80
4.7 Experimental Evaluation....80
4.7.1 Experiment Methodology....81
4.7.2 Overall Performance....83
4.7.3 Robustness Evaluation....86
4.7.4 Micro Benchmarks....88
4.7.5 Case Study: System-level Closed-loop Flight Control....91
4.8 Discussion....92
4.9 Summary....96
CHAPTER 5 MULTI-AGENT-ORIENTED ENGAGED COLLABORATION METHOD....97
5.1 Introduction....98
5.2 System Overview....103
5.2.1 Framework Goal....103
5.2.2 Framework Overview....104
5.3 Error-aware Joint Location Estimation Model....105
5.3.1 Key Definitions....105
5.3.2 Uncertainty-aided Inference Method....107
5.3.3 Joint Estimation of BMAVs' Location....108
5.4 Similarity-instructed Adaptive Grouping-scheduling Strategy....109
5.4.1 Problem Formulation....110
5.4.2 Graph-based Adaptive Grouping....110
5.4.3 Search Tree-based Non-myopic Scheduling....112
5.4.4 Scheduling of BMAV....114
5.5 Experimental Evaluation....115
5.5.1 Implementation and Methodology....115

5.5.2 Overall Performance 117
5.5.3 System Robustness Evaluation 121
5.5.4 System Micro-benchmark 122
5.5.5 System Efficiency 124
5.6 Discussion 124
5.7 Summary 128
CHAPTER 6 CONCLUSION AND FUTURE WORK 129
6.1 Conclusion 129
6.1.1 Holistic Adaptive Mode-switching Strategy 130
6.1.2 Broader Applicability Beyond Terminal Landing 131
6.1.3 Summary 132
6.2 Future Work 132
6.2.1 Integration of Enacted Perception and Engaged Coordination 132
6.2.2 Toward the Evolutionary Dimension 133
REFERENCES 137
APPENDIX A SUPPLEMENTAL MATERIAL OF MME-LOC 150
RESUME 157

LIST OF FIGURES AND TABLES

Figure 1.1 Illustration of low-altitude intelligent agent. 2
Figure 1.2 Leveraging bio-inspired mechanisms to realize the intelligent perception for drones..... 3
Figure 1.3 The main research content of the thesis..... 7
Figure 3.1 Snapshot of drone landing phase, drone airport, and performance of different sensors.....23
Figure 3.2 Benchmark study on delivery drone localization. 25
Figure 3.3 System architecture of mmE-Loc.....27
Figure 3.4 Distance calculation by frequency difference. 29
Figure 3.5 Direction calculation by phase difference.....30
Figure 3.6 Event tracking model. 31
Figure 3.7 Illustration of synchronous images and asynchronous events.....32
Figure 3.8 Event filtering performance.....33
Figure 3.9 Consistency-instructed measurements filter. 33
Figure 3.10 The structure of both the quadcopter and hexacopter drones features propellers arranged in axial symmetry.....34
Figure 3.11 Illustration of relationship between *GAJO*, event tracking model and radar tracking model. 36
Figure 3.12 Illustration of Graph-informed Adaptive Joint Optimization. 38
Figure 3.13 Experimental setup and scenarios of mmE-Loc.....41
Figure 3.14 Overall performance comparison of mmE-Loc and four related works...44
Figure 3.15 Robustness evaluation on drone type and env. & illu.....46
Figure 3.16 Robustness evaluation on background and distance. 47
Figure 3.17 Robustness evaluation on occlusion and velocity. 49
Figure 3.18 Ablation study on sensor fusion and impact of different modules. 50
Figure 3.19 Ablation study on performance of *CCT* and *GAJO*.....50
Figure 3.20 System efficiency evaluation. 52
Figure 3.21 Case study: delivery drone airport. 53
Figure 4.1 EV-Pose estimates the drone's 6-DoF pose by redesigning the drone-oriented VPS with event cameras. 60

Figure 4.2 Motivation study of EV-Pose. 61
Figure 4.3 Current VPS and comparison of frame camera-based VPS & event camera-enhanced VPS. 64
Figure 4.4 Principles of frame cameras and event cameras. 65
Figure 4.5 System architecture of EV-Pose. 67
Figure 4.6 Illustration of SP-TS, temporal distance field, and 3D point map. 70
Figure 4.7 Mechanism of the adaptive event filtering and buffer zone strategy. 74
Figure 4.8 Illustration of the graph-informed joint fusion and optimization. 76
Figure 4.9 Testbed configuration of EV-Pose platform. 79
Figure 4.10 Overview of the real-world experimental setups and challenging test scenarios. 81
Figure 4.11 Real-world indoor experiments. 82
Figure 4.12 Real-world outdoor experiments. 83
Figure 4.13 Overall performance of *EV-Pose* and three SOTA 6-DoF pose tracking systems on VECtor dataset. 84
Figure 4.14 Overall performance of *EV-Pose* and three SOTA 6-DoF pose tracking systems on TUM-VIE dataset. 85
Figure 4.15 Impact of drone moving velocity. 86
Figure 4.16 Impact of drone estimation frequency. 86
Figure 4.17 Impact of HDR. 87
Figure 4.18 Impact of scene dynamics. 87
Figure 4.19 Impact of modules. 89
Figure 4.20 Comparison of different fusion frameworks. 89
Figure 4.21 Resource overhead. 90
Figure 5.1 Introduction of a heterogeneous multi-MAV system composed of two types: AMAVs and BMAVs. 98
Figure 5.2 Motivating example of TransformLoc. 100
Figure 5.3 Demonstration of the TransformLoc framework. 104
Figure 5.4 The error-aware joint pose estimation model. 108
Figure 5.5 Similarity-instructed adaptive grouping-scheduling strategy. 111
Figure 5.6 Experimental setup and implementation of a heterogeneous multi-MAV system. 115

Figure 5.7 Overall performance in in-field experiments with two AMAVs and six BMAVs. 118
Figure 5.8 Overall performance in in-field experiments with two AMAVs and six BMAVs. 119
Figure 5.9 System robustness evaluation with five AMAVs and twenty BMAVs. ... 120
Figure 5.10 System micro-benchmark. 122
Figure 5.11 System efficiency. 123

Table 4.1 Latency of different systems. 82

LIST OF SYMBOLS AND ACRONYMS

6-DoF	Six-degree-of-freedom
AI	Artificial Intelligence
AIoT	Artificial Intelligence of Things
AMAV	Advanced MAV
ARE	Absolute Rotation Error
ATE	Absolute Translation Error
BMAV	Basic MAV
CDF	Cumulative Distribution Function
COTS	Commercial Off-the-Shelf
CSI	Channel State Information
EKF	Extended Kalman Filter
FAA	Federal Aviation Administration
FoV	Field of View
GPS	Global Positioning System
HDR	High Dynamic Range
IF	Frequency Signal
IMU	Inertial Measurement Unit
IoT	Internet of Things
LPF	Low-pass Filter
MAV	Micro Aerial Vehicle
MoCap	Motion Capture
NRTK	Network RTK
PF	Particle Filter
RF	Radio Frequency
RRE	Relative Rotation Error
RSSI	Received Signal Strength Indicator
RTE	Relative Translation Error
RTK	Real-Time Kinematic
SAE	Surface of Active Events
SLAM	Simultaneous Localization and Mapping

SoTA	State-of-the-Art
TDoA	Time-Difference-of-Arrival
ToA	Time-of-Arrival
ToF	Time-of-Flight
TS	Time Surface
UWB	Ultra-Wideband
VIO	Visual Inertial Odometry
VPS	Visual Positioning Service

CHAPTER 1 INTRODUCTION

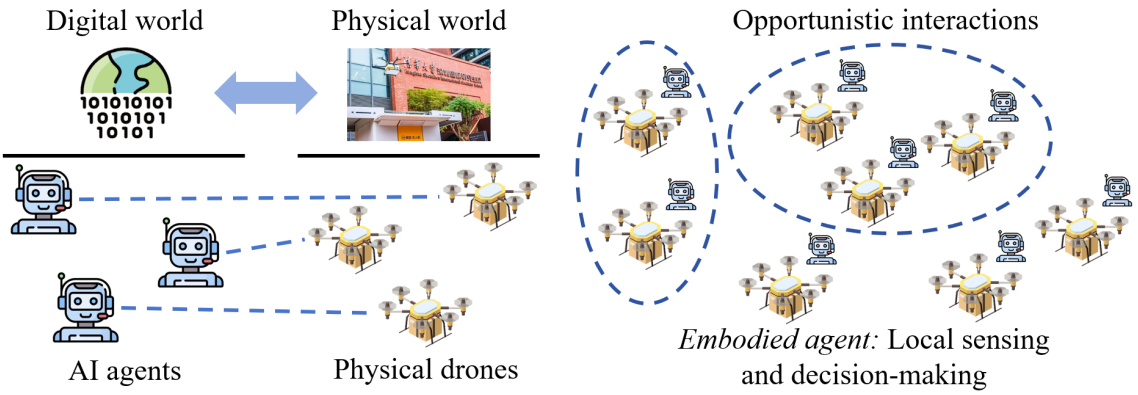
1.1 Research Background

The low-altitude economy is a comprehensive economic model that leverages low-altitude flight activities to drive industrial integration and innovation. Compared to traditional economic models that rely on ground-based infrastructure (e.g., roads and railways), the low-altitude economy exploits the underutilized airspace below 1000 meters, significantly extending the boundaries of economic activities and reinforcing the vision of harnessing the sky for productivity. As this sector rapidly expands, drones or Micro Aerial Vehicles (MAVs) have emerged as key enablers of various low-altitude applications.

Drone technology has also garnered attention from national and international organizations. Notably, China's Outline of the National Comprehensive Three-Dimensional Transportation Network Plan, issued by the CPC Central Committee and the State Council in 2021, underscores the strategic importance of low-altitude economic and drone development. Similarly, the FAA in the United States released the Near-Term Implementation Plan for Advanced Air Mobility in 2023, while the European Union introduced the European Air Traffic Management Master Plan 2025. From 2024 to 2025, policy initiatives and industrial advancements are progressing in tandem, with various regions across China actively establishing new frameworks for low-altitude economic development.

However, the Implementation Plan for Innovation and Application of General Aviation Equipment (2024-2030), jointly issued by the Ministry of Industry and Information Technology and three other departments, identifies several critical challenges in drone technology: reliable localization, precise environmental perception, and intelligent swarm coordination. Specifically, as the low-altitude economy accelerates, drone operational environments are transitioning from open airspace to complex and extreme urban settings. In these scenarios, traditional localization and navigation paradigms face fundamental challenges characterized by severe perception bottlenecks:

- *Inadequate dynamic adaptability at the individual dimension:* Conventional frame-based vision systems are constrained by a strictly discrete spatiotemporal sampling mechanism. Restricted by fixed sampling frequencies and exposure mechanisms, it frequently suffers from motion blur and high latency (typically $>$ 30ms) during high-dynamic flight, where drone flight control response demands exceed 150Hz.



(a) Devices are "Enacted" in real world (b) Agents are "Engaged" in interactions

Figure 1.1 Illustration of low-altitude intelligent agent. (a) Enacted Autonomy: Hardware units transform into self-governing entities by tightly coupling sensory inputs, physical actuation, and continuous environmental engagement. (b) Engaged Coordination: Agents dynamically establish decentralized, intermittent communication links that are strictly governed by real-time spatio-temporal conditions.

This discrete sampling logic fails to satisfy the drone's requirement for continuous, real-time physical interaction.

- *Isolation and vulnerability at the system dimension:* Current drone localization solutions rely heavily on external timing signals (e.g., GNSS) and are largely confined to closed-loop individual perception. In extreme environments with signal masking or limited sensing, the lack of physical-layer information symbiosis and synergy between drones or between drones and their environment renders these systems devoid of biological-like resilience.

These technological bottlenecks may constrain the long-term development of the low-altitude economy, necessitating intensified research and development efforts to enhance core technological capabilities.

To address the aforementioned bottlenecks, this research adopts the emerging paradigm of the intelligent perception[1-2]. Under the paradigm of intelligent perception, drones must possess the following two crucial capabilities to successfully evolve into low-altitude agents, as illustrated in Figure 1.1: *(i) Enacted*, where intelligence manifests through perception-action loops rather than rigid heuristics; *(ii) Engaged*, characterized by decentralized, event-driven, and spatio-temporally constrained peer-to-peer interactions.

By leveraging bio-inspired dynamic vision (event camera) ① to realize the intelligent perception for drones, this study seeks to effectively transform conventional drones into autonomous low-altitude agents capable of Enacted perception and Engaged coordination, thereby transcending the inherent limitations of discrete and isolated perception. The

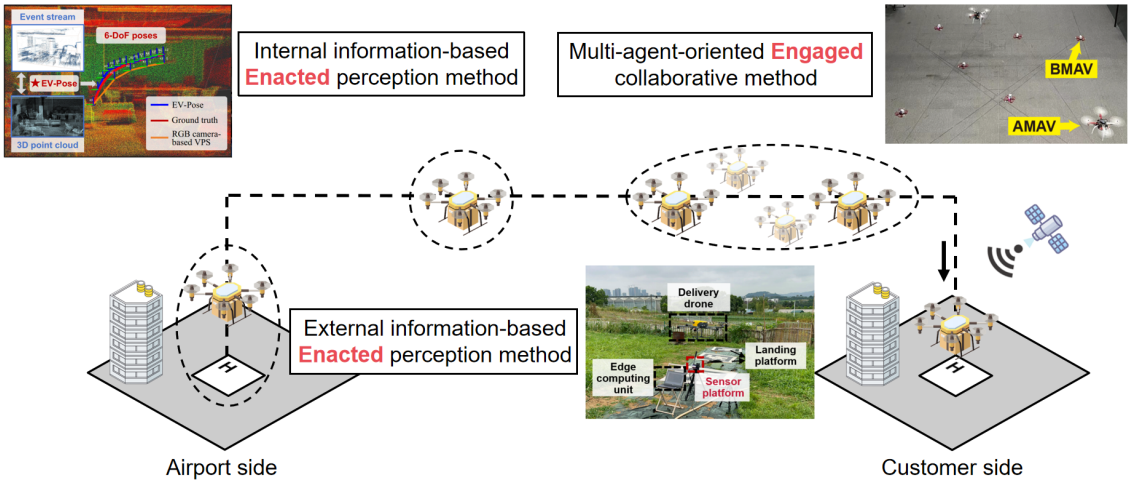


Figure 1.2 Leveraging bio-inspired mechanisms to realize the intelligent perception for drones.

specific objectives are structured into two core dimensions, as shown in Figure 1.2:

- *Developing Enacted Perception Method:* To address the inefficiencies and high latency of traditional perception, this research introduces event cameras in drone perception. By leveraging asynchronous spike mechanisms to emulate the biological retina, this thesis aims to upgrade drone's perception-action loop. This grants the agent μs-level temporal resolution, achieving physiological-grade responsiveness.
- *Establishing Engaged Agents Collaboration:* To overcome individual FoV constraints and environmental uncertainties, this research transcends single-agent boundaries to investigate how multiple drones execute intermittent interactions governed by physical constraints and environmental context. By establishing a heterogeneous symbiosis, this thesis enables the spatio-temporal flow of multi-source information, constructing a collaborative localization space.

Scope of this thesis. The current scope of this research focuses on the realization of the enacted perception and engaged interaction phases. Building upon these foundations, this thesis will further explore the self-organization and adaptive evolution of the multi-drone system through continuous environmental interaction in future work. This trajectory aims to ultimately drive the transition of aerial platforms from rigid, constrained algorithms toward autonomous, self-evolving embodied swarm intelligence.

① Principle of event camera: Unlike conventional cameras that capture synchronous frames, neuromorphic event cameras are bio-inspired sensors mimicking the biological retina. Each pixel operates independently, asynchronously triggering an 'event' only when detecting a local change in light intensity. This sparse, event-driven mechanism provides microsecond-level temporal resolution and HDR, effectively eliminating motion blur and bypassing the latency bottlenecks of traditional vision to perfectly align with the high-frequency demands of drone flight control.

1.2 Research Problems and Current Status

While leveraging the intelligent perception paradigm to transform conventional drones into autonomous low-altitude agents capable of enacted perception and engaged coordination offers immense potential to overcome the inherent limitations of discrete and isolated perception, significant technical hurdles remain. First, enhancing a drone's enacted perception through bio-inspired dynamic vision is complicated by the asynchronous event-triggering mechanism and the sparse semantic information characteristic of event cameras. Second, establishing a heterogeneous symbiosis to facilitate intermittent interactions among multiple drones introduces complex challenges in resource allocation and systemic management, particularly as the swarm scales. Specifically,

(1) *Utilizing event cameras to enable external information-based enacted perception method:* The lack of semantics and scale in event streams makes it difficult to extract drone features. The inherent scarcity of semantic information complicates the extraction of drone-specific features, thereby impeding the achievement of low-latency, high-precision localization. Specifically, the output of event cameras lacks inherent drone semantic information, complicating the retrieval of the drone's two-dimensional pixel coordinates within the event camera reference frame. Additionally, the immense, noisy output of event cameras overwhelms critical drone information, making target extraction more difficult. Moreover, the absence of scale information prevents direct estimation of the drone's absolute position in three-dimensional space. To align with the high temporal resolution of event cameras, mmWave radar, which also operates at a high frequency, can be introduced to supplement scale information while maintaining a consistent data acquisition rate, enabling low-latency drone localization. However, the tracking results from these two sensors exhibit significant discrepancies in terms of precision, data density, and scale, making existing Extended Kalman Filter (EKF) or Particle Filter (PF)-based approaches prone to cumulative drift errors, thus failing to achieve effective multimodal sensor fusion for precise localization. Furthermore, given the limited computational resources available on landing platforms, it is crucial to optimize the efficiency of fusion to enable low-latency drone ground localization.

(2) *Leveraging event cameras to empower internal information-based enacted perception method:* The spatiotemporal sparsity of event streams makes it difficult to align environmental features. The inherent semantic sparsity within asynchronous event streams poses significant hurdles for reliable environment feature extraction. This complication

hinders efficient and precise data association with the prior 3D point map, ultimately constraining the attainment of high-fidelity drone localization. Specifically, unlike RGB frames generated by frame cameras, which contain rich semantic information, event cameras produce asynchronous and noisy event streams that lack inherent semantic content. This fundamental difference renders traditional feature extraction methods designed for RGB frames ineffective for event data. As a result, critical features within the event stream that are essential for 3D point map matching often cannot be extracted, leading to frequent matching failures and underutilization of the 3D point maps. Furthermore, event cameras are highly sensitive to environmental changes, where even slight movements trigger numerous pixels and generate hundreds of events. As the drone moves, rapid scene variations lead to event bursts, producing thousands of event reports in a short period. Directly utilizing all events during the matching process significantly degrades the computational efficiency of pose estimation. More critically, the inherent lack of semantic content and scale ambiguity in event cameras results in incorrect matching when dealing with complex structures (e.g., repetitive textures), leading to a decline in localization accuracy.

(3) *Establishing a heterogeneous symbiosis to facilitate multi-agent-oriented engaged collaboration method:* It is challenging to establish intermittent interactions among multi-agent systems. Leveraging Advanced Micro Aerial Vehicles (AMAVs) as external information sources for multi-MAV systems introduces two primary challenges. First, it must account for the unknown and heterogeneous localization errors inherent in Basic MAVs (BMAVs). Second, it necessitates solving the complexities of resource allocation under highly dynamic operational constraints. First, BMAV localization errors accumulate due to sensing noise and environmental variations, making accurate state estimation increasingly difficult. Since each AMAV has a limited field of view (FoV), it can only assist a subset of BMAVs at any given time. Ideally, AMAVs should provide external observations to correct significant localization errors in BMAVs. However, real-time derivation of BMAV localization errors is challenging due to the lack of static reference points. Without timely corrections, localization errors may persistently accumulate, degrading the overall accuracy of the swarm. Second, optimizing AMAV resource allocation—balancing sensing and computing resources across a large fleet of BMAVs—requires decision-making in a high-dimensional space. The challenge is further compounded by three factors: *(i)* an exponential growth in search space as the number of AMAVs increases, *(ii)* constantly shifting BMAV observability, as both AMAVs and BMAVs are in continuous motion,

demanding dynamic adjustments in allocation strategies, and *(iii)* observation accuracy variations due to differences in distance and bearing angles between AMAVs and BMAVs, directly affecting localization performance. These interconnected factors make real-time, efficient resource allocation highly complex within a large-scale multi-MAV system.

This thesis conducts an in-depth investigation into low-altitude intelligent perception with bio-inspired dynamic vision. By developing specialized algorithms for event cameras and designing reliable resource allocation methods for multi-drone systems, this research enables both external and internal information-based enacted perception methods, while establishing a heterogeneous symbiosis to facilitate engaged intermittent interactions in multi-agent systems. Through a vertical exploration, ranging from low-level biomimetic perceptual fusion to system-level heterogeneous synergy, this work establishes a dual-driven "Perception-Interaction" framework for aerial intelligent perception. The proposed framework is specifically designed to overcome technical bottlenecks of perceptual ambiguity, computational latency, and coordination complexity faced by aerial agents.

1.3 Research Content and Contribution

As illustrated in Figure 1.3, this research addresses the critical challenges of inadequate dynamic adaptability at the individual dimension, as well as isolation and vulnerability at the system dimension. This is achieved by upgrading drones into low-altitude agents, which enables enacted perception and engaged intermittent interactions within multi-agent systems. This integrated approach specifically mitigates the technical bottlenecks of perceptual ambiguity, computational latency, and coordination complexity that aerial agents frequently encounter in extreme environments. Research content includes:

(1) *On the external information-based enacted perception method front*, this thesis proposes mmE-Loc, a ground-based localization method integrated with event cameras and mmWave radar. This method is designed to transform drones into agents with biological-like perceptual characteristics by leveraging external sensory sources. To address the challenges of GNSS-denied environments and motion blur in conventional vision, particularly during critical flight phases such as takeoff and landing, this thesis introduces a ground-based "enacted sensory" solution. By harnessing the temporal consistency and spatial complementarity of event cameras and mmWave radar, this approach effectively converts ground infrastructure into sensitive external information sources, achieving centimeter-level precision and millisecond-level latency. Consequently, it provides

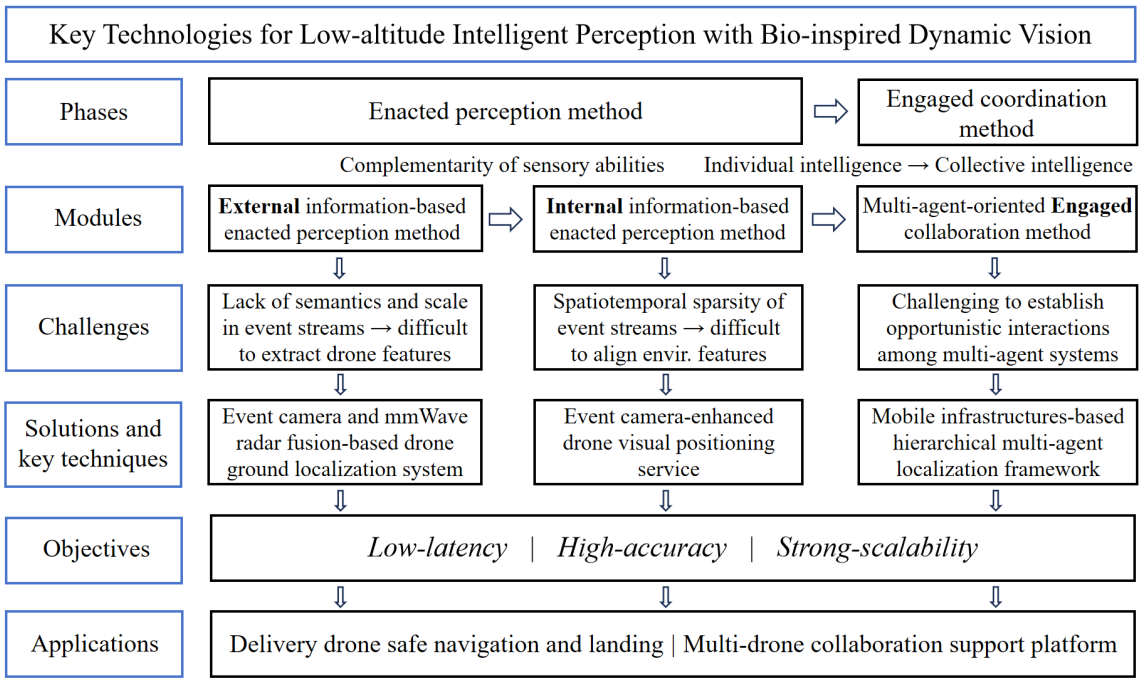


Figure 1.3 The main research content of the thesis.

highly reliable external feedback to ensure the safe operation of aerial agents.

(2) *On the internal information-based enacted perception method front*, this thesis proposes EV-Pose, an event-camera-enhanced visual positioning service. This method aims to transform drones into agents with biological-like perceptual traits by leveraging onboard multi-sensory sources. By re-designing the underlying perceptual logic, EV-Pose achieves a real-time response rate that matches high-frequency flight control demands. Specifically, to address the requirements for high-mobility autonomous navigation in GNSS-denied environments, EV-Pose constructs a lightweight onboard endogenous sensing method. By utilizing event streams to emulate the spiking mechanisms of the biological retina, EV-Pose developed Spatio-Temporal Features to surmount the challenges of semantic sparsity. Furthermore, EV-Pose implements a hierarchical fusion mechanism characterized by perception-action coupling. This architecture effectively establishes a compact onboard perception-action loop.

(3) *On the multi-agent-oriented engaged collaboration method front*, this thesis proposes TransformLoc, a hierarchical localization framework for multi-MAV systems leveraging mobile infrastructure. This framework aims to transcend the physical boundaries of individual perception, constructing a resilient collaborative localization space through physical linkages and dynamic interactions between agents. Specifically, to address the collaborative challenges of large-scale heterogeneous multi-agent systems in complex en-

vironments, this framework introduces a hierarchical architecture based on "participatory interaction." Furthermore, the framework exploits physical spatial constraints among agents to facilitate intermittent information alignment, providing on-demand observational corrections for resource-constrained lower-level agents. This approach reduces overall system hardware costs while enhancing the swarm's collective survivability in areas with degraded sensing coverage.

The perception-decision-action closed loop. Within the context of intelligent perception, the perceptual and coordination methods proposed in this thesis are not isolated estimators; they are deeply coupled with the physical action capabilities of the drones. At the individual agent level, the ultra-low latency and high-frequency pose estimations generated by the enacted perception modules (e.g., mmE-Loc, EV-Pose) are continuously streamed to the drone's underlying flight control stack (e.g., the PX4 position and attitude controllers). By significantly reducing the perceptual latency compared to traditional frame-based vision, these methods fundamentally minimize the phase margin delay within the control feedback loop. Furthermore, at the swarm level, the TransformLoc framework explicitly extends this closed loop by feeding synergistic localization outputs directly into the multi-drone trajectory planning modules. This tight 'perception-decision-action' coupling, validated through system-level closed-loop flight experiments across all proposed frameworks, directly translates to enhanced control stability. It allows both individual agents and the collective multi-agent system to execute highly adaptive, dynamic physical adjustments in real time when interacting with complex environments.

Specific research content and results. The following sections will detail the specific research content and present the corresponding experimental results.

(1) *External information-based enacted perception method.* This thesis introduces mmE-Loc, an active, high-precision, and low-latency localization method designed to enable external information-based enacted aerial agents by integrating mmWave radar and event cameras (Chapter 3). By transforming ground infrastructure into a bio-inspired "external sensory organ", mmE-Loc empowers drones with the perceptual feedback necessary for embodied autonomy. The microsecond-level temporal resolution of event cameras aligns seamlessly with the high-frequency acquisition of mmWave radar, while their 2D imaging capabilities compensate for the radar's inherent spatial resolution constraints. To harness the temporal coherence and spatial complementarity of these heterogeneous modalities, mmE-Loc develops two core algorithmic modules: (1) Consistency-Instructed

Collaborative Tracking, which enhances the precision of drone-specific measurement extraction through bio-inspired spatiotemporal filtering; (2) Graph-Informed Adaptive Joint Optimization, which facilitates efficient, multi-modal sensor fusion for high-fidelity pose estimation. The mmE-Loc method is fully realized using commercial off-the-shelf (COTS) components. Comprehensive evaluations—encompassing over 30 hours of testing across diverse indoor and outdoor scenarios—demonstrate its superiority over four SOTA techniques. The method achieves an average accuracy of 0.083 m with a latency of 5.15 ms, outperforming existing approaches by over 48% in precision and 62% in responsiveness. Furthermore, a 10-hour deployment at a commercial drone delivery hub validates its operational feasibility for high-safety real-world landing maneuvers, effectively bridging the gap between theoretical paradigms and practical aerial applications.

(2) *Internal information-based enacted perception method.* This thesis presents EV-Pose, an event camera-enhanced Visual Positioning Service (VPS) that transforms drones into autonomous enacted agents by enabling high-precision, low-latency 6-DoF pose tracking (Chapter 4). By aligning neuromorphic event streams with a prior 3D point map, EV-Pose establishes a reliable onboard perception-action loop independent of external timing signals. To surmount the challenges of semantic sparsity and high-dynamic instability, EV-Pose introduces two pivotal modules: (1) Spatio-Temporal Feature-Guided Pose Estimation, which leverages a temporal distance field extracted from asynchronous event streams to facilitate high-fidelity alignment with 3D point maps; (2) Motion-Aware Hierarchical Fusion and Optimization, which exploits kinetic cues—derived from both early-stage event filtering and final pose refinement—to enhance systemic robustness and computational efficiency. The EV-Pose method is fully implemented and integrated into ArduPilot, a leading open-source flight control ecosystem. Extensive evaluations conducted across public datasets featuring extreme lighting, high-velocity maneuvers, and complex dynamic environments demonstrate its superiority. Benchmarks against three state-of-the-art (SOTA) methods show that EV-Pose achieves a rotation accuracy of 1.76° and a translation accuracy of 7.5mm with a latency of 10.08ms, outperforming baselines by >40% and enabling accurate drone landings.

(3) *Multi-agent-oriented engaged collaboration method.* This thesis presents TransformLoc, a novel framework that repurposes Advanced Micro Aerial Vehicles (AMAVs) as dynamic mobile localization infrastructures, specifically optimized for low-cost, resource-constrained Basic Micro Aerial Vehicles (BMAVs) (Chapter 5). By bridging

the gap between high-performance and resource-limited platforms, TransformLoc facilitates engaged intermittent interactions and ensures strong scalability within large-scale multi-drone systems. To achieve high-fidelity collaborative positioning, TransformLoc introduces two pivotal components: (1) an Error-Aware Joint Localization Model, which performs intermittent joint state estimation to systematically mitigate the diverse localization uncertainties inherent in BMAVs; (2) a Similarity-Guided Adaptive Grouping and Scheduling Strategy, which dynamically optimizes AMAV resource allocation based on real-time BMAV localization demands and spatio-temporal constraints. Through the integration of these modules, TransformLoc establishes a collaborative, adaptive, and cost-effective localization paradigm, realizing a heterogeneous symbiosis that is uniquely suited for large-scale operations. This thesis comprehensively evaluates TransformLoc against state-of-the-art (SOTA) baselines using an extensive experimental suite, including a real-time hardware testbed and high-fidelity physical simulations. Experimental results indicate that TransformLoc maintains a mean localization error below 1.0 m for BMAVs operating with minimal onboard perception, surpassing existing baselines by up to 68%. This technical improvement translates to a 60% increase in navigation success rates, significantly enhancing the collective survivability and deployment feasibility of multi-drone swarms. Furthermore, large-scale stress tests under varying fleet sizes and measurement noise conditions validate the framework's robustness and adaptability, while detailed ablation studies underscore the efficacy of its hierarchical design.

In summary, the three aforementioned research thrusts exhibit a logically progressive structure: Research Thrusts I and II focus on the Enacted dimension, re-engineering the agent's perception system through "external-source assistance" and "endogenous autonomy," respectively. Research Thrust III extends the scope to the Engaged dimension, leveraging "multi-agent synergy" to transcend the inherent perceptual boundaries of individual platforms. Together, these components constitute a multi-dimensional low-altitude agents intelligent perception technical framework, spanning ground-based assistance, onboard autonomy, and swarm coordination. This integrated method provides a comprehensive solution to the spatio-temporal perceptual constraints faced by aerial agents in complex and dynamic environments.

1.4 Thesis Structure

The structure of this dissertation is logically organized into six chapters:

Chapter 2: Literature Review. This chapter provides a comprehensive synthesis of existing methodologies regarding external and internal information-based aerial agents and the current landscape of multi-agent coordination. It identifies the fundamental gaps between conventional paradigms and the proposed intelligent perception with bio-inspired dynamic vision, establishing the necessity for subsequent research.

Chapter 3: External Information-based Enacted Perception Method. This chapter proposes mmE-Loc, a ground-based drone localization method utilizing mmWave radar and event camera fusion. By synergizing the high-frequency sampling of radar with the microsecond-level temporal resolution of event streams, this method transforms ground infrastructure into an "external sensory organ". This paradigm empowers drones with biological-like perceptual characteristics, and its operational feasibility is validated through deployment in a commercial drone delivery hub.

Chapter 4: Internal Information-based Enacted Perception Method. Shifting the focus to onboard autonomy, this chapter develops EV-Pose, an event-camera-enhanced visual positioning service. By leveraging neuromorphic vision and prior 3D point maps, the method enables drones to extract semantic features from asynchronous event streams for robust data association, transforming drones into agents with biological-like perceptual traits. The method's efficacy is demonstrated on a custom-built hardware platform, showcasing its resilience under high-dynamic maneuvers and semantic sparsity.

Chapter 5: Multi-agent-oriented Engaged Collaboration Method. To transcend the physical and computational boundaries of individual sensing, this chapter introduces TransformLoc, a hierarchical localization framework rooted in hierarchical symbiosis. By repurposing a subset of resource-rich agents as dynamic mobile infrastructures, the framework facilitates engaged intermittent interactions and provides on-demand corrections for resource-limited drones. This symbiotic approach ensures systemic scalability and collective survivability, validated on a large-scale, self-built multi-drone testbed.

Chapter 6: Conclusions and Future Directions. This chapter summarizes the core contributions of this dissertation and synthesizes the findings within the broader intelligent perception of aerial agent architecture. Finally, it outlines potential avenues for future research, focusing on the evolutionary adaptation and self-organization of aerial swarms to achieve higher-level collective intelligence.

CHAPTER 2 RELATED WORK

This chapter provides a comprehensive synthesis of the research landscape relevant to this dissertation, systematically categorizing existing methodologies into external information-based and internal information-based enacted perception, as well as multi-drone engaged coordination. First, it examines external information-based enacted perception, which relies on exogenous infrastructure to enable enacted aerial agents. This section focuses on satellite-based GNSS, ground-based optical systems, and wireless signal-based approaches, identifying their vulnerabilities in signal-masked or high-dynamic environments where external "senses" become unreliable. Next, the review explores internal information-based enacted perception, highlighting recent advancements in onboard frame-based vision and LiDAR-based SLAM. This analysis emphasizes the constraints of discrete sampling and computational bottlenecks that hinder the realization of truly autonomous, bio-inspired enacted agents in resource-constrained scenarios. Finally, the chapter discusses multi-drone system-oriented engaged coordination, with a particular focus on how the reliance on static infrastructure limits the engaged coordination and operational range of the multi-drone system. By synthesizing these perspectives, this chapter highlights the need for a low-altitude intelligent perception paradigm, spanning individual enacted autonomy to collective engaged interaction, proposed in the subsequent research.

2.1 External Information-based Enacted Perception Methods

External information-based localization systems leverage exogenous sensing modalities to augment the positional awareness of aerial agents, particularly in GNSS-denied or perception-challenged environments. Within the low-altitude embodied agents framework, these systems function as external sensory organs, providing the necessary environmental grounding to enable enacted aerial agents. Current methodologies can be broadly categorized into satellite-based, optics-based, and wireless signal-based approaches, each presenting distinct trade-offs between precision, coverage, and latency.

2.1.1 Satellite-based Systems

The GPS provides meter-level accuracy outdoors by measuring the time-of-flight (ToF) of signals from at least four satellites to estimate the receiver's location via tri-

lateration[3-4]. However, standard GPS suffers from multiple sources of error, including satellite clock drift, ionospheric and tropospheric delays, multipath interference, and receiver noise. These errors can accumulate, leading to location deviations that are especially problematic in environments with obstructed satellite visibility, such as urban canyons and dense forests.

Real-Time Kinematic (RTK) enhances GPS accuracy to the centimeter-level by utilizing carrier-phase measurements and differential corrections from a reference base station. Unlike standard GPS, which relies solely on pseudorange measurements, RTK exploits phase of GPS signal's carrier wave, which has a much shorter wavelength. By continuously comparing phase of carrier wave received at a mobile receiver and a base station, RTK can resolve integer cycle ambiguities and achieve high-precision locationing.

RTK operates in two primary modes: *(i) Single-base RTK*: A fixed ground station provides real-time differential corrections to a receiver, which refines its location estimate. This setup requires a stable communication link and is sensitive to base station availability and transmission delays. *(ii) Network RTK (NRTK)*: Instead of relying on a single base station, NRTK leverages a network of continuously operating reference stations to model and compensate for atmospheric errors over a wider area, extending the effective operational range beyond single-base RTK.

Despite its high precision, RTK faces several challenges: *(i) Dependency on base stations*: RTK requires a stable connection to a reference station, which can be disrupted in areas with poor communication infrastructure. *(ii) Sensitivity to multipath interference*: In dense urban environments, reflected signals can introduce phase ambiguities, degrading accuracy. *(iii) Initialization time and cycle slips*: The integer ambiguity resolution process requires an initialization period and can be disrupted by signal obstructions, necessitating re-initialization. These limitations make RTK less reliable in highly dynamic environments or areas with frequent signal obstructions, such as urban canyon environments and under dense foliage[5-7].

2.1.2 Optics-based Systems

Optics-based localization systems leverage visual or infrared markers, structured light, or passive image-based techniques to achieve high-precision location estimation. These systems include motion capture (MoCap), LiDAR-based localization system, and visual-based localization system, each with distinct advantages and limitations. *(i) Motion Capture (MoCap) systems*: MoCap systems, such as Vicon[8] and OptiTrack[9], use mul-

tiple infrared cameras to track reflective markers attached to an object. By triangulating marker locations from multiple viewpoints, these systems can achieve sub-millimeter accuracy in controlled indoor environments. However, MoCap requires precise calibration, a well-defined tracking volume, and a line-of-sight between cameras and markers, making it impractical for outdoor use or large-scale deployment[10]. *(ii) LiDAR-based localization system:* LiDAR sensors emit laser pulses and measure their ToF to construct a 3D map of the environment and localize the drone[11]. However, LiDAR suffers from high computational costs and degraded performance in featureless environments. *(iii) Visual-based localization system*: Visual-based localization system fuses data from cameras to estimate the motion of the drone[12]. Feature-based methods track visual keypoints to infer pose changes of the drone[13-15]. While these systems work in GPS-denied environments, they suffer from scale drift without external corrections and are affected by lighting conditions and motion blur.

Despite their high precision, optics-based localization systems face several challenges, making optics-based methods less suitable for large-scale or outdoor deployments: *(i) Environmental constraints:* MoCap requires a controlled indoor setup. Meanwhile, cameras and LiDARs, with low frame rates ($< 50 Hz$), may miss rapid movements during frames, reducing localization accuracy. *(ii) Calibration complexity:* Accurate localization depends on precise camera calibration and sensor synchronization. *(iii) Scalability issues:* Expanding MoCap coverage requires additional cameras, and LiDAR/cameras may struggle in dynamic or unstructured environments.

2.1.3 Wireless Signal-based Systems

Wireless signal-based localization techniques, including radar[16-17], acoustic systems[5], and radio frequency (RF) methods such as Wi-Fi and Ultra-Wideband (UWB), are often integrated with SLAM or deep learning algorithms[18-23] to enhance drone localization. However, the limited spatio-temporal resolution of these sensors affects their ability to achieve precise, low-latency localization, particularly for landing applications.

Radar-based systems. mmWave radar offers high depth estimation accuracy along the radial axis but struggles with tangential motion tracking due to inherent angular resolution constraints[24-25]. While mmWave radar's sub-millimeter wavelength improves sensitivity[26-28], accurately localizing a drone's center remains challenging, as drones (e.g., 80 cm across) appear as non-uniform radar reflections[24,29-30]. Furthermore, multipath scattering can obscure direct signals, and the low spatial resolution of radar can lead

to unstable tracking results with frequent outliers[31]. Deep learning-based radar processing has been explored[32-34], but these methods require extensive pre-training for specific drone models and often fail to generalize across environments.

Acoustic-based systems. Acoustic localization leverages time-of-arrival (ToA), time-difference-of-arrival (TDoA), or phase-based methods to estimate location by analyzing sound wave propagation. These systems are advantageous in GPS-denied environments, as sound waves can travel through materials that block electromagnetic signals[5,35]. However, they face significant challenges: *(i) Environmental noise sensitivity:* Acoustic signals are highly susceptible to interference from background noise, including wind, machinery, and even human speech, which can distort signal measurements. *(ii) Multipath effects:* Reflections from surrounding surfaces introduce signal distortions, affecting ToA and TDoA accuracy. *(iii) Limited range and resolution:* High-frequency acoustic waves provide better resolution but attenuate rapidly over distance, reducing their effectiveness in large-scale outdoor environments. *(iv) Doppler effects:* Rapid drone movements cause frequency shifts in acoustic signals, complicating tracking and requiring compensation algorithms.

Radio frequency-based systems. Wi-Fi and Ultra-Wideband signals provide another approach to wireless localization by leveraging existing RF infrastructure[36]. *(i) Wi-Fi-based localization:* Wi-Fi localization methods typically use Received Signal Strength Indicator (RSSI), ToF, or Channel State Information (CSI) to estimate drone locations. While Wi-Fi networks are widely available, they suffer from low localization accuracy (meter-level) due to signal fluctuations, interference, and multipath propagation in cluttered environments. Moreover, real-time performance is often limited by network congestion and latency. *(ii) UWB-based localization:* UWB systems offer superior localization accuracy (centimeter-level) by utilizing high-bandwidth signals and precise ToA measurements. UWB is robust against multipath effects and interference, making it ideal for indoor environments. However, its reliance on pre-deployed anchors limits scalability in large outdoor spaces, and the high energy consumption of UWB transmitters poses challenges for drone applications.

2.2 Internal Information-based Enacted Perception Methods

Internal information-based localization empowers drones with endogenous autonomy by utilizing onboard sensors, such as RGB-D cameras and LiDAR, in conjunction with

prior environmental maps for precise pose estimation. Within the low-altitude embodied agents framework, these onboard modalities function as the primary "sensory organs" for enacted aerial agents, enabling self-contained navigation without reliance on exogenous signals. However, existing internal sensing paradigms face significant hurdles in achieving the high-frequency, low-latency feedback required for agile embodied intelligence.

2.2.1 Onboard Frame Camera-based Localization

Vision-based approaches using RGB or RGB-D cameras track relative poses via feature matching[37-38]. Inertial Measurement Unit (IMU)-based methods infer the rotation and motion of drone from the built-in gyroscope and accelerometer, though cost-effective and widely used, suffer from severe cumulative drift[39-41]. Recent researches integrate visual odometry with IMU for Visual Inertial Odometry (VIO) for better performance[42-43]. Modern VPS for drone 6-DoF pose tracking further integrates infrastructure of multi-source visual cues and prior environmental maps (e.g., feature point maps or 3D point map) to enhance global pose estimation and drift correction. These systems typically rely on the following key components: *(i) 2D-3D feature matching:* By extracting keypoints from an onboard RGB or RGB-D camera, VPS matches them against a pre-built 3D map (e.g., Structure-from-Motion or LiDAR-based maps) to estimate the absolute pose of the drone[44-45]. *(ii) Keyframe-based SLAM:* Some VPS solutions incorporate keyframe selection strategies to maintain a sparse but informative set of visual landmarks, improving robustness against occlusions and dynamic scenes. *(iii) Optimization-based pose refinement:* To mitigate local tracking errors, VPS employs pose graph optimization or bundle adjustment techniques, refining estimates based on spatial constraints and temporal consistency. However, despite their advantages, these systems encounter significant accuracy and latency constraints in drone applications: *(i) Motion blur and rapid maneuvers:* High-speed drone motion leads to severe motion blur, degrading feature quality and reducing tracking reliability, especially in low-light or textureless environments. *(ii) Computational overhead:* The process of feature extraction, descriptor matching, and map-based pose optimization is computationally expensive. Even with optimized implementations, VIO-based tracking is typically constrained to $< 50Hz$, while global map-based pose updates operate at only $< 1Hz$, insufficient for real-time drone navigation.

2.2.2 Onboard LiDAR-based Localization

These methods utilize LiDAR sensors mounted on drones to gather environmental data, which is subsequently aligned with infrastructure-provided references (e.g., 3D point maps) to estimate the drone's spatial pose. Among point cloud registration techniques, ICP[46] remains the most commonly employed, with various adaptations[47]. NDT[48], on the other hand, models point distributions as Gaussians to facilitate faster alignment. While both are widely applied in robotics, ICP necessitates a substantial degree of overlap and close initial proximity between point clouds, whereas NDT also depends on well-aligned starting conditions for reliable performance. Feature-based approaches such as ICL[49], and FGR[50] extract key features and establish correspondences. However, their accuracy is often compromised by feature ambiguity, especially in environments with repetitive structures like building facades. Deep learning-based registration methods[51] perform well on synthetic datasets such as ModelNet[52] and controlled indoor scenarios. Nonetheless, their application to large-scale, unstructured outdoor point clouds remains challenging due to high computational demands and scalability limitations.

2.2.3 Multi-Modal Sensor Fusion for Drone Localization

Multi-modal sensor fusion techniques are widely adopted in the localization of drones to overcome the perceptual limitations of individual sensors[53-56]. Existing multi-modal approaches can be broadly categorized as follows: *(i) Traditional pipeline fusion methods.* Conventional approaches typically integrate dense point clouds from sensors (e.g., LiDAR) to provide spatial depth information alongside dense photometric pixel data from standard frame cameras[57-58]. However, these alignment and filtering frameworks are fundamentally designed for synchronous, dense data streams. Consequently, they cannot directly accommodate the highly asynchronous, microsecond-resolution, and spatially sparse data outputs characteristic of event cameras and mmWave radars without severe temporal discretization and information loss. *(ii) Deep learning-based fusion methods.* To handle heterogeneous sensor data, recent advancements have explored learning-based architectures for fusing mmWave radar, frame cameras, and IMUs in drone contexts[12,59]. While these methods offer promising end-to-end results, they typically require extensive, domain-specific labeled training data[32,60-61]. Furthermore, their inherent black-box nature often leads to severe performance degradation and poor generalization when encountering unseen, highly dynamic flight environments[62-63]. *(iii) Event-involved multi-modal*

fusion. Given the advantages of neuromorphic vision, emerging works have attempted to integrate event cameras with other modalities, most notably combining events with IMUs (Event-VIO) or standard RGB frames[64-66]. However, literature addressing the fusion of event cameras with structurally disparate modalities—such as mmWave radar or pre-built 3D point maps—remains extremely sparse in drone applications. The severe cross-modality gap, particularly the absence of photometric semantics in asynchronous events and the spatial resolution mismatch with radar or 3D point clouds, renders traditional feature matching and fusion algorithms ineffective. This fundamental gap necessitates the development of novel cross-modality alignment and fusion frameworks, which form a core motivation for the methodologies proposed in Chapters 3 and 4 of this thesis.

2.3 Multi-drone System-oriented Engaged Coordination Methods

Multi-drone system-oriented localization is a fundamental prerequisite for achieving coordinated autonomy and collaborative sensing. Within the low-altitude embodied agents framework, this necessitates engaged coordination, where individual agents transcend their local perceptual boundaries through dynamic interactions. Current methodologies for multi-agent localization can be categorized into decentralized collaborative SLAM and static infrastructure-based approaches, each presenting unique constraints regarding scalability and flexibility.

2.3.1 Multi-drone System-based Collaborative Sensing

The convergence of AI advancements with the manufacturing sector has ignited a surge in research exploration within multi drone system[67-68], unlocking augmented sensing capabilities and accelerated response rates during critical situations through their synchronized operational framework and collaborative synergy[69-71]. These multi drone systems are outfitted with a diverse array of sensors, spanning from cameras[72] to Lidar[53], Radar[32], IMU[73-74], acoustic sensors[5], and gas sensors[75], empowering them to undertake an extensive range of emergency sensing tasks across challenging environments like urban search and rescue operations[76-77], hazardous gas detection scenarios[78-80], and post-disaster data transmission endeavors[81-84].

The data harvested by drones is seamlessly transmitted through communication channels[85], offering valuable insights into the multi drone system's mission objectives and the real-time status of individual drones[86-87], inclusive of their dynamic motion strate-

gies[88-90]. Such real-time data exchange facilitates coordinated decision-making, ensuring efficient navigation and task completion, even in highly unpredictable and resource-constrained environments. The ability to leverage advanced AI algorithms for data fusion further enhances the precision of sensing and situational awareness, enabling drones to autonomously adapt to complex and evolving circumstances.

Nevertheless, the adoption of drones equipped with such sophisticated sensor suites entails considerable financial investment, often rendering the integration of a vast number of drone nodes within a multi-drone system economically impractical, thereby imposing constraints on multi-drone system applications. This challenge necessitates the development of cost-effective solutions that balance sensor functionality with system scalability. Approaches such as collaborative sensing, where fewer but more capable drones share data and tasks, and resource-efficient sensor fusion techniques, are actively being explored. These innovations aim to optimize multi-drone system performance while mitigating the high costs associated with large-scale deployments. Furthermore, advancements in lightweight sensor technologies and computational efficiency hold promise for lowering the overall expenses of multi-drone systems, thus paving the way for more widespread applications in emergency response and industrial settings.

2.3.2 Collaborative SLAM-based Multi-drone System Localization

Collaborative SLAM (Simultaneous Localization and Mapping) is a technique where multiple robots or drones work together to simultaneously localize themselves and build a shared map of their environment. In this approach, each robot generates a partial map based on its own sensor data, such as visual, Lidar, or inertial measurements. These partial maps are then shared and integrated by the multi-drone system to create a global map, enhancing the accuracy and efficiency of the mapping process. This collaboration allows robots to cover larger areas more quickly and reliably than a single robot, making Collaborative SLAM highly beneficial in applications like exploration and surveillance.

One example of this approach is SwarmMap[21], a framework for multi-robot systems that enables real-time map-building and sharing. This system improves the efficiency of navigation tasks and increases the accuracy of the maps by distributing the computational load and allowing robots to complement each other's sensor coverage. This capability is particularly useful in environments that are large, dynamic, or difficult to access, where a single robot might struggle to gather sufficient data or maintain its localization.

Despite its advantages, Collaborative SLAM faces several challenges. The primary

limitation is the high cost of the sensors required for accurate mapping and localization. Advanced sensors such as Lidar, high-resolution cameras, and IMUs are necessary to achieve high-quality results, but these sensors can be expensive, limiting the number of robots that can be deployed in a multi drone system. This makes large-scale deployment economically impractical for many applications[91].

In addition to the cost, the communication and computational overhead involved in sharing and merging map data can be significant. drone need to continuously exchange their sensor data and map updates, which can be bandwidth-intensive and introduce delays, especially in environments with limited communication infrastructure. As the size of the multi drone system increases, the challenges of synchronizing the data and maintaining consistency across the entire map also grow, complicating the system and potentially leading to inaccuracies in the global map. These challenges make scaling Collaborative SLAM to large multi-drone systems difficult and raise concerns about its real-time performance in large, complex environments.

2.3.3 Infrastructure-based Multi-drone System Localization

Various approaches integrate infrastructure to enhance the localization of multi drone systems, leveraging external resources to improve accuracy, efficiency, and scalability. (i) One category directly utilizes infrastructure to determine the precise location of drones, such as satellite-based navigation systems (e.g., GPS, GNSS). These methods provide global coverage and high accuracy under favorable conditions but can suffer from signal degradation in environments with obstructions, such as urban canyons or indoor settings. (ii) Another category leverages infrastructure to support drone localization more interactively. This can involve fusing onboard sensor data with infrastructure-based information, such as pre-mapped environmental features, to refine location estimation. Additionally, offloading computationally intensive tasks—such as SLAM, state estimation, or trajectory optimization—to infrastructure-based computing resources, such as edge servers, can significantly reduce onboard processing demands. This offloading enhances real-time performance and enables drones to operate with more sophisticated algorithms that would otherwise be computationally prohibitive. For instance,[6] introduced Edge-SLAM, which utilizes edge computing resources to offload SLAM computations from drones, improving their efficiency and extending operational endurance. Such methods demonstrate how infrastructure can complement drone localization, particularly in large-scale deployments where onboard computational limitations are a bottleneck.

However, despite their advantages, infrastructure-dependent approaches impose inherent constraints on multi-drone systems. The effectiveness of these methods is tightly coupled with the availability and reliability of external infrastructure, restricting operations in remote, infrastructure-sparse, or adversarial environments. This dependency raises challenges related to scalability, robustness, and autonomy, motivating research into hybrid or infrastructure-independent localization solutions that can enable multi-drone systems to operate reliably in diverse and dynamic conditions.

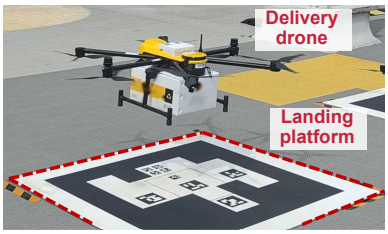
2.4 Summary

By synthesizing the existing methodologies in autonomous drone localization, this chapter identifies a critical research gap in the current low-altitude embodied agents landscape. While conventional sensing paradigms have made significant strides, they remain constrained by a fundamental "trilemma": perceptual ambiguity in extreme environments, computational latency in high-dynamic maneuvers, and coordination rigidity in large-scale multi-drone systems. Specifically, the limitations of discrete sampling in frame-based vision and the spatial tethering of static infrastructure hinder the realization of truly resilient enacted aerial agents. Furthermore, the lack of engaged symbiotic interactions between heterogeneous platforms restricts the scalability of multi-drone systems in wide-area operations. To surmount these challenges, the subsequent chapters of this thesis will systematically introduce the bio-inspired low-altitude embodied agents framework for aerial agents.

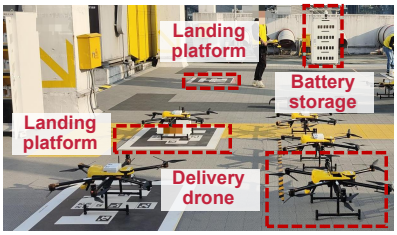
CHAPTER 3 EXTERNAL INFORMATION-BASED ENACTED PERCEPTION METHOD

Building upon the concept of "enacted perception" defined in Chapter 1, this chapter introduces mmE-Loc, a ground-based localization method that physically realizes this paradigm through external infrastructure. To surmount the pervasive challenges of GNSS-denied environments and motion blur during high-risk terminal maneuvers such as takeoff and landing, mmE-Loc integrates event cameras with mmWave radar. By synergizing the microsecond-level temporal consistency of event streams with the spatial complementarity of active radar, the proposed approach effectively reconfigures ground infrastructure into a high-fidelity external information source. This cross-modal integration achieves centimeter-level precision and ultra-low latency, thereby directly providing a perception-action feedback loop to optimize the drone's underlying flight control.

For precise, efficient, and safe drone landings, ground platforms should accurately locate descending drones in real time and guide them to designated spots. While mmWave sensing combined with cameras improves localization accuracy, the lower sampling frequency of traditional frame cameras compared to mmWave radar creates bottlenecks in system throughput. This work upgrades a traditional frame camera with an event camera, a novel sensor that harmonizes in sampling frequency with mmWave radar within a ground platform setup, and introduces mmE-Loc, a high-precision, low-latency ground localization system designed for precise drone landings. To fully exploit the *temporal consistency* and *spatial complementarity* between these two modalities, this work proposes two innovative modules: *(i)* the Consistency-instructed Collaborative Tracking module, which further leverages the drone's physical knowledge of periodic micro-motions and structure for accurate measurements extraction, and *(ii)* the Graph-informed Adaptive Joint Optimization module, which integrates drone motion information for efficient sensor fusion and drone localization. Extensive experiments (30+ hours) demonstrate that mmE-Loc attains $0.083m$ localization accuracy and $5.12ms$ end-to-end latency, outperforming four state-of-the-art methods by over 48% and 62%, respectively.



(a) Landing of the drone



(b) Real-world delivery drone airport

Sensor type	2D imaging	Depth sensing	Output rate compatible with flight controller (Over 150Hz)
Frame camera	✔	✗	✗
mmWave radar	✗	✔	✔
mmWave radar + Frame camera	✔	✔	✗
Event camera	✔	✗	✔
mmWave radar + Event camera	✔	✔	✔

(c) Sensing capability of different sensor type

Figure 3.1 Snapshot of drone landing phase, delivery drone airport, and performance of different sensors. (a) A delivery drone lands on the landing platform. (b) The real-world drone airport is equipped with multiple drones for package delivery. (c) Integrating mmWave radar with an event camera combines reliable depth sensing and 2D imaging at ultra-high sampling frequencies. This solution achieves high spatio-temporal resolution 2D sensing and precise depth sensing for drone ground localization, while maintaining full compatibility with flight controllers by operating at update frequencies exceeding 150Hz.

3.1 Introduction

Projected to soar to a $1 trillion market by 2040[92], the drone-driven low-altitude economy is transforming sectors with revolutionary applications such as on-demand delivery[71], meticulous industrial inspections[21], and rapid relief-and-rescue[90,93]. Of paramount importance within this burgeoning sector is the *landing phase*, where ground platforms locate drones descending from below 10 meters and guide them to accurately land at designated spots (Figure 3.1(a))[5,35]. Situated near populated and commercial zones, these operations emphasize safety and reliability: the research with a leading drone delivery company reveals that a landing bias of just $10cm$ will result in drones missing charging ports, damaging delivery targets, or even posing risks to people[94]. Such inaccuracies undermine the operational efficiency of this rapidly growing sector, potentially leading to severe consequences.

Widely adopted and straightforward approaches involve installing cameras at the center or edges of drone landing pads and employing computer vision algorithms for drone localization[95-97]. However, traditional frame cameras' Achilles heel is capturing only 2D images without depth information, leading to scale uncertainty that limits the 3D local-

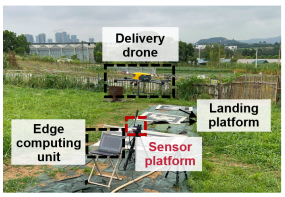
ization accuracy[98-100]. To address this shortcoming, current practices have incorporated mmWave sensing to provide the lacking depth information for better localization accuracy and reliability in various conditions[16-17,28,32,101-103].

Albeit inspiring, the benchmark study with a world-leading drone delivery company in landing scenarios (Figure 3.1(b)) reveals another critical drawback (Figure 3.1(c)): the long exposure times of frame cameras ($> 20ms$) prevent their sampling rates from matching the high frequency of mmWave radars (e.g., $200Hz$). This limitation creates system efficiency and throughput bottlenecks, restricting drone location updates to $< 50Hz$. In contrast, drone flight controllers typically require location input rates $> 150Hz$ to precisely adjust the drone's flight attitude for safe landing[16,104]. Therefore, most of the time, drone flight controllers can only adjust the flight attitude based on uncalibrated coarse location updates from mmWave radar or onboard IMU. However, during the landing phase, precise attitude control is crucial. Such coarse updates may lead to a high failure rate in landing, potentially resulting in serious accidents or injuries. The inefficiency of the system originates from the inherent physical limitations of conventional frame cameras and cannot be easily solved by software solutions.

Upgrade frame camera to event camera. Event cameras are bio-inspired sensors that report pixel-wise intensity changes with *ms*-level resolution[65,105], capturing high-speed motions without blurring[106], ideal for fast-tracking tasks[107]. Event cameras offer *ms*-level sampling latency, which harmonizes exceptionally with the high sampling frequencies of mmWave radar. Their 2D imaging capability also complements radars' limited spatial resolution, similar to how traditional frame cameras operate. Such *temporal-consistency* and *spatial-complementarity* across event camera and mmWave radar inspire us to upgrade the frame camera with an event camera, pairing it with the mmWave radar to enable accurate and low-latency drone localization.

Proposed work. Following the above insight, this thesis presents **mmE-Loc**, the first active, high-precision, and low-latency landing drone ground localization system that enhances mmWave radar functionality with event cameras. mmE-Loc works in scenarios where urban canyon environments degrade the accuracy of GPS or RTK systems as altitude decreases, rendering them nearly ineffective for the landing phase. With mmE-Loc, drones achieve reliable localization even under challenging conditions (e.g., weak illumination), ensuring stable and efficient landing.

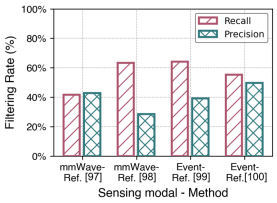
However, the benchmark study at a real-world drone delivery airport (Figure 3.2(a))



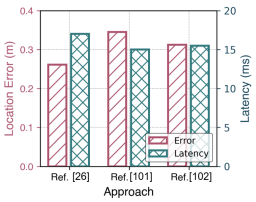
(a) Localization of the drone

Sensor Modal	Data Generating Speed	
	Background	Drone
Event Camera (*e/ms*)	764 (25.3%)	2253 (74.7%)
mmWave Radar (*#/ms*)	22 (71.5%)	5 (18.5%)

(b) Sensitivity to background noise



(c) Noise filtering rate



(d) Localization accuracy-latency

Figure 3.2 Benchmark study on delivery drone localization. (a) This work conducts the benchmark study at a real-world drone delivery airport. (b) Both the event camera and mmWave radar are sensitive to environmental variations, leading to significant sensing noise. (c) Comparison of noise filtering performance in terms of Recall and Precision. Existing algorithms rely solely on either mmWave radar or event cameras, failing to exploit their cross-modal correlations, which limits noise filtering capability. (d) These algorithms are prone to cumulative drift and experience considerable delays.

highlights several challenges that have been solved in making **mmE-Loc** a viable system outdoors: (*i*) *How to accurately extract drone-related measurements* given the immense noisy output of event cameras and mmWave radars, which also lack inherent drone semantic information and differ greatly in dimension and pattern? Both modalities are sensitive to environmental variations (e.g., changes in lighting conditions), as shown in Figure 3.2(b). Existing algorithms[108-111] are typically designed for single-modality, resulting in low noise filtering rates (recall and precision < 65% in Figure 3.2(c)). (*ii*) *How to efficiently fuse event camera and mmWave readings* that are heterogeneous in measurement precision, scale, and density? Existing EKF or PF-based approaches[25,112-113], suffer from cumulative drift errors, making them insufficient for precise localization (Figure 3.2(d)). (*iii*) *How to optimize the efficiency of the fusion algorithm* to achieve high-frequency drone ground localization, given the limited computational resources on landing platforms? Existing methods experience significant processing delays, rendering them unsuitable for low-latency localization tasks (Figure 3.2(d))[25,112-113].

To solve the above challenges, the design and implementation of mmE-Loc excel in the three aspects of drone ground localization:

- *On system architecture front.* Upgrading the frame camera to an event camera with *ms*-level latency to pair with the mmWave radar, mmE-Loc improves accuracy and efficiency of drone ground localization at the data source level. The system architecture tightly integrates both modalities, from early-stage noise filtering and drone detection to later-stage fusion and optimization, fully leveraging the advantages of both sensors (§3.2).
- *On system algorithm front.* This work introduces a Consistency-instructed Collaborative Tracking (*CCT*) module, which leverages the drone's physical knowl-

edge of periodic micro-motions and structure, along with cross-modal *temporal-consistency*, to filter environment-induced noise and achieve accurate drone detection (§3.3). This work then presents a Graph-informed Adaptive Joint Optimization (*GAJO*) module, which fuses *spatial-complementarity* with a novel factor graph to enhance the accuracy of drone ground localization, while incorporating drone motion information as a constraint to improve efficiency, resulting in a trajectory with minimal bias and low cumulative drift (§3.4).

- *On system implementation front.* This work further analyzes the sources of latency and proposes an incremental optimization method to further improve the efficiency of the *GAJO* algorithm. This approach allows *GAJO* to dynamically optimize a set of locations, maintaining accuracy while reducing latency (§3.5).

This thesis fully implements mmE-Loc using a COTS event camera and mmWave radar. Over 30+ hours of indoor and outdoor experiments under various drone flight conditions assess its localization accuracy and end-to-end latency performance against four SOTA methods. mmE-Loc achieves an average localization accuracy of $0.083m$ and latency of $5.12ms$, surpassing baselines by >48% and >62%, respectively, and showing minimal sensitivity to drone type and environment. This thesis also deploys mmE-Loc at a real-world drone delivery airport (Figure 3.2(a)) for 10+ hours, demonstrating its practicality for commercial-level drone landing requirements.

In summary, this chapter makes the following contributions.

1. This work explores a novel sensor configuration, event camera plus mmWave radar, that harmonizes ultra-high sampling frequencies and proposes mmE-Loc, a ground localization system for drone landings that delivers cm accuracy and ms latency.
2. This work presents CCT, which leverages *temporal consistency* and the drone's physical knowledge of periodic micro-motions and structure for precise drone detection; and $GAJO$, which employs *spatial complementarity* with a novel factor graph, along with drone motion information to enhance localization.
3. This work implements and extensively evaluates mmE-Loc by comparing it with four SOTA methods, showing its effectiveness. This work also deploys mmE-Loc in a real-world drone delivery airport, demonstrating the feasibility of mmE-Loc.

The remaining sections of this chapter are organized as follows: Section 3.2 provides an overview of mmE-Loc. Sections 3.3 and 3.4 describe the consistency-instructed collaborative tracking method and the graph-informed adaptive joint optimization method,

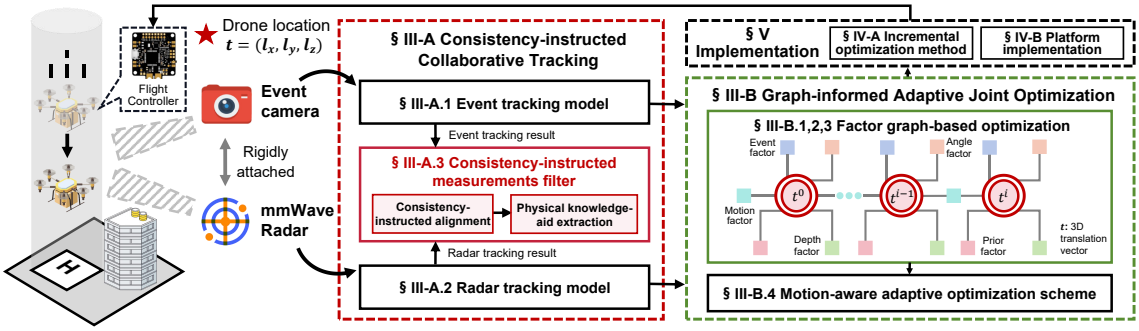


Figure 3.3 System architecture of mmE-Loc. mmE-Loc consists of two main components: Consistency-instructed Collaborative Tracking and Graph-informed Adaptive Joint Optimization.

which efficiently fuse event camera and mmWave radar data. Section 3.5 presents system implementation and evaluation. Section 3.6 discusses a case study at a real-world drone airport. Section 3.7 examines key design choices, and Section 3.8 concludes the chapter.

3.2 System Overview

mmE-Loc integrates an event camera with mmWave radar for accurate, low-latency drone ground localization, enabling rapid position adjustments for precise landing. For safety-critical commercial operations, mmE-Loc can complement RTK or visual markers to ensure precise landing. This section formally defines the problem mmE-Loc addresses and provides an overview of the system design.

3.2.1 Problem Formulation

In this section, this work illustrates key variables in mmE-Loc and introduces the system's inputs and outputs.

Reference systems. There are four reference (a.k.a. , coordinate) systems in mmE-Loc: (i) the Event camera reference system $\mathtt{E}$; (ii) the Radar reference system $\mathtt{R}$; (iii) the Object reference system $\mathtt{O}$; (iv) the Drone reference system $\mathtt{D}$. Note that a drone can be considered an object. For clarity, before an object is identified as a drone, this work utilizes $\mathtt{O}$. Once confirmed as a drone, this work uses $\mathtt{D}$ for the drone and continues using $\mathtt{O}$ for other objects. Throughout the operation of the system, $\mathtt{E}$ and $\mathtt{R}$ remain stationary and are rigidly attached, while $\mathtt{O}$ and $\mathtt{D}$ change in accordance with the movement of the object and the drone, respectively. The transformation from $\mathtt{R}$ to $\mathtt{E}$ can be readily obtained from calibration[114].

Goal of mmE-Loc. The goal of mmE-Loc is to determine the 3D location of the drone, defined as $t_{\mathtt{ED}}$, the translation from coordinate system $\mathtt{D}$ to $\mathtt{E}$. Specifically, mmE-

Loc optimizes and reports the 3D location of the drone (l_x, l_y, l_z) at each timestamp i with input of event stream and radar sample. t_{ED} and (l_x, l_y, l_z) are equivalent representations of the drone's location and can be inter-converted with Rodrigues' formula[115]. The former representation is adopted in the paper, as it is commonly used in flight control systems.

3.2.2 Overview

As illustrated in the Figure 3.3, mmE-Loc comprises two key modules and a task-oriented implementation:

• The *CCT* (**C**onsistency-instructed **C**ollaborative **T**racking) for event camera and mmWave radar noise filtering, drone detection, and preliminary localization of the drone. This module utilizes time-synchronized event streams and mmWave radar measurements as inputs. The *radar tracking model* processes radar measurements to generate a sparse 3D point cloud. Meanwhile, the *event tracking model* takes into the stream of asynchronous events for event filtering, drone detection, and tracking. Finally, *consistency-instructed measurements filter* aligns the outputs of both tracking models by leveraging *temporal-consistency* between the two modalities. It then utilizes the drone's physical knowledge to extract drone-specific measurements and achieve drone preliminary localization. Specifically, this method exploits two key characteristics of the landing drone: (*i*) the high-frequency event bursts caused by the rapid micro-motions of the propellers, and (*ii*) the drone's symmetrical structure causes these events to exhibit axial symmetry, both of which serve as distinctive signatures for identifying the drone.

• The *GAJO* (**G**raph-informed **A**daptive **J**oint **O**ptimization) for fine and low-latency localization and trajectory optimization of the drone. Based on the operational principles of two sensors and their respective noise distributions, *GAJO* incorporates a meticulously designed *factor graph-based optimization* method. This module employs the *spatial-complementarity* from both modalities to unleash the potential of event camera and mmWave radar in drone ground localization. Specifically, *GAJO* jointly fuses the preliminary location estimation from the *event tracking model* and the *radar tracking model* and refines them. To reduce latency, this work thoroughly analyzes its underlying sources and introduces a motion-aware adaptive optimization scheme. By exploiting the drone's inherent motion information, this method triggers optimization at appropriate times and adaptively adjusts the optimization window size, achieving precise localization with *ms*-level latency.

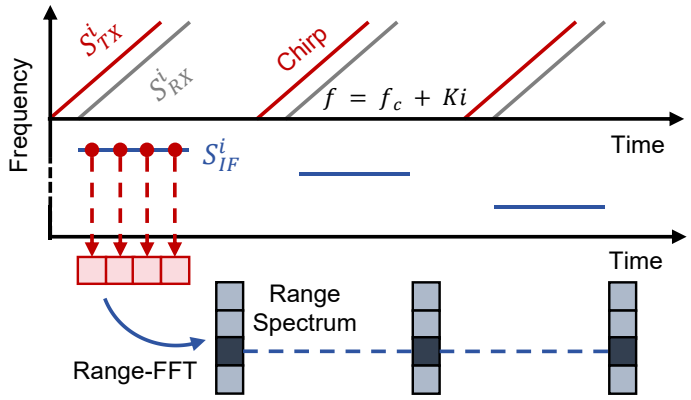


Figure 3.4 Distance calculation by frequency difference.

3.3 Consistency-instructed Collaborative Tracking

The mmWave radar is prone to signal multipath effects, leading to inaccurate point cloud data. Meanwhile, the event camera captures per-pixel brightness changes asynchronously, which are frequently influenced by non-drone factors such as shadows. However, the absence of intrinsic drone semantic information, combined with significant differences in dimension and patterns between these two modalities, presents challenges for noise filtering. This results in drone detection bottlenecks, which reduce the efficiency and accuracy of localization. Therefore, in this part, this work focuses on enhancing noise filtering and drone detection, while providing preliminary localization of the drone.

To address this challenge, this work examines the operational principles of both sensors. The design is guided by two key observations: *(i)Event cameras and mmWave radar exhibit temporal consistency while operating via distinct mechanisms.* Both maintain *ms*-level latency, with event cameras immune to multipath effects and mmWave radar unaffected by changes in brightness. *(ii) Drones display periodic micro-motion features (e.g., propeller rotation) and a symmetrical structure*, providing stable and distinctive cues. The high-speed rotation of propellers induces rapid light changes, generating numerous events, which exhibit axial symmetry due to the drone's structure.

To realize this idea, this work designs *CCT*, a lightweight cross-modal drone detector and tracker. *CCT* includes three components: (*i*) a radar tracking model (§3.3.1) that provides sparse point clouds encoding the distance and direction of objects; (*ii*) an event tracking model (§3.3.2) for event filtering, detection, and tracking of objects; (*iii*) a consistency-instructed measurements filter (§3.3.3) that leverages temporal consistency between modalities and the drone's periodic micro-motion features to extract drone-related events and point clouds, enabling preliminary localization.

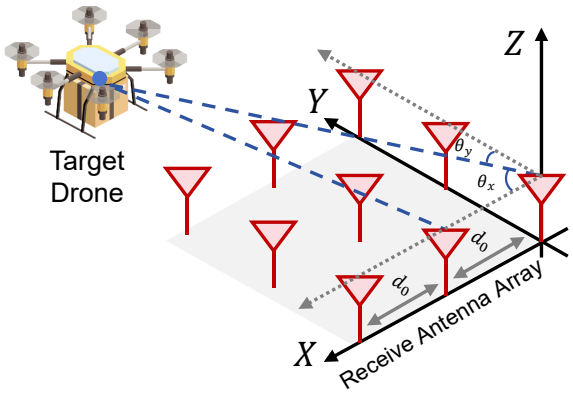


Figure 3.5 Direction calculation by phase difference.

3.3.1 Radar Tracking Model

In this part, this work calculates the distance D and direction vector $\vec{v}$ between the radar and objects, along with a preliminary estimate of the object's location.

Distance calculation. As shown in Figure 3.4, the frequency difference between the transmitted (TX) and received (RX) signals indicates the signal propagation time, revealing the distance between the object and the radar. Denoting D^i as the distance at time i, TX and RX signals as:

$$S^i_{TX} = \exp\left[j\left(2\pi f_c i + \pi K i^2\right)\right], S^i_{RX} = \alpha S_{TX}\left[i - \frac{2D^i}{c}\right], \tag{3.1}$$

where α denotes the attenuation rate, f_c is the initial frequency, K represents the chirp slope of FMCW signal, and c stands for light speed. The TX and RX signals from Eqn.(3.1) undergo mixing and a low-pass filter (LPF) to extract the intermediate frequency signal (IF signal) $s(t)$:

$$S^i_{IF} = LPF(S^{i*}_{TX} S^i_{RX}) \approx \alpha \exp\left[j2\pi\left(\frac{2KD^i}{c}\right) i\right]. \tag{3.2}$$

The frequency value f_{IF} within S^i_{IF} encapsulates distance information. After the Range-FFT operation of S^i_{IF}, f_{IF} is extracted, facilitating distance calculation as

$$D^i = \frac{c f_{IF}}{2K}. \tag{3.3}$$

Direction calculation. Using a fixed antenna array, the mmWave radar determines the object's direction by employing two orthogonal linear arrays. As depicted in Figure 3.5, each linear array captures an Angle of Arrival (AoA), calculated from the phase difference between adjacent antennas spaced apart by d as $cos\theta = \Delta\phi\lambda/2\pi d$, where θ represents AoA, λ denotes the wavelength and $\Delta\phi$ indicates the phase difference. With two orthogonal arrays, the radar obtains two AoAs, θ_x and θ_y. The unit vector indicating

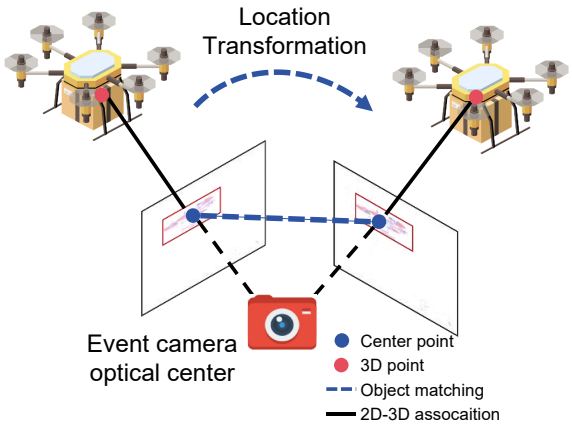


Figure 3.6 Event tracking model.

the object's direction i is given by

$$\vec{v}^i = [\cos\theta_x \cos\theta_y \sqrt{1 - \cos^2\theta_x - \cos^2\theta_y}]^{\mathrm{T}}. \tag{3.4}$$

Using the distance and angle information obtained above, along with the spatial relationship between the radar and the event camera, this work can determine the preliminary 3D location estimation of the object in E as $P_E = D\vec{v} + t_{ER}$. This work then leverages the radar for object 3D location tracking, estimating translation t_{EO} of object at time i:

$$t^i_{\mathrm{EO}} = t^{i-1}_{\mathrm{EO}} + U^i_{\mathrm{E}} + w^i + w^{i-1}, U^i_{\mathrm{E}} = \left(P^i_{\mathrm{E}} - P^{i-1}_{\mathrm{E}}\right), \tag{3.5}$$

where w_i and w_{i-1} signify measurement noise.

Drawback of mmWave radar. Although mmWave radars are effective for estimating object depth along the radial direction, their accuracy degrades when capturing motion in the horizontal and vertical (tangential) directions[116]. To address this limitation, this work integrates an event camera, which features comparable latency but relies on a fundamentally different sensing mechanism. With its high spatial and temporal resolution, the event camera enhances object detection and provides precise motion information in the tangential plane, effectively complementing the mmWave radar.

3.3.2 Event Tracking Model

In this part, this work demonstrates the process of noise filtering from a stream of asynchronous events, and how to detect and track objects with the filtered events, as depicted in Figure 3.6. As illustrated in Figure 3.7, unlike conventional frame-based cameras that rely on a global shutter to capture images at fixed intervals, event cameras asynchronously record per-pixel intensity changes with μs-level temporal resolution. This enables high-speed motion capture without motion blur, but also introduces challenges in

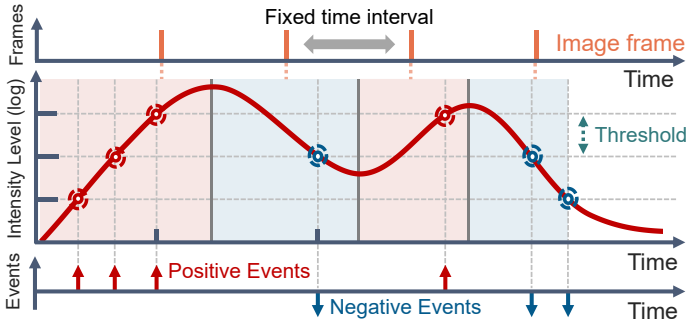


Figure 3.7 Illustration of synchronous images and asynchronous events. Frame cameras capture images at fixed intervals using a global shutter. In contrast, each pixel in an event camera responds independently, generating events whenever intensity changes exceed a threshold.

noise suppression and object detection.

Similarity-informed event filtering. Event cameras are prone to noise from transistor circuits and other non-idealities, requiring pre-processing filtering. For the i^{th} event $e^i_{(x,y)}$ with the timestamp $t^i_{(x,y)}$, this work assesses the timestamp ($t^i_{n(x,y)}$) of the most recent neighboring event in all directions. Events with a time difference less than the threshold T_n are retained, indicating object activity, while those exceeding it are discarded as noise (Figure 3.8(b), Figure 3.8(c)). This work utilizes the Surface of Active Events (SAE)[117] to manage events, mapping coordinates (x, y) to timestamps (t_l, t_r). Upon a new event's arrival, t_l updates accordingly, and t_r updates only if the previous event at the same location occurred outside the time window T_k or had a different polarity. Events that update the value of t_r are retained. The event stream, segregated by polarity, is processed with distinct SAEs. This method ensures precise spatial-temporal representation, reducing events and conserving computational resources.

Filter-based detection and tracking. This work employs a grid-based method to cluster events to facilitate object detection. The camera's field of view is partitioned into elementary cells sized $c_w \times c_h$. For each cell, this work compares the event count within a specified time interval ($c_{\Delta t}$) to an activation threshold c_{thres}. Cells surpassing c_{thres} are marked as active and connected to form clusters, serving as object detection results, including those generated by the drone. For tracking, this work deploys Kalman filter-based trackers with a constant velocity motion model, which provides low-latency estimates. Specifically, the tracker predicts the current state of the object and associates it with the input cluster that has the largest Intersection over Union (IoU). The selected input cluster then updates the tracker state, resulting in bounding box estimation of moving objects.

Using bounding box proposals and a pinhole camera model with projection function

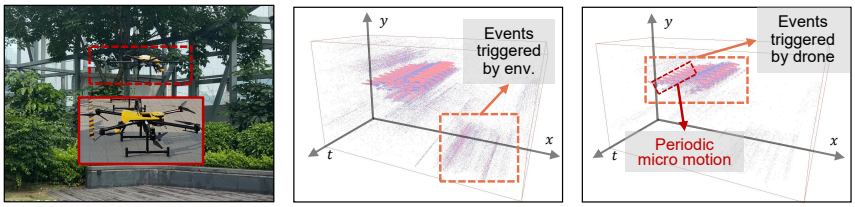


Figure 3.8 Event filtering performance. mmE-Loc filters event noise via neighborhood similarity and manages events with SAEs, ensuring precise spatio-temporal representation while reducing event load and computation. (a) RGB frame. (b) Origin event stream. (c) Filtered event stream.

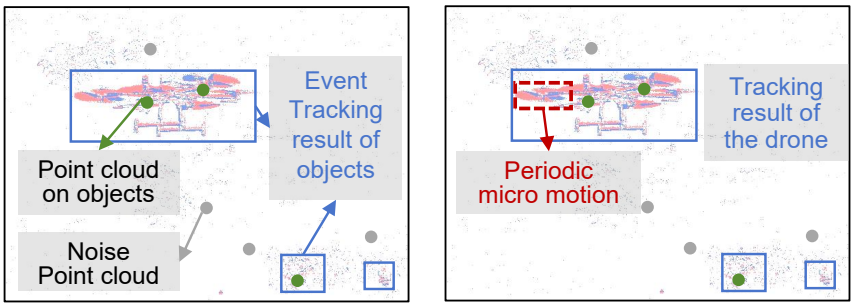


Figure 3.9 Consistency-instructed measurements filter. (a) Leveraging the temporal consistency of event camera and mmWave radar, this work aligns the outputs of two sensors, enabling the detection of the drone and other objects. (b) The rotation of drone's propellers induces light intensity changes, generating numerous events with elliptical patterns. The drone's symmetrical structure yields symmetric event patterns, providing a key feature for distinguishing it from other objects.

π, this work estimates preliminary 3D locations of objects. Specifically, π transforms a 3D point $\mathbf{X}_{\mathrm{E}} = [X_{\mathrm{E}}, Y_{\mathrm{E}}, Z_{\mathrm{E}}]^T$ into a 2D pixel x in image plane as:

$$x = \pi\left(\mathbf{X}_{\mathrm{E}}\right) = [f_x \frac{X_{\mathrm{E}}}{Z_{\mathrm{E}}} + c_x, f_y \frac{Y_{\mathrm{E}}}{Z_{\mathrm{E}}} + c_y]^T, \tag{3.6}$$

where $[f_x, f_y]^T$ is focal length of event camera, and $[c_x, c_y]^T$ denotes principal point, both being intrinsic parameters.

Then, the object's preliminary location at time i is estimated using the center point of the bounding box proposal x^i as:

$$x^i = \pi(\mathbf{X}^i_{\mathrm{E}}) + v^i = \pi(\mathbf{X}^i_{\mathrm{O}} + t^i_{\mathrm{EO}}) + v^i, \tag{3.7}$$

where $\mathbf{X}^i_{\mathrm{O}}$ represents the corresponding 3D point of center point x^i in the object reference O, v^i denotes the random noise. When extracting center points from bounding box proposals, this work first undistorts their coordinates.

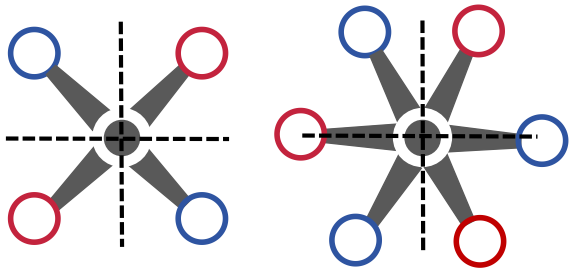


Figure 3.10 The structure of both the quadcopter and hexacopter drones features propellers arranged in axial symmetry. (a) Propeller arrangement of a quadcopter drone. (b) Propeller arrangement of a hexacopter drone.

3.3.3 Consistency-instructed Measurements Filter

The *event tracking model* detects drones and other objects causing light variations, such as indicator lights or shadows, and needs to distinguish drone-induced events from background noise. Similarly, the *radar tracking model* produces a 3D point cloud containing valid reflections and multipath artifacts, requiring extraction of drone-specific points.

Consistency-instructed alignment. Utilizing the *temporal-consistency* from the event camera and mmWave radar, and their distinct mechanisms respond to dynamic objects, this work filters event camera results affected by lighting variations on stationary objects and vice versa for radar points influenced by multipath effects. Specifically, this work aligns temporal-synchronized radar points to event bounding boxes using spatial relationships of event camera and mmWave radar (Figure 3.9(a)). Using the event camera's projection function Eqn.(3.6), this work determines that the object's location lies along the ray from the optical center through the bounding box center. The system then identifies the nearest radar points along this ray to isolate the object-associated points. If no point is found, the bounding box is considered noise and discarded.

Drone physical knowledge-aid extraction. Since each platform supports one drone landing at a time, this work needs to identify features of the landing drone and use them to extract landing drone-specific measurements from the aligned tracking results. The findings reveal three distinctive features for landing drones: *(i)* periodic micro-motions (e.g., propeller rotation) observed as temporal oscillations in the event stream; *(ii)* structural event patterns (e.g., circular patterns) produced by those micro-motions; and *(iii)* symmetry-induced features (e.g., axial symmetry) arising from the drone's multi-propeller design. Together, these complementary features provide cues for reliably detecting a landing drone from surrounding objects.

(*i*) *Periodic micro motion-based extraction.* The rapid periodic micro-motions of the

propellers produce high-frequency bursts of events, leading to a substantially larger event count for drones compared to surrounding objects. Further details are provided in Appendix A. Leveraging this property, this work transforms the spatio-temporal distribution of events into a heatmap and applies statistical metrics to isolate drone-induced measurements. Specifically, within each time window $[i, i + \delta i]$, events are aggregated into a 2D histogram, where each bin corresponds to a spatial region (e.g., 5×5pixels). Bins containing propeller rotations accumulate more events due to rapid light intensity changes. Moreover, propeller motion produces bipolar events within a bin, while background motion and noise typically generate unipolar events (e.g., flying birds). This work, therefore, identifies propeller-related bins by considering both event count and positive-to-negative event ratio, selecting bins with higher counts and a more balanced polarity distribution.

(ii) *Drone structure-based extraction.* To further suppress irrelevant measurements, this work exploits both the distinct structural patterns of propeller-induced events and the inherent symmetry of the drone's propeller configuration. Specifically, for selected bins, this work analyzes their connectivity to form clusters and fits ellipses to the structure of the resulting clusters. Given the drone's propeller axial symmetry (Figure 3.10), this work evaluates the symmetry of ellipse-fitted clusters to determine whether a bounding box originates from the drone. Bounding boxes with axially symmetric clusters are retained as drone-generated.

In this way, this work associates the event tracking results with corresponding point clouds, and extracts drone tracking outputs from two models for preliminary localization, as shown in Figure 3.9(b). When multiple drones are scheduled to land, they descend sequentially. This method identifies landing drone and extracts relevant measurements.

3.4 Graph-informed Adaptive Joint Optimization

The preliminary drone location estimations from the event and radar tracking models suffer from biases. Specifically, event camera estimations face scale uncertainty, while radar estimations struggle with limited spatial resolution, scatter center drift, and accumulating drift. Additionally, estimations from different models are heterogeneous in precision, scale, and density, complicating the fusion and optimization. Therefore, in this part, this work prioritizes accurate drone ground localization and trajectory tracking.

The design is based on the insight that *the event and radar tracking models provide spatially complementary features*. The 2D imaging capability of the event camera

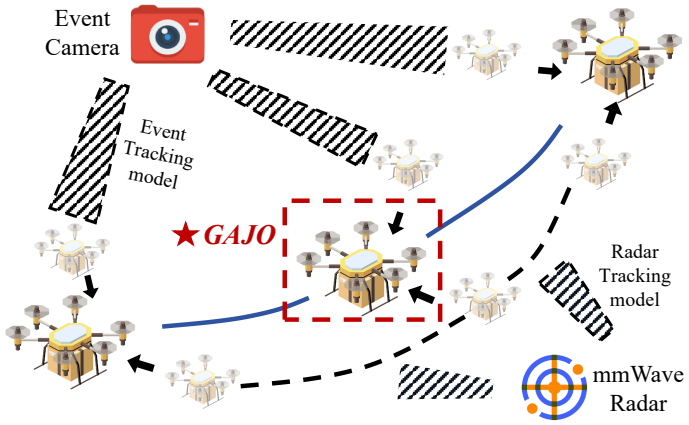


Figure 3.11 Illustration of relationship between *GAJO*, event tracking model and radar tracking model. The *GAJO* module harnesses *spatial-complementarity* of both modalities through a factor graph-based join optimization, where the event camera provides 2D imaging and mmWave radar delivers depth information.

and the depth sensing of the mmWave radar can effectively compensate for each other, as illustrated in Figure 3.11. Because both the event stream and mmWave samples are drone-related measurements, fully exploiting their *spatial complementarity* through joint optimization can enhance localization performance, leading to trajectories with reduced bias and minimized cumulative drift.

To realize this idea, this work proposes *GAJO*, a factor graph-based framework for low-latency and accurate 3D drone localization (§3.4.1). *GAJO* consists of two tightly coupled modules: (*i*) short-term inter-SAE tracking and (*ii*) long-term local location optimization, which together improve localization precision (§3.4.2). It also incorporates prior knowledge of flight dynamics to refine the trajectory. To reduce computational cost, *GAJO* leverages motion cues to evaluate localization quality, deciding when long-term optimization is needed and adjusting the optimization window size to minimize overhead.

3.4.1 Factor Graph-based Optimization

A factor graph comprises variable nodes, indicating the states to be optimized (e.g., t^i_{ED}), and factor nodes, representing the probability of certain states given a measurement result. In mmE-Loc, measurements are derived from the Event Tracking (ET) model (x^i from Eqn.(3.7)) and Radar Tracking (RT) model (D^i from Eqn.(3.3), $\vec{v}^i$ from Eqn.(3.4), and U^i_E from Eqn.(3.5)). To estimate the values of a set of variable nodes $\mathcal{X} = \{t^i_{ED} | i \in \mathcal{T}\}$ given measurements $\mathcal{Z} = \{x^i, D^i, \vec{v}^i, U^i_E | i \in \mathcal{T}\}$, *GAJO* optimizes connected factors based on maximum a posteriori estimation:

$$\hat{\mathcal{X}} = \arg\max_{\mathcal{X}} p(\mathcal{X}) \prod_{i \in \mathcal{T}} p\left(x^i, D^i, \vec{v}^i, U^i_E \mid t^i_{ED}\right), \tag{3.8}$$

where $p(\mathcal{X})$ represents the prior term, encoding the drone's flight dynamics and motion constraints. The $p\left(x^i \mid t^i_{ED}\right)$ corresponds to the likelihood of the ET model, capturing the measurement uncertainty in the SAE plane. The $p\left(D^i, \vec{v}^i, U^i_E \mid t^i_{ED}\right)$ denotes the likelihood of the RT model, which integrates distance, direction, and coarse translation to constrain the drone's 3D position. Further details are provided in Appendix B.

3.4.2 Probabilistic Representation

Inferring the drone's location requires incorporating both the prior and likelihood terms in Eqn.(3.8). This section provides a formulation of these terms.

Prior term. The prior term $p(t^i_{ED})$ models the location probability distribution of the drone at time i independent of current measurements. It is derived from a constant velocity motion model, which assumes that the drone maintains a steady speed over short time intervals. Based on this assumption, the prior location can be predicted as:

$$\vec{t}^i_{\mathrm{ED}} - t^{i-1}_{\mathrm{ED}} = t^{i-1}_{\mathrm{ED}} - t^{i-2}_{\mathrm{ED}}. \tag{3.9}$$

ET model likelihood. The likelihood $p(x^i|t^i_{ED})$ from the ET model characterizes the distribution of the detected center point given the drone's location at time i. Following common practice in tracking systems[118], the center point noise v^i is modeled as Gaussian, which has proven effective in capturing localization uncertainty. Therefore, the likelihood of the ET model is formulated as:

$$p(x^i|t^i_{ED}) \sim \mathcal{N}(\pi(\mathbf{X}^i_E), \sigma_{ET}), \tag{3.10}$$

where σ_{ET} denotes standard deviation of center point noise.

RT model likelihood. The likelihood $p(D^i \mid t^i_{ED})$, $p(\vec{v}^i \mid t^i_{ED})$, and $p(U^i_E \mid t^i_{ED})$ describe the probability distributions of the measured distance, angle, and motion given the drone's location at time i. These measurements are modeled as Gaussian distributions:

$$\begin{gathered} p(D^i \mid t^i_{ED}) \sim \mathcal{N}(||t^i_{ED}||, \sigma_D), p(\vec{v}^i \mid t^i_{ED}) \sim \mathcal{N}(\vec{v}_{t^i_{ED}}, \sigma_{\vec{v}}), \\ p(U^i_E \mid t^i_{ED}) \sim \mathcal{N}(t^i_{ED} - t^{i-1}_{ED}, \sigma_{U_E}), \end{gathered} \tag{3.11}$$

where σ_D, $\sigma_{\vec{v}}$, and σ_{U_E} denote the standard deviations of the distance, angle, and motion measurements, respectively.

3.4.3 Fusion-based Tracking

In mmE-Loc, two fusion schemes are employed for sensor fusion and optimization, as depicted in Figure 3.12. The first, short-term inter-SAE tracking, provides instantaneous

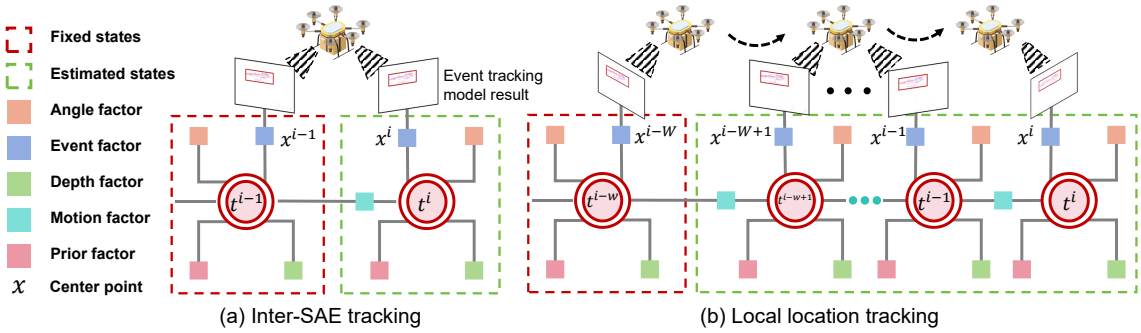


Figure 3.12 Illustration of Graph-informed Adaptive Joint Optimization. The module consists of two parts. The first part is (a) Inter-SAE Tracking, which aims to perform short-term joint optimization for low-latency output. However, the localization results from this part may accumulate errors over time. Therefore, the second part, (b) Local Location Optimization, performs long-term joint optimization by jointly considering multiple locations and associated constraints, resulting in a trajectory with reduced bias and minimized cumulative drift.

location estimates by minimizing errors across different tracking models. The second, long-term local location optimization, improves overall trajectory accuracy through joint optimization of a selected set of locations.

Short-term inter-SAE tracking. Once the measurements of the ET model and the RT model $(x^i, \boldsymbol{D}^i, \vec{v}^i, \boldsymbol{U}_E^i)$ are received, the prior factor, ET factor, and the RT factor are formulated using Eqn.(3.9), Eqn.(3.10), and Eqn.(3.11) as follows:

$$\begin{aligned} E^i_{\text{Prior}} &= -\log p\left(t^i_{ED}\right), E^i_{\text{ET}} = -\log p\left(x^i \mid t^i_{ED}\right), \\ E^i_{\text{RT}} &= -\log p\left(\boldsymbol{D}^i, \vec{v}^i, \boldsymbol{U}^i_E \mid t^i_{ED}\right). \end{aligned} \tag{3.12}$$

Building on the factors in Eqn.(3.12), the inter-SAE tracking in Figure 3.12(a) is performed to provide an instantaneous location estimate according to Eqn.(3.8) as follows:

$$\hat{t}^i_{ED} = \arg\min_{\boldsymbol{t^i_{ED}}} \left(E^i_{\text{prior}} + E^i_{\text{ET}} + E^i_{\text{RT}}\right). \tag{3.13}$$

Further details and derivations are provided in Appendix C.

Local location optimization. To effectively mitigate cumulative localization drift, this work performs a local location optimization procedure that refines the estimated positions by exploiting the temporal consistency across multiple consecutive SAEs. Specifically, this process involves the joint optimization of a set of SAE locations, denoted as $\mathcal{X} = \bigcup_{i\in\mathcal{T}} \left\{t^i_{ED}\right\}$, as illustrated in Figure 3.12(b), where $W = |\mathcal{T}|$ represents the number of SAEs included in the optimization window. The optimization problem is formulated based on Eq. (3.8) as follows:

$$\hat{\mathcal{X}} = \arg\min_{\mathcal{X}} \sum_{i\in\mathcal{T}} \left(E_i^{\text{prior}} + E_i^{\text{ET}} + E_i^{\text{RT}}\right). \tag{3.14}$$

Further details and derivations are provided in Appendix D. It is worth noting that (i) the

conditions under which local location optimization is triggered, (ii) the size of the optimization window $\mathcal{T}$ ($W = |\mathcal{T}|$), and (iii) the strategy for solving the inter-SAE tracking and local location optimization problems have a direct impact on both the latency and the accuracy of localization. To address these issues, this work enhances the efficiency of GAJO by introducing a motion-aware adaptive optimization scheme to handle (i) and (ii), and an incremental optimization method to efficiently solve (iii).

3.4.4 Motion-aware Adaptive Optimization Scheme

In this section, this work addresses two key issues: (i) determining when to trigger the local location optimization, and (ii) defining the size of the optimization window $\mathcal{T}$. To tackle these problems, this work leverages the drone's onboard motion information. Specifically, when mmE-Loc computes the drone's position, this estimated position is sent to the flight controller. Meanwhile, the flight controller typically employs an Inertial Measurement Unit (IMU), which continuously provides the drone's acceleration measurements. By integrating the acceleration over time, this work can estimate the drone's displacement between two consecutive timestamps. Using this motion information, this work combines the location estimates from mmE-Loc at time $i - j$ ($j \in \{1, ..., W\}$) to predict the drone's position at time i. If the predicted position at time i deviates from the mmE-Loc output at the same time by more than a predefined threshold Δ, the local location optimization is triggered for the location set $\mathcal{X} = \bigcup_{k \in [i-j,i]} \{t_{ED}^k\}$.

3.5 Implementation

3.5.1 Incremental Optimization Method

To efficiently solve the nonlinear least-squares estimation problems, this work formulates them within a factor graph framework and employs QR-based matrix factorization to estimate the optimal locations. Nonetheless, the conventional approach of re-linearizing and regenerating the square root information matrix R upon receiving new measurements incurs considerable computational cost, hindering real-time applicability. To overcome this limitation, this work develops an incremental optimization method that exploits the insight that new measurements predominantly affect localized regions of the factor graph. By incrementally updating R for these regions and selectively re-generating it under significant structural changes, proposed method effectively balances computational efficiency

and estimation accuracy. The detailed algorithmic steps are presented in Appendix E.

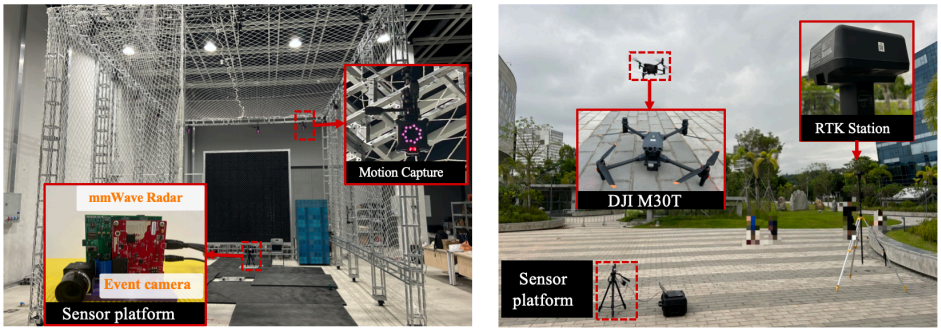
3.5.2 Platform Implementation

Sensor platform configuration. As illustrated in Figure 3.13, this work implements the sensing platform with multiple sensors, including (*i*) A Prophesee EVK4 HD evaluation kit, featuring the IMX636ES event-based vision sensor for HD event data (1280 × 720 pixels) with 47.0° FoV. (*ii*) A Texas Instruments (TI) IWR1843 board for transmitting and receiving mmWave signals within the frequency range of 76 GHz to 81 GHz with three transmitting antennas and four receiving antennas. These antennas are arranged in two linear configurations on the horizontal plane. (*iii*) An Intel D435i Depth camera for RGB image capture used in the baseline method.

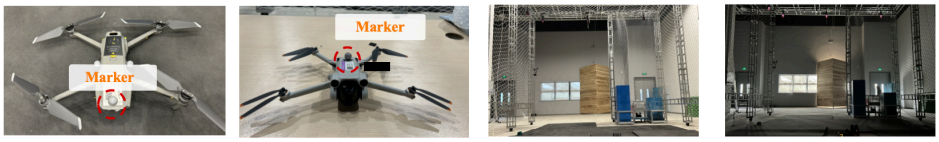
Deployment detail. During the experiments, the sensor platform is placed at the center of the landing pad. In the real-world case study conducted at an airport, the platform is positioned at the edge of the pad for safety reasons, with the distance from the platform to the pad's center being less than 1 meter. Once the drone takes off, it enters the event camera's field of view (FoV). All sensors are synchronized via the Robot Operating System (ROS). mmE-Loc runs on a PC with Ubuntu 20.04 operating system, equipped with an Intel i7 CPU, 32GB of RAM, and an NVIDIA GeForce GTX 1070 GPU. For practical deployments, the sensor platform is ideally positioned at the center of the landing pad, with its height aligned to the pad's, ensuring that the drone remains within the event camera's FoV at all times for continuous localization.

3.6 Experimental Evaluation

The evaluation of mmE-Loc is comprehensive and grounded entirely in real-world experimentation. This work begins with experimental settings in Section 3.6.1. Then, in Section 3.6.2, this work focuses on key performance metrics: localization accuracy and latency. Section 3.6.3 delves into external factors impacting mmE-Loc, such as drone characteristics and environmental conditions. Section 3.6.4 assesses the benefits of sensor fusion and the performance of various modules. Finally, in Section 5.5, this work evaluates system load, including latency, CPU usage, and memory consumption.



(a) Indoor experiment scenario (b) Outdoor experiment scenario



(c) DJI MAVIC 2 (d) DJI MINI 3 Pro (e) Normal Illumination (f) Weak Illumination

Figure 3.13 Experimental setup and scenarios of mmE-Loc. (a) A laboratory scenario with a motion capture system for ground-truth collection. (b) An outdoor scenario with RTK system for ground-truth collection. (c)-(f) Different drones with different sizes (DJI MAVIC 2 and DJI MINI 3 Pro), and different illumination of the laboratory (normal and weak), respectively.

3.6.1 Experimental Methodology

Experiment setting. This work uses a deployment setup and drone equipment that closely mimic real-world applications. Figure 3.13 shows the experimental scenarios in both an *indoor* laboratory and an *outdoor* flight test site. The setup in Figure 3.13(a) includes the event camera and mmWave radar mounted together *at the ground of the experimental area*. This work evaluates mmE-Loc on various target drones: (*i*) *DJI Mini 3 Pro*: $0.25m \times 0.36m \times 0.07m$ with unfolded propellers. (*ii*) *DJI MAVIC 2*: $0.32m \times 0.24m \times 0.08m$ with unfolded propellers. (*iii*) *DJI M30T*: $0.49m \times 0.61m \times 0.22m$ with unfolded propellers. By default, in the indoor experiments, the drone flew above the sensor platform at an altitude of 3–4 m, following a zigzag trajectory that covered an area of 2 m × 3 m. This setup ensured that the system could continuously localize the drone during its final landing phase, while the motion capture system simultaneously recorded ground-truth trajectories. In the outdoor experiments, the drone flew at an altitude of 5– 6m, covering the same 2 m × 3 m region with a similar zigzag pattern to maintain reliable localization until touchdown. The DJI Mini 3 Pro drone was used in the indoor and robustness experiments conducted under controlled conditions, whereas the DJI M30T drone was employed for the outdoor experiments. A DJI Mavic 2 drone was introduced to evaluate the system's localization performance across different drone models. This work conducts extensive experiments over *30* hours, collecting *more than 400GB* of raw data.

Ground truth. In the indoor scenario, this work uses a NOKOV motion capture

system equipped with fourteen cameras. The system covers an $8\ m \times 8\ m \times 8\ m$ area and provides localization accuracy within *1 mm*, enabling precise performance evaluation under controlled conditions. In the outdoor scenario, this work conducts experiments at an open test site with excellent GPS signal reception. A DJI D-RTK2 Real-Time Kinematic (RTK) base station is deployed to provide carrier-phase corrections to the Matrice M30T drone, ensuring high-precision positioning[119]. The RTK system achieves a localization error of *1 cm (horizontal)* and *2 cm (vertical)*. These RTK results were used as the ground truth for the outdoor experiments.

Comparative methods. This work re-implements and adapts the following state-of-the-art (SOTA) algorithms to the experimental setup for fair comparison:

- *mDrone*[25]: a SOTA point-cloud-based drone localization system using single-chip mmWave radar, originally designed for ground localization. This work adjusts the parameter settings for the radar configuration and ground-based testing. This method is chosen as a baseline to validate the effectiveness of using an event camera for 2D spatial calibration of the mmWave radar and its contribution to improving localization accuracy.
- *EventDodge-S*[112]: a SOTA **S**ingle event camera-based onboard obstacle localization system applies to obstacles with known geometries. Since the original code is not publicly available, this work re-implements its monocular 3D localization module and adapts it for *ground-based* drone localization, excluding ego-motion compensation for noise reduction.
- *EventDodge-D*[112]: a SOTA **D**ual event camera-based onboard obstacle localization system. The original method uses onboard stereo event cameras for obstacle localization, but its code is unavailable. This work implements its stereo 3D location estimation module, excluding ego-motion compensation, for *ground-based* drone localization. These two methods are chosen as baselines to validate the effectiveness of introducing the mmWave radar for providing depth information to the event camera and to assess its contribution to the overall localization accuracy.
- *milliEye*[60]: a deep learning-based multimodal localization system that integrates monocular images and mmWave point clouds. This work pre-trains the neural networks following the original design and incorporates a Kalman Filter to adapt the method for 3D *ground-based* drone localization. This method is selected as a baseline to demonstrate the significance of upgrading the frame camera to an event cam-

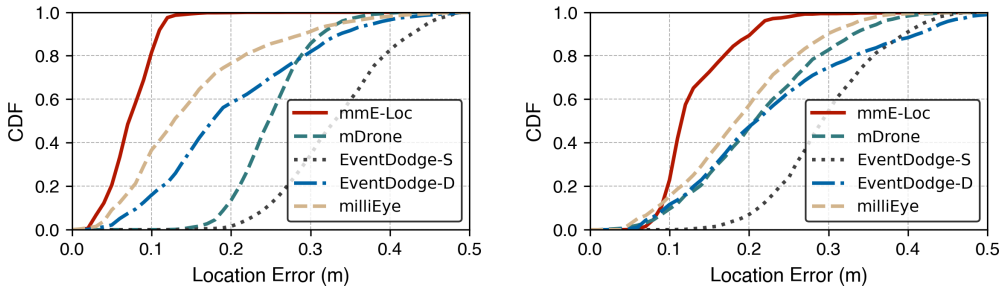
era, showing that this upgrade leads to substantial improvements in both localization accuracy and latency.

Evaluation metrics. The mmE-Loc continuously reports the drone's location estimation with low latency, ensuring accurate and responsive localization. Similar to related works, this work measures location estimation error in meters and processing latency in milliseconds, allowing for a direct comparison with existing methods in terms of both accuracy and computational efficiency.

Robustness experiments. During drone landing, localization may occur under varying conditions. To assess mmE-Loc's adaptability, this work conducts experiments across diverse scenarios, including different environments, drone models, lighting conditions (Figure 3.13a-f), and background dynamics (e.g., moving objects like balls). This work also evaluates the impact of distance, occlusions (e.g., partial obstruction of the drone in the event camera's FoV), and velocity to demonstrate robustness. It is worth noting that the robustness evaluation in this work primarily focuses on the operation scenarios of delivery drones or those performing drone-driven low-altitude economy activities. This means that the main objective is to guide cooperative drones to accurately and reliably land on the designated landing pad. *Owing to the considerable latency introduced by the frame camera's exposure and readout processes (as illustrated in Figure 3.1), milliEye is excluded from the robustness experiments.* The resulting delay, typically tens of milliseconds, leads to poor temporal responsiveness under dynamic conditions. Since the objective is to enable accurate and low-latency drone localization, Baseline-IV does not align with the intended low-latency evaluation focus.

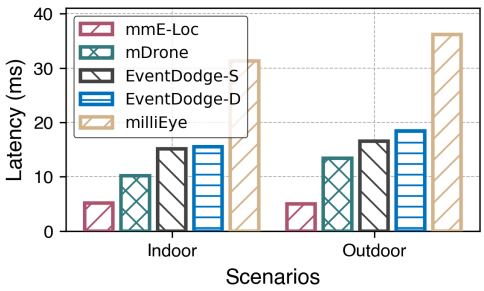
3.6.2 Overall Performance

Drone localization. Figure 3.14(a) shows the localization performance of mmE-Loc in an indoor environment using a DJI Mini 3 Pro drone, compared to four baselines. mmE-Loc achieves an average end-to-end localization error of $0.083m$, outperforming the baselines of mDrone, EventDodge-S, EventDodge-D, and milliEye with errors of $0.261m$, $0.345m$, $0.209m$, and $0.160m$, making it well-suited for landing assistance. Figure 3.14(b) presents mmE-Loc's localization performance in an outdoor setting with a DJI M30T drone. Here, mmE-Loc achieves the lowest average error of $0.135m$, outperforming the baselines by 39.2%, 56.0%, 43.8%, and 31.8%, respectively. mDrone suffers from point cloud errors caused by phase center offset, resulting in deviations from the drone's geometric center. In indoor environments, these errors are exacerbated by specu-



(a) Indoor accuracy comparison

(b) Outdoor accuracy comparison



(c) Latency comparison

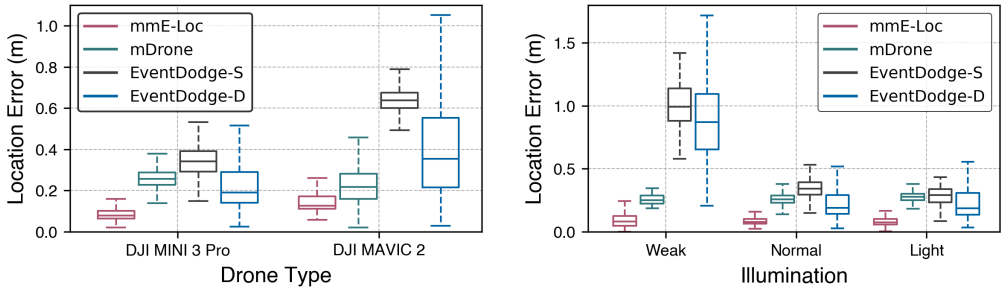
Figure 3.14 Overall performance comparison of mmE-Loc and four related works.

lar reflections, diffraction, and multi-path effects that further degrade radar measurements. EventDodge-S employs a single event camera with prior knowledge of drone geometry, while EventDodge-D uses a stereo setup with dual event cameras for depth estimation. Both approaches are vulnerable to environmental noise (e.g., birds), which can lead to inaccurate depth estimation. milliEye relies on deep learning models, which may exhibit higher errors when deployed in unseen environments due to limited generalization. In contrast, mmE-Loc significantly improves performance by leveraging the complementary strengths of mmWave radar and event cameras. It operates without the need for pre-training, enabling robust and broadly applicable localization. A more detailed analysis is provided in Appendix F.

End-to-end latency. This work evaluates end-to-end latency, including the *CCT* and *GAJO* phases. As shown in Figure 3.14(c), mmE-Loc achieves 5.12ms latency indoors, outperforming baselines of mDrone, EventDodge-S, EventDodge-D, and milliEye at 10.19ms, 15.12ms, 15.52ms, and 31.2ms, respectively. In outdoor scenarios with higher complexity, baseline latencies increase, and mmE-Loc maintains the lowest latency, outperforming them by 62.5%, 69.6%, 72.7%, and 83.3%. Baselines face increased latency due to environmental noise and more optimization parameters. mmE-Loc establishes a tight coupling between the event camera and mmWave radar. It incorporates *CCT* module

to exploit temporal consistency, drone's periodic micro-motions, and structure for drone detection. Then, *GAJO* module, combined with an incremental optimization approach, enhances spatial complementarity while minimizing system latency.

Comparison with Lidar-based method. To further demonstrate the advantages of mmE-Loc, this work conducts indoor experiments using the same setup as mmE-Loc and a DJI Mini 3 Pro drone (measuring 0.25 m × 0.36 m × 0.07 m with unfolded propellers). In this experiment, this work compares the localization accuracy and latency of mmE-Loc against a LiDAR-based method[120]. [120] leverages onboard LiDAR to detect and localize flying objects. Specifically, its detector employs 3D voxel mapping for object detection, achieving high localization accuracy and robustness to environmental variations. Furthermore, the method integrates a clustering-based tracker to suppress false positives and to estimate and predict the target's state. For a fair comparison with mmE-Loc, this work configures[120]'s observer as static, detecting and localizing a flying drone to simulate a drone landing scenario. The LiDAR used was a RoboSense Helios-16-line LiDAR with a scanning rate of 10 Hz. Both methods were executed on the same computing platform, a PC equipped with an Intel i7 CPU. When using only the detector, the LiDAR-based method achieved an average localization error of 0.071 m and an average latency of 658.82 ms. After integrating the tracker, the average localization error increased to 0.165 m, while the latency decreased to 29.74 ms. This trade-off occurs because the tracker predicts the target's position, thereby reducing latency but inevitably introducing additional estimation errors. The residual localization error also arises from the LiDAR's limited field of view, as each scan only captures the target from one side, leading to uneven point cloud sampling and a positional bias within the range of $[0, a]$, where a denotes the radius of the target's bounding sphere. Such behavior aligns well with the description in[120]. As shown by the results, although incorporating LiDAR enables a localization accuracy comparable to that of mmE-Loc, its latency remains more than 5x higher, even with tracker assistance. In realistic landing scenarios, achieving both rapid and reliable localization updates is critical. Hence, a hybrid deployment that combines the strengths of the LiDAR-based method and mmE-Loc could be a promising solution. The LiDAR-based approach can offer precise position estimation, while mmE-Loc can complement it by providing low-latency updates between LiDAR scans, jointly supporting efficient drone landing guidance.



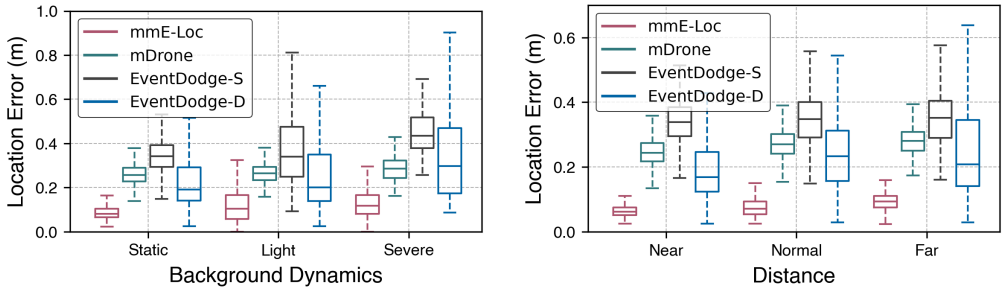
(a) Impact of drone type (b) Impact of env. & illu.

Figure 3.15 Robustness evaluation on drone type and env. & illu.

3.6.3 System Robustness Evaluation

Impact of drone type. This work evaluates the impact of different drone types under controlled indoor conditions, using a DJI Mini 3 Pro (Figure 3.13(c)) and a DJI Mavic 2 (Figure 3.13(d)). The results are presented in Figure 3.15(a). The average error results demonstrate that mmE-Loc outperforms all baselines when using the DJI Mini 3 Pro. For the DJI Mavic 2, EventDodge-S and EventDodge-D exhibit larger localization errors, primarily due to changes in the drone's geometry that impact depth estimation. mmE-Loc's error of $0.135m$ remains within an acceptable range, which outperforms all baselines with $0.222m$, $0.639m$, and $0.403m$, demonstrating its effectiveness across drone types.

Impact of environment & illumination. The performance of mmE-Loc in different environments with varying lighting conditions (Figure 3.13(e) and Figure 3.13(f)) is shown in Figure 3.15(b). The experiments were conducted in an indoor environment, where environmental variables could be precisely controlled. The experimental settings are defined as follows: Weak refers to an indoor scene with an illumination level of approximately 50 lux, Normal corresponds to about 400 lux, and Light indicates a bright indoor condition of around 1000 lux. In outdoor scenarios, the ambient illumination level during drone delivery tasks generally exceeds 300 lux, indicating that the defined lighting condition effectively encompasses the illumination range commonly experienced in real-world drone delivery environments. Although event cameras offer a high dynamic range, the events generated by drones are still influenced by illumination conditions. As lighting diminishes, EventDodge-S and EventDodge-D suffer from increased localization errors due to degraded depth estimation performance. In contrast, mDrone shows little variation in performance, as mmWave signals are inherently robust to illumination changes. In comparison, mmE-Loc sustains a relatively low average error even under low illumination, recording an average error of $0.103m$ and a maximum error of $0.27m$. This can be



(a) Impact of background (b) Impact of distance

Figure 3.16 Robustness evaluation on background and distance.

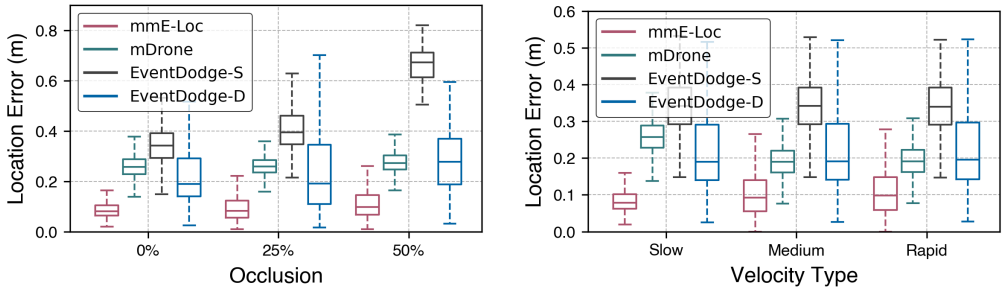
attributed to two main reasons: (*i*) The depth estimation in mmE-Loc primarily relies on the mmWave radar, which is unaffected by lighting conditions. In other words, mmE-Loc assigns greater weight to mmWave data when computing depth. (*ii*) The event camera mainly perceives the drone in the 2D spatial domain. Under weak illumination, its bounding box center may deviate from the drone's true geometric center, leading to an increase in localization error. Nevertheless, by detecting the periodic micro-motion of the drone, mmE-Loc can still produce accurate bounding boxes, enabling effective spatial calibration for the mmWave radar. As a result, even in low-light conditions, the maximum error remains within a reasonable range and still outperforms localization using mmWave radar alone. mmE-Loc's consistent performance across different lighting conditions, without requiring pre-training or prior knowledge, makes it a versatile solution.

Impact of background motion. This work further evaluates mmE-Loc under dynamic background motion, as illustrated in Figure 3.16(a). The level of background dynamics is controlled by introducing moving objects of different sizes into the scene (e.g., a tennis ball with a diameter of approximately 7 cm and a football with a diameter of approximately 20 cm). Specifically, in the Static condition, no other moving objects are present besides the drone; in the Light condition, a tennis ball is repeatedly tossed above the drone; and in the Severe condition, both a tennis ball and a football are continuously thrown across the drone's flight area. In real-world delivery scenarios, the disturbances that affect drone landings are predominantly caused by pedestrians, most commonly through object-throwing or transient movements. To emulate such realistic interference, the experimental setup is designed to cover representative forms of background motion frequently encountered during drone delivery landings. As shown in the results, all baselines exhibit increased localization errors as background motion intensifies, primarily due to radar and vision misidentifications. Under the Light and Severe conditions, EventDodge-S and

EventDodge-D are more severely affected than mDrone, since overlapping trajectories between the balls and the drone distort the event camera's bounding box generation and degrade depth estimation accuracy. In contrast, the mmWave radar in mmE-Loc remains more robust, as the drone's propeller rotation induces micro-motions at a much higher frequency than the movement of the balls, making the radar response more sensitive to the drone. Despite the strong interference, mmE-Loc maintains an average localization error of only 0.129 m in the most challenging Severe scenario. This robustness arises from its consistency-instructed measurement filter, which exploits the drone's unique periodic micro-motion and structural priors to distinguish it from other moving objects. If the background contains additional fast-moving objects (e.g., fans), mmE-Loc further analyzes the structural characteristics of the motion (whether other objects in the vicinity exhibit similar periodic micro-motions, and these objects possess symmetry) to determine whether the observed motion originates from the drone, thereby ensuring reliable tracking and localization.

Impact of distance. This work investigates the impact of drone-to-platform distance indoors with a DJI Mini 3 Pro ($0.25m \times 0.36m \times 0.07m$). Since the application scenario of mmE-Loc focuses on landing guidance when a delivery drone approaches the landing pad, the experiments are conducted under three distance settings: Near (<3 m), Normal (3 m – 6 m), and Far (>6 m). These settings correspond to typical approach phases of a drone during descent and enable a comprehensive evaluation of mmE-Loc's localization performance under different proximity conditions. Figure 3.16(b) shows the results. As distance increases, all methods show higher errors. mmE-Loc achieves an average error of $0.102m$ for far distances, outperforming other baselines. mmE-Loc combines mmWave radar for depth sensing with an event camera for high-resolution 2D imaging, effectively addressing the low spatial resolution of mmWave radar and the scale ambiguity of event cameras. Recognizing the critical need for precise landings, mmE-Loc serves as a complementary system to existing solutions. When integrated with RTK and visual markers, it enhances localization reliability and accuracy in real-world applications.

Impact of occlusion. This work validates mmE-Loc's performance with partially occluded drones. Occlusion is introduced by partially obstructing the drone within the event camera's field of view, including cases where the drone appears only partially in FoV. Figure 3.17(a) shows that with 25% occlusions, mmE-Loc maintains high performance with an average error of $0.094m$. At 50% occlusion, the average error of mmE-Loc increases



(a) Impact of occlusion (b) Impact of velocity

Figure 3.17 Robustness evaluation on occlusion and velocity.

to $0.12m$. EventDodge-S and EventDodge-D show larger errors due to incorrect depth estimation caused by occlusion. mmE-Loc harnesses the strengths of both modalities in accurately tracking the drone's location, even under partial occlusion.

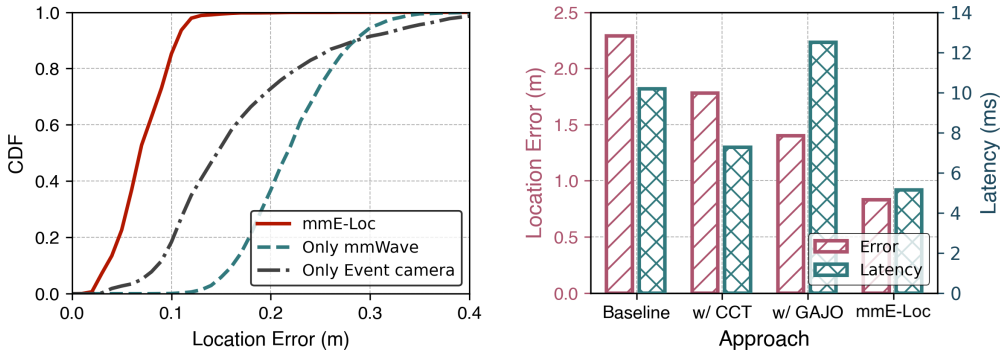
Impact of drone velocity. This work further evaluates mmE-Loc's performance under different drone velocities in Figure 3.17(b). The application scenario of mmE-Loc focuses on landing guidance when a delivery drone approaches the landing pad. During this phase, the drone typically reduces its speed to ensure a precise and stable landing. Accordingly, this work categorizes the flight velocities into three levels: slow ($v < 0.5$ m/s), medium (0.5 m/s $\leq v <$ 1 m/s), and rapid (1 m/s $\leq v <$ 1.5 m/s), corresponding to different stages of the drone's landing process. Although errors of all methods increased with speed, mmE-Loc maintained an average error of $0.11m$ even in rapid speed scenarios, outperforming baselines by 44%, 68.5%, and 52.4%. This benefit stems from the high sampling rates of both the event camera and mmWave radar, which enable more frequent observations of the drone, including both 2D imaging and depth measurements.

3.6.4 Ablation Study

This work experimentally analyzes the core components of mmE-Loc, focusing on both their individual performance and improvements they contribute to overall system.

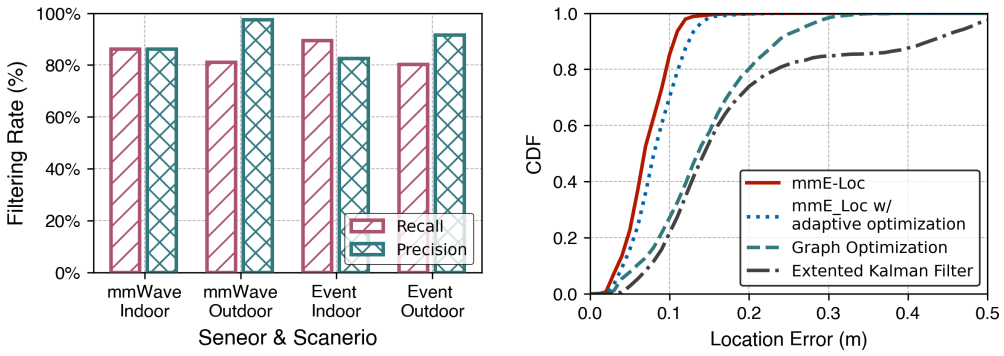
Effectiveness of multi-modal fusion. This work demonstrates the superiority of fusing radar and event cameras over using each sensor individually. Figure 3.18(a) shows that the fusion-based approach significantly outperforms both radar-only and event camera-only methods in terms of location error. mmE-Loc outperforms the event-only approach by 63.6% and exceeds the radar-only method by 52.9%.

Contributions of each module. This work investigates the contributions of *CCT* and *GAJO* to mmE-Loc by gradually integrating them with the event camera into the



(a) Effectiveness of sensor fusion (b) Impact of different module

Figure 3.18 Ablation study on sensor fusion and impact of different modules.



(a) Performance of *CCT* (b) Performance of *GAJO*

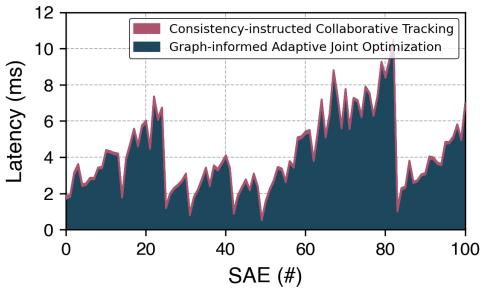
Figure 3.19 Ablation study on performance of *CCT* and *GAJO*.

baseline system (i.e., the radar-only-based method) and assessing localization accuracy and end-to-end latency. Figure 3.18(b) illustrates that without these modules, the baseline method achieves a localization error of 0.229m and a latency of 10.19ms. Integrating the event camera with the *CCT* module reduces the localization error to 0.178m and decreases latency to 7.27ms. This is because the event camera, together with the *CCT* module, filters out the noisy point cloud generated by the radar. Integrating the event camera with *GAJO* further reduces the error to 0.139m, although the delay increases due to the absence of an efficient detection mechanism. This is because the *GAJO* jointly optimizes multiple locations to improve accuracy; however, without a robust detection strategy, the system may also optimize the locations of non-drone objects, leading to additional computational overhead. Finally, integrating both *CCT* and *GAJO* minimizes both the error and latency.

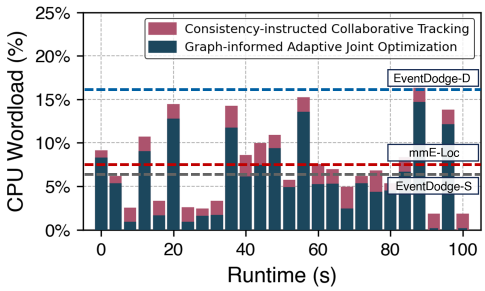
Performance of *CCT*. This work evaluates the filtering performance of the proposed *CCT* module on both mmWave radar and event data in indoor and outdoor environments. As illustrated in Figure 3.19(a), a higher recall indicates that more true drone-triggered

data are correctly identified (fewer are missed), while a higher precision reflects that fewer background data are mistakenly classified as drone data (fewer false alarms). In the indoor scenario, *CCT* achieves recalls above 86% for mmWave radar and 89% for the event camera, with corresponding precisions of over 86% and 82%. The slightly lower filtering rate in indoor conditions can be attributed to two factors: First, the denoising algorithm may inadvertently remove weak perception signals from the edges of the drone's propellers. Second, the complex indoor background, particularly the overhead mesh-like protective structures, can introduce spurious detections for both sensors. In contrast, under outdoor conditions, the recall remains above 80% for both sensing types, and the precision exceeds 90%, demonstrating the robustness of *CCT*. Overall, these results arise from two key design advantages of the *CCT* module. First, it effectively exploits the complementary sensing characteristics of the mmWave radar and event camera, achieving consistent filtering performance across different environments. Second, the module further incorporates the drone's intrinsic physical priors, including its periodic micro-motions and characteristic structural patterns, to differentiate genuine drone responses from irrelevant background signals. In practical terms, this physical knowledge allows the system to recognize and suppress detections that do not exhibit the periodic micro-motions or geometric structure typical of a flying drone. By leveraging these motion and structural cues, the module effectively filters out spurious responses caused by environmental noise or static reflections. Consequently, the integration of these physical priors contributes to a consistent improvement in precision for both sensing modalities under both indoor and outdoor conditions, demonstrating the effectiveness of the proposed filtering mechanism.

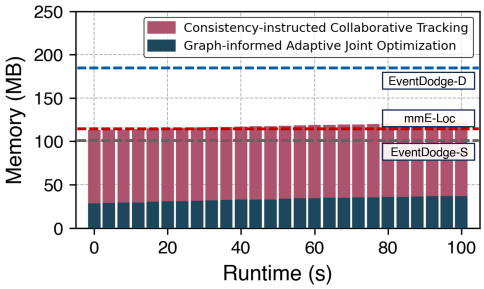
Performance of *GAJO*. This work compares the performance of different multimodal fusion strategies. Specifically, this work evaluates mmE-Loc against two widely used approaches: the extended Kalman filter (EKF)[121] and Graph Optimization (GO) [121]. As shown in Figure 3.19(b), mmE-Loc enhances localization performance by over 57.9% compared to EKF and 47.1% compared to GO, due to its tightly coupled multi-modal fusion and the incremental optimization method. This work further evaluates the effect of enabling the motion-aware adaptive optimization scheme in mmE-Loc. This scheme dynamically adjusts the optimization window based on the drone's motion state. Experimental results show that enabling this scheme leads to a 14.9% increase in localization error, mainly due to the smaller optimization window, which limits the available temporal information for refinement. However, it also yields an 11.5% reduction in CPU usage,



(a) System latency



(b) CPU workload



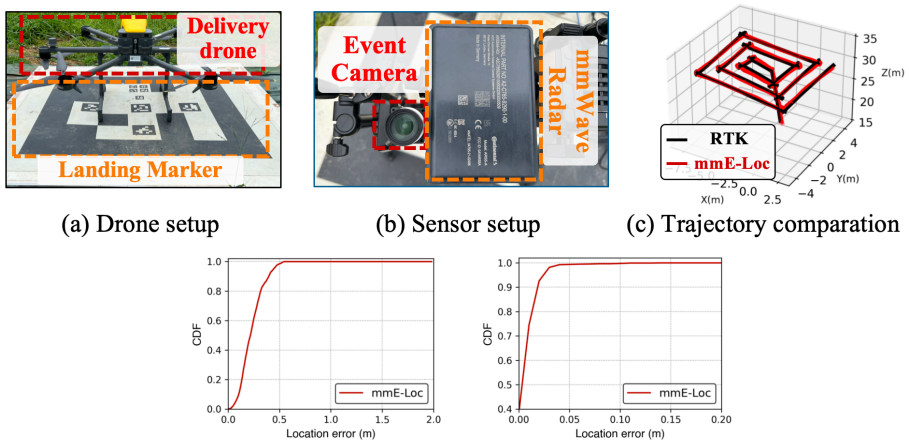
(c) Memory usage

Figure 3.20 System efficiency evaluation.

demonstrating that the adaptive mechanism effectively decreases computational overhead. These findings reveal a clear trade-off between localization accuracy and computational efficiency. While a smaller optimization window leads to less accurate but faster updates, it reduces runtime overhead. Consequently, under resource-constrained conditions, such as embedded processors, enabling this scheme can provide a practical balance, maintaining reasonable localization accuracy while improving real-time performance.

3.6.5 System Efficiency Study

The mmE-Loc distinguishes itself from existing models and learning-based methods due to its low latency and minimal resource overhead. Figure 3.20(a) presents the end-to-end latency of the typical localization process, which includes delays from both CCT and GAJO modules. During drone localization, mmE-Loc's latency may vary because its motion-aware adaptive optimization scheme optimizes different sets of locations at different times, and the incremental optimization method dynamically decides whether to perform incremental or full optimization at each time step. As the size of the location set increases, the optimization time also grows. Nevertheless, thanks to the motion-aware adaptive optimization scheme and the incremental optimization method, mmE-Loc maintains low latency overall. Nonetheless, mmE-Loc's latency remains suitable for use



(a) Drone setup (b) Sensor setup (c) Trajectory comparation

(d) Absolute location error (e) Relative location error

Figure 3.21 Case study: delivery drone airport.

in flight control loops. Figure 3.20(b) and Figure 3.20(c) show that the CPU utilization of mmE-Loc remains below 18%, while its memory consumption stays under 120 MB. Moreover, the memory usage increases slightly as the size of the location set grows. To further highlight the efficiency of mmE-Loc, this work compares its CPU and memory usage with those of baseline methods (mDrone and milliEye are excluded from this comparison since they rely on GPU acceleration). As shown in Figure 3.20(b), mmE-Loc exhibits an average CPU usage of 7.47%. The CPU usage of EventDodge-S is about 15% lower than that of mmE-Loc, primarily because mmE-Loc incorporates mmWave radar for depth estimation, which introduces additional computational overhead while yielding more accurate localization results. In contrast, EventDodge-D, which employs two event cameras, shows a 125.9% increase in CPU usage compared to mmE-Loc. Although multiple event cameras improve localization accuracy, they also introduce significant overhead due to multi-camera target detection and joint position optimization. As shown in Figure 3.20(c), the average memory usage of mmE-Loc is 115 MB. EventDodge-S consumes about 13% less memory, mainly due to the reduced amount of sensor data. In contrast, EventDodge-D exhibits a 60% higher memory footprint than mmE-Loc, as it processes a larger volume of event camera data and involves more optimization factors.

3.7 Case study: Real-world Drone Airport

As shown in Figure 3.21(a), to verify the system's practical usability and its "perception-decision-action" capability, this work conducts a system-level closed-loop experiment using a custom drone equipped with six propellers and managed by a PX4

flight controller ①. This drone is developed by a world-class delivery company exploring the feasibility of instant deliveries. Crucially, rather than merely collecting data for offline open-loop evaluation, the high-precision, low-latency localization outputs from mmE-Loc are continuously transmitted to the drone via a wireless data link. The onboard PX4 flight controller utilizes these real-time coordinate updates to correct internal cumulative drift and dynamically adjust its motor outputs, successfully executing autonomous, closed-loop landing maneuvers in a real-world dynamic environment.

The experiment takes place at a real-world delivery drone airport. To enable drone localization over a larger area, this work employs an ARS548 mmWave radar, as depicted in Figure 3.21(b). Figure 3.21(c), Figure 3.21(d), and Figure 3.21(e) illustrate the localization results as the drone follows a square spiral trajectory at an altitude of 30 m. In the qualitative analysis, the left image in Figure 3.21(c) shows the localization results from RTK, while the middle image presents the results from mmE-Loc. As illustrated in the right image of Figure 3.21(c), mmE-Loc produces smooth trajectories that closely align with those of RTK, demonstrating its potential to complement RTK-based localization. In the quantitative analysis, the results in Figure 3.21(d) and Figure 3.21(e) show that in real-world scenarios, mmE-Loc achieves high localization precision, maintaining a maximum absolute location error below $0.5m$ and relative location error below $0.1m$. mmE-Loc has significant potential as a complementary system to RTK, aiding drones in challenging environments (e.g., urban canyons, where RTK may be compromised by signal blockage).

3.8 Discussion

This work provides a detailed explanation of the rationale behind key design decisions in the design and implementation of mmE-Loc.

How to integrate ground-based and onboard localization paradigms? While mmE-Loc is designed as an active ground-based localization system, it is not intended to replace onboard localization paradigms; rather, they are strictly complementary. In real-world deployments, these two paradigms operate in parallel during the terminal landing phase. As the drone descends into the landing zone, the onboard sensors (e.g., IMU and visual odometry) continuously maintain autonomous, high-frequency relative state estimation. Simultaneously, the ground-based mmE-Loc tracks the drone and generates high-precision, drift-free absolute global coordinates. These external measurements are

① Experiment video: https://mme-loc.github.io/

transmitted to the drone via a low-latency wireless data link. Onboard the drone, the flight controller's state estimator (e.g., an Extended Kalman Filter) systematically fuses the continuous internal relative motion data with the absolute external position updates provided by mmE-Loc. This parallel integration effectively eliminates onboard cumulative drift while satisfying the high-frequency demands of the control loop, ensuring a robust and precise touchdown.

How can mmE-Loc address system deployment cost and calibration complexity in large-scale deployments? While the fusion of event cameras and mmWave radar provides exceptional perceptual performance for terminal landing phases, scaling the mmE-Loc system across large-scale low-altitude infrastructure introduces practical challenges regarding calibration maintenance and computational overhead. To enhance the universality of this technology for wide-area deployment, two strategic optimizations can be implemented: (*i*) Automated calibration schemes are essential for multi-node networks. Traditional manual calibration using checkerboards and corner reflectors is unfeasible for large-scale infrastructure. An alternative approach is to leverage the delivery drones themselves as dynamic, moving spatial anchors. Crucially, this calibration should be performed opportunistically during the drones' high-altitude cruising phase rather than during the descent phase. In open airspace, the drone's onboard RTK system receives unobstructed signals, providing a highly reliable global trajectory. As these drones routinely fly over ground nodes, the system can perform cross-modal self-calibration by aligning the event-radar observations with the drone's globally anchored trajectory, effectively eliminating manual maintenance costs without being affected by the signal degradation typical in urban canyons. (*ii*) Software-hardware co-optimization is crucial for mitigating computational overhead, particularly regarding the processing of event streams. Traditional synchronous architectures (e.g., standard CPUs or GPUs) are fundamentally inefficient for processing the asynchronous, sparse data generated by event cameras, often wasting clock cycles and power on inactive pixels. To address this, future deployments can integrate neuromorphic computing chips. These event-driven processors mimic biological neural networks, consuming power only when processing active events. By executing early-stage filtering (such as the CCT module) directly on neuromorphic hardware, individual ground nodes can achieve microsecond-level latency and ultra-low power consumption. Meanwhile, heavier backend tasks, such as the Graph-informed Adaptive Joint Optimization (GAJO), can be offloaded to centralized edge servers within an edge-cloud collaborative

architecture. This co-optimization significantly reduces the unit cost and energy footprint of ground-based nodes, facilitating sustainable and large-scale urban deployment.

How does mmE-Loc relate to visual markers? Currently, delivery drones utilize visual markers and onboard cameras for self-localization. In contrast, mmE-Loc focuses on ground-based drone localization, enabling the landing pad to determine the spatial relationship between the drone and itself for precise adjustments. In practice, mmE-Loc operates alongside visual markers to enhance reliability of localization service.

Is it feasible to design an onboard sensor system for drone landing localization? It's theoretically feasible. However, designing an onboard system for drone localization using an event camera and mmWave radar presents several challenges. Given the continuous motion of the drone, the system must address: (*i*) motion compensation for event data, (*ii*) reliable feature extraction and matching within the event stream despite its lack of semantic information, and (*iii*) the sparsity of radar measurements and noise induced by specular reflections, diffraction, and multi-path effects.

How does network latency impact the system, and how can potential delay-related issues be addressed? mmE-Loc is designed to assist ground platforms in locating drones and guiding them to land accurately at designated spots. Network latency may impact the location update rate for the drone. To mitigate this issue, drone airports usually deploy access points near the landing pad and utilize multiple wireless links, including Wi-Fi and cellular networks, to ensure reliable and fast communication. Additionally, an optics-based communication system could be integrated into the system to ensure a high communication frequency[122].

Will there be cases that the drone is not fully visible in the camera frame, as mentioned in[123]? Yes, such cases may occur under certain extreme conditions. For instance, when the drone descends to a very low altitude (e.g., around 1 m) and is about to land on the pad, it may not be fully visible in the camera frame. At this stage, mmE-Loc has already guided the drone directly above the pad, and the final descent can be safely completed using the onboard IMU. Similarly, when the drone is at high altitude and just entering mmE-Loc's operational range, partial visibility may also occur. In practice, given the critical importance of flight safety in commercial operations, multiple localization systems, such as visual markers and visual positioning services, are used together. Thus, even if mmE-Loc does not provide optimal results, stable and safe flight remains ensured.

Does mmE-Loc address orientation ambiguities that arise from the use of some

fiducial markers, as observed by[124]? (*i*) During visual-marker-assisted landing, the drone did not encounter orientation ambiguities, as it employs a fixed, downward-facing frame camera. Unlike gimbal-mounted cameras, this configuration maintains a constant orientation relative to the drone body, preventing cases where a marker appears identical from different viewing angles. (*ii*) Likewise, in mmE-Loc-assisted landing, no orientation ambiguities were observed. Before transmitting localization results, this work transforms them into the global coordinate system, as the mmE-Loc deployment location is known in advance. Operating within this unified frame, the drone can accurately adjust its position and achieve a stable, precise landing.

Why is the mmWave + event camera configuration preferable despite higher cost and limited FoV? First, the goal is to achieve accurate and low-latency drone localization for safe landings, which only the event camera and mmWave radar can provide. Frame cameras and LiDARs suffer from low frame rates, limiting real-time performance. Second, as the drone-driven low-altitude economy is projected to reach $1 trillion by 2040, one event camera + mmWave radar setup per airport is economically acceptable since it serves multiple drones. Finally, with the rapid advancement of event-based sensors, the cost is steadily decreasing[125]. Regarding the FoV, mmE-Loc focuses on landing phase, where such coverage is sufficient.

How cost-effective are existing methods, and what scenarios can the new combination uniquely address? Current drone localization methods mainly rely on frame cameras, LiDAR, or mmWave radars, each with notable limitations. Frame cameras, such as the Logitech C920E (1080p, 30Hz), provide sufficient spatial resolution but suffer from long exposure times, low frame rates (typically below 50Hz), and sensitivity to lighting conditions, which reduces localization accuracy in high-speed or low-altitude operations. LiDAR sensors, such as Velodyne VLP-16 or Livox Mid-40, generate dense 3D point clouds but are limited by low scanning frequencies (typically below 20Hz), sparse returns at long ranges, high cost, and heavy onboard processing requirements. mmWave radars, e.g., TI IWR1843 BOOST, achieve μs-level temporal resolution and reliable radial depth estimation, yet their horizontal and vertical spatial resolution is limited, constraining 3D localization performance. The proposed mmE-Loc fuses an event camera with mmWave radar. Event cameras provide high-temporal-resolution, HDR-enabled visual sensing, complementing the radar's radial measurements and limited spatial resolution. Both sensors produce sparse outputs focusing on dynamic objects, reducing pro-

cessing demands while enabling rapid, low-latency tracking of drones, thus supporting faster and more efficient drone operations. The fusion supports operation under varying illumination, including low-light or nighttime conditions. Although the combined system has higher upfront costs than a single sensor, one event + mmWave unit can serve multiple drones in shared operational areas. Given the projected $1 trillion low-altitude drone economy by 2040 and decreasing event camera costs [125], this investment is economically justified. The combination offers unique advantages of low-latency, high-accuracy, sparse representation, and suitability for high-speed operations.

How does mmE-Loc handle multi-drone discrimination and tracking? mmE-Loc distinguishes and tracks multiple drones by exploiting their distinct periodic micro-motion patterns and the corresponding signatures captured by the mmWave radar and event camera. Specifically, drones exhibit unique periodic motion characteristics when performing different maneuvers (e.g., ascending, descending, or lateral motion). Meanwhile, the bounding box size and position observed in the event camera (representing spatial cues), as well as the range measurements from the mmWave radar (representing distance cues), also differ among drones. By associating each drone with a unique combination of periodic micro-motion pattern, bounding box characteristics, and range information, mmE-Loc can reliably identify and track individual drones. In real-world drone delivery scenarios, a landing pad typically allows only one drone to land at a time, while others hover nearby awaiting clearance. In such cases, mmE-Loc utilizes periodic micro-motion patterns to distinguish between the landing drone and hovering drones, and further discriminates among multiple hovering drones based on their bounding box positions and sizes in event camera and their range measurements in the mmWave radar.

3.9 Summary

This chapter explores a novel sensor configuration combining event camera and mmWave radar, harmonizing ultra-high sampling frequencies, and proposes mmE-Loc, a ground localization system for drone landings, achieving *cm* accuracy and *ms* latency. The innovation of mmE-Loc lies in two aspects: (*i*) a Consistency-Instructed Collaborative Tracking module that leverages cross-modal *temporal consistency* for accurate drone detection, and (*ii*) a Graph-Informed Adaptive Joint Optimization module that boosts localization and reduces latency by utilizing cross-modal *spatial complementarity*. Evaluations conducted across various scenarios demonstrate the performance of mmE-Loc.

CHAPTER 4 INTERNAL INFORMATION-BASED ENACTED PERCEPTION METHOD

Building upon the concept of "enacted perception" defined in Chapter 1, this chapter introduces EV-Pose, an event-camera-enhanced VPS designed to empower drones with endogenous onboard perception. Moving beyond traditional frame-based constraints, EV-Pose is developed to facilitate high-mobility autonomous navigation in GNSS-denied environments through a lightweight sensing architecture. By exploiting the microsecond-level temporal resolution of asynchronous event streams, this research introduces a novel spatio-temporal feature extraction method to surmount the challenges of semantic sparsity. Crucially, through a hierarchical fusion mechanism, EV-Pose generates ultra-low latency, high-frequency pose estimations that directly optimize the drone's underlying flight control stack. This tight coupling effectively closes the "perception-decision-action" feedback loop, ensuring highly adaptive and resilient aerial operations.

Real-time 6-DoF drone pose tracking enables precise flight control and accurate drone landing. With the widespread availability of urban 3D maps, the VPS has been adapted for drone landing to overcome GPS unreliability, serving as a global localization system that yields absolute global coordinates by continuously aligning visual features with the prior 3D map. However, deploying conventional vision-based VPS on highly dynamic drones faces bottlenecks in both pose estimation accuracy and efficiency. This work pioneers EV-Pose, an event camera-enhanced VPS designed to deliver drift-free, *absolute global coordinates* at ultra-high frequencies for drones. EV-Pose addresses the 2D-3D modality gap via a novel Spatio-Temporal Feature-instructed Pose Estimation module, which extracts a Temporal Distance Field (TDF) to enable continuous, differentiable matching with a prior 3D point map for pose estimation. To fully exploit this formulation, this work proposes a Motion-aware Hierarchical Fusion and Optimization scheme. This architecture utilizes onboard IMU motion for fine-grained early-stage event filtering and seamlessly optimizes pose estimation in a later-stage factor graph. Evaluation shows that EV-Pose achieves a rotation accuracy of 1.76° and a translation accuracy of 7.5mm with a latency of 10.08ms, outperforming baselines by >40% and enabling accurate drone landings.

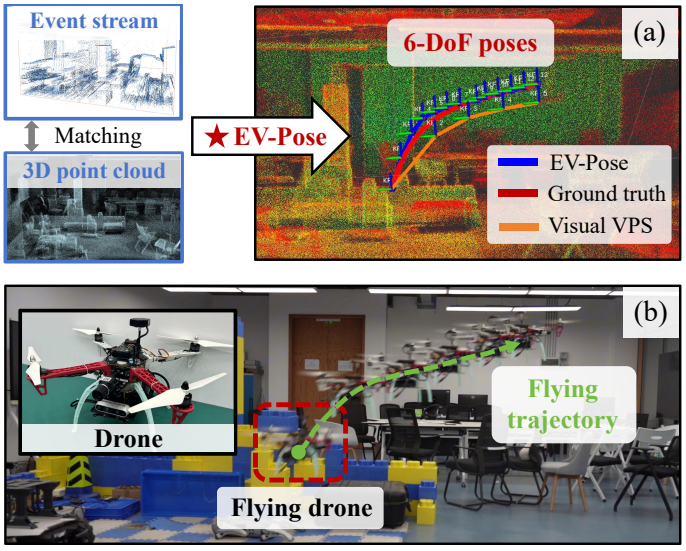


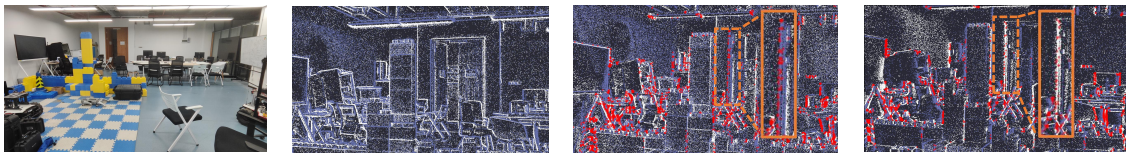
Figure 4.1 EV-Pose estimates the drone's 6-DoF pose by redesigning the drone-oriented VPS with event cameras. As shown in (a), compared to conventional VPS systems, EV-Pose enables high-frequency drone pose tracking, ensuring precise flight control and landing as shown in (b).

4.1 Introduction

Drone-based delivery has been revolutionizing logistics by reducing delivery times, enhancing accessibility to congested areas in rush hours, and providing a sustainable alternative to conventional transportation[126-127]. Central to the success of drone-based delivery is the *landing phase*, in which accurate real-time six-degree-of-freedom (6-DoF) pose estimation plays a pivotal role by enabling precise navigation, effective obstacle avoidance, and reliable package drop-off[128-130]. However, conventional positioning systems such as GPS are inherently unreliable for accurate pose estimation during landing in urban areas, due to severe signal attenuation[131].

Recently, as urban 3D maps have become more widely accessible through platforms such as Google Earth and Mapbox, Visual Positioning Services (VPS) are increasingly being integrated into drones to achieve precise 6-DoF pose estimation during landing in urban areas[44,132-133]. VPS uses video frames captured by an RGB camera. It extracts visual feature points from frames and matches them against a pre-built 3D environmental map (typically represented as a 3D point cloud) to accurately determine the drone's real-time 6-DoF pose[11,134-135]. It further improves the accuracy of these pose estimations by integrating data from onboard IMU sensors using Visual-Inertial Odometry (VIO)[136].

While VPS shows great promise for accurate drone pose estimation during landing, our field studies conducted in collaboration with a leading drone delivery company uncovers two major limitations when applying current VPS to drones:



(a) RGB image (b) Noisy event stream (c) Features in left-to-right motion (d) Features in right-to-left motion

Figure 4.2 Motivation study of EV-Pose. (a) RGB image. (b) Noisy event stream. (c) In left-to-right motion, event features extracted by Arc*, an event-based feature extraction algorithm[139]. (d) Arc* features extracted during right-to-left motion.

(*i*) *Motion blur negatively impacts the drone pose estimation accuracy*. Drone delivery systems need to rapidly and precisely adjust their pose during the landing process to ensure accurate touchdown at the designated location. However, during flight, the drone's rapid motion causes video frames to be highly susceptible to severe motion blur, impairing feature point detection and pose estimation algorithms, ultimately resulting in increased errors in estimating drone pose[137].

(*ii*) *The sampling rate mismatch between sensor modalities delays drone pose estimation.* For precise flight control, drone pose estimation must operate in real-time (e.g., >100 Hz) to provide immediate feedback to the flight controller[16]. The onboard IMU typically operates at <200 Hz, whereas the camera captures frames at <30 Hz. The camera's limited sampling rate constrains VIO pose updates to <30 Hz, forcing the controller to rely on IMU over extended periods, leading to accumulated errors and potential loss of control. Even worse, in practice, since urban maps are typically stored as 3D point clouds, the computational overhead required to match 2D images with 3D digital maps often limits drone pose estimation to operating at <1 Hz[44,132].

All these issues necessitate a comprehensive rethinking of VPS designed for drone landing in urban areas, spanning from hardware components to algorithmic designs.

This paper explores the feasibility of redesigning drone-oriented VPS with *event cameras*. Event cameras are bio-inspired sensors that capture pixel-level intensity changes with a high dynamic range[138]. Compared to low frame-rate RGB cameras, they offer *ms*-level resolution and latency, thus can capture scene dynamics without blurring[112]. As shown in Figure 4.1, the absence of motion blurring and high temporal resolution makes event cameras a better alternative for addressing the aforementioned limitations with the current VPS.

Nevertheless, simply replacing RGB cameras with event cameras without adapting current algorithms in VPS systems would not work due to the unique hardware characteristics of event cameras. This work summarizes two major challenges below.

• **C1**: *Extraction of repeatable vision features from the event stream for cross-modality vision feature matching.* Event cameras detect illumination changes at pixel-level granularity, enabling easy identification of object edges within a scene (Figure 4.2(a)-(b)). However, unlike RGB images, where visual features remain consistent regardless of the camera's viewpoint, event cameras produce significantly different event patterns depending on the drone's direction of flight relative to the observed object. Consequently, existing feature extraction algorithms, even those tailored to event cameras, face difficulties in consistently extracting repeatable features, as shown by Figure 4.2(c)-(d). This results in numerous noisy points, negatively affecting the accuracy of matching visual features with the pre-built 3D point map.

• **C2**: *Joint optimization of efficiency and accuracy in map matching-based pose estimation.* Event cameras are highly sensitive to environmental changes, with even slight movements triggering numerous pixels and generating hundreds of events. As the drone moves, rapid scene changes lead to event bursts, generating thousands of event reports in a short time. Utilizing all reported events for matching would introduce computational overhead, delaying pose estimations. Moreover, the inherent lack of semantic content in the event stream can cause incorrect matching when encountering complex structures, leading to an accuracy decline.

To conquer these challenges, this work introduces **EV-Pose**, an **E**vent camera-enhanced **V**isual Positioning Service, which is an accurate and high-frequency event-based drone 6-DoF **Pose** tracking system that provides drift-free, *absolute global coordinates* for precise drone landing (Figure 4.1). EV-Pose is designed to function in an urban canyon, where satellite-based systems lose accuracy, rendering them nearly useless.

To address **C1**, this work designs a *Spatio-temporal Feature-instructed Pose Estimation (STPE)* module (§4.4). Leveraging temporal relationships in an event stream, STPE resolves the cross-modality matching bottleneck by introducing a Separated Polarity Time-Surface (SP-TS) and extracting a Temporal Distance Field (TDF). This representation elegantly bridges the modality gap, aligning 2D asynchronous events directly with a 3D point map to provide reliable global exteroceptive drone pose measurements.

To address **C2**, this work proposes a *Motion-aware Hierarchical Fusion and Optimization (MHFO)* scheme (§4.5) to enhance pose estimation in *STPE*. Leveraging the temporal consistency between event camera and IMU, MHFO proposes a two-stage fusion scheme. At *early stage*, it fuses motion information from IMU and 3D map structural

data to predict event density and polarity for event filtering, boosting estimation *efficiency*. At *later stage*, a factor graph tightly couples proprioceptive motion measurements with exteroceptive STPE outputs, ensuring *accurate* global pose optimization.

This work integrates EV-Pose into the ArduPilot flight controller and evaluates it across 20+ hours of field studies and challenging public datasets. Benchmarked against three SOTA systems, EV-Pose achieves an average rotation error of 1.76°, a translation error of 7.5mm, and a latency of 10.08ms, outperforming baselines by >40%. Field experiments validate its robustness under various dynamic and complex illumination conditions.

Summary. The contributions of this work are as follows:

(1) This work pioneers *EV-Pose*, the first event-enhanced VPS designed for drones. Departing from traditional relative odometry (VIO), EV-Pose provides drift-free, *absolute global coordinates* at ultra-high frequencies.

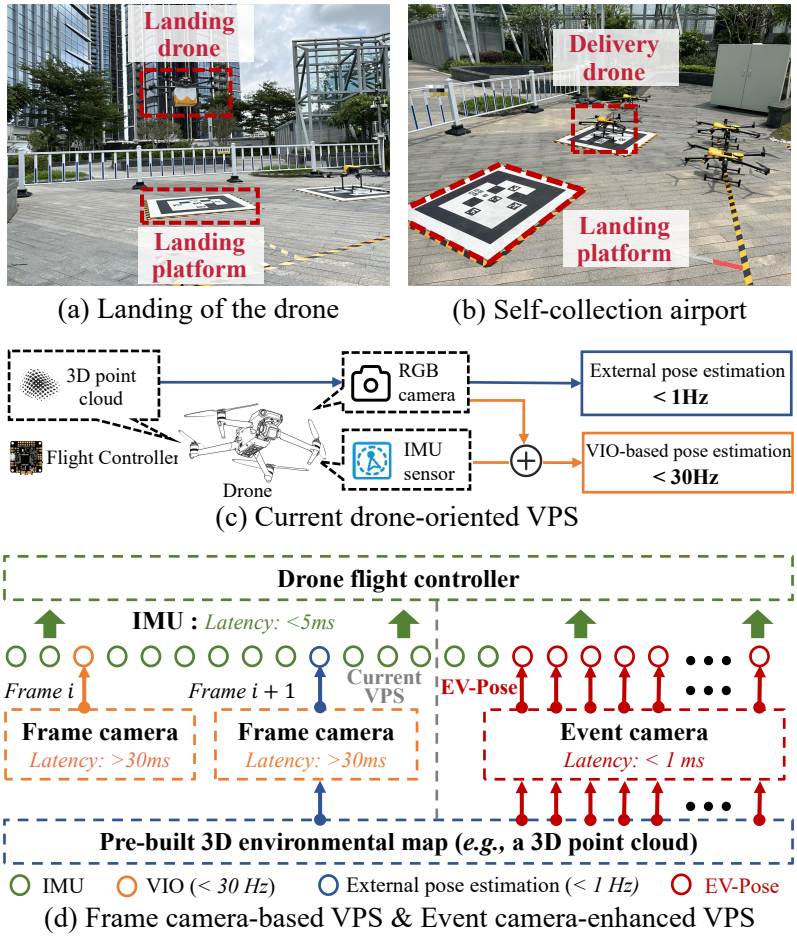
(2) This work formulates an algorithmic challenge of cross-modality matching by designing the *STPE* module, which extracts a TDF to elegantly align 2D asynchronous events with a prior 3D point map. To fully exploit this, this work proposes the *MHFO* scheme, a proactive architecture that deeply integrates IMU motion to instruct early-stage event filtering and seamlessly optimizes the poses in the later-stage factor graph.

(3) This work implements EV-Pose and evaluates its performance through field studies and experiments. Comparisons with SoTA systems demonstrate its advantages.

The remainder of this chapter is organized as follows: Section 4.2 presents the background and motivation. Section 4.3 provides an overview of the system. Sections 4.4 and 4.5 introduce the Spatio-temporal Feature-Instructed Pose Estimation method and the Motion-Aware Hierarchical Fusion and Optimization method, respectively. Subsequently, Sections 4.6 and 4.7 detail the system's implementation and evaluation. Finally, Section 4.8 discusses the key design choice, and Section 4.9 concludes this chapter.

4.2 Background and motivation

This section establishes the motivation for transitioning from traditional visual systems to neuromorphic sensing. Section 4.2.1 first reviews the research background of RGB camera-based VPS and identifies its critical limitations in high-dynamic drone applications. Building upon this analysis, Section 4.2.2 articulates the necessity and inherent superiority of event camera-enhanced VPS, highlighting its potential to overcome the perceptual bottlenecks of conventional frame-based paradigms.



(a) Landing of the drone (b) Self-collection airport

(c) Current drone-oriented VPS

(d) Frame camera-based VPS & Event camera-enhanced VPS

Figure 4.3 Current VPS and comparison of frame camera-based VPS & event camera-enhanced VPS. (a) Landing of a drone. (b) Self-collection airport and landing platform. (c) Current VPS uses an RGB camera, an IMU, and 3D point clouds for pose estimation. (d) EV-Pose leverages event cameras for accurate and low-latency 6-DoF drone pose tracking.

4.2.1 RGB Camera-based VPS

Background. Projected to reach a $1 trillion market by 2040[127], the drone-driven low-altitude economy is revolutionizing industries with applications like on-demand delivery[140]. At its core, 6-DoF drone pose tracking provides real-time drone location and orientation, supporting flight control, path planning, and landing, as shown in Figure 4.3(a) and Figure 4.3(b)[37,141]. With urban 3D maps widely available, VPS, a mobile pose estimation system leveraging environmental visual cues, has been adapted for drone 6-DoF tracking, particularly in GPS-denied environments[133,142]. Current VPS integrates an RGB camera, IMU, and pre-existing environmental data (e.g., a 3D point cloud)[44,136,143]. As shown in Figure 4.3(c), the RGB camera and IMU facilitate VIO at <30 Hz, while the 3D point cloud supports external pose estimation at <1 Hz[144]. Specifically, the external pose estimation is performed by extracting local features from a camera

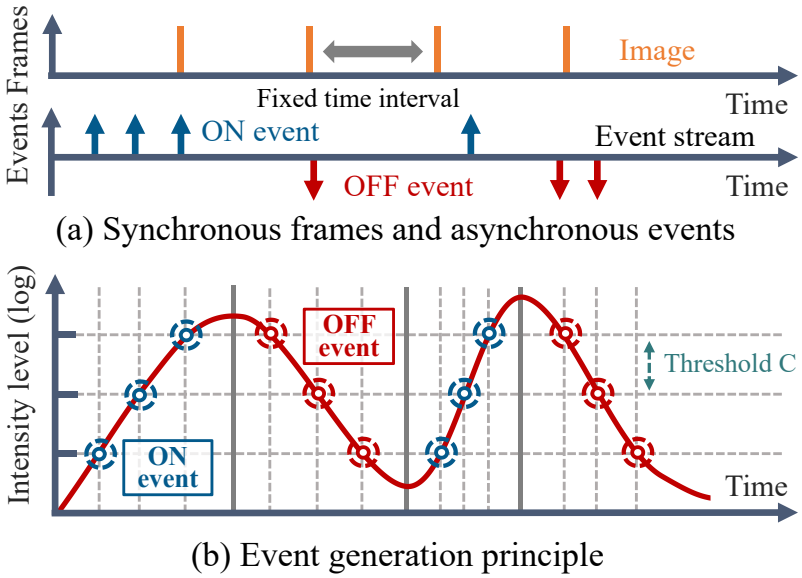


Figure 4.4 Principles of frame cameras and event cameras. (a) A frame camera uses a global shutter to capture synchronous frames. Each pixel of the event camera operates independently, generating events asynchronously. (b) Each pixel in event cameras generates events when intensity changes exceed a threshold: [ON] for increases, [OFF] for decreases.

image and matching 2D descriptors to 3D map points. The prior map, typically pre-stored in the flight controller, is often generated from images or LiDAR data[57,145-146].

Limitations of current VPS on drone. Despite the ability of VPS to provide accurate pose estimation for pedestrian navigation[133], the high flight speed of drones presents significant challenges for frame camera-based VPS. Specifically, the motion blur in frame images disrupts feature point detection. As a result, pose estimation algorithms fail to establish correspondences between feature points and the map, degrading estimation accuracy. At the same time, drone pose estimation must operate in real-time (>100 Hz) to provide immediate feedback to the flight controller for precise control. However, despite onboard IMU running at <200 Hz, the long exposure time of frame cameras (>30 ms) limits VIO operating at <30 Hz; and the computational overhead of 2D-3D matching limits 3D point cloud-based global pose estimation to updates at <1 Hz. As shown in Figure 4.3(d), both VIO and map-based estimation rates are incompatible with the flight controller, forcing reliance on IMU and reducing accuracy.

4.2.2 Event Camera-enhanced VPS

Principle of event camera. Event cameras are bio-inspired sensors that differ from traditional frame cameras[105]. Specifically, frame cameras capture synchronous images with a global shutter at fixed time intervals, while the output of an event camera is event

streams with *ms*-level resolution, as shown in Figure 4.4(a)[106]. Each pixel in an event camera independently responds to changes in brightness asynchronously. Each event $e = (\boldsymbol{x}, i, p)$ represents a pixel at $\boldsymbol{x} = (u, v)$ that has undergone a predefined magnitude change in brightness at a time i, as shown in Figure 4.4(b). p is the polarity of intensity change, [ON] for brighter and [OFF] for darker.

Advantages of event camera. The *ms*-level resolution and latency of event cameras make them highly suitable for capturing fast motion without motion blur, a common issue in frame cameras[107]. Additionally, due to their logarithmic intensity recording, event cameras achieve a 120 dB dynamic range, compared to 60 dB of frame cameras[105], ensuring performance in extreme conditions (e.g., low light scenes).

Redesign drone-oriented VPS with event camera. As shown in Figure 4.3(d), event cameras, with advantages in temporal resolution, hold great potential for revolutionizing drone-oriented VPS systems. 3D point clouds are pre-stored in the flight controller. By redesigning the VPS with event cameras, this work can leverage these prior 3D point clouds while ensuring temporal consistency with the IMU, enabling accurate and low-latency 6-DoF pose tracking for precise drone pose adjustment and landing[147-148]. However, inherent characteristics of event cameras (e.g., motion-dependent, output of pixel-level intensity changes, and scale uncertainty) pose challenges for event-based tracking[149]. This work presents EV-Pose, which extracts spatio-temporal features from event streams and matches them against a 3D point map for pose estimation. Then, it enhances the accuracy and efficiency of the above estimation by mitigating event bursts and scale uncertainty through a hierarchical integration of motion information.

4.3 System Overview of EV-Pose

This section provides a description of the *EV-Pose* system architecture. Section 4.3.1 first defines the problem statement, formally outlining the challenges and variables involved in absolute 6-DoF pose tracking. Subsequently, Section 4.3.2 presents a high-level overview of the *EV-Pose* framework, introducing the functional interconnections between the spatio-temporal feature extraction and the hierarchical fusion modules.

4.3.1 Problem Statement

Variables description. The 6-DoF pose of drone includes its 3D location (l_x, l_y, l_z) and 3D orientation $(\theta_x, \theta_y, \theta_z)$. In EV-Pose, IMU is a standard sensor on a drone to provide

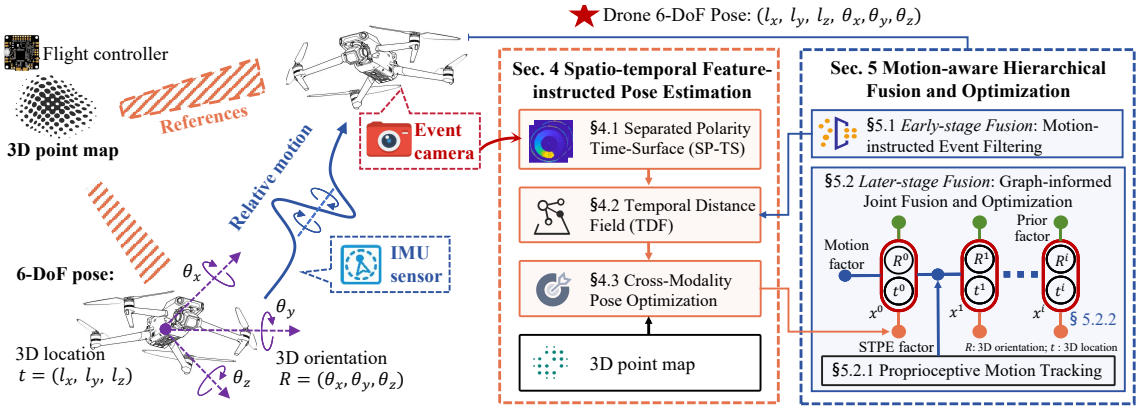


Figure 4.5 System architecture of EV-Pose.

motion information, and there are three coordinate systems: (*i*) event camera coordinate system, E; (*ii*) IMU coordinate system, M; and (*iii*) 3D point map coordinate system, P. The 3D point map is typically stored in the flight controller with known global coordinates. Therefore, the 6-DoF pose of drone x^i at each time i can be treated as transformation from E to P, described using a combination of rotation matrix (R^i_{PE}) and translation vector (t^i_{PE}). The event camera and IMU are fixed on the drone, with relative pose remaining unchanged during movement, described as (R^i_{EM}, t^i_{EM}). Since global coordinates of 3D point map are known, (R^i_{PE}, t^i_{PE}) is equivalent to ($l_x, l_y, l_z, \theta_x, \theta_y, \theta_z$). This work uses the former for description, as it's widely used in flight control systems.

Problem statement. EV-Pose leverages 3D point maps stored in flight controller, denoted as $\mathcal{P} = \{p^k \mid k \in \mathcal{N}\}$, and $\mathcal{N}$ is number of points. Subsequently, EV-Pose reads time-series signals from the event camera and IMU during a period $\mathcal{I}$, denoted as $\mathcal{E} = \{e^i \mid i \in \mathcal{I}\}$ and $\mathcal{M} = \{m^i \mid i \in \mathcal{I}\}$, respectively. Using these three sets of data $\{\mathcal{P}, \mathcal{E}, \mathcal{M}\}$ as input, EV-Pose calculates 6-DoF pose of drone $x^i = \{R^i_{IE}, t^i_{IE}\}$.

4.3.2 Overview

From a top-level perspective, this work designs EV-Pose, an event-based 6-DoF pose tracking system for drones that redesigns VPS with event cameras. EV-Pose leverages prior 3D point maps and temporal consistency between the event camera and IMU to achieve accurate, efficient drone 6-DoF pose tracking. As shown in Figure 4.5, EV-Pose has two key parts:

• **Spatio-Temporal Feature-instructed Pose Estimation (STPE)** module (§4.4). This module first introduces the separated polarity time-surface, a novel spatio-temporal representation for event streams (§4.4.1). Subsequently, it leverages the temporal rela-

tionships among events encoded in the time-surface to generate a temporal distance field, which is used as a feature representation for the event stream (§4.4.2). Finally, the cross-modality pose optimization module models the drone pose estimation problem, which aligns the event stream's temporal distance field with the 3D point map, thus facilitating absolute pose estimation of the drone (§4.4.3).

• **Motion-aware Hierarchical Fusion and Optimization (MHFO)** scheme (§4.5). This scheme first introduces motion-optical flow-instructed event filtering (§4.5.1), which combines drone motion information with structural data from the 3D point map to predict event density and polarity for fine-grained event filtering. This approach fuses the event camera with IMU at *early stage* of raw data processing, improving *efficiency* of matching-based pose estimation. This scheme then introduces a graph-informed joint fusion and optimization module (§4.5.2), This module first infers the drone's relative motion through proprioceptive tracking and then introduces a factor graph to fuse these measurements with exteroceptive data from the STPF module. This fusion, performed at the *later stage* of pose estimation, further improves the *accuracy* of matching-based pose estimation.

Relationship between STPE and MHFO. STPE serves as the exteroceptive frontend, extracting the TDF from the raw event stream and aligning it with a prior 3D point map for visual pose estimation. To fundamentally enhance both efficiency and global accuracy, EV-Pose wraps STPE within the MHFO framework to form a tightly-coupled feedback loop. Specifically, MHFO leverages drone motion kinematics for *early-stage event filtering*, drastically reducing the data bandwidth required for STPE matching. Subsequently, it performs *later-stage joint optimization* to fuse the STPE output with proprioceptive measurements, yielding a globally consistent, drift-free 6-DoF trajectory.

4.4 Spatio-Temporal Feature-instructed Pose Estimation

Challenge. Event cameras produce asynchronous and texture-sparse event streams that lack inherent photometric semantics. Consequently, directly matching these discrete 2D events with a dense 3D spatial point cloud presents a profound cross-modality matching bottleneck. Traditional feature extraction and alignment techniques are fundamentally inapplicable to such heterogeneous data structures.

Methodological formulation. This work observes that the spatio-temporal distribution of events encapsulates rich geometric structure priors. Rather than treating events as isolated features, this work mathematically models their spatio-temporal relationships as

a continuous distance field. By explicitly extracting this Temporal Distance Field (TDF), this work bridges the 2D-3D modality gap, enabling an optimizable framework for global drone pose estimation.

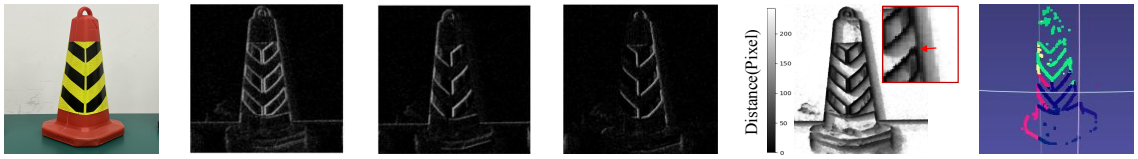
To operationalize this theoretical formulation, this work designs the STPE module to execute the 2D-3D alignment. This module operates through a rigorous three-stage pipeline: First, to preserve the inherent high-temporal resolution without geometric distortion, it introduces a novel representation termed the Separated Polarity Time-Surface (SP-TS) (§4.4.1), mapping asynchronous events strictly according to their polarity. Second, by exploiting the local spatio-temporal correlations encoded within the SP-TS, it extracts the continuous TDF (§4.4.2), successfully transforming the discrete event streams into an optimizable feature space. Finally, utilizing a pre-built 3D point map as a static spatial reference, §4.4.3 mathematically formulates the cross-modality alignment as a differentiable global pose estimation problem.

4.4.1 Separated Polarity Time-Surface (SP-TS)

The high volume of noisy events makes it time-consuming to directly extract features from 3D meta event streams, as hundreds of events accumulate within a brief period. Therefore, it is essential to develop a representation that can be used to effectively extract reliable features, enabling accurate pose estimation via 2D event-3D point map matching.

Consistency-based event filtering. Event cameras are prone to noise from transistor circuits and other non-idealities, requiring pre-processing filtering at first. For the k^{th} event e^k with the timestamp i, this work assesses the timestamp of the most recent neighboring event in all directions ($i_{n(x)}$). Events with a time difference to the most recent neighboring event less than the threshold T are retained, indicating object activity, while those exceeding T are discarded as noise.

Spatio-temporal representation of events. This work then introduces *Separated Polarity Time-Surface (SP-TS)*, a lightweight event stream representation which preserves rich spatio-temporal information (e.g., edges, the most descriptive areas in the event stream). Time Surface (TS) is a 2D map in which each pixel value represents the timestamp of the most recent event at that location, and SP-TS maps events of two different polarities onto two separate 2D maps. In this way, the SP-TS emphasizes recent events rather than past ones. When using an exponential decay kernel, recent events are further highlighted, enhancing their significance. Specifically, if i_{last} is timestamp of the most recent event at each pixel $x = (u, v)^T$, then at time i ($i \geq i_{last}(x)$), TS is defined



(a) RGB image (b) Original time surface (c) $\mathcal{T}_{posi}$ in SP-TS (d) $\mathcal{T}_{nega}$ in SP-TS (e) Temporal distance field (f) 3D point map

Figure 4.6 Illustration of SP-TS, temporal distance field, and 3D point map. (a) Frame-based image; (b) Original time surface; (c) $\mathcal{T}_{posi}$ in SP-TS; (d) $\mathcal{T}_{nega}$ in SP-TS; (e) Temporal distance field; (f) 3D sparse point reference map.

as $\mathcal{T}(x,i) = exp\left(-\frac{i-i_{last}(x)}{\eta}\right)$, where η denotes constant decay rate (Figure 4.6(b)). For positive and negative events, SP-TS maintains separate TSs: (i) $\mathcal{T}_{posi}$ is the TS of positive events (Figure 4.6(c)), and (ii) $\mathcal{T}_{nega}$ is the TS of negative events (Figure 4.6(d)). Constructing an SP-TS for each event stream requires $O(N)$ time, as it involves only a single pass through the events.

Difference between SP-TS and original TS. Original TS disregards event polarity when projecting event stream onto a 2D map, despite polarity encoding event features that can facilitate event filtering. In contrast, SP-TS retains polarity by maintaining separate TS for different polarities. As elaborated in §4.5.1, SP-TS can leverage polarity information in conjunction with the drone's motion and environmental structure to refine event filtering, reducing the computational cost associated with 2D event-3D point map matching.

4.4.2 Novel Event-based Feature: Temporal Distance Field

Conventional event-based tracking often relies on Contrast Maximization (CM) to aggregate events. However, CM is fundamentally ill-posed for the VPS system, as it assumes photometric consistency within a purely 2D temporal stream, failing to address the cross-modality alignment between 2D events and a 3D structural map. To overcome this, this work leverages the SP-TS to construct a novel spatial metric.

The SP-TS captures the motion history of scene edges by tracking the most recent event timestamp at each pixel. Crucially, the temporal values in the TS $\mathcal{T}$ exhibit an *anisotropic spatial decay*: they decrease gradually along the trajectory of past motion, while declining sharply in the opposite direction. This physical property inherently forms an anisotropic distance field, where the spatial gradient smoothly points toward the edge's most current location.

Temporal Distance Field (TDF) generation. To adapt this property for rigorous pose optimization, this work constructs the Temporal Distance Field (TDF) by inverting and offsetting the TS: $\overline{\mathcal{T}} = 1-\mathcal{T}$. In this transformed continuous field, pixels exactly at the

current edge possess the minimum value (approaching zero), and the values monotonically increase corresponding to the spatial-temporal distance from the edge (as shown in Figure 4.6(e)). Consequently, the TDF acts as a dense, differentiable 2D feature map to evaluate the reprojection of the prior 3D point map.

Benefits of TDF. Traditional direct 2D-3D edge matching is highly non-convex due to the discrete, asynchronous nature of raw events, inevitably leading to severe local minima. By mapping events into the TDF, this work formulates the cross-modality alignment as a continuous minimization problem. The pixel values in the TDF naturally serve as the optimization residuals, while the continuous spatial gradient $\nabla\overline{\mathcal{T}}$ provides a reliable analytical Jacobian. This anisotropic gradient significantly broadens the basin of attraction for the non-linear solver, ensuring that the objective function remains locally convex and converges rapidly, even when initialized with substantial pose errors.

4.4.3 Cross-Modality Pose Optimization via TDF Alignment

Leveraging the continuous, differentiable nature of the TDF ($\overline{\mathcal{T}}$), this work formalizes the cross-modality matching between the 2D event stream and the prior 3D point cloud $\mathcal{P}$ (Figure 4.6(f)) as a direct, non-linear optimization problem.

Problem formulation and analytical optimization. Given a candidate drone pose $\hat{x}^i$, this work warps the global 3D points $p \in \mathcal{P}$ onto the 2D plane of the event camera. The optimal absolute pose is estimated by minimizing residuals sampled directly from TDF:

$$\hat{x}^i = \underset{R^i_{PE}, t^i_{PE}}{\arg\min} \sum_{p\in\mathcal{P}} \rho\left(\overline{\mathcal{T}}(\pi(W(p; R^i_{PE}, t^i_{PE})))^2\right), \tag{4.1}$$

where ρ is a robust loss function (e.g., Huber loss), π represents the camera projection model, and $W(p; R^i_{PE}, t^i_{PE}) = R^i_{PE} \cdot p + t^i_{PE}$ applies the rigid SE(3) transformation to p.

Crucially, solving this non-linear least-squares problem[150] requires computing the Jacobian of the residual with respect to pose increment δx. Thanks to anisotropic spatial decay of TDF, this work can formulate an exact, analytical Jacobian using chain rule:

$$\boldsymbol{J} = \frac{\partial\overline{\mathcal{T}}}{\partial\delta x} = \nabla\overline{\mathcal{T}} \cdot \frac{\partial\pi(p_c)}{\partial p_c} \cdot \frac{\partial p_c}{\partial\delta x}, \tag{4.2}$$

where $\nabla\overline{\mathcal{T}}$ is the 2D continuous spatial gradient provided by the TDF, and p_c is the transformed 3D point in the camera frame. This analytical continuous gradient avoids the numerical approximation errors common in discrete event tracking, ensuring rapid convergence and robust optimization even under aggressive maneuvers.

Inherent limitations of purely exteroceptive tracking. Theoretically, the microsec-

ond resolution of events enables ultra-high-frequency pose updates. However, optimizing against every raw event without structural priors is computationally prohibitive, reducing estimation efficiency. Furthermore, under extreme dynamic scenarios or visually degenerate conditions, pure geometric alignment may suffer from local minima, severely degrading accuracy. Therefore, to fundamentally overcome these bottlenecks, this work must transition from purely reactive visual matching to a proactive paradigm by incorporating the drone's intrinsic kinematic motion and environmental structure priors.

4.5 Motion-aware Hierarchical Fusion and Optimization

Challenge. Rapid drone maneuvers trigger massive "event bursts," imposing severe computational bottlenecks on raw event processing. Furthermore, the inherent lack of photometric semantics in events exacerbates data association ambiguities, frequently causing incorrect cross-modality matching and tracking degradation.

Methodological insight. The design is driven by two theoretical insights: (i) Coupling 3D map spatial gradients with real-time IMU kinematics provides a deterministic physical before proactively predicting event density and polarities for dynamic noise filtering. (ii) Synergistically fusing absolute *exteroceptive* poses from STPE with the high-frequency *proprioceptive* relative constraints from the IMU mutually regularizes the state manifold, ensuring a drift-free trajectory.

To formalize these insights, this work proposes the *Motion-aware Hierarchical Fusion and Optimization (MHFO)* scheme via a rigorous dual-stage pipeline. At the *early stage* (§4.5.1), it deeply couples IMU data with 3D map priors to proactively filter events via motion-optical flow prediction, maximizing computational *efficiency*. At the *later stage* (§4.5.2), a graph-informed optimization seamlessly integrates proprioceptive IMU constraints with exteroceptive STPE measurements. This factor graph fusion mutually calibrates both modalities, significantly enhancing global pose *accuracy* and robustness.

4.5.1 Early-stage Fusion: Motion-optical Flow-instructed Event Filtering

In this part, this work fuses the event camera and IMU sensor at the early stage of raw data processing to enhance the efficiency of the 2D event–3D map matching–based pose estimation process. As shown in Line 1–2 of the Alg.4.1, this work first calculates the SP-TS, which results in $\mathcal{T}_{posi}$ and $\mathcal{T}_{nega}$. Subsequently, by negating and offsetting $\mathcal{T}_{posi}$

Algorithm 4.1 Tightly-Coupled Pose Estimation via Adaptive Event Filtering

Data: Event stream $\{\mathcal{E}^i\}$; IMU prior $\mathbf{m}^i$; 3D Map $\mathcal{P}$.
Result: Exteroceptive absolute pose trajectory $\{\hat{x}^i\}$.

Represent $\mathcal{E}$ as SP-TS, $\{\mathcal{T}_{posi}, \mathcal{T}_{nega}\}$;
Extract TDF of $\{\mathcal{T}_{posi}, \mathcal{T}_{nega}\}$: $\overline{\mathcal{T}}_{posi}$ and $\overline{\mathcal{T}}_{nega}$;
for *each interval i* **do**
  Extract prior $\bar{x}^i$ and $[\mathbf{v}, \boldsymbol{\omega}]^T$ from $\mathbf{m}^i$;
  Initialize optimization variable $\hat{x}^i_{iter} \leftarrow \bar{x}^i$;
  while *optimization not converged* **do**
    % Motion-Instructed Event Filtering (§ 5.1)
    Initialize aligned event TDF $\overline{\mathcal{T}}_{align} \leftarrow \varnothing$;
    for *each point* $p^k \in \mathcal{P}$ *visible in* $\hat{x}^i_{iter}$ **do**
      Project to $\mathbf{u}_k$ and compute $\mathbf{v}_{flow} = \mathbf{J}_{img}(\mathbf{u}_k)[\mathbf{v}_{iter}, \boldsymbol{\omega}_{iter}]^T$;
      % Retain events from structural edges
      if *kinematic saliency* $|\nabla S \cdot \mathbf{v}_{flow}| > \epsilon_{kinematic}$ **then**
        Retain local event TDF patch of $\overline{\mathcal{T}}_{posi}$ and $\overline{\mathcal{T}}_{nega}$ at $\mathbf{u}_k$ into $\overline{\mathcal{T}}_{align}$;
      end
      % Retain events from photometric texture
      if *map has texture* ***and*** *raw events match polarity of* $\nabla L \cdot \mathbf{v}_{flow}$ **then**
        Retain local event TDF patch of the corresponding polarity at $\mathbf{u}_k$ into $\overline{\mathcal{T}}_{align}$;
      end
    end
    % 2D Event - 3D Map Alignment (§ 4.3)
    for *each valid point* p^k **do**
      Compute residual r_k from $\overline{\mathcal{T}}_{align}(\mathbf{u}_k)$ and Jacobian $\mathbf{J}_k$;
    end
    Solve increment: $\Delta x^* = \arg\min_{\Delta x} \sum \rho\left(\|r_k\|^2_{\Sigma_{STPE}}\right)$;
    Update pose manifold: $\hat{x}^i_{iter+1} \leftarrow \hat{x}^i_{iter} \boxplus \Delta x^*$;
  end
  Yield absolute pose $\hat{x}^i \leftarrow \hat{x}^i_{iter}$ for later-stage Factor Graph;
end

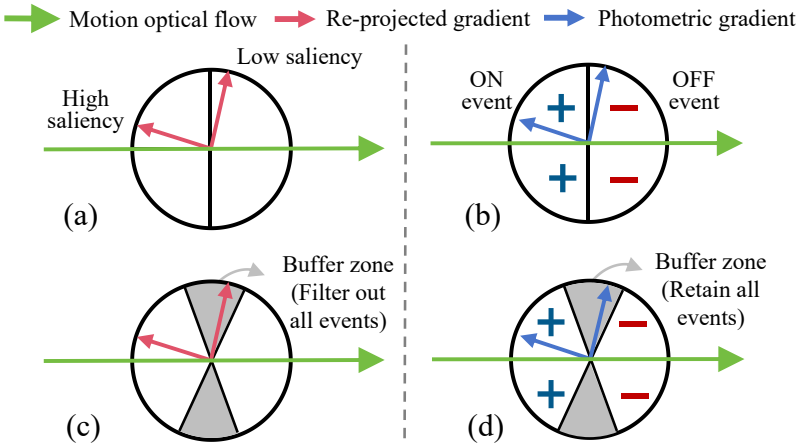


Figure 4.7 Mechanism of the adaptive event filtering and buffer zone strategy. (a) Kinematic Saliency: Evaluated by the alignment between the structural gradient and optical flow. (b) Photometric Polarity: Predicted via the inner product of the photometric gradient and optical flow. (c) Kinematic Margin: Events in the near-orthogonal buffer zone (gray area) are strictly filtered out due to zero saliency. (d) Photometric Margin: In ambiguous near-zero regions, strict polarity matching is suspended to conservatively retain all events.

and $\mathcal{T}_{nega}$, this work obtains the TDFs: $\overline{\mathcal{T}}_{posi}$ and $\overline{\mathcal{T}}_{nega}$.

Motion-instructed event filtering (Line 7-19). As detailed in Algorithm4.1, distinct from conventional sequential pipelines, this work embeds the event filtering directly *inside* the non-linear optimization loop to form a tightly-coupled architecture. Initially, the raw event stream is asynchronously converted into positive and negative TDFs ($\overline{\mathcal{T}}_{posi}$ and $\overline{\mathcal{T}}_{nega}$). During each iteration, this work projects the 3D structural gradients (e.g., geometric edges or local normals) of the point map onto the 2D image plane using the current pose estimate $\hat{x}^i_{iter}$. This work then estimates the dense optical flow $\mathbf{v}_{flow}$ by tightly coupling the point depth with the updated IMU kinematics[151]. Since events are predominantly triggered by structural edges moving perpendicular to their orientation, this work evaluates the inner product between the projected 2D structural gradient and the local optical flow. As illustrated in Figure 4.7(a), structures whose projected gradients align closely with the optical flow (high kinematic saliency) are expected to trigger strong event responses. Conversely, events generated in regions where the structural gradient is nearly orthogonal to the optical flow are physically incongruent with the static map; thus, they are identified as dynamic noise and rigorously filtered out. Furthermore, this formulation is highly extensible: should the prior 3D map encode photometric information (e.g., textured point clouds), the exact photometric gradient could be utilized to explicitly predict the sign of the brightness change, enabling more fine-grained event filtering through strict polarity matching (Figure 4.7(b)).

Adaptive buffer zone. In highly dynamic flight, inherent numerical errors in the estimated optical flow are inevitable. To prevent the erroneous rejection of valid edges, this work introduces an adaptive buffer zone strategy that handles physical ambiguity through conservative retention, seamlessly adapting to the prior map's modality (Figure 4.7(c)). For a purely geometric LiDAR map, this work deploys a *kinematic margin* ($\epsilon_{kinematic}$). This work rigorously filters out events falling strictly within the orthogonal band (near 90°, indicating zero kinematic saliency). For regions exceeding this margin, this work conservatively retains the local TDF patches of *both* polarities ($\overline{\mathcal{T}}_{posi}$ and $\overline{\mathcal{T}}_{nega}$) to preserve potential geometries. Crucially, this formulation is highly extensible to textured maps via an additional *photometric filter*. By evaluating the exact inner product of the photometric gradient ∇L and the optical flow, the system can explicitly predict the event polarity. If the raw events strictly match this predicted sign of brightness change, only the TDF patch of the *corresponding polarity* is extracted. The buffer zone is then applied near the zero-crossing of brightness change: events with high absolute inner products receive strict polarity filtering, while events within the near-zero buffer zone—representing ambiguous, low-contrast variations—are conservatively retained without strict polarity assumptions. Operating in parallel, these two filters cumulatively assemble a fine-grained, noise-free aligned TDF ($\overline{\mathcal{T}}_{align}$) (Figure 4.7(d)).

Enhancing the efficiency of SE(3) optimization (Line 20-26). By dynamically assembling the structurally salient event subset ($\overline{\mathcal{T}}_{align}$) at each iteration, this mechanism drastically reduces the data bandwidth required for 2D-3D matching. In the alignment stage of Algorithm4.1, the residuals r_k and continuous Jacobians $\mathbf{J}_k$ are directly queried from the filtered $\overline{\mathcal{T}}_{align}(\mathbf{u}_k)$. The optimal pose increment Δx^* is then solved via a nonlinear least squares formulation, iteratively updating the SE(3) pose manifold ($\hat{x}^i_{iter} \boxplus \Delta x^*$) until convergence. Because optical flow prediction and event filtering become progressively more accurate as the pose converges, this tightly-coupled formulation maximizes the computational efficiency of drone pose estimation detailed in §4.4.3.

4.5.2 Later-stage Fusion: Graph-informed Joint Fusion and Optimization

4.5.2.1 Proprioceptive Motion Tracking

The IMU provides high-frequency proprioceptive measurements in the body frame $\mathtt{M}$. By applying the extrinsic spatial calibration between $\mathtt{M}$ and the event camera frame $\mathtt{E}$, this

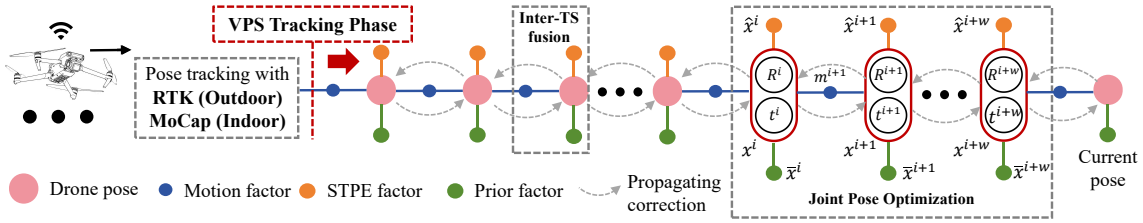


Figure 4.8 Illustration of the graph-informed joint fusion and optimization. The drone initially acquires its global pose via GPS (outdoors) or a Motion Capture system (indoors) before transitioning to the VPS tracking phase. Within the factor graph, the drone pose is jointly optimized by fusing exteroceptive measurements (STPE factors), proprioceptive kinematics (motion factors), and a constant velocity assumption (prior factors). The system executes Inter-TS fusion for immediate updates and Joint Pose Optimization over a sliding window to mitigate accumulated drift.

work maps these inertial measurements directly into the visual tracking space. The raw acceleration $\hat{\mathbf{a}}^i$ and angular velocity $\hat{\boldsymbol{\omega}}^i$ at time i are formulated by the standard kinematic sensor model: $\hat{\mathbf{a}}^i = \mathbf{a}^i + \mathbf{g} + \mathbf{b}_a + \mathbf{n}_a$ and $\hat{\boldsymbol{\omega}}^i = \boldsymbol{\omega}^i + \mathbf{b}_\omega + \mathbf{n}_\omega$, where $\mathbf{a}^i$ and $\boldsymbol{\omega}^i$ are the true kinematic states, $\mathbf{g}$ is the gravity vector, $\mathbf{b}_a, \mathbf{b}_\omega$ are the dynamic biases, and $\mathbf{n}_a, \mathbf{n}_\omega$ are Gaussian noise vectors.

To synchronize these signals with the asynchronous event streams (e.g., bridging consecutive time surfaces $\mathcal{T}^i_{posi}$ and $\mathcal{T}^{i+1}_{posi}$), this work incorporates the continuous-time quaternion-based IMU pre-integration approach[152]. This methodology allows us to analytically encapsulate the dense inertial readings into a single relative kinematic constraint, avoiding repetitive integration during state optimization. Consequently, this work compute relative motion prior $\mathbf{m} = \{\Delta\mathbf{x}^i, \Delta\mathbf{v}^i\}$, where spatial transform $\Delta\mathbf{x}^i = \{\Delta\mathbf{t}^i, \Delta\mathbf{R}^i\}$ strictly regularizes drone's pose increment.

4.5.2.2 Joint Fusion and Optimization

Leveraging the exteroceptive absolute pose from the STPE module, the proprioceptive relative motion from the IMU, and kinematic priors (e.g., drone flight characteristics), this work has designed a tightly coupled, factor graph-based joint fusion and optimization framework. This architecture integrates these independent yet complementary measurements to significantly enhance the accuracy and robustness of drone pose estimation[153].

Probabilistic formulation of pose optimization. Formally, estimating the optimal drone pose is equivalent to a Maximum A Posteriori (MAP) estimation. Assuming the sensor measurements are conditionally independent, the joint posterior probability factorizes into the product of individual likelihoods. By taking the negative logarithm, this maximization elegantly transforms into minimizing the sum of negative log-likelihoods (denoted as

energy terms E):

$$\begin{aligned}\hat{x}^i &= \arg\max_{x^i} \Big(p(x^i \mid x^{i-1}, x^{i-2}) \cdot p(\hat{x}^i \mid x^i) \\ &\qquad \cdot p(\Delta x^i \mid x^i, x^{i+1}) \Big) \\ &= \arg\min_{x^i} \Big(-\log p(x^i \mid x^{i-1}, x^{i-2}) \\ &\qquad -\log p(\hat{x}^i \mid x^i) - \log p(\Delta x^i \mid x^i, x^{i+1}) \Big) \\ &= \arg\min_{x^i} \Big(E^i_{Prior} + E^i_{STPE} + E^i_{Motion} \Big).\end{aligned} \tag{4.3}$$

Under the standard assumption that sensor measurements are corrupted by independent, zero-mean Gaussian noise, these negative log-likelihoods mathematically correspond to squared Mahalanobis distances, defined as $\|\mathbf{e}\|^2_{\boldsymbol{\Sigma}} \triangleq \mathbf{e}^T \boldsymbol{\Sigma}^{-1}\mathbf{e}$. This derivation seamlessly bridges the probabilistic formulation with the geometric L_2 norm space. Specifically, the factor graph minimizes the following three covariance-weighted heterogeneous factors:

• **Prior factor.** The prior factor represents the predictive probability distribution of the drone's pose at time i, denoted as $p(x^i)$, conditioned on previous states but independent of current sensor measurements. To derive this distribution, this work incorporates a constant velocity model—a well-established kinematic assumption in SLAM—which assumes that the drone maintains near-linear translational and rotational velocities over short intervals. Beyond providing a motion baseline, this prior acts as a crucial kinematic constraint that regularizes the trajectory, ensuring that the estimated states remain physically plausible during high-frequency update cycles. Therefore, the drone's prior state $\bar{x}^i = \{\bar{R}^i_{IE}, \bar{t}^i_{IE}\}$ is described by $\bar{R}^i_{IE} = R^{i-1}_{IE}(R^{i-2}_{IE})^T R^{i-1}_{IE}$ and $\bar{t}^i_{IE} = 2t^{i-1}_{IE} - t^{i-2}_{IE}$, with $p(x^i \mid x^{i-1}, x^{i-2}) \sim \mathcal{N}(\bar{x}^i, \boldsymbol{\Sigma}_{Prior})$. The prior factor is formulated as:

$$E^i_{Prior} = -\log p(x^i \mid x^{i-1}, x^{i-2}) \propto \|x^i - \bar{x}^i\|^2_{\boldsymbol{\Sigma}_{Prior}}. \tag{4.4}$$

• **STPE factor: *Exteroceptive* measurements.** The STPE likelihood, $p(\hat{x}^i \mid x^i)$, models the probability distribution of the exteroceptive absolute pose $\hat{x}^i$ given the true state x^i. Assuming this measurement is corrupted by Gaussian noise with a covariance matrix $\boldsymbol{\Sigma}_{STPE}$, the STPE factor is defined as:

$$E^i_{STPE} = -\log p(\hat{x}^i \mid x^i) \propto \|x^i - \hat{x}^i\|^2_{\boldsymbol{\Sigma}_{STPE}}. \tag{4.5}$$

• **Motion factor: *Proprioceptive* measurements.** The motion tracking likelihood, $p(\Delta x^i \mid x^i, x^{i+1})$, represents the distribution of the proprioceptive relative motion provided by the IMU. Since IMU pre-integration is subject to Gaussian noise with a covari-

ance matrix $\boldsymbol{\Sigma}_{Motion}$, this work formulates the motion factor as:

$$E^i_{Motion} = -\log p(\Delta x^i \mid x^i, x^{i+1}) \propto \|x^{i+1} - x^i - \Delta x^i\|^2_{\boldsymbol{\Sigma}_{Motion}}. \tag{4.6}$$

Figure 4.8 illustrates the construction and optimization of the proposed factor graph. This work aims to compute the optimal node values (poses) by jointly minimizing all heterogeneous factors through a two-stage process:

• **Inter-TS Fusion: Immediate state update.** Upon receiving the preliminary absolute pose from the STPE module, the Inter-TS fusion executes a lightweight, high-frequency state update to satisfy the strict latency demands of the flight controller. Specifically, under the MAP formulation, this immediate fusion problem is equivalent to minimizing the sum of the respective negative log-likelihoods (the Mahalanobis distances) over the current state:

$$\begin{aligned} \hat{x}^i &= \arg\min_{x^i} \left(E^i_{Prior} + \rho\left(E^i_{STPE}\right) + E^i_{Motion} \right), \\ x^i &= \left\{ R^i_{IE}, t^i_{IE} \right\}, \end{aligned} \tag{4.7}$$

where E^i_{Prior} and E^i_{Motion} serve as the kinematic and proprioceptive Mahalanobis residuals, and E^i_{STPE} is the exteroceptive measurement residual derived from the TDF alignment. Crucially, $\rho(\cdot)$ represents a robust loss function (e.g., Huber loss) applied to the visual residuals to mitigate the impact of outlier matches during severe dynamic bursts.

• **Joint Pose Optimization: sliding-window refinement.** To ensure long-term global consistency and mitigate accumulated drift, a periodic joint pose optimization is triggered. Operating within a sliding window framework, this backend module jointly refines all historical poses inside a given window $\mathcal{W}$ by minimizing the sum of the heterogeneous Mahalanobis cost terms. The joint pose optimization problem is rigorously modeled as:

$$\begin{aligned} \hat{\mathcal{X}} &= \arg\min_{\mathcal{X}} \sum_{i \in \mathcal{W}} \left(E^i_{Prior} + \rho\left(E^i_{STPE}\right) + E^i_{Motion} \right), \\ \mathcal{X} &= \bigcup_{i \in \mathcal{W}} \left\{ R^i_{IE}, t^i_{IE} \right\}. \end{aligned} \tag{4.8}$$

Both optimization problems Eqn.(4.7) and Eqn.(4.8) are efficiently solved within a tightly coupled incremental factor graph framework. By iteratively updating the SE(3) state manifold based on these heterogeneous factors, the solver bounds the estimation error, yielding a drift-free and globally consistent absolute trajectory.

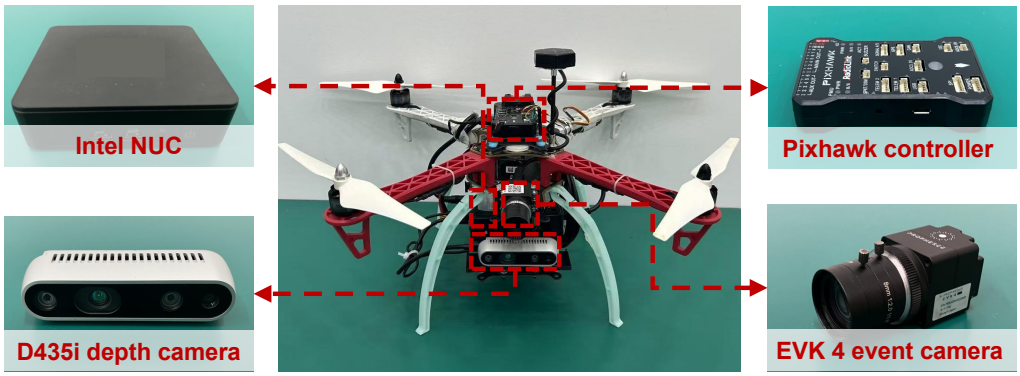


Figure 4.9 Testbed configuration of EV-Pose platform.

4.6 System Implementation

4.6.1 Testbed Configuration

As illustrated in Figure 4.9, EV-Pose is implemented using a 450 mm-wide drone equipped with (*i*) a Prophesee EVK4 HD event camera with 1280 × 720 resolution; (*ii*) a D435i depth camera for frame images capture; and (*iii*) a Pixhawk flight controller for drone control and IMU measurements. EV-Pose runs on an Intel NUC with a Core i7 CPU, 16GB RAM, and Ubuntu 20.04 operating system. All EV-Pose algorithms are implemented in C++ and ROS[154].

Prior map construction and evaluation protocol. To rigorously isolate and evaluate the exact cross-modality matching accuracy of the proposed system, the prior 3D point cloud map must be globally consistent. Therefore, this work constructs the prior map by accumulating the raw LiDAR scans strictly based on the high-fidelity 6-DoF Ground Truth poses provided by the indoor MoCap and outdoor RTK systems. This procedure eliminates any underlying mapping inaccuracies, ensuring that the prior map is flawlessly registered to the global GT coordinate frame.

4.6.2 Map Optimization

To better match the point cloud and event feature, this work introduces a map construction method that emphasizes object contours over textures by fitting planes to raw point cloud data and extracting contours from these planes. Specifically, point cloud data, represented as $p_i = (x_i, y_i, z_i)$, is processed using Principal Component Analysis (PCA) to compute normals and curvatures. For each point p_i, its k-nearest neighbors p_j define the covariance matrix, where eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2$ and eigenvectors v_0, v_1, v_2 are obtained via singular value decomposition. The normal is given by v_0 and the curvature at p_i

is: $\sigma(p_i) = \lambda_0/(\lambda_0 + \lambda_1 + \lambda_2)$. Plane fitting is performed using a region-growing algorithm that groups points based on the angular difference between their normals. Smoothness constraints identify clusters, forming segmented planes. A contour-tracking algorithm then extracts closed boundary contours from the segmented regions.

4.6.3 Push Limit of Accuracy and Efficiency

Ego-motion instructed infrastructure extraction. Moving objects within the FoV often disrupt alignment and need to be filtered. This work proposes a motion analysis method to identify and prioritize stationary objects, which provide reliable edge features for localization. For a sequence of events $\mathcal{E}$ and IMU data $\mathcal{M}$ over an interval $[0, \delta i]$, each event $e = (\boldsymbol{x}, i, p)$, where $\boldsymbol{x}$ and p represent pixel coordinates and polarity, is analyzed. The predicted location $\hat{\boldsymbol{x}}_{\delta i}$ is derived by integrating IMU data, followed by coordinate transformations. If $e_{\delta i}$ is triggered by stationary objects, predicted location $\hat{\boldsymbol{x}}_{\delta i}$ should align with actual location $\boldsymbol{x}_{\delta i}$, as changes in stationary objects are due to drone motion.

Perspective movement in point cloud. In Sec. 4, this work detailed the 2D event-3D point map algorithm. To further improve pose estimation efficiency, once the initial pose is established, this work restricts the point cloud range using the drone's current pose and FoV. Only points within the FoV are used for alignment. As the drone moves, the perspective dynamically shifts to include new regions of the point cloud for pose estimation. To accommodate potential errors in pose estimation, this work expands the perspective range. Also, this work integrates occlusion handling by prioritizing nearby point clouds to the estimated pose, improving matching efficiency.

4.7 Experimental Evaluation

This section presents a comprehensive evaluation of the *EV-Pose* system through both field studies and dataset-based benchmarks. Section 4.7.1 first details the experimental methodology, including the testbed configuration and the evaluation metrics employed. The subsequent subsections provide an in-depth analysis of the empirical results: Section 4.7.2 evaluates the overall performance of *EV-Pose* against SoTA baselines, while Section 4.7.3 examines the system's robustness and accuracy under various extreme operational conditions. Finally, Section 4.7.4 conducts micro-benchmarks to validate the specific contributions of each algorithmic module within the hierarchical architecture.



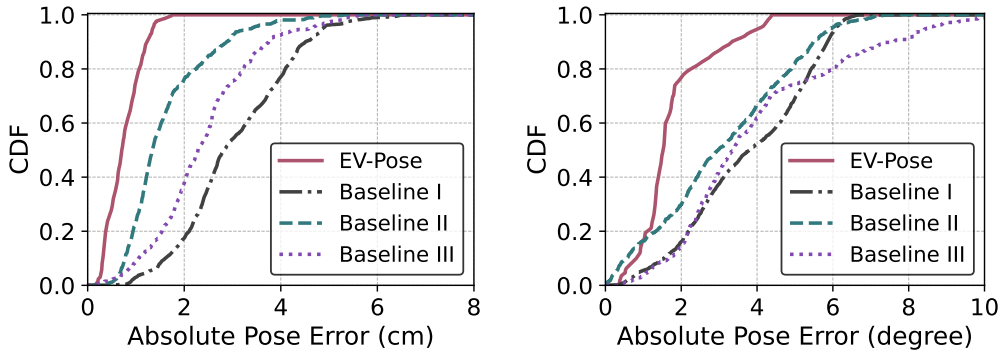
(a) Indoor experiment setup (b) Outdoor experiment setup (c) Window contrast (d) Flickering light (e) Rotating fan (f) Walking people

Figure 4.10 Overview of the real-world experimental setups and challenging test scenarios. (a) Indoor experiment setup equipped with a motion capture system and a prior 3D point map. (b) Outdoor experiment setup utilizing an RTK station and an outdoor 3D point map. (c) Testing scenario featuring window contrast with high luminance differences. (d) Testing scenario with a flickering light generated by a controllable spotlight. (e) Dynamic interference scenario with a rotating fan and its corresponding generated events. (f) A dynamic environment scenario involving walking people.

4.7.1 Experiment Methodology

Field studies. To comprehensively evaluate the pose tracking performance of EV-Pose, this work conducts extensive field studies in both indoor and outdoor environments, as illustrated in Figure 4.10. The system operates within globally aligned, pre-built 3D point cloud maps for absolute pose estimation. Over the course of the evaluation, this work accumulates more than 20 hours of flight experiments, yielding over 200 GB of raw sensor data. *(i) Ground Truth and Initialization:* For the indoor evaluations, experiments are conducted in an 8 × 8 × 8 m^3 test field. Ground truth is captured by a high-fidelity motion capture system equipped with 16 infrared cameras operating at 210 FPS[155]. In the absence of GPS, this system also provides the precise initial 6-DoF pose required to bootstrap the EV-Pose tracking, mirroring the role of the RTK station in the outdoor setup. Conversely, in outdoor settings, this work deploys a private RTK base station (Hi-Target D8) to provide both the initial absolute pose and the highly accurate ground truth trajectory[156]. *(ii) Trajectory Design:* To rigorously assess the system's robustness under continuous dynamic motion, the drone is commanded to execute a square spiral trajectory. Specifically, the spiral expands outward from the center, with the side length increasing by 1 m per loop up to a maximum boundary of 5 × 5 m. Each flight trial lasts for approximately 60-90 seconds, covering a continuous trajectory distance of around 60 meters.

Dataset-based studies. This work also performs extensive experiments on public datasets to evaluate the performance of EV-Pose thoroughly. Specifically, this work tests on two datasets to demonstrate its effectiveness across different scenarios: (*i*) the TUM-VIE dataset (denoted as T)[157], (*ii*) the VECtor dataset (denoted as V)[158]. The 3D point reference maps for both datasets are provided by depth cameras.



(a) CDF of ATE (b) CDF of ARE

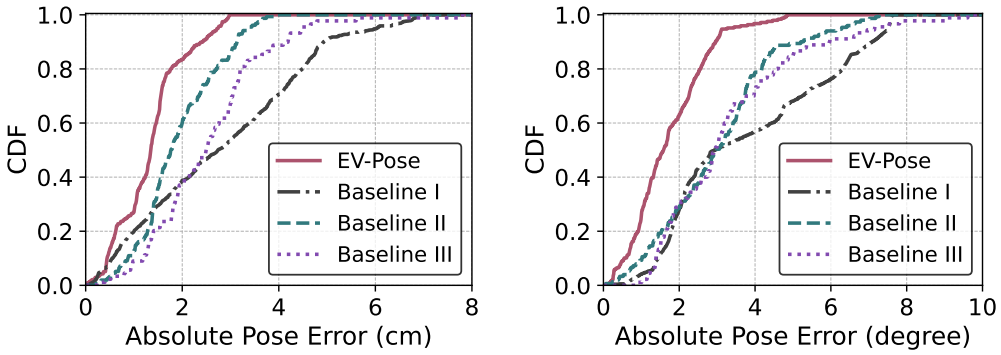
Figure 4.11 Real-world indoor experiments.

EV-Pose	Baseline-I	Baseline-II	Baseline-III
10.08 *ms*	23.11 *ms*	31.76 *ms*	29.27 *ms*

Table 4.1 Latency of different systems.

Comparative methods. To comprehensively evaluate the performance of EV-Pose, which is the first *event camera-enhanced VPS*, this work compares it with three systems: ***(i)*** *Baseline I[159]: A SOTA event camera-based VIO*, which uses event streams and IMU data for drone pose estimation (the event–IMU version of[159]), validating the benefit of incorporating prior 3D point maps in EV-Pose. Since[159] does not provide publicly available code, for a fair comparison, this work implements it based on[64] by adding line-based event features to the feature set used in VIO-based pose estimation. ***(ii)*** *Baseline II[44]: A SOTA frame camera-based VPS*. This method first extracts point and line features from frame images and integrates IMU data for VIO. By leveraging prior 3D point maps, it then establishes 2D-3D correspondences through feature matching between 2D images and 3D map points, enabling drone pose estimation. This method is used to evaluate the effectiveness of EV-Pose in redesigning drone-oriented VPS with event cameras. Since[44] does not provide publicly available code, for a fair comparison, this work implements it based on[160] by adding point and line features-based VIO. ***(iii)*** *Baseline III[118]: A SOTA frame camera-based SLAM method*, which estimates the drone's pose using monocular images and IMU measurements. This method is used to assess the benefits of upgrading frame cameras to event cameras and incorporating prior 3D point maps.

Evaluation protocol and metrics. To ensure a rigorous and fair comparison, all systems are evaluated offline using the identical sensor datasets recorded during the real-



(a) CDF of ATE (b) CDF of ARE

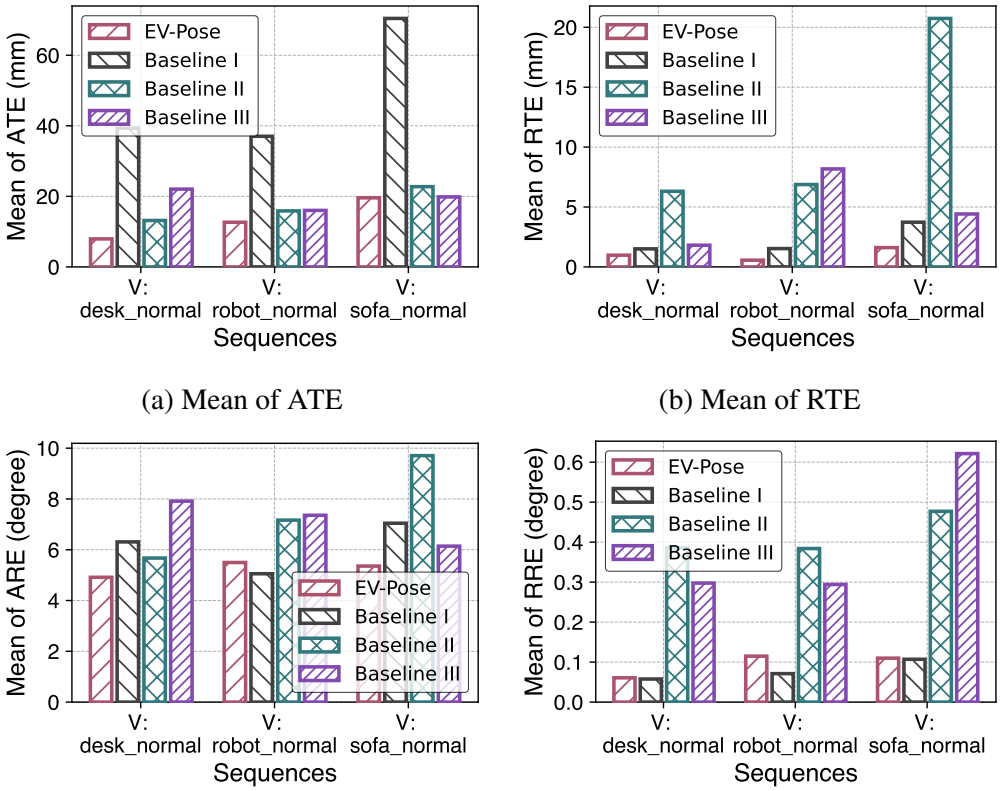
Figure 4.12 Real-world outdoor experiments.

world flights. Specifically, the frame-based baselines are fed with standard images and IMU data, while the event-based methods (Baseline I and EV-Pose) process the corresponding event streams and IMU measurements. To thoroughly assess the 6-DoF pose estimation performance, this work employs four quantitative metrics: (*i*) absolute translation error (ATE); (*ii*) absolute rotation error (ARE); (*iii*) relative translation error (RTE); and (*iv*) relative rotation error (RRE). The absolute errors (ATE, ARE) measure the global accuracy of the drone's estimated trajectory, while the relative errors (RTE, RRE) assess the local consistency and smoothness of pose estimates over time—both of which are fundamentally crucial for stable drone flight control. Because EV-Pose and Baseline II are absolute map-based localization systems, they naturally yield poses in the global frame. Consequently, to establish rigorous performance comparison, this work computes their trajectory directly against the GT trajectory *without* any SE(3) trajectory alignment, thereby avoiding any partial compensation for global coordinate discrepancies.

Robustness evaluation. To validate the robustness of EV-Pose, this work conducts field studies in a controlled indoor environment as shown in Figure 4.10(c)-Figure 4.10(f), evaluating its performance under various dynamic conditions, including different drone velocities, frequencies, HDR conditions, and scene dynamics.

4.7.2 Overall Performance

Field studies result. This work first evaluates EV-Pose's pose tracking performance in an indoor environment of Figure 4.10(a). As shown in Figure 4.11, EV-Pose achieves a mean ATE of 7.5*mm*, outperforming Baseline I, Baseline II, and Baseline III by 75.2%, 53.3%, and 68.5%, respectively. For rotation, its mean ARE is 1.76°, with improvements of 53.3%, 42.4%, and 55.5% over all baselines. Furthermore, this work assesses EV-



(a) Mean of ATE

(b) Mean of RTE

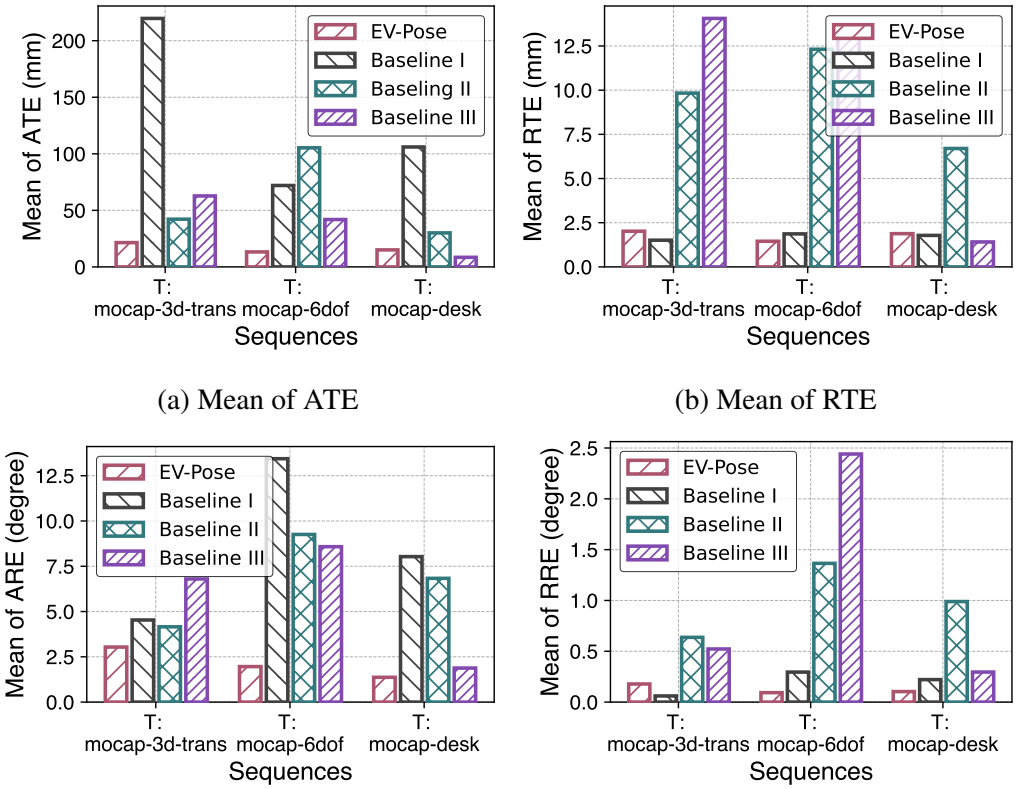
(c) Mean of ARE

(d) Mean of RRE

Figure 4.13 Overall performance of *EV-Pose* and three SOTA 6-DoF pose tracking systems on VECtor dataset.

Pose in an outdoor environment of Figure 4.10(b). As shown in Figure 4.12, EV-Pose achieves a mean ATE of 12.9 *mm*, again surpassing the three baselines by 54.2%, 32.6%, and 48.4%, respectively. For rotation, its mean ARE is 1.73 °, with improvements of 54.1%, 41.1%, and 47.6% over all baselines. During operation, EV-Pose maintains a latency of 10.08*ms*, compared to Baseline I, II, and III latencies of 23.11*ms*, 31.76*ms*, and 29.27*ms*, as shown in Tab. 4.1. The superior performance of EV-Pose stems from two key factors: (*i*) The integration of a 3D point reference map and 2D event-3D point matching significantly enhances pose tracking, outperforming Baseline I and III through the use of external information. (*ii*) Motion blur from drone flight impairs edge detection in RGB images, while their low frame rate causes large pose changes in dynamic scenes. In contrast, event cameras with *ms*-level latency, unaffected by motion blur, enable rapid and accurate edge tracking. EV-Pose's high pose tracking frequency leads to smaller pose changes, further enhancing tracking accuracy.

Dataset-based studies result. This work further evaluates the performance of EV-Pose using public datasets. Figure 4.13 and Figure 4.14 illustrate the overall 6-DoF pose



(a) Mean of ATE

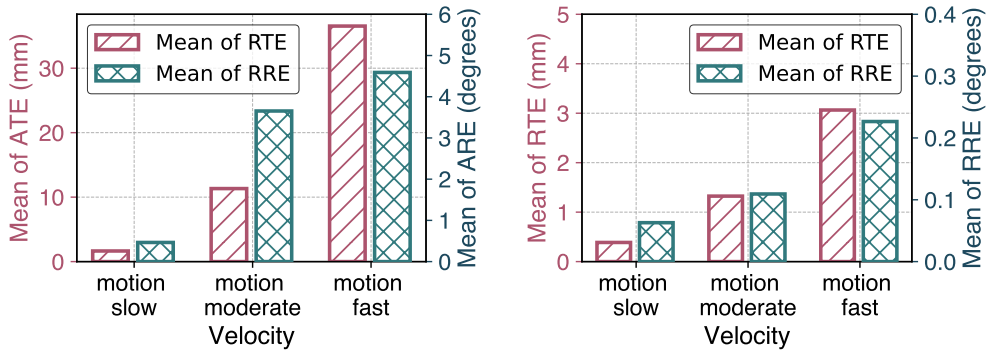
(b) Mean of RTE

(c) Mean of ARE

(d) Mean of RRE

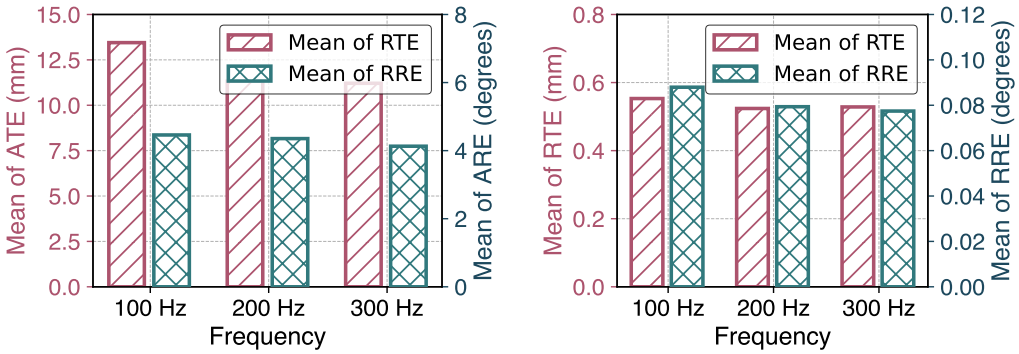
Figure 4.14 Overall performance of *EV-Pose* and three SOTA 6-DoF pose tracking systems on TUM-VIE dataset.

tracking performance of EV-Pose compared to three competitive SOTA methods. Figure 4.13 shows EV-Pose's performance on the VECtor dataset, where it consistently achieves superior accuracy across various scenarios. EV-Pose has a mean ATE of 13.19*mm*, outperforming Baseline I, Baseline II, and Baseline III by 72.63%, 22.42%, and 30.62%, respectively. EV-Pose's mean of ARE is 1.05°, outperforming the same methods by 53.33%, 90.72%, and 78.13%. For RTE, EV-Pose's mean of RTE is 5.25*mm*, surpassing the others by 14.36%, 30.09%, and 26.37%. Lastly, its mean of RRE is 0.095 °, beating other methods by up to 77.14%. Figure 4.14 presents results on the TUM-VIE dataset, showing similar trends. The mean of ATE is 16.59*mm*, ARE is 2.12°, RTE is 1.78*mm*, and RRE is 0.12°, with EV-Pose outperforming Baseline I, II, and III in each metric. EV-Pose excels in challenging environments with dynamic changes, fast motions, and HDR conditions. By upgrading the frame camera to an event camera, EV-Pose consistently demonstrates resilience across various conditions and achieves lower errors. This underscores its robustness and reliability as a solution for real-world 6-DoF pose tracking applications.



(a) Absolute estimation error (b) Relative estimation error

Figure 4.15 Impact of drone moving velocity.



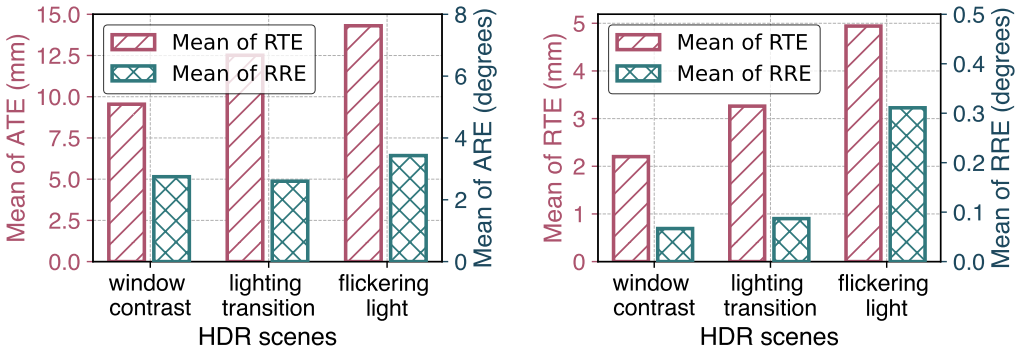
(a) Absolute estimation error (b) Relative estimation error

Figure 4.16 Impact of drone estimation frequency.

4.7.3 Robustness Evaluation

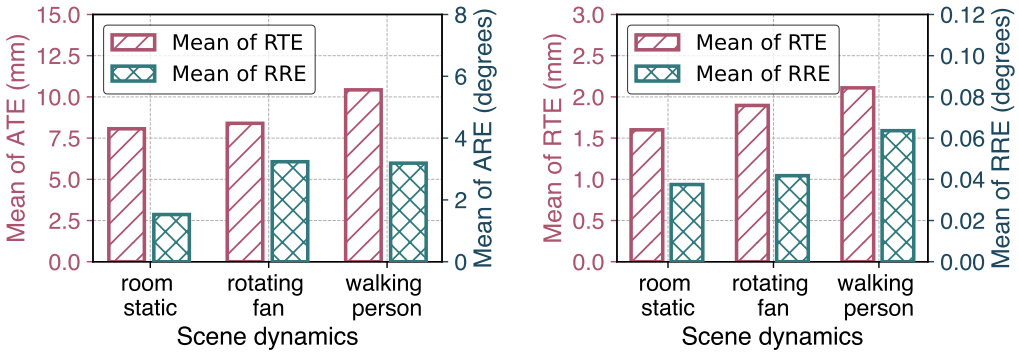
The robustness experiments primarily focus on evaluating EV-Pose's performance under different extreme conditions.

Impact of drone moving velocity. To evaluate the impact of drone velocity on EV-Pose, this work conducts experiments at various speeds. As shown in Figure 4.15, this work separately measures both absolute and relative errors across different translation and rotation velocities. For translation velocity, this work categorizes motion into slow (v<2 m/s), moderate (2 m/s < v < 4 m/s), and fast (v > 4 m/s) to assess performance. The mean ATE of EV-Pose is 1.6 *mm*, 11.3 *mm*, and 36.7 *mm* for slow, moderate, and fast motion, respectively. Regarding relative errors, the mean RTE is 0.38 *mm*, 1.32 *mm*, and 3.06 *mm* for corresponding speed categories. Similarly, for rotation velocity, this work classifies motion into slow ($\omega < 10°$), moderate ($10° < \omega < 30°$), and fast ($\omega > 30°$). The mean ARE is 0.46°, 3.64°, and 4.58°, while mean RRE is 0.06°, 0.109°, and 0.26°. These results highlight the robustness of EV-Pose across different velocities, thanks to its



(a) Absolute estimation error (b) Relative estimation error

Figure 4.17 Impact of HDR.



(a) Absolute estimation error (b) Relative estimation error

Figure 4.18 Impact of scene dynamics.

integration of an event camera with a 3D point map, which enables tracking at high speeds without being affected by motion blur.

Impact of estimation frequency. This work examines the impact of estimation frequency on the performance of EV-Pose, as shown in Figure 4.16. The experiments include varying frequencies: $100Hz$, $200Hz$, and $300Hz$. As the frequency increased from $100Hz$ to $300Hz$, the mean of ATE decreased from $13.4mm$ to $11.2mm$, the mean of ARE from 4.46° to 4.13°, the mean of RTE from $0.55mm$ to $0.52mm$, and the mean of RRE from 0.087° to 0.077°. Higher frequencies result in smaller changes between drone poses, providing EV-Pose with more information for accurate pose estimation. The results demonstrate the robustness of EV-Pose to varying frequencies, making it a promising solution for high-frequency pose estimation in fast-moving drones.

Impact of HDR. This work investigates the influence of HDR on EV-Pose. Various HDR scenarios are triggered: (*i*) *window contrast*: luminance differences between different areas (Figure 4.10(c)); (*ii*) *lighting transition*: switching from lights-on to lights-off

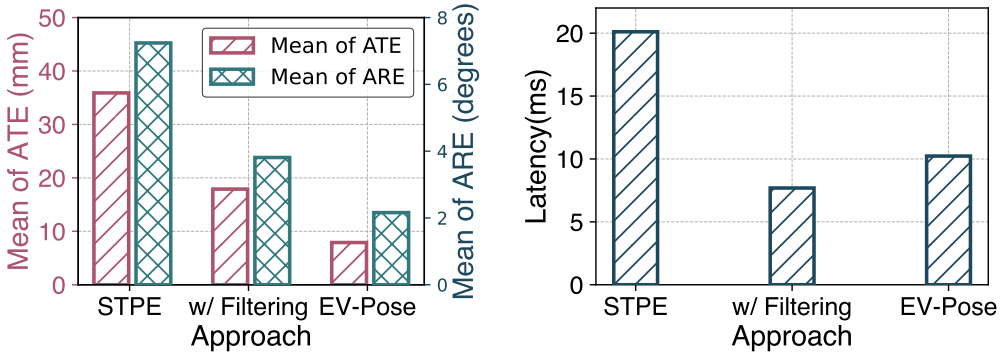
in area of Figure 4.10(a); (*iii*) *flickering light*: the flickering light generated by rapidly switching a controllable spotlight (Figure 4.10(d)). As shown in Figure 4.17, EV-Pose's mean of ATE, ARE under different scenarios are 9.5*mm*, 12.5*mm*, 14.3*mm*; 2.74°, 2.60°, 3.43°; mean of RTE, and RRE are 2.2*mm*, 3.26*mm*, 4.9*mm*; and 0.067°, 0.087°, 0.31°, respectively. EV-Pose demonstrates stability under HDR, highlighting its potential for reliable performance in real-world environments with varying lighting conditions.

Impact of scene dynamics. This work finally assesses the impact of scene dynamics on performance. The setup included: (*i*) *room static*: a completely static room (Figure 4.10(c)); (*ii*) *rotating fan*: a running fan produces numerous events, moderately rendering scene modeling outdated (Figure 4.10(e)); (*iii*) *walking people*: several people walking across the test area (Figure 4.10(f)). Results are shown in Figure 4.18. The mean of ATE for EV-Pose in each scenario is 8.1*mm*, 8.4*mm*, and 10.4*mm*, respectively. The mean of ARE is 1.53°, 3.24°, and 3.19°. The mean of RTE is 1.6*mm*, 1.9*mm*, and 2.1*mm*. The mean of RRE is 0.037°, 0.042°, and 0.063°. EV-Pose demonstrates resilience against these challenges. By combining the drone's motion information as *proprioceptive* measurements with 3D point map–based *exteroceptive* measurements, EV-Pose effectively isolates and removes the impact of dynamic objects. This ensures accurate and consistent results under varying scene dynamics and mitigates issues caused by outdated scene models, strengthening EV-Pose's reliability in dynamic environments.

4.7.4 Micro Benchmarks

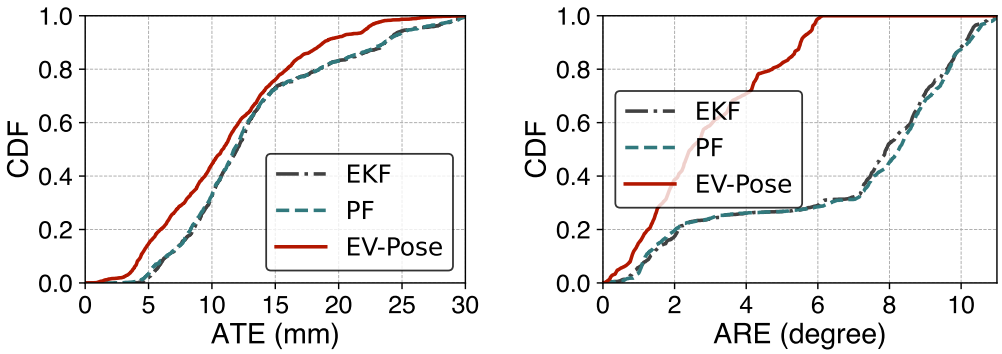
Validating the hierarchical architecture. To rigorously evaluate how each algorithmic component contributes to the EV-Pose framework, this work conducts an incremental ablation study on the pure-event baseline. The configurations are designed to validate the theoretical claims: (*i*) *Pure exteroceptive tracking (STPE only)*: Relies solely on 2D event-3D map cross-modality alignment without kinematic priors. (*ii*) *w/ early-stage event filtering*: Integrates early-stage IMU motion information to structurally filter asynchronous events. (*iii*) *Full EV-Pose*: Applies the complete MHFO, coupling the front-end with the later-stage SE(3) manifold optimization.

As shown in Figure 4.19(a), the baseline (STPE only) yields a mean ATE of 35.9 *mm* and an ARE of 7.12 °. By introducing the front-end filtering (§5.1), these errors gracefully decrease to 24.2 *mm* and 6.67°. Ultimately, with the full tightly-coupled MHFO (§ 5.3), EV-Pose achieves an exceptional ATE of 6.9 *mm* and ARE of 1.34°. Furthermore, the latency analysis in Figure 4.19(b) reveals a profound system insight: incorporating mo-



(a) CDF of ATE (b) CDF of ARE

Figure 4.19 Impact of modules.



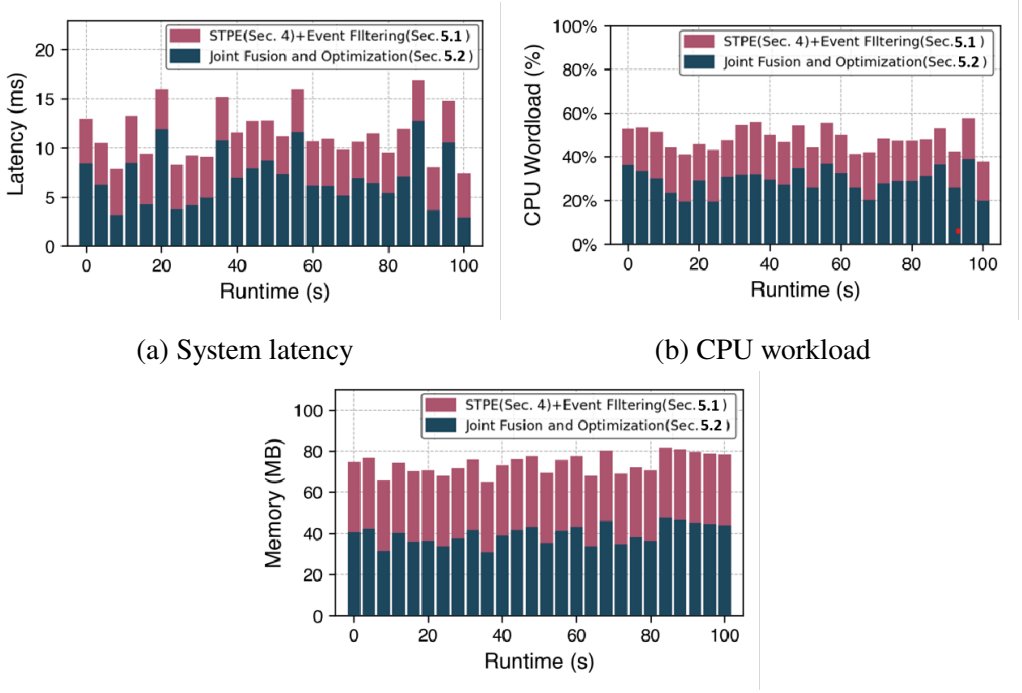
(a) CDF of ATE (b) CDF of ARE

Figure 4.20 Comparison of different fusion frameworks.

tion information structurally reduces the event processing burden, slashing latency from 20.11 *ms* to 7.69 *ms*. The subsequent factor graph optimization only marginally increases the latency to 10.08*ms*, proving that the hierarchical design successfully strictly bounds absolute error without compromising the microsecond-level agility of the event camera.

Impact of event filtering on TDF generation. Beyond mere computational reduction, Figure 4.19 illustrates the mathematical necessity of early-stage filtering for the cross-modality alignment. By proactively rejecting over *40%* of dynamically irrelevant events (e.g., noise bursts), the front-end prevents the corruption of the SP-TS. This ensures the extracted TDF maintains clean, anisotropic spatial gradients, fundamentally preventing the non-linear solver from falling into local minima during the 2D-3D matching phase.

Non-linear factor graph vs. linearized filtering. To validate the choice of the back-end solver, this work benchmarked the MHFO against traditional recursive filtering frameworks: the Particle Filter (PF) and the Extended Kalman Filter (EKF). As shown in Figure 4.20, EV-Pose drastically outperforms PF and EKF, improving translation tracking by over



(a) System latency

(b) CPU workload

(c) Memory usage

Figure 4.21 Resource overhead.

25.41% and 25.7%, and rotation tracking by 57.81% and 58.43%.

Comparison of fusion frameworks. This work compared the factor graph-based fusion framework with two other methods: particle filter (PF) and extended Kalman filter (EKF). As shown in Figure 4.20, EV-Pose improves translation tracking by over 25.41% and 25.7% compared to PF and EKF, and rotation tracking by over 57.81% and 58.43%. This performance is attributed to the MHFO module's ability to tightly integrate exteroceptive and proprioceptive measurements.

This substantial performance gap highlights a critical theoretical advantage of the architectural design. EKF inherently relies on a single-step, first-order linearization based on the Markov assumption. If the preliminary pose from the STPE frontend contains transient noise due to extreme maneuvers, EKF commits to this suboptimal linearization point, causing irreversible accumulation of error. In contrast, the factor graph framework formulates the fusion as a tightly-coupled, sliding-window optimization. By iteratively re-evaluating the non-linear IMU pre-integration and exteroceptive pose residuals until convergence, MHFO gracefully smooths out high-frequency frontend noise and strictly preserves global consistency on the exact SE(3) manifold.

Latency analysis. As illustrated in Figure 4.21(a), this work evaluates the end-to-end execution latency of EV-Pose, which comprises the processing times of both the *STPE* +

Event Filtering and the *Joint Fusion and Optimization* modules. The system achieves an overall average latency of 10.08 ms, with the former contributing 3.05 ms and the latter 7.03 ms. This performance readily satisfies the stringent real-time requirements of high-frequency drone flight control loops. As shown in Figure 4.21(a), the execution latency of EV-Pose occasionally exceeds 15 ms under worst-case conditions. This is attributed to the fact that, although event filtering in the STPE effectively reduces the time required for map matching, the backend factor graph incorporates several poses for joint optimization to enhance estimation accuracy, which consequently leads to increased optimization time.

Resource Overhead. Figure 4.21(b) and Figure 4.21(c) show that EV-Pose's CPU usage remains <60%, with average memory usage < 80 MB, including 7 MB for storing point maps. These results highlight EV-Pose's minimal resource overhead, making it suitable for resource-constrained drone systems.

4.7.5 Case Study: System-level Closed-loop Flight Control

To fully demonstrate the "perception-decision-action" capability of the proposed EV-Pose framework, this work conducts preliminary system-level closed-loop flight control experiments. The multimedia demonstrations of these real-world flights are publicly available on our project website①. Crucially, in this experiment, EV-Pose is not evaluated as an offline, open-loop trajectory estimator. Instead, the real-time, high-frequency 6-DoF pose estimations generated by the onboard event camera are continuously streamed to the drone's PX4 flight controller via the MAVROS protocol. Acting as the primary external vision state estimator, the flight controller's Extended Kalman Filter fuses these ultra-low latency updates to dynamically generate motor control commands (PWM signals). Relying on the event-driven perception loop, the drone successfully executes autonomous physical maneuvers, including stable hovering and dynamic trajectory tracking, without the assistance of GPS or conventional frame-based visual odometry. The successful execution of these physical actions validates that EV-Pose effectively closes the control feedback loop. Fundamentally minimizing the perceptual phase delay, it provides a "enacted" hardware foundation for intelligent agents operating in real-world environments.

① Project website: https://ev-pose.github.io/

4.8 Discussion

Why do drones need to introduce event cameras to upgrade existing VPS? Existing VPS based on RGB frame cameras face two critical physical bottlenecks during drone landings: First, they are highly susceptible to severe motion blur under rapid movement or complex lighting conditions, which disrupts feature matching and causes pose estimation errors to surge. Second, their low sampling rates and the high latency of 2D-3D global map matching are severely disconnected from the flight controller's rigid demand for high-frequency, real-time feedback. This forces the drone to rely on IMU-based "blind flight" for extended periods, increasing the risk of losing control due to accumulated errors. The introduction of event cameras aims to shatter these bottlenecks at the hardware level. By asynchronously responding to pixel-level brightness changes, event cameras boast an ultra-high dynamic range (120 dB) and complete immunity to motion blur. Furthermore, their microsecond-level temporal resolution reduces tracking latency. This enables ultra-high-frequency, low-latency 6-DoF pose tracking perfectly aligned with the flight controller's requirements, empowering drones to break free from the "slow landing" compromise imposed by traditional visual systems and achieve safer, more agile, and highly efficient precision landings.

However, a potential counter-argument is that global shutter cameras can be used to mitigate motion blur, and that any residual motion blur after using global shutter cameras is likely minimal since drones move at low speeds during landing. To address this concern, the response is as follows:

(i) The fundamental difference between rolling shutter effects and motion blur in real-world lighting. While the authors agree that global shutter cameras successfully eliminate rolling shutter artifacts (e.g., the "jello" effect), they fundamentally cannot eliminate motion blur if the exposure time is non-negligible. In real-world urban delivery scenarios, drones frequently land in sub-optimal lighting conditions, such as within deep urban canyons, under building shadows, or during twilight. To maintain an acceptable signal-to-noise ratio in these environments, frame-based cameras are forced to increase their exposure time (e.g., to 20-30 ms). Within this prolonged exposure window, any relative motion between the camera and the scene inherently produces motion blur, which degrades feature extraction and tracking. Event cameras, with their HDR and microsecond-level temporal resolution, are completely immune to this exposure-time tradeoff, guaranteeing sharp edge capture regardless of lighting conditions.

(ii) The impact of high-frequency angular velocity and proximity optical flow during the landing phase. Regarding landing speeds, it is critical to distinguish between macro-level translational velocity and micro-level angular velocity. Even if a drone descends slowly (e.g., < 1 m/s), it inevitably encounters complex aerodynamics near the ground, such as the "ground effect" or sudden urban wind gusts. To maintain stability, the flight controller must execute high-frequency attitude adjustments (pitch and roll). These rapid angular motions generate substantial rotational velocity. Furthermore, as the drone approaches the landing pad, the distance to ground features shrinks. At proximity, even slight translational movements induce massive pixel-level optical flow. Both of these factors easily overwhelm standard frame cameras, causing feature tracking failures that event cameras can effortlessly bypass.

(iii) Breaking the "slow landing" compromise for higher efficiency. The authors respectfully argue that the current practice of landing at low speeds is not an optimal design, but rather a necessary compromise dictated by the tracking limitations of existing frame-based visual systems. In commercial drone delivery applications, operational efficiency and throughput are primary objectives. Forcing drones to artificially slow down during the landing phase to accommodate the low frame rate and high latency of RGB cameras restricts overall delivery efficiency. A fundamental motivation behind EV-Pose is to break this hardware bottleneck. By providing ultra-low-latency and high-frequency pose updates, EV-Pose not only ensures safety under challenging dynamics but also actively empowers drones to land efficiently, paving the way for high-throughput delivery systems.

(iv) Guaranteeing flight safety during the landing phase. Unlike general flight scenarios, drone landing involves precise and rapid attitude control because the size of the landing pad is limited, and the drone must land accurately on the pad itself rather than merely in its vicinity. Failure to do so may lead to severe consequences, such as equipment damage or personal injury. This is especially true in urban deployment scenarios (e.g., delivery lockers installed in crowded areas), where even a single landing failure is unacceptable. EV-Pose provides pose updates at >100 Hz, ensuring that the drone can continuously perform fine-grained pose adjustments using high-quality pose estimation data. This prevents the system from relying solely on the onboard IMU—which provides only uncalibrated, coarse location updates—thereby maximizing flight safety during the landing process.

What is the difference between Event-enhanced VPS from Event-based VIO?

While event-based Visual-Inertial Odometry (VIO) has seen tremendous progress over the past decade, providing highly robust relative ego-motion estimation, EV-Pose fundamentally diverges from these systems: EV-Pose is not an Event-based VIO system that estimates drone *relative ego-motion*; rather, it is the first event-enhanced Visual Positioning Service (VPS) which provides the *absolute global-coordinates* required for precise drone landing. This work advances the field not merely by changing the operational objective, but through specific algorithmic innovations designed for cross-modality global matching. The core differences are outlined as follows: *(i) Operational objective: relative odometry vs. absolute global positioning.* Traditional event-based VIO systems track features across continuous temporal event streams to compute relative motion. While effective, they inevitably suffer from cumulative drift and cannot anchor the drone to a specific global coordinate. EV-Pose addresses the exact needs of accurate drone landing by functioning as a Visual Positioning Service (VPS), estimating drift-free, absolute 6-DoF poses directly matching against a pre-built global 3D point cloud map. *(ii) Algorithmic innovation in matching: temporal tracking vs. cross-modality alignment.* The transition from VIO to VPS introduces the unique challenge of cross-modality matching. Existing VIOs track visual patterns between consecutive event frames. EV-Pose, however, must align 2D asynchronous, texture-sparse events with a static 3D spatial point cloud, a scenario where standard photometric descriptors fail. To bridge this modality gap, this work proposes the Spatio-Temporal Feature-instructed Pose Estimation (STPE) module. By introducing the Separated Polarity Time-Surface (SP-TS) and extracting a Temporal Distance Field (TDF), the algorithm successfully transforms sparse asynchronous events into an anisotropic distance field, formulating the 2D-3D alignment as a solvable minimization problem. *(iii) Algorithmic innovation in fusion: reactive tracking vs. proactive hierarchical optimization.* In standard event-IMU setups, IMU data is predominantly used for reactive backend pre-integration. EV-Pose innovates the system architecture via the Motion-aware Hierarchical Fusion and Optimization (MHFO) scheme. At the early stage, it deeply integrates the prior 3D map into the frontend, leveraging IMU kinematics and 3D structural gradients to proactively predict optical flow and filter out dynamic noise. At the later stage, it employs a customized factor graph that explicitly incorporates a constant velocity prior, seamlessly regularizing the state manifold to ensure numerical stability even during visually degenerate periods.

What is the impact of event bursts during highly dynamic scenes on EV-Pose

optimization time? The computational latency of EV-Pose primarily stems from three sequential stages: *(i) Event representation:* Processing the raw event stream to generate the Separated Polarity Time-Surface (SP-TS). *(ii) Event filtering and map matching:* Leveraging motion data and the 3D environmental point cloud to filter the event stream. The filtered events are then used to generate a Temporal Distance Field (TDF), which is matched against the 3D point map to yield a coarse pose estimation. *(iii) Factor graph optimization:* Jointly optimizing the coarse pose estimates alongside motion measurements within a factor graph to produce the final fine pose estimation. Notably, event bursts during highly dynamic scenes predominantly exacerbate the latency during the second stage. Specifically, the combination of high-speed ego-motion and complex environmental textures triggers massive event bursts. Consequently, EV-Pose must evaluate and filter a larger volume of noisy events to construct the TDF and perform efficient 3D map matching for pose estimation. Although the current filtering mechanism in EV-Pose successfully eliminates a substantial portion of redundant events, thereby reducing matching latency, further mitigation of the computational burden induced by severe event bursts remains a viable direction for future improvement. A viable strategy is to optimize the map construction phase by retaining only rigid, static structures (e.g., buildings) and preemptively removing non-rigid point clouds (e.g., trees, foliage, and pedestrians). Furthermore, during the active event filtering phase, events located in patches without any projected 3D map points could be directly discarded from the map matching stage. This dual-pronged approach would drastically reduce the total volume of processed events, thereby shrinking the search space and accelerating the optimization process.

What is the role of the constant velocity prior in the factor graph formulation? A key design consideration in the factor graph is the explicit inclusion of a constant velocity prior alongside IMU pre-integration. While IMU signals provide high-frequency proprioceptive measurements, the performance of an event-based VPS can be sensitive to the spatial distribution of events. Specifically, this prior serves three critical functions: *(i) Kinematic Regularization:* Event cameras respond exclusively to intensity changes, meaning the exteroceptive constraints from the STPE module can become sparse or geometrically degenerate (e.g., when facing texture-less surfaces). In such cases, the constant velocity prior prevents the optimizer from converging to erratic solutions. *(ii) Compensating for Sensor Noise:* Over the short intervals typical of event-based updates, the integration of high-frequency IMU noise may lead to rapid drift. The prior provides a smooth

motion that suppresses these short-term fluctuations. *(iii) Numerical Stability:* By constraining the state transitions within a physically plausible range, the prior ensures the positive definiteness of the information matrix in the factor graph, thereby accelerating convergence and enhancing the numerical stability of the incremental solver (e.g., iSAM2 or Ceres) during aggressive maneuvers.

Hardware SWaP and flight dynamics. While the Intel NUC provides reliable computational headroom for rapid prototyping, it introduces Size, Weight, and Power (SWaP) considerations. Regarding power draw, the NUC's electrical consumption ($\sim$40 W) is an order of magnitude lower than the drone's mechanical propulsion power (>400 W), having a negligible impact on overall endurance. Specifically, using a 4S 5200mAh battery, the fully-loaded platform safely sustains 12-15 minutes of dynamic flight, which is sufficient for evaluation trials. Regarding flight dynamics, the added payload ($\sim$500 g) slightly increases mechanical inertia. The PID control loops of the flight controller efficiently compensate for this static weight, ensuring stable trajectory tracking without control degradation. Since the current focus is on algorithmic validation, the x86-based NUC was chosen to prevent hardware bottlenecks. Transitioning this validated framework to SWaP-optimized ARM platforms is a direct avenue for future edge deployment.

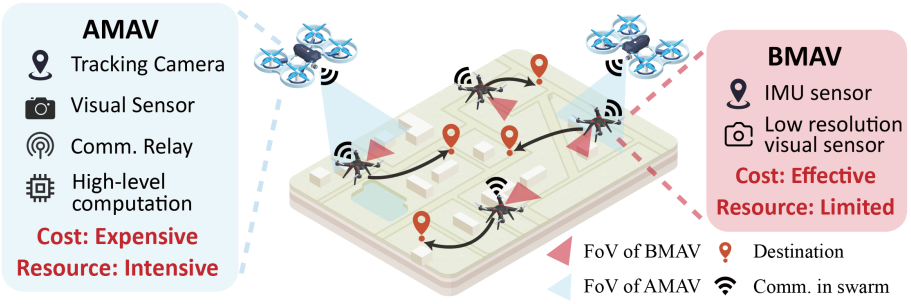
4.9 Summary

This paper introduces EV-Pose, pioneering an event-enhanced Visual Positioning Service (VPS) that fundamentally overcomes the motion blur and sampling rate bottlenecks of conventional RGB-based VPS systems. The primary objective is to provide drift-free, *absolute global coordinates* at ultra-high frequencies for agile drone landing. To address the first major challenge of cross-modality matching, this work proposes the STPE module. By extracting a continuous Temporal Distance Field (TDF), EV-Pose successfully reformulates the heterogeneous alignment between 2D asynchronous events and a static 3D point map into an optimizable mathematical problem. To tackle the second challenge of jointly optimizing efficiency and accuracy, this work develops the MHFO scheme. This proactive hierarchical architecture tightly couples IMU kinematics with spatial priors, executing efficient early-stage event filtering, and employing a later-stage factor graph to seamlessly regularize the SE(3) manifold. Extensive real-world evaluations demonstrate that EV-Pose achieves superior accuracy and ultra-low latency.

CHAPTER 5 MULTI-AGENT-ORIENTED ENGAGED COLLABORATION METHOD

Building upon the concept of "engaged coordination" defined in Chapter 1, this chapter proposes TransformLoc, a hierarchical localization framework for multi-MAV systems. Transitioning from individual enacted perception to swarm-level collaboration, TransformLoc addresses the collaborative challenges of large-scale heterogeneous systems in complex environments by leveraging high-performance AMAVs as mobile infrastructures. By exploiting the latent spatio-temporal constraints among agents, the framework provides intermittent, on-demand observational corrections for resource-constrained lower-level agents. This symbiotic approach effectively reduces overall system hardware costs while enhancing the swarm's collective survivability in regions with degraded sensing coverage. Crucially, by feeding these synergistic localization outputs directly into the multi-agent trajectory planning modules, TransformLoc explicitly extends the "perception-decision-action" feedback loop from the individual agent to the swarm level, enabling highly adaptive and resilient collective maneuvers.

A heterogeneous multi-MAV system consists of resource-intensive but expensive AMAVs and resource-limited but cost-effective BMAVs, offering opportunities in diverse fields. Accurate and real-time localization is crucial for multi-MAV systems, but current practices lack a low-cost, high-precision, and real-time solution, especially for lightweight BMAVs. This work finds an opportunity to accomplish the task by transforming AMAVs into mobile localization infrastructures for BMAVs. However, translating this insight into a practical system is challenging due to issues in estimating locations with diverse and unknown localization errors of BMAVs, and allocating resources of AMAVs considering interconnected influential factors. This work introduces TransformLoc, a new framework that transforms AMAVs into mobile localization infrastructures, specifically designed for low-cost and resource-constrained BMAVs. This work designs an error-aware joint location estimation model to perform intermittent joint estimation for BMAVs and introduces a similarity-instructed adaptive grouping-scheduling strategy to allocate resources of AMAVs dynamically. TransformLoc achieves a collaborative, adaptive, and cost-effective localization system suitable for large-scale heterogeneous multi-MAV systems. This work implements and validates TransformLoc on industrial drones. Results



Type	Localiaztion accuracy (Absolute trajectory error)	Runtime	Responsible task
AMAV	< 0.05 m	<150 s	Data management, Comm., etc.
BMAV	> 0.36 m	> 770 s	Environment exploration, etc.

Figure 5.1 Introduction of a heterogeneous multi-MAV system composed of two types: AMAVs and BMAVs. AMAVs are resource-intensive, offering lower localization error and reduced latency, whereas BMAVs, with their resource limitations, experience higher error and latency ①.

show it outperforms all baselines by up to 68% in localization performance, improving navigation success rates by 60%. Extensive robustness and ablation experiments further highlight the superiority of its design.

5.1 Introduction

Heterogeneous multi-MAV systems demonstrate transformative potential in addressing 4D (deep, dull, dangerous, dirty) tasks, particularly in scenarios requiring prompt responses, such as search and rescue during diverse disasters[161], gas leak source detection[162-164], and wildfire suppression[165]. These swarms capitalize on inherent advantages, including working scalability[166], flexibility[69], and adaptability[84]. Industry forecasts project that the market size for applications supported by heterogeneous multi-MAV systems will reach approximately $279 billion by 2032.[127].

A heterogeneous multi-MAV system typically comprises: (*i*) a limited number of advanced MAVs (AMAVs) endowed with intensive capabilities (e.g., sensing, computing, etc.), albeit at a high cost[169]; and (*ii*) a larger cohort of resource-limited yet cost-effective basic MAVs (BMAVs)[170], as illustrated in Figure 5.1. In a heterogeneous multi-MAV system, the AMAV primarily manages data, communication, and intricate computational tasks, while the latter is dispersed into (hazardous) target areas for data collection and environmental exploration[171]. The collaboration paradigm between AMAVs and

① Figure 5.1 and Figure 5.2(a) present results obtained using MH_02 from the EuRoC dataset[167] and CCM-SLAM[168]. The AMAV is equipped with an Intel(R) i7-8750H processor and 32GB RAM, while the BMAV is equipped with a Cortex-A53 processor and 1GB RAM.

BMAVs achieves an optimal balance between overall capabilities and cost, facilitating its widespread adoption in extensive applications, including Amazon disaster relief[172], DJI industrial inspection[173], and aerial imaging[174].

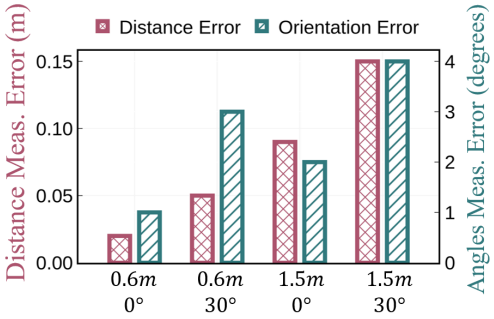
Nevertheless, the disparate onboard resources result in imbalanced localization performance between AMAVs and BMAVs, a critical capability for flight control[73], obstacle avoidance[6], and more. Specifically, the BMAV experiences rapid accumulation of localization errors due to its noisy sensing data and limited computational capability, impeding its ability to achieve precise and real-time localization comparable to the AMAV. As illustrated in Figure 5.1, for the same localization and navigation task, the AMAV completes it in <150 seconds with a localization error of <0.05 meters. In contrast, the BMAV takes >770 seconds and has a localization error of >0.36 meters. This stark difference in sensing and computation capabilities results in significantly lower localization efficiency for the BMAV compared to the AMAV.

Regrettably, existing approaches fail to provide viable solutions for BMAV localization in quick response scenarios, which can be divided into two distinct types:

- **Extra infrastructure-based solutions**. These approaches leverage external devices to furnish reference signals, such as GPS[131,175], RTK[5], radio[36,176], or acoustic[5], for MAV localization. They necessitate: (*i*) deploying expensive localization infrastructure in the operation site, often in a dense configuration, and (*ii*) maintaining a line-of-sight connection between the localization infrastructure and MAVs. However, in quick response scenarios like urban disaster relief, fulfilling either requirement becomes challenging, resulting in localization failures[74].
- **Intra on-board sensor-based solutions.** These approaches utilize onboard sensors, including cameras[21], spectral camera[177-178], IMU[179], LiDAR[180], and Radar[32], in conjunction with SLAM techniques, to achieve precise location estimation[103,181]. Moreover, leveraging image enhancement, these methods can adapt to complex environments[182-184]. These methods demand substantial onboard sensing and computing capabilities, rendering them suitable only for AMAVs. Conversely, the limited on-board resources of BMAVs result in computation delays or considerable error accumulation[37]. As depicted in Figure 5.2(a), this work tests different localization algorithms on BMAVs. When employing CCM-SLAM, the BMAV achieved a low localization error (>0.3 cm) but required a long runtime (>5 s). In contrast, when using Dead-Reckoning, the error increased (>9 cm), but the

Algorithm	Drift	Latency
CCM-SLAM	< 0.3 cm	> 5 s
Dead-Reckoning	> 9 cm	< 0.05 s

(a) Accuracy & latency on BMAV



(b) Accuracy of AMAV observation

Figure 5.2 Motivating example of TransformLoc. (a) BMAVs contend with constrained onboard resources, necessitating a balance between accuracy and latency. (b) The precision of observations is influenced by the distances and angles between MAVs.

latency is significantly reduced (<0.05 s). Therefore, BMAVs face limited onboard resources, necessitating a balance between latency and accuracy.

Therefore, this paper aims to enhance the localization accuracy and efficiency of BMAVs, leveraging limited onboard sensing and computing capabilities without dependence on pre-deployed localization infrastructures. The **key insight** involves repurposing a select number of AMAVs as aerial shepherds, functioning as *mobile localization infrastructures*, which redistribute their sensing and computing capabilities to enhance the location estimation of a larger fleet of BMAVs. Specifically, AMAVs contribute external observations (via visual sensors, e.g., event camera) to rectify cumulative location estimation errors[185-186]. In addition, to optimize real-time performance, the most intricate computational tasks should be primarily executed within the AMAVs.

However, converting this insight into a functional system presents significant complexity, as it necessitates the resolution of two pivotal technical challenges:

- **(C1) Unknown & diverse localization errors of BMAVs.** On one hand, diverse factors such as sensing noises and varying operational conditions contribute to the accumulation of location estimation errors among BMAVs. On the other hand, the limited FoV of each AMAV restricts its ability to serve as a localization infrastructure for only a subset of BMAVs. Therefore, AMAVs ought to furnish external observations to aid in correcting estimation errors of BMAVs exhibiting significant deviations. However, the real-time derivation of BMAVs' localization errors presents challenges due to the absence of static location references. Consequently, without timely external observations from AMAVs, localization errors of BMAVs may persistently accumulate. Therefore, it is crucial to address how to correct BMAVs'

locations with AMAVs' observations.

- **(C2) Resource allocation given coupled influential factors.** Optimizing the allocation of sensing and computing resources from a limited number of AMAVs to a larger fleet of BMAVs as localization infrastructure entails optimization in a high-dimensional decision space. Firstly, even if the initial constraint (**C1**) is resolved, the involvement of multiple AMAVs leads to an exponential growth in search space. Secondly, given the perpetual motion of both AMAVs and BMAVs, the set of observable BMAVs for each AMAV constantly shifts, demanding dynamic adjustments in resource allocation strategies. Thirdly, the accuracy of AMAVs' observations, influenced by varying distances and bearing angles between AMAVs and BMAVs, further complicates the search space for resource allocation, directly impacting localization effectiveness. As illustrated in Figure 5.2(b), this work conducts measurement experiments with a camera and AprilTag. At a short distance and small angle (0.6 meters and 0 degrees), the observed distance and angle errors are minimal (0.02 meters and 1 degree). However, as the distance or angle increases, the errors grow significantly (at 1.5 meters and 30 degrees, the errors reach 0.15 meters and 4 degrees). These intertwined factors render resource allocation exceedingly intricate for implementation on MAVs.

Proposed work. To address these challenges, this paper introduces *TransformLoc*, an innovative collaborative and adaptive location estimation framework designed for heterogeneous multi-MAV systems. TransformLoc dynamically converts AMAVs, equipped with intensive onboard capabilities, into aerial shepherds and mobile localization infrastructures to enhance location estimation for resource-constrained BMAVs. By concentrating intensive sensing and computing resources on a select few AMAVs, TransformLoc ensures accurate and real-time localization for the entire swarm, eliminating the need for external localization infrastructures. As a result, this framework promotes the extensive adoption of swarm technology.

To tackle **C1**, this work devises an *error-aware joint location estimation* model. This model incorporates an uncertainty-aided inference method, allowing AMAVs to prioritize providing observations to BMAVs with greater errors. Subsequently, it integrates inaccurate estimations from BMAVs with intermittent external observations from AMAVs to facilitate periodic joint location estimation for BMAVs.

To address **C2**, this work introduces a *similarity-instructed adaptive grouping-*

scheduling strategy. Initially, MAVs are dynamically grouped in accordance with the principle of proximity, transforming the many-to-many resource allocation challenge into multiple one-to-many resource allocation tasks. Then, employing a several-step lookahead approach for BMAVs, each AMAV is scheduled in a non-myopic manner, optimizing both distance and angle. Concurrently, the strategy utilizes a similarity-instructed method to identify and eliminate redundant search space, retaining a smaller subset of decision space with significant distinctions. Ultimately, the computational complexity is diminished from an exponential to a linear scale.

This work assesses the performance of TransformLoc and benchmarks it against baselines, including the state-of-the-art (SOTA) method, using comprehensive experiments conducted on both a real-time testbed and physical feature-based simulations. The findings reveal that TransformLoc achieves an average localization error of under $1m$ for BMAVs with limited onboard resources in real-time, outperforming baselines by up to 68%. This enhancement translates to a potential increase of up to 60% in navigation success rates over baselines. Additionally, to explore the boundaries of TransformLoc, this work conducts extensive robustness evaluations, testing the system's resilience to varying numbers of MAVs and measurement noise. Furthermore, this work performs ablation experiments, which underscore the superiority of TransformLoc's design through comprehensive results.

In summary, the primary contributions of this chapter include:

(1) This work introduces TransformLoc, a novel framework that dynamically converts AMAVs into aerial shepherds and mobile localization infrastructures, improving localization accuracy and real-time performance for lightweight BMAVs.

(2) This work develops an *error-aware joint location estimation* aimed at enhancing the accuracy of location estimation for BMAVs utilizing observations from AMAVs.

(3) This work formulates a *similarity-instructed adaptive grouping-scheduling* strategy to decouple the intertwined resource allocation challenge influenced by coupled factors.

(4) This work verifies the effectiveness of the solution through real-world experiments conducted on a real heterogeneous multi-MAV testbed and extensive physical feature-based simulations.

The subsequent sections of the paper are structured as follows: Section 5.2 presents an overview of TransformLoc, followed by detailed descriptions of the error-aware joint location estimation model in Section 5.3 and the similarity-instructed adaptive grouping-

scheduling strategy in Section 5.4. The implementation and evaluation of TransformLoc are showcased in Section 5.5, while Section 5.6 delves into the discussion of the framework, concluding the TransformLoc in Section 5.7.

5.2 System Overview

This section provides a high-level overview of the *TransformLoc* architecture, which is designed to coordinate AMAVs as mobile infrastructure for BMAVs. Section 5.2.1 first defines the primary objective of the framework: to transform AMAVs into mobile localization infrastructures specifically tailored to overcome the resource limitations of BMAVs. Subsequently, Section 5.2.2 introduces the structural components of the framework, outlining the synergistic relationship between the joint estimation model and the adaptive scheduling strategy.

5.2.1 Framework Goal

From a top-level perspective, TransformLoc is designed to convert AMAVs into mobile localization infrastructures tailored for BMAVs. This specialized adaptation is meticulously designed to accommodate the economic and resource limitations inherent to BMAVs, thus effectively overcoming the challenges posed by these constraints. The framework design aims to address the following questions:

- *How to enhance the location estimation accuracy of BMAVs despite diverse and unknown localization errors?* TransformLoc is tasked with inferring the localization errors of BMAVs and leveraging observations provided by AMAVs to aid in localization error reduction effectively.
- *How to allocate sensing resources of AMAVs, considering intertwined influential factors to support BMAVs?* TransformLoc should disentangle the resource allocation challenge and guide AMAVs in a non-myopic manner, ensuring minimal overall localization errors for BMAVs.

Within the TransformLoc architecture, a strict division of operational roles is established to maximize system efficiency. The massive fleet of BMAVs is tasked with the direct execution of distributed missions, such as environmental exploration or target search. Conversely, the AMAVs are conceptually designated as dynamic mobile infrastructures and system-level supervisors. Rather than participating in the direct physical exploration, the primary mission of the AMAVs is to optimize their spatial distribution and trajectories

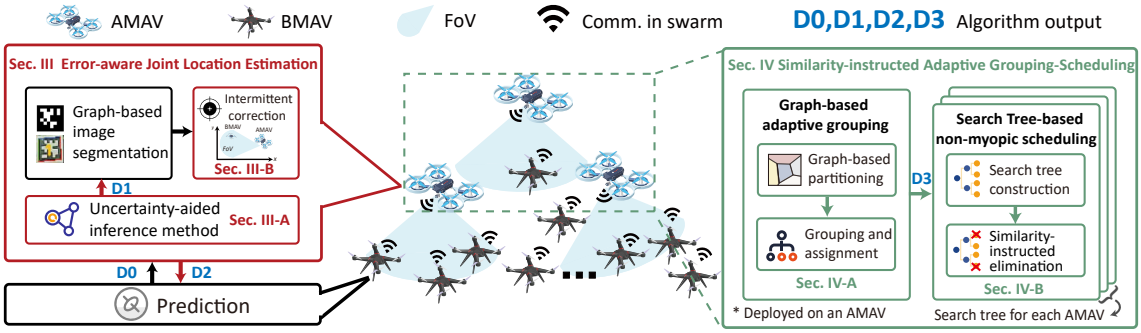


Figure 5.3 Demonstration of the TransformLoc framework. Initially, BMAVs estimate locations amidst noisy measurements. Leveraging an uncertainty-aided inference method, AMAV generates intermittent observations to facilitate intermittent joint location estimation for BMAVs. Following this, TransformLoc optimizes resource allocation for AMAVs through adaptive grouping and scheduling. This entails dynamically grouping MAVs and subsequently scheduling AMAVs in a non-myopic fashion, followed by a similarity-instructed elimination method to identify and eliminate redundant search space.

to provide continuous communication relay, data aggregation, and on-demand localization assistance to the BMAV swarm.

5.2.2 Framework Overview

The architecture of TransformLoc is depicted in Figure 5.3. It comprises two primary components: the *Error-aware Joint Location Estimation* model and the *Similarity-instructed Adaptive Grouping-Scheduling* strategy. In the following, this work will provide an overview of these two components:

- *Error-aware Joint Location Estimation* model. In the initial step, BMAVs employ the state model and motion commands to generate predictions, transmitting the resulting prior distribution of location to AMAVs (*D0* in Figure 5.3). Subsequently, the uncertainty-aided inference method (Section 5.3.2) discerns instances where BMAVs require assistance (*D1* in the Figure 5.3). Leveraging observations from the visual fiducial system, AMAVs collaboratively refine the location estimations of BMAVs requiring assistance (Section 5.3.3), transmitting the resulting posterior of BMAVs' locations and commands back to BMAVs (*D2* in the Figure 5.3).
- *Similarity-instructed Adaptive Grouping-Scheduling* strategy. With a limited number of AMAVs, this strategy is tasked with allocating sensing resources from AMAVs to BMAVs. Initially, this strategy dynamically groups MAVs using a graph-based approach based on the spatial relationships among AMAVs (Section 5.4.2). Leveraging the grouping outcomes (*D3* in Figure 5.3), TransformLoc then orchestrates the scheduling of each AMAV in a non-myopic manner by construct-

ing search trees for individual AMAVs. To enhance the efficiency of the search tree, TransformLoc further employs a similarity-instructed method to identify and eliminate redundant nodes in search trees (Section 5.4.3). Ultimately, TransformLoc generates motion commands for AMAVs to distribute resources to BMAVs.

5.3 Error-aware Joint Location Estimation Model

The BMAVs present a challenge due to their diverse and unknown localization errors, complicating joint location estimation with AMAV assistance. In this section, this work introduces an error-aware joint location estimation model rooted in the Kalman filter, focusing on localization error inference and joint location estimation for BMAVs.

Initially, this work delineates the key definitions, which encompass the state model of both BMAVs and AMAVs, as well as the observation model of AMAVs (Section 5.3.1). Subsequently, to facilitate AMAVs in providing observations for BMAVs with higher errors, this work introduces a measure of uncertainty to reflect the accuracy of BMAVs' location estimation (Section 5.3.2). Concurrently, this work undertakes two interconnected operations for BMAV location estimation (Section 5.3.3): (1) *Prediction from motion model*, which utilizes noisy motion measurements and the state model of BMAVs to forecast location estimation. (2) *Correction from observation*, incorporating linearized observations generated by AMAVs to refine the location estimation of the BMAVs.

5.3.1 Key Definitions

Environmental description. Consider Ω as a bounded space within $\mathbb{R}^3$ with dimensions defined by length (L), width (W), and height (H), where the heterogeneous multi-MAV system operates. This swarm consists of a predetermined quantity of BMAVs and AMAVs, which operate autonomously. For the sake of simplicity, this work assumes that both types of MAVs operate at an equal altitude throughout the paper.

State model of AMAV. The heterogeneous multi-MAV system comprises M AMAVs ($M > 1$). At time t, the state of each AMAV A_j encompasses both its location and motion command. The location is represented as $\boldsymbol{x}_{j,t} = (x^1_{j,t}, x^2_{j,t}, \phi_{j,t})$, where $\phi_{j,t}$ denotes the orientation angle, and the distance between two locations is denoted as d_χ. Regarding the motion command, $\boldsymbol{u}_{j,t} = (u_{j,t}, \omega_{j,t})$, where $u_{j,t}$ and $\omega_{j,t}$ represent translational and rotational velocities, respectively. The AMAVs achieve accurate location estimation through advanced sensing capabilities and adhere to the motion model $\boldsymbol{x}_{j,t} = f(\boldsymbol{x}_{j,t-1}, \boldsymbol{u}_{j,t-1})$,

which can be expressed as follows:

$$\begin{pmatrix} x_{j,t}^1 \\ x_{j,t}^2 \\ \phi_{j,t} \end{pmatrix} = \begin{pmatrix} x_{j,t-1}^1 \\ x_{j,t-1}^2 \\ \phi_{j,t-1} \end{pmatrix} + \begin{pmatrix} u_{j,t-1} \cos\left(\phi_{j,t-1}\right) \\ u_{j,t-1} \sin\left(\phi_{j,t-1}\right) \\ \omega_{j,t-1} \end{pmatrix}. \tag{5.1}$$

State model of BMAV. The heterogeneous multi-MAV system comprises N ($N > 1$) BMAVs. The number of BMAVs is greater than or equal to the number of AMAVs ($N \geq M$). Each BMAV B_i's state at time t includes both its location and motion command. The location of B_i is denoted as $\boldsymbol{y_{i,t}} = (y_{i,t}^1, y_{i,t}^2)$, and its motion command is $\boldsymbol{v_{i,t}} = (v_{i,t}^1, v_{i,t}^2)$. The B_i follows double integrator dynamics with Gaussian noise, consistent with the model used in other work[187]:

$$\begin{gathered} \begin{pmatrix} y_{i,t+\delta}^1 \\ y_{i,t+\delta}^2 \end{pmatrix} = \begin{pmatrix} y_{i,t}^1 \\ y_{i,t}^2 \end{pmatrix} + \delta \begin{pmatrix} v_{i,t}^1 + n_{i,t}^1 \\ v_{i,t}^2 + n_{i,t}^2 \end{pmatrix}, \\ n_{i,t}^1 \text{ is drawn from } p(n^1), \\ n_{i,t}^2 \text{ is drawn from } p(n^2). \end{gathered} \tag{5.2}$$

The motion noise terms $n_{i,t}^1$ and $n_{i,t}^2$ are drawn from motion noise models $p(n^1)$ and $p(n^2)$ respectively, with δ ($\delta > 1$) representing the time interval between two commands. The distributions $p(n^1)$ and $p(n^2)$, empirically obtained from the testbed, are specified as normal distributions with mean $\mu = 0$ and variance σ, expressed as a percentage of $v_{i,t}^1$ and $v_{i,t}^2$, consistent with the approach outlined in[188]. The location estimation of B_i is denoted as $\hat{\boldsymbol{y}}_{\boldsymbol{i,t}}$, and its covariance matrix is denoted as $\Sigma_{i,t}$. The noisy motion measurement from B_i's Inertial Measurement Unit(IMU) is represented as $\dot{\boldsymbol{v}}_{\boldsymbol{i,t}}$.

Observation model of AMAV. The observation angle of the visual sensor on AMAVs is denoted by ϕ, and at time t, the Field of View (FoV) of A_j is defined as follows:

$$\begin{aligned} F_{j,t} = \{(x, y) \mid & \\ 0 < \sqrt{\left(y - x_{j,t}^2\right)^2 + \left(x - x_{j,t}^1\right)^2} < r_m & \quad \bigcap \\ -\frac{\phi}{2} < \arctan\left(\left(y - x_{j,t}^2\right)\left(x - x_{j,t}^1\right)\right) - \phi < \frac{\phi}{2}\}, & \end{aligned} \tag{5.3}$$

where r_m represents the maximum observable distance constrained by the sensor's observation technique.

When B_i falls within the FoV of A_j at t, A_j generates a noisy observation $z_{i,j,t}$ for B_i,

comprising the range $r_{i,j,t}$ and bearing $\alpha_{i,j,t}$. Both $r_{i,j,t}$ and $\alpha_{i,j,t}$ are relative to A_j.

$$\begin{aligned} z_{i,j,t} &= h(\boldsymbol{x_{j,t}}, \boldsymbol{y_{i,t}}) + \boldsymbol{n_{i,j,t}}, \\ h(\boldsymbol{x_{j,t}}, \boldsymbol{y_{i,t}}) &:= \begin{bmatrix} r_{i,j,t} \\ \alpha_{i,j,t} \end{bmatrix} \\ &:= \begin{bmatrix} \sqrt{\left(y^1_{i,t} - x^1_{j,t}\right)^2 + \left(y^2_{i,t} - x^2_{j,t}\right)^2} \\ \arctan\left(\left(y^2_{i,t} - x^2_{j,t}\right)\left(y^1_{i,t} - x^1_{j,t}\right)\right) - \phi \end{bmatrix} \\ \boldsymbol{n_{i,j,t}} &:= [n^r_{i,j,t}, n^\alpha_{i,j,t}]^T, \\ & \quad n^r_{i,j,t} \text{ is drawn from } p(n^r), \\ & \quad n^\alpha_{i,j,t} \text{ is drawn from } p(n^\alpha). \end{aligned} \tag{5.4}$$

The noise models for range and bearing measurements are represented by $p(n^r)$ and $p(n^\alpha)$, respectively, both empirically obtained from the testbed. These models, $p(n^r)$ and $p(n^\alpha)$, are defined as normal distributions with mean $\mu = 0$ and variance σ, presented as a percentage of $r_{i,j,t}$ and $\alpha_{i,j,t}$. The above model applies to both frame-based cameras and event cameras. In this work, frame-based cameras will be used as an example.

Linearization of observation model. To linearize the observation model, this work computes the Jacobian matrix of $h(\boldsymbol{x}, \boldsymbol{y})$ using Taylor expansion, by calculating the gradient with respect to the location estimation of $\boldsymbol{y}$:

$$\nabla_{\boldsymbol{y}} h(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{r(\boldsymbol{x}, \boldsymbol{y})} \begin{bmatrix} \left(y^1 - x^1\right) & \left(y^2 - x^2\right) \\ -\sin\left(\phi + \alpha(\boldsymbol{x}, \boldsymbol{y})\right) & \cos\left(\phi + \alpha(\boldsymbol{x}, \boldsymbol{y})\right) \end{bmatrix}. \tag{5.5}$$

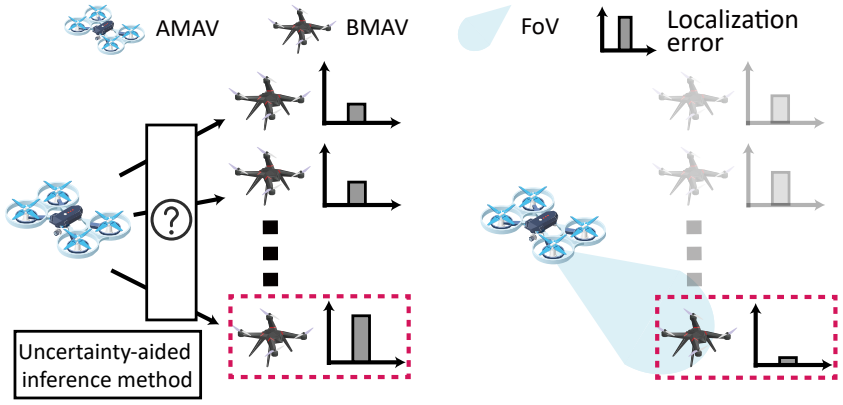
To simplify notation, this work lets $\boldsymbol{y_t} := [\boldsymbol{y^T_{1,t}}, \ldots, \boldsymbol{y^T_{N,t}}]^T$, $\boldsymbol{x_t} := [\boldsymbol{x^T_{1,t}}, \ldots, \boldsymbol{x^T_{M,t}}]^T$, $\boldsymbol{u_t} := [\boldsymbol{u^T_{1,t}}, \ldots, \boldsymbol{u^T_{M,t}}]^T$, $\hat{\boldsymbol{y}}_{\boldsymbol{t}} := [\hat{\boldsymbol{y}}^{\boldsymbol{T}}_{1,t}, \ldots, \hat{\boldsymbol{y}}^{\boldsymbol{T}}_{N,t}]^T$, $\Sigma_t := diag(\Sigma^T_{1,t}, \ldots, \Sigma^T_{N,t})^T$.

5.3.2 Uncertainty-aided Inference Method

TransformLoc aims to efficiently allocate the sensing resources of AMAVs to generate observations for BMAVs, thereby minimizing localization errors. The localization error of a BMAV B_i at time t can be formally represented as:

$$\psi(i,t) := E\left[||y_{i,t} - \hat{y}_{i,t}||^2\right]. \tag{5.6}$$

In this context, $y_{i,t}$ denotes the actual location of B_i, while $\hat{y}_{i,t}$ represents the estimated location of B_i. However, obtaining the actual locations of BMAVs is impractical due to their limited capabilities and the absence of external localization infrastructure. Incorporating the Kalman filter-based model for estimating the location of BMAVs, this work



(a) Determine BMAV with significant error (b) AMAV assist BMAV in localization

Figure 5.4 The error-aware joint pose estimation model operates as follows: (a) With the aid of the uncertainty-aided inference method, the AMAV identifies the BMAV with a higher error, and (b) subsequently generates observations for correction.

relies on a measure of estimation uncertainty to gauge the quality of BMAVs' location estimations. This approach eliminates the need to access the actual locations of BMAVs to evaluate localization quality. To quantify the estimation uncertainty of BMAVs, this work opts for the trace of the covariance matrix of the BMAV's state estimator[189]. This is mathematically expressed as follows:

$$\psi(i,t) = tr\left(\Sigma_{i,t}\right), \tag{5.7}$$

where $\Sigma_{i,t}$ represents the trace of the covariance matrix of B_i at time t. This indicator serves as a quantitative measure of estimation uncertainty, where a lower value signifies higher confidence in the current pose estimate. By leveraging this metric, the AMAV can prioritize its observational resources to assist BMAVs exhibiting larger localization errors. Such a strategy facilitates on-demand observational corrections tailored to the varying error profiles within the BMAVs, as illustrated in Figure 5.4(a) and Figure 5.4(b).

5.3.3 Joint Estimation of BMAVs' Location

The location estimation framework for BMAVs comprises two operations: (1) prediction based on the state model and noisy velocity measurements of BMAVs, which computes the prior distribution of BMAVs' location; (2) correction based on observations from AMAVs, which computes the posterior distribution of BMAVs' location. Further details regarding the joint estimation of BMAVs' location are elucidated in Figure 5.1.

- *Prediction of BMAVs from motion model* is outlined in Lines (1-3) of Figure 5.1. During this operation, the motion model, along with the noisy motion measurement $\dot{v}_{i,t}$ and the previous location estimation $\hat{\boldsymbol{y}}_{i,t-1}$ of BMAV B_i at time t, are utilized

to compute the prior distribution and the covariance matrix, denoted as $\boldsymbol{y}_{i,t}^{-}$ and $\Sigma_{i,t}^{-}$, respectively.

- *Correction of BMAVs from observations* is depicted in Lines (4-11) of Figure 5.1. During this operation, when BMAV $\boldsymbol{B}_i$ falls within the FoV of AMAV $\boldsymbol{A}_j$ at time t, the observation $z_{i,j,t}$ and the prior distribution of $\boldsymbol{B}_i$'s location are utilized to compute the posterior distribution and the covariance matrix at time t, denoted as $\boldsymbol{y}_{i,t}^{+}$ and $\Sigma_{i,t}^{+}$, respectively. The symbol η in Line 8 represents a normalization constant. If AMAV $\boldsymbol{A}_j$ generates an observation for $\boldsymbol{B}_i$, the location estimation $\hat{\boldsymbol{y}}_{i,t}$ and the covariance matrix of estimation $\Sigma_{i,t}$ for $\boldsymbol{B}_i$ at time t are drawn from the posterior distribution $\boldsymbol{y}_{i,t}^{+}$ and $\Sigma_{i,t}^{+}$; otherwise, they are drawn from the prior distribution $\boldsymbol{y}_{i,t}^{-}$ and $\Sigma_{i,t}^{-}$.

Algorithm 5.1 $\boldsymbol{A}_j$ assists $\boldsymbol{B}_i$ for localization using noisy motion measurements and observations.

Input: Prior estimate $\hat{\boldsymbol{y}}_{i,t-1}$, covariance matrix $\Sigma_{i,t-1}$, noisy motion measurements $\dot{\upsilon}_{i,t-1}$, AMAV state $\boldsymbol{x}_{j,t-1}$, AMAV motion command $u_{j,t-1}$
Output: Updated estimate $\hat{\boldsymbol{y}}_{i,t}$, updated covariance matrix $\Sigma_{i,t}$
% Prediction of BMAV from motion model
1: Update prior distribution of $\boldsymbol{B}_i$'s location at time t: $\boldsymbol{y}_{i,t}^{-} = \int p\left(\boldsymbol{y}_{i,t} \mid \boldsymbol{y}_{i,t-1}, \dot{\upsilon}_{i,t-1}\right) \hat{\boldsymbol{y}}_{i,t-1} d\boldsymbol{y}_{i,t-1}$;
2: Update covariance matrix $\Sigma_{i,t-1}^{-}$ from $\boldsymbol{y}_{i,t}^{-}$;
3: Update $\hat{\boldsymbol{y}}_{i,t}$ and $\Sigma_{i,t}$ from $\boldsymbol{y}_{i,t}^{-}$ and $\Sigma_{i,t}^{-}$;
% Correction of BMAV from observations
4: Update the FoV of $\boldsymbol{A}_j$, $F_{j,t}$, according to Eq. (5.3);
5: **if** $\boldsymbol{B}_i$ is in the FoV of $\boldsymbol{A}_j$ **then**
6: Update observation $\boldsymbol{z}_{i,j,t}$ according to Eq. (5.4);
7: Linearize the observation according to Eq. (5.5);
8: Update posterior distribution of $\boldsymbol{B}_i$'s location at time t: $\boldsymbol{y}_{i,t}^{+} = \eta p\left(\boldsymbol{z}_{i,j,t} \mid \boldsymbol{y}_{i,t}\right) \boldsymbol{y}_{i,t}^{-}$;
9: Update covariance matrix $\Sigma_{i,t}^{+}$ from $\boldsymbol{y}_{i,t}^{+}$;
10: Update $\hat{\boldsymbol{y}}_{i,t}$ and $\Sigma_{i,t}$ from $\boldsymbol{y}_{i,t}^{+}$ and $\Sigma_{i,t}^{+}$;
11: **end if**

5.4 Similarity-instructed Adaptive Grouping-scheduling Strategy

The BMAVs are dispersed across different locations and exhibit varying levels of localization errors. Due to the limited FoV and velocity of AMAV, it is crucial to identify the BMAVs and allocate localization resources accordingly. Furthermore, since both AMAVs and BMAVs are continuously moving, the set of observable BMAVs for each AMAV is constantly changing. This dynamic nature necessitates the adaptation of resource allocation strategies, involving optimization in a high-dimensional decision space.

In this section, this work presents a *Similarity-instructed adaptive grouping-scheduling* strategy for allocating sensing and computing resources of AMAVs to assist BMAVs in localization. This work begins by outlining the mathematical formulation of the resource allocation problem (Section 5.4.1), followed by an introduction to the strategy. This strategy initially dynamically groups AMAVs and BMAVs based on spatial distance, leveraging the Voronoi diagram (Section 5.4.2). This step converts the complex many-to-many resource allocation problem into multiple one-to-many resource allocation subproblems. Subsequently, to plan the trajectory of each AMAV in a non-myopic manner and determine the optimal distance and angle, the strategy constructs search trees for each AMAV by involving a several-step lookahead about BMAVs. With the aim of further reducing search space, the strategy selectively prunes spatially close nodes with similar covariances related to BMAVs, thereby eliminating redundant search space (Section 5.4.3). Finally, this work delineates the planning method for BMAVs (Section 5.4.4).

5.4.1 Problem Formulation

The objective is to optimize the trace of the covariance of BMAVs' location estimation Σ_t to constrain the localization error within a finite time horizon T, taking into account the motion commands of each AMAV within this time span. The mathematically formulated sensing resource allocation problem is as follows:

$$\min_{\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{T-1}} \quad \xi(T) = \sum_{t=1}^{T} \mathrm{tr}(\Sigma_t). \tag{5.8a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \quad t \in \{0, \ldots, T-1\}, \tag{5.8b}$$

$$0 \le x^1_{j,t}, y^1_{i,t} \le L, \quad \forall i, j, t \in \mathcal{S}, \tag{5.8c}$$

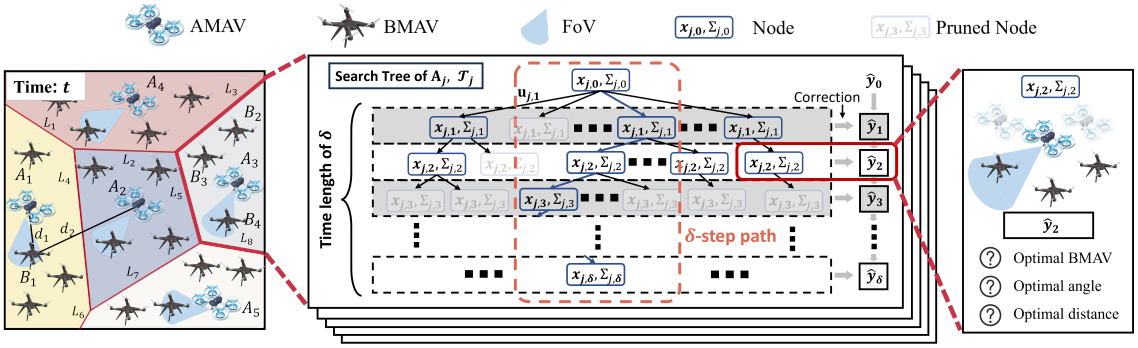
$$0 \le x^2_{j,t}, y^2_{i,t} \le W, \quad \forall i, j, t \in \mathcal{S}, \tag{5.8d}$$

$$\Sigma_{t+1} = \rho^e_{t+1}\left(\rho^p_t(\Sigma_t), \boldsymbol{x}_{t+1}\right), \quad t \in \{0, \ldots, T-1\}, \tag{5.8e}$$

where $\mathcal{S} = \{i \in \{0..M\}, j \in \{0..N\}, t \in \{0..T-1\}\}$. Eqn.(5.8b) delineates the state transition equation of the AMAV, while Eqn.(5.8c) and (5.8d) characterize the motion range of the AMAV. In Eqn.(5.8e), ρ^e_t and ρ^p_{t+1} denote the correction and prediction steps, respectively, elucidated in Section 5.3.3.

5.4.2 Graph-based Adaptive Grouping

The BMAVs are located in various locations with varying localization errors. When AMAVs generate observations for BMAVs, a single AMAV may waste sensing resources



(a) Grouping result at time t (b) Tree-based scheduling and pruning of each AMAVs for δ steps

Figure 5.5 The similarity-instructed adaptive grouping-scheduling strategy aims to address the resource allocation challenge. (a) Initially, it leverages the Voronoi diagram to group MAVs, effectively decoupling the resource allocation problem. (b) Subsequently, the strategy constructs search trees for each AMAV, employing lookahead techniques regarding BMAVs, and employs a similarity-instructed method to prune redundant nodes. This results in δ-step paths for AMAVs. Here, $x_{j,0}$ represents the state of A_j, $\Sigma_{j,0}$ signifies the covariance of BMAVs' estimation maintained by A_j, and $\hat{y}_0$ denotes the estimation location of BMAVs.

by traversing between different BMAVs.

In this section, this work proposes a method to dynamically group BMAVs, enabling each AMAV to concentrate its sensing resources on a single group. However, grouping BMAVs and allocating them to different AMAVs poses a combinatorial optimization challenge, particularly with a large number of BMAVs and AMAVs, given its inherently NP-hard nature. TransformLoc initially partitions the entire area into non-overlapping regions based on the locations of AMAVs. Subsequently, each AMAV allocates its sensing resources to BMAVs within the nearest region for a duration of δ, which represents the control command interval of BMAVs. Consequently, all BMAVs are divided into non-overlapping groups, each assigned to a different AMAV.

Graph-based region partitioning. The partitioning method is grounded in the Voronoi diagram[190]. Under this approach, each AMAV is assigned a region containing points equidistant to or closer to it than to any other AMAVs. The steps followed to create the Voronoi regions include:

1. *Seed Point Selection*, which identifies the positions of AMAVs in operational area. These positions serve as seed points or generating sites for the Voronoi diagram.
2. *Distance Calculation*, where the Euclidean distance from each point in the space to every seed point is computed, determining the closest seed point for each location.
3. *Bisector Construction*, which calculates the perpendicular bisector of the line segment connecting each pair of seed points. These bisectors define the boundaries where points are equidistant from two seed points.

4. *Region Formation*, where all bisectors are combined to partition the space into non-overlapping regions. Each region is bounded by segments of bisectors and contains all points closest to its corresponding seed point.
5. *Boundary Handling*, which constrains the Voronoi diagram to the predefined operational area for finite spaces, clipping any regions that extend beyond these boundaries to ensure practical applicability.

This approach ensures an efficient and equitable partitioning of the operational area, facilitating effective task allocation among AMAVs. Figure 5.5(a) illustrates this partitioning method with five AMAVs and several BMAVs. The outcome is depicted by region boundaries labeled as *L1 - L8*, separated by perpendicular bisectors of neighboring AMAVs. Any BMAV within region of A_1 is closer to it than to any other AMAV, indicating the distance between the BMAV and A_1 is shorter than its distance to any other AMAV (i.e., $d1 < d2$).

Group and assignment of MAVs. BMAVs located within the corresponding region of an AMAV are grouped, and the AMAV allocates its sensing resources to aid in their localization for a duration of δ. For example, AMAV A_3 generates observations for BMAVs B_2, B_3, and B_4 within its region in Figure 5.5(a). The group refreshes after an interval of δ. It is important to note that the boundaries of a region for a given AMAV are determined solely by the locations of neighboring AMAVs. These boundaries can be computed by identifying the perpendicular bisectors between pairs of adjacent AMAVs. If no BMAVs are present within an AMAV's region, it allocates sensing resources to all BMAVs over the duration of δ.

5.4.3 Search Tree-based Non-myopic Scheduling

The AMAV encounters two primary challenges when generating observations for its assigned BMAVs:

- Observation noise is proportional to the distance between the AMAV and the BMAV (Eqn.(5.4)). Closer proximity reduces noise, but the number of BMAVs that can be assisted by an AMAV is constrained by both the size of the BMAV and the AMAV's FoV (Eqn.(5.3)). Balancing this trade-off is essential for the precise localization process of BMAVs.
- BMAVs move to destinations using Eqn.(5.2) while the AMAVs generate observations, necessitating the AMAVs to adjust for displacement. This calls for non-myopic AMAV planning, which anticipates movements of BMAVs several steps

ahead. However, the exponential increase in the AMAV's action space with the planning horizon results in high computational complexity.

In this section, the incorporation of a search tree-based scheduling strategy is introduced for the non-myopic resource allocation of AMAVs. The key concepts are as follows:

- BMAVs receive commands at time intervals of δ, enabling AMAV to acquire BMAV motion commands within δ. These motion commands are utilized to schedule paths of AMAVs with a δ-step lookahead for BMAVs.
- To address the high computational complexity, the search tree is pruned. Specifically, if two nodes in the search tree have similar AMAV locations, the node with the higher BMAV uncertainty may be pruned from the search tree.

Search tree construction. This work constructs search trees for each AMAV. As depicted in Figure 5.5(b), this work illustrates an example of constructing a search tree $\mathcal{T}_j$ for A_j. $\mathcal{T}_j$ encompasses potential trajectories that A_j can pursue, originating from an initial location and covariance pair $(\boldsymbol{x}_{\boldsymbol{j},0}, \Sigma_{j,0})$. Here, $\boldsymbol{x}_{\boldsymbol{j},0}$ denotes the starting location of A_j, and $\Sigma_{j,0}$ represents the initial covariance of the location estimations of the BMAVs assigned to A_j. The nodes within search tree at level $t \leq \delta$ correspond to reachable locations for A_j and are denoted as $(\boldsymbol{x}_{\boldsymbol{j},t}, \Sigma_{j,t})$. At each location, AMAV assesses distance and angle to observable BMAVs to determine the optimal distance and angle for generating observations for BMAVs with significant errors.

This work discretizes the control space of the AMAV, where A_j possesses a finite set of control options $\mathcal{U}$. For each option $\boldsymbol{u}_{\boldsymbol{j},t}$, there exists an edge originating at node $(\boldsymbol{x}_{\boldsymbol{j},t}, \Sigma_{j,t})$ and leading to node $(\boldsymbol{x}_{\boldsymbol{j},t+1}, \Sigma_{j,t+1})$, determined by evaluating the state model of the AMAV. The estimation location of the corresponding BMAVs, $\hat{\boldsymbol{y}}_{\boldsymbol{t}}$, is computed by evaluating the motion model of the BMAV, the observation model between the BMAV and AMAV, and the estimation process shown in Figure 5.1.

Similarity-instructed search tree pruning. The final level of $\mathcal{T}_j$ encompasses $\mathcal{O}\left(\mathcal{U}^{\delta}\right)$ nodes, rendering it impractical to compute in real-time for extended horizons δ by the AMAV. To tackle this challenge, this work employs pruning strategies to reduce the computational burden. During the construction of $\mathcal{T}_j$, if two nodes exhibit identical physical locations at a particular level t but differ in estimation covariance of the assigned BMAVs' location $(\Sigma_{j,t})$, one of the nodes is deemed less informative and is thus considered for deletion. This pruning process is carried out according to the following criteria:

• *Definition 1 (ϵ-Algebraic Redundancy)*[191]: Given $\epsilon \geq 0$ and a finite set $\left\{\Sigma_i\right\}_{i=1}^{K} \subset$

S, a matrix $\Sigma \subset S$ is ϵ-Algebraic Redundancy with respect to $\left\{\Sigma_i\right\}_{i=1}^{K}$ if there exist non-negative constants $\{\alpha_i\}_{i=1}^{K}$ such that

$$\sum_{i=1}^{K} \alpha_i = 1 \quad \text{and} \quad \Sigma + \epsilon I_n \geq \sum_{i=1}^{K} \alpha_i \Sigma_i, \tag{5.9}$$

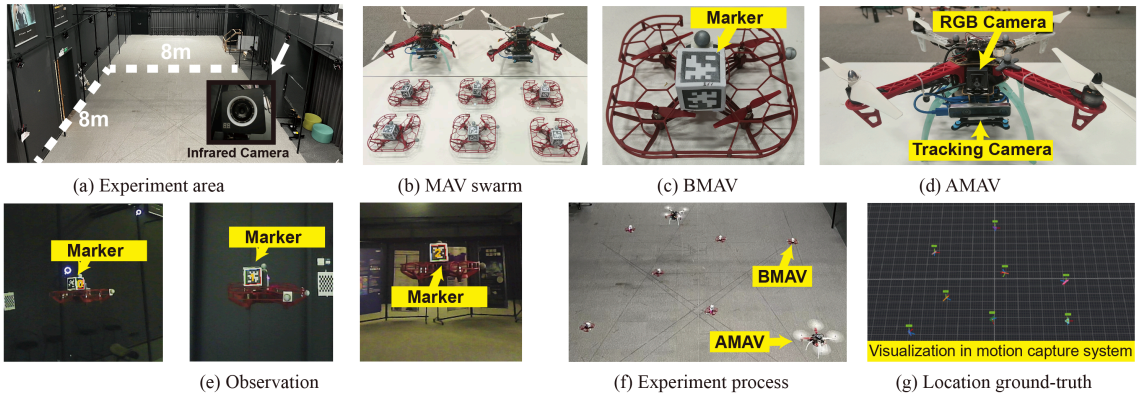
where I_n denotes the identity matrix. A node in the search tree can be pruned if it exhibits ϵ-Algebraic Redundancy with other nodes on the same level, as it provides less information compared to the remaining nodes.

• *Definition 2 (Trajectory σ-Crossing)*[192]: Two trajectories π^1 and π^2 are said to σ-cross at time $t \in [1, \delta]$ if their distance $d_\chi(\pi^1, \pi^2)$ is less than or equal to σ for $\sigma \geq 0$, where d_χ measures the distance between trajectories. If a node in the search tree $\mathcal{T}_j$ is σ-crossing with other nodes at the same level, it implies that these nodes are located in proximity to each other.

This work applies ϵ-Algebraic Redundancy and Trajectory σ-Crossing criteria to determine which nodes at level t of the search tree can be pruned. Increasing ϵ leads to fewer nodes being retained in each level of the tree, approaching a greedy policy, while decreasing ϵ preserves more nodes. As illustrated in Figure 5.5(b), this work demonstrates the construction and pruning of level 3 of AMAV A_j's search tree to exemplify the application of these criteria. Initially, this work retains the node with the minimum $tr(\Sigma_{j,3})$. Subsequently, this work assesses other nodes at level 3. If a node does not σ-cross with the nodes reserved, or if it does σ-cross but its estimation covariance of BMAVs' location is not ϵ-redundant with the reserved nodes, this node is reserved. Otherwise, this node is pruned. This process eliminates less informative nodes that are spatially close. Finally, a node with minimum $tr(\Sigma_{j,\delta})$ at level δ and schedules A_j for a period of δ is selected.

5.4.4 Scheduling of BMAV

The BMAVs' locations are initialized before takeoff, providing accurate initial localization. However, their sensing capabilities are limited, relying solely on the dead-reckoning algorithm, which integrates data from IMUs and optical flow sensors. Over time, the accuracy of their localization decreases due to reliance on this algorithm, leading to increased localization errors. Additionally, BMAVs have constrained computing capabilities, making advanced methods like SLAM using heavy sensors impractical. Although AMAVs can offer observations for BMAVs, they cannot provide real-time assistance to all BMAVs due to the outnumbering of BMAVs by AMAVs.



(a) Experiment area (b) MAV swarm (c) BMAV (d) AMAV

(e) Observation (f) Experiment process (g) Location ground-truth

Figure 5.6 Experimental setup and implementation of the heterogeneous multi-MAV system. (a) The 8 m × 8 m indoor experiment area is equipped with a motion capture system. (b) The heterogeneous MAV swarm. (c) Hardware configuration of a BMAV equipped with an AprilTag marker. (d) Hardware configuration of an AMAV equipped with an RGB camera and a tracking camera. (e) Visual observations of BMAV markers. (f) Snapshot of the dynamic experiment process. (g) Visualization of the location ground-truth in the motion capture system.

To tackle these challenges, TransformLoc utilizes location estimation for BMAV navigation. A lightweight planning algorithm based on artificial potential fields is employed, ensuring collision avoidance. TransformLoc maintains BMAVs' location estimates and generates motion commands at intervals of δ based on their distance from the destination and the obstacles. BMAVs navigate within a force field, where the destination exerts an attractive force proportional to the distance, while obstacles generate repulsive forces proportional to the distance. This approach allows BMAVs to dynamically adjust their motion as they approach the destination, reducing velocity and improving navigation success.

5.5 Experimental Evaluation

This section presents a comprehensive experimental evaluation of *TransformLoc*. Section 5.5.1 first details the system implementation and the experimental methodology employed for testing. The following subsections systematically assess the framework from multiple dimensions: Section 5.5.2 evaluates the overall localization performance, while Section 5.5.3 examines the system's robustness under varying operational conditions. Finally, Sections 5.5.4 and 5.5.5 conduct micro-benchmarks to analyze the contribution of individual modules and the computational efficiency of the system, respectively.

5.5.1 Implementation and Methodology

Testbed implementation. As illustrated in Figure 5.6, this work implements TransformLoc using DJI Robomaster TTs (BMAVs) and industry drones (AMAVs) built on

Pixhawk, one of the most widely used autopilot systems, to validate its performance in the real world. The AMAV is equipped with an Intel(R) T265 tracking camera for localization and an RGB camera with a FoV of 120 degrees for observation generation. When the AMAV is equipped with an event camera, this work will utilize event camera-based object detection algorithms to detect the BMAV and estimate its orientation[113]. Subsequently, by combining the actual size of the BMAV with the detection results, this work can calculate the distance between the AMAV and the BMAV. In this work, frame-based cameras will be used for evaluation. Each BMAV is equipped with an IMU and a downward-facing optical flow sensor. Additionally, each BMAV mounts a $3cm \times 3cm$ AprilTag for recognition and observation generation[185] (Figure 5.6(b)). The AMAV utilizes an ArduPilot flight controller for motion control (Figure 5.6(c)). Ground truth data with millimeter-level accuracy is provided by a motion capture system operating at 240 FPS in the experiment area measuring $8m \times 8m$ (Figure 5.6a and Figure 5.6(d)). This work assesses the robustness of TransformLoc using a physical-feature-based simulator in an experiment area similar to the experiment area shown in Figure 5.6(a).

Experiment setting. The BMAVs are capable of reaching a maximum velocity of 0.5 m/s and operate with a command interval δ set to 5 steps. AMAVs possess an observation angle θ of 120 degrees and a maximum observation distance r_m of 1 m. Control commands for AMAVs utilize motion primitives $\{(u, \omega)\}$ where $u \in \{0, 1, 3\}$ m/s and $\omega \in \{0, \pm 1, \pm 3\}$ rad/s. Performance evaluation was conducted through in-field experiments involving two AMAVs and six BMAVs, alongside simulations featuring five AMAVs and twenty BMAVs. During experiments, BMAVs navigated towards the perimeter of the room for environmental sensing. Simulations are conducted over a duration of 420 seconds, corresponding to the typical battery life of the DJI Robomaster TT drones used in the in-field experiments. The parameters for pruning the search tree, ϵ and σ, are set to 1 and 10, respectively. Noise models for BMAVs' motion and AMAVs' range and bearing measurements are determined empirically during in-field experiments and set to 20%, 10%, and 5% of the measured values, respectively.

Comparative methods. This work conducts a comprehensive evaluation of *TransformLoc (TL)* by comparing it against three baseline methods that operate without the need for a dedicated localization infrastructure.

1. *Dead-Reckoning (DR)* relies solely on measurements from motion sensors to estimate the locations of BMAVs[193].

2. *Station (ST)* employs stationary AMAVs positioned at fixed locations to provide observations for BMAVs[194].
3. *H-SwarmLoc (SL)* represents a state-of-the-art method, where a single AMAV is guided by reinforcement learning to support BMAVs in localization. To ensure a fair comparison, this work modifies SL to group MAVs in a predefined manner and direct AMAVs to generate observations[170].
4. Additionally, this work assesses the latency performance of TransformLoc against *CCM-SLAM*[168], a homogeneous collaborative SLAM method leveraging monocular cameras. The CCM-SLAM represents the first instance of a collaborative SLAM system that demonstrates bidirectional communication between agents and a central server during real-world flights involving three UAVs. The CCM-SLAM achieves swarm localization through multi-UAV collaborative mapping, whereas TransformLoc realizes swarm localization by transforming the sensing and computational resources of AMAVs to BMAVs via observation. Thus, comparing TransformLoc with CCM-SLAM highlights the advantages of TransformLoc's resource transformation method for swarm localization in terms of latency, because it avoids collaborative mapping with high computational overhead.

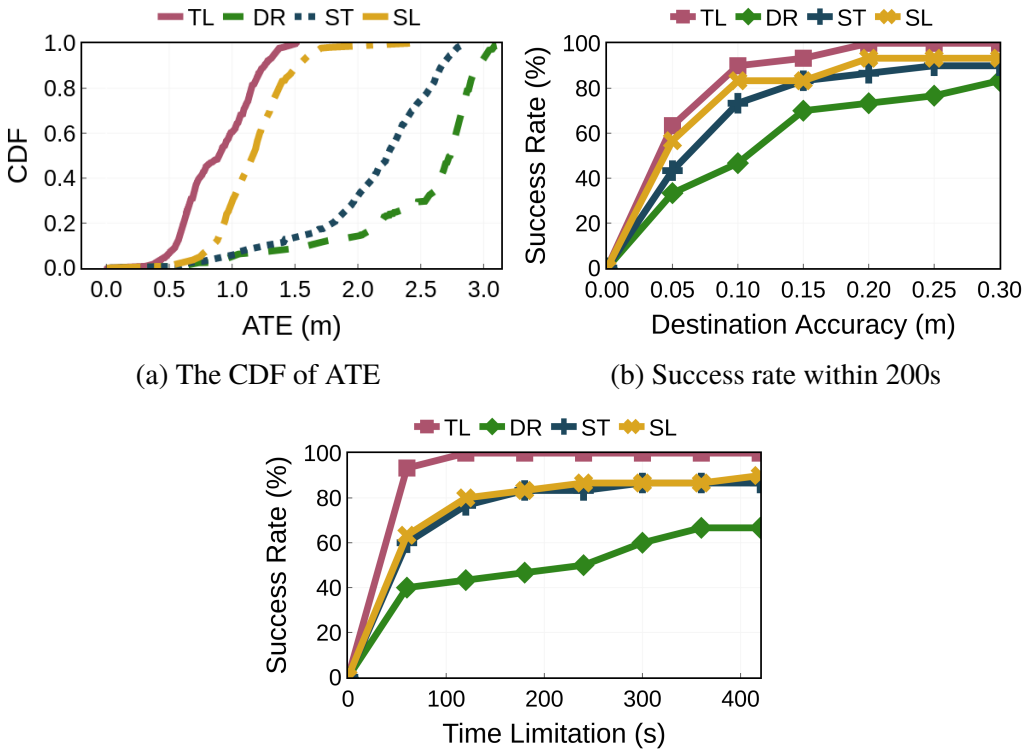
Evaluation metrics. The primary objective of TransformLoc is to enhance both the localization accuracy and navigation success rate of BMAVs. To assess its effectiveness, this work employs two key performance metrics:

- *Localization error:* This metric evaluates the localization accuracy of BMAVs by comparing their estimated locations against ground truth data at each timestep. This work quantifies the localization error using the absolute trajectory error (ATE).
- *Success rate:* The success rate metric measures the proportion of BMAVs that successfully reach their destinations within predefined time and destination constraints.

5.5.2 Overall Performance

In-field experiments. Figure 5.7 provides a detailed analysis of the outcomes obtained from in-field experiments. To evaluate localization accuracy, this work conducts a 420-second random walk for BMAVs, plotting the cumulative distribution function (CDF) of the ATE. TransformLoc achieves an ATE below $1.5m$, outperforming SL, ST, and DR, which achieve ATEs below $2.4m$, $2.8m$, and $3m$, respectively (as depicted in Figure 5.7(a)).

Regarding navigation performance, this work assesses the success rate of BMAVs over a 200-second duration under varying destination accuracy constraints (as illustrated



(a) The CDF of ATE

(b) Success rate within 200s

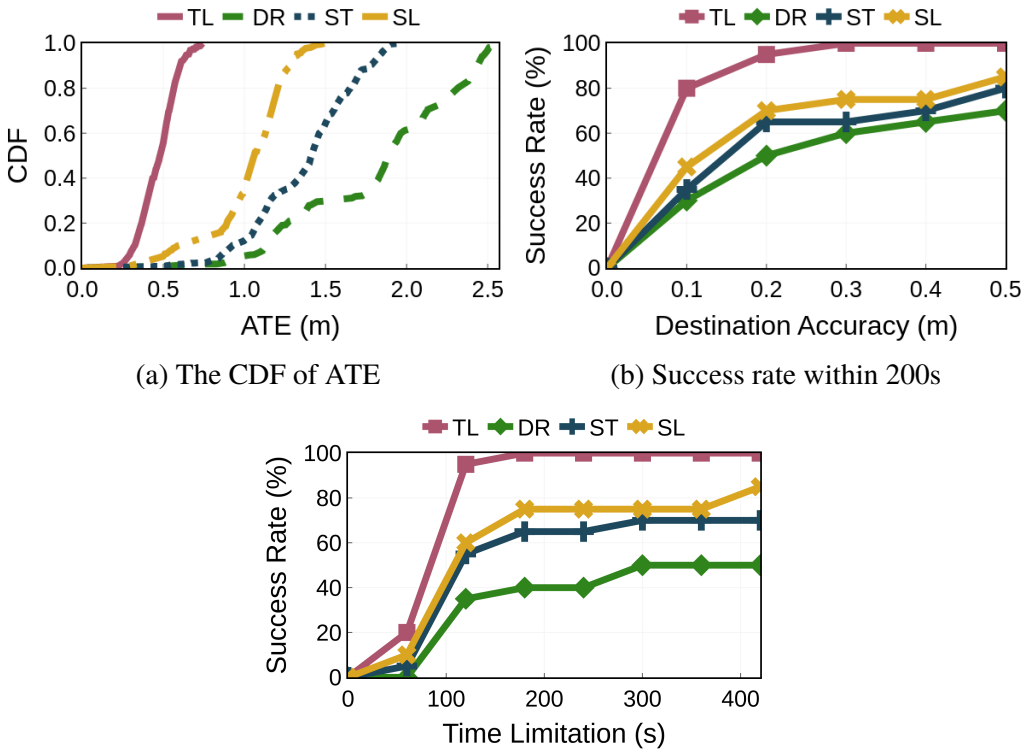
(c) Success rate with 0.2m accuracy

Figure 5.7 Overall performance in in-field experiments with two AMAVs and six BMAVs.

in Figure 5.7(b)). TransformLoc outperforms all baselines, achieving a remarkable 63% success rate with a stringent destination accuracy constraint of 0.05m. Furthermore, under looser destination accuracy limits (ranging from 0.2m to 0.3m), TransformLoc attains a 100% navigation success rate, surpassing the performance of the baselines.

Additionally, as the time limitation increases, the success rate of BMAVs with a destination accuracy constraint of 0.2m improves for all methods, benefiting from more time to reach the target (as shown in Figure 5.7(c)). Notably, TransformLoc exhibits superior performance compared to the baselines, achieving a navigation success rate 22.3% higher than SL, 25.6% higher than ST, and 55.6% higher than DR within strict time constraints (ranging from 60 to 180 seconds). These results underscore the efficacy of TransformLoc in proficiently allocating AMAVs' resources to enhance the localization and navigation capabilities of BMAVs.

Physical feature-based simulations experiments. The simulations in this section evaluate the navigation success rate of the BMAVs under different destination constraints and time constraints. In the meantime, the localization error of all BMAVs is evaluated. To further verify TransformLoc's localization performance, this work initiates a 420-second random walk for each BMAV. Figure 5.8(a) shows the CDF of localization errors. Trans-



(a) The CDF of ATE

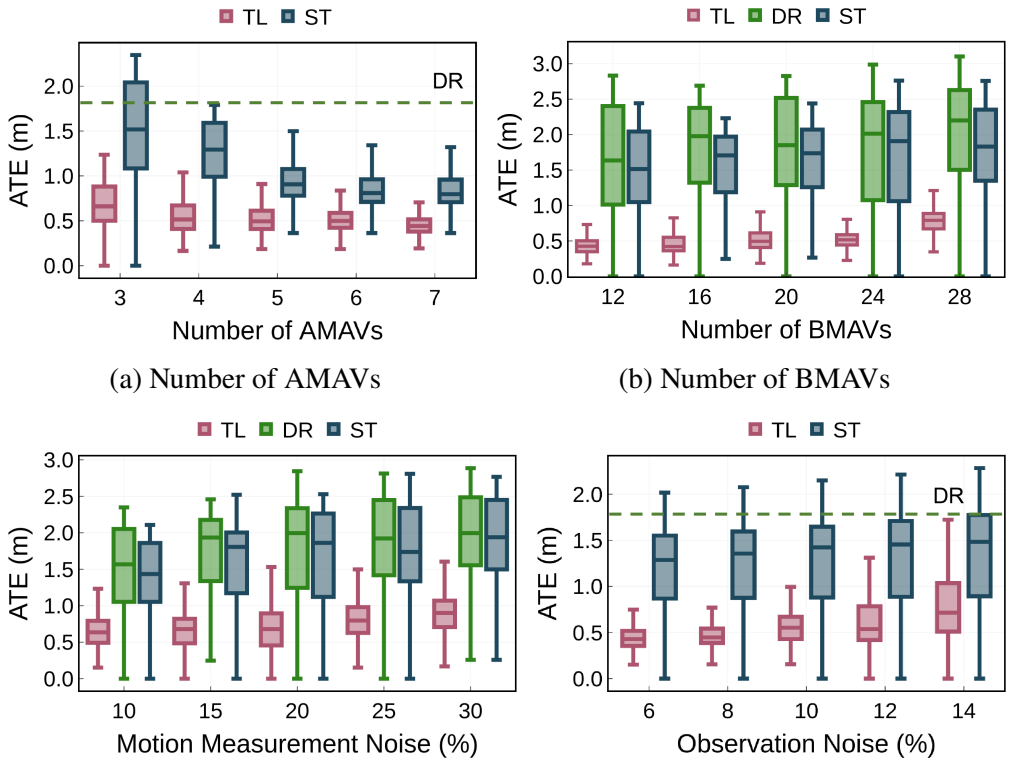
(b) Success rate within 200s

(c) Success rate with 0.2m accuracy

Figure 5.8 Overall performance in in-field experiments with two AMAVs and six BMAVs.

formLoc achieves an impressive ATE below 0.7m, while SL, ST, and DR maintain ATE below 1.5m, 2m, and 2.5m respectively.

This work evaluates TransformLoc's navigation performance across various destination accuracy constraints by analyzing the success rate of BMAVs' navigation within a 200-second duration, as depicted in Figure 5.8(b). The success rate of BMAVs' navigation increases with increased destination accuracy, as the increase in destination accuracy means the farther away the destination is when the BMAV is successfully navigated. TransformLoc consistently outperforms all baselines across different destination accuracy conditions. For strict destination accuracy (0.2m), TransformLoc achieves an impressive 80% success rate, while all baselines fall below 50%. Under less constrained destination accuracy (ranging from 0.4 to 0.6 m), TransformLoc maintains a consistent 30% higher success rate than the baselines. At the loosest destination accuracy (0.8m), TransformLoc achieves a flawless 100% navigation success rate, while the baselines remain below 80%. On average, TransformLoc exhibits substantial performance advantages over SL, ST, and DR, with margins of 25%, 26.67%, and 33.3%, respectively. These findings underscore TransformLoc's effectiveness in accommodating varying destination accuracy levels and achieving consistently high navigation success rates.



(a) Number of AMAVs (b) Number of BMAVs

(c) Motion measurement noise (d) Observation noise

Figure 5.9 System robustness evaluation with five AMAVs and twenty BMAVs.

To validate TransformLoc's navigation performance under various time constraints, this work analyzes the navigation success rate of BMAVs with a destination accuracy constraint of 0.2m, as depicted in Figure 5.8(c). As the time limitation increases, all methods exhibit improved navigation success rates for BMAVs, given the extended duration to reach their destinations. TransformLoc consistently outperforms the other baselines across different time constraints. Under strict time constraints (60-180 seconds), TransformLoc achieves a navigation success rate nearly 10% higher than ST and SL. As time constraints become more relaxed (240-420 seconds), TransformLoc achieves a perfect 100% navigation success rate, surpassing all other baselines. On average, TransformLoc demonstrates significant performance advantages over SL, ST, and DR by 22.8%, 25.6%, and 42.5%, respectively. By strategically dispatching AMAVs to provide observations for BMAVs near their final destination, TransformLoc effectively reduces localization errors and enhances the navigation success rate. In summary, TransformLoc demonstrates the capability to efficiently allocate AMAVs' sensing resources, thereby improving BMAVs' navigation success rates across varying time constraints. In conclusion, TransformLoc dispatches AMAVs to provide timely observations for BMAVs, resulting in lower errors.

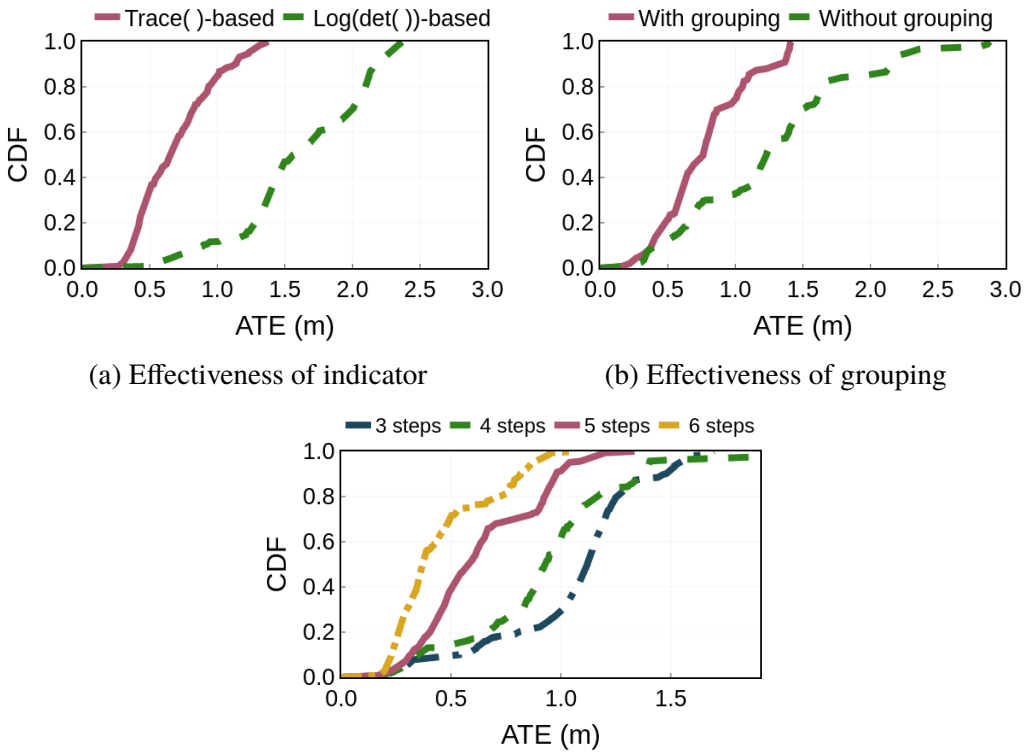
5.5.3 System Robustness Evaluation

Impact of number of AMAVs. Figure 5.9a presents a comparison of the ATE of BMAVs across different methods and varying numbers of AMAVs. TransformLoc consistently achieves lower errors compared to DR, indicating that the inclusion of AMAVs contributes to the reduction in ATE. As the number of AMAVs increases from 3 to 7, TransformLoc demonstrates a notable decrease in ATE to below $0.8m$. This reduction is substantial compared to ST, which only assists BMAVs when traversing over fixed AMAVs. These findings underscore the significant impact of increasing the number of AMAVs on decreasing the ATE of BMAVs. Overall, the results highlight the effectiveness of TransformLoc in reducing localization errors through the allocation of AMAVs.

Impact of number of BMAVs. Figure 5.9b depicts the performance of different methods in localizing varying numbers of BMAVs. As the number of BMAVs increases from 12 to 28, the ATE for all methods also increases. This trend is attributed to a smaller percentage of BMAVs being assisted by AMAVs as the overall BMAV count rises. However, even with the increasing number of BMAVs, TransformLoc consistently achieves an ATE of less than $1.2m$. This result is more than 57% lower than any other baseline method. These experimental findings highlight the efficacy of TransformLoc in efficiently distributing the sensing resources of AMAVs to cater to different numbers of BMAVs, thereby ensuring a consistently low localization error.

Impact of motion measurement noise of BMAV. Motion measurement noise plays a pivotal role in determining the accuracy of BMAV localization. As this noise level increases, so does the localization error of BMAVs. The performance of TransformLoc under varying levels of motion measurement noise is assessed by plotting the ATE in Figure 5.9(c). Even amidst increasing noise levels, TransformLoc consistently achieves a lower ATE compared to other baseline methods. Notably, TransformLoc maintains an error of less than $1.6m$ even when motion measurement noise reaches 30%. These results underscore TransformLoc's ability to allocate AMAVs' sensing resources for timely observation generation, particularly in scenarios with significant motion measurement noise.

Impact of observation noise of AMAV. Experimental results from the testbed highlight the substantial influence of range measurement noise on observation quality [195]. To assess the effect of range measurement noise, this work analyzes the ATE under different noise conditions, as depicted in Figure 5.9(d). As the level of range measurement noise rises, the discrepancy between observed and actual BMAV locations widens, resulting



(a) Effectiveness of indicator

(b) Effectiveness of grouping

(c) Motion command interval

Figure 5.10 System micro-benchmark.

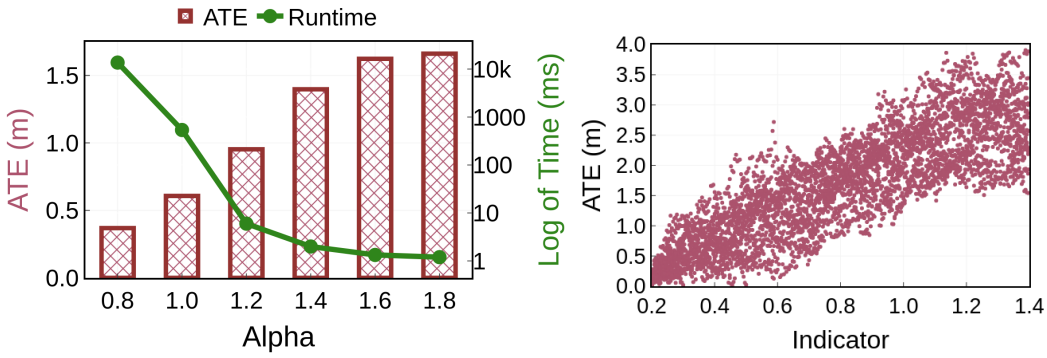
in an increase in ATE across all methods. TransformLoc exhibits superior performance compared to the baselines across various noise conditions. By strategically planning the trajectories of AMAVs to provide observations in closer proximity to BMAVs, TransformLoc effectively mitigates the impact of noise on localization accuracy of BMAVs.

5.5.4 System Micro-benchmark

This work conducts additional experimental analyses to scrutinize key components of TransformLoc, focusing on the individual performance enhancements contributed by each component to the overall system.

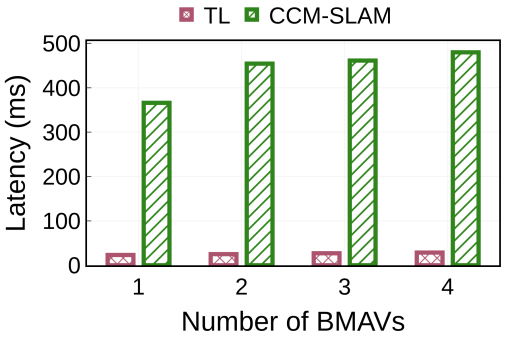
Effectiveness of indicator. In Figure 5.10(a), this work presents the CDF of the ATE using two different indicators: the trace-based and the logarithm determinant-based[170]. The results reveal that the trace-based indicator consistently yields lower errors. Particularly noteworthy is the sharper incline in the CDF curve based on the trace-based indicator, even for ATE values below $1.0m$, compared to the slower growth observed with the logarithm determinant-based indicator. This difference suggests that trace-based indicator provides a more precise estimation for BMAVs, resulting in enhanced performance.

Effectiveness of grouping module. In Figure 5.10(b), this work analyzes the influ-



(a) Impact of parameter

(b) Correlation of ATE and indicator



(c) Localization latency

Figure 5.11 System efficiency.

ence of the grouping module on the overall localization performance. The results demonstrate that the presence of the grouping module consistently leads to lower ATE values. Specifically, the system incorporating the grouping module exhibits a maximum ATE lower than 1.4m, while the system without it experiences a maximum error nearly twice as large. These findings underscore the effectiveness of the grouping module in efficiently allocating AMAVs' sensing resources, thereby enhancing utility and ensuring a timely response for BMAVs' localization.

Motion command interval of BMAVs. In Figure 5.10(c), the CDFs of ATE are depicted for four different command intervals, demonstrating that larger intervals correspond to lower localization errors. A larger interval allows TransformLoc to anticipate the motion of BMAVs across multiple future states, providing richer information for the strategic planning of AMAVs. Consequently, this facilitates more accurate localization for BMAVs in the multi-MAV system.

5.5.5 System Efficiency

Impact of parameter. The AMAV search tree pruning method involves eliminating nodes in the search tree that are physically close and exhibit similar BMAV covariances. Evaluating ϵ-algebraic redundancy constitutes the computationally intensive core operation in this pruning process. To investigate the interplay between ATE, computational time, and the pruning parameter ϵ, Figure 5.11(a) is generated with varying values of ϵ. The results illustrate that increasing ϵ leads to reduced computational time, owing to the decrease in the number of search tree nodes. However, this reduction in time comes at the cost of increased ATE, as the policy tends towards a more greedy approach, posing challenges in achieving non-myopic objectives. Notably, when ϵ is set to 1.2, the AMAV planning process requires less than $10ms$ to execute, with further reductions in time attainable by increasing ϵ.

Correlation of ATE and indicator. To gauge the efficacy of the indicator, this work delves deeper into the insights provided by Figure 5.11(b). The visual representation of the indicator ($tr(\Sigma_{j,t})$) unveils a conspicuous association with the ATE of BMAVs. This elucidation underscores the indicator's capacity to faithfully mirror the degree of localization error experienced by the BMAVs, affirming its utility as a reliable metric for assessing positional accuracy.

Localization latency. To evaluate the latency of TransformLoc, this work analyzes the data presented in Figure 5.11(c). The findings unveil that TransformLoc exhibits remarkable efficiency, swiftly generating observations and localizing four BMAVs within a mere $28.58ms$. This swift processing time underscores the agility of TransformLoc in handling localization tasks. In stark contrast, the comparative method, CCM-SLAM, lags significantly behind, demanding a substantially longer time of $479.93ms$ for aligning point clouds to facilitate localization. This notable discrepancy highlights the superior speed and efficiency of TransformLoc in comparison to traditional methods like CCM-SLAM.

5.6 Discussion

In this part, this work explores several influential factors affecting TransformLoc's operational performance.

(i) **Communication overload.** BMAVs regularly transmit location estimations (mean and covariance) to the AMAVs. With twenty BMAVs, each AMAV processes and transmits less than 10KB of data every δ seconds. Similarly, during grouping, an AMAV

gathers location information from other AMAVs, resulting in a total data amount of less than 10KB with twenty AMAVs. Thus, the communication load remains acceptable.

(*ii*) **Localization accuracy of AMAV.** When the localization error of an AMAV is less than $10cm$, the system significantly enhances the localization accuracy of BMAVs. Achieving this level of accuracy is feasible with an AMAV equipped with advanced sensors like depth cameras.

(*iii*) **The quantity of AMAV and BMAV.** In the best-case scenario, the number of BMAVs equals the number of AMAVs ($N = M$). Under this condition, each AMAV can consistently observe a single BMAV, transforming its sensing and computational resources to it, thereby maintaining the swarm's localization error at a relatively low level. As the ratio of BMAVs to AMAVs (N/M) increases, the spatial dispersion of BMAVs grows, leading to some BMAVs always falling outside the AMAVs'sensing range, which can degrade localization accuracy. Speed limitations exacerbate this issue: AMAVs may struggle to track dispersed BMAVs, while AMAVs waste energy maneuvering without effectively reducing localization errors. Additionally, the limited FOV of AMAVs restricts their ability to observe multiple BMAVs simultaneously, particularly in dense or scattered configurations, resulting in unmonitored regions and increased localization error. These factors collectively undermine the swarm's accuracy and stability. The experiments in an $8m \times 8m$ environment show that when the number of AMAVs is 5, the localization error increases significantly if the number of BMAVs exceeds 25, with a notable rise in collision risk. Thus, the worst-case scenario occurs when the number of BMAVs is at least five times that of AMAVs ($N \geq 5M$), where both error and collision risk escalate.

(*iv*) **Scalability of TransformLoc.** When enhancing the scalability of TransformLoc to larger swarms or complex environments[196], several factors need to be considered:

1. *Resource allocation efficiency.* The adaptive grouping-scheduling strategy has demonstrated effectiveness in allocating sensing and computing resources of AMAVs within moderate-sized swarms. For larger swarms, techniques such as hierarchical grouping or distributed scheduling could enhance scalability. Hierarchical grouping organizes MAVs into nested subgroups, allowing localized decision-making and reducing the burden on central coordination[197]. Distributed scheduling, on the other hand, leverages decentralized algorithms to enable individual MAVs or swarms to independently manage their tasks, thereby enhancing responsiveness[198]. Additionally, learning-based grouping-scheduling[199], which lever-

ages attention mechanisms to model the dynamic relationships among MAVs or swarms, offers another promising approach for adaptive grouping and allocation efficiency enhancement.

2. *Computational overhead.* Increasing the number of MAVs introduces higher computational demands for joint location estimation and scheduling. To address these challenges, strategies such as pruning in the search tree and parallel processing can be highly effective. Pruning in the search tree reduces computational overhead by eliminating suboptimal or redundant branches early, thereby focusing resources on promising solutions[192]. Parallel processing, meanwhile, leverages the computational power of multi-core processors or distributed systems to execute location estimation and scheduling tasks simultaneously, dramatically improving efficiency. Additionally, hybrid approaches that integrate heuristic optimization algorithms or learning-based methods, such as reinforcement learning[200], can further enhance scalability and adaptability, especially in resource-constrained environments.
3. *Environmental complexity*. In more dynamic or complex environments, additional constraints such as obstacle avoidance or varying communication reliability must be addressed. The modular design of TransformLoc allows for the integration of corresponding solutions into the framework to address these constraints. For example, advanced obstacle avoidance algorithms for both AMAVs and BMAVs[162,201] can be embedded into the framework, leveraging real-time sensor data and predictive modeling to enable safe navigation in cluttered or rapidly evolving environments. To enhance scalability, adaptive and efficient communication media are key. Wi-Fi mesh networks expand coverage dynamically[202], LoRa supports large-scale deployments with low-power, long-range communication[203], and 5G leverages massive device connectivity and network slicing for scalability enhancement[204]. Combining Wi-Fi for bandwidth with LoRa for coverage could ensure scalable performance in complex environments. Furthermore, implementing more stable and adaptive communication methods, such as multi-hop relay protocols, can enhance connectivity in scenarios where communication reliability fluctuates[196].

(υ) **Time complexity.** TransformLoc consists of two main components: Error-Aware Joint Location Estimation and Similarity-Instructed Adaptive Grouping-Scheduling.

1. *Error-Aware Joint Location Estimation*. In the Prediction Step, the algorithm performs two operations: state propagation through the motion model and covariance

propagation, with respective time complexities $O(n^2)$ and $O(n^3)$, where n represents the state dimension. These complexities arise due to matrix multiplication and inversion. For N BMAVs, the total time complexity of this step is $O(Nn^3)$. The BMAV state includes position information, denoted as $y_{i,t}$. In the Correction Step, the operations include checking if a BMAV is within an AMAV's FoV, observation model updates, and Kalman corrections. For N BMAVs and M AMAVs, these operations have respective time complexities of $O(NM)$, $O(n^2)$, and $O(n^3)$. Thus, the time complexity of this step is $O(NMn^3)$.

2. *Similarity-Instructed Adaptive Grouping-Scheduling*. In the Graph-Based Adaptive Grouping, the operations include computing Voronoi diagrams for AMAV grouping and assigning BMAVs to AMAV regions. For N BMAVs and M AMAVs, these operations have time complexities of $O(MlogM)$ and $O(N)$, respectively, resulting in a total time complexity of $O(MlogM + N)$ for this step. In the Search Tree-Based Non-Myopic Scheduling, the operations involve constructing the search tree and pruning nodes. For each AMAV with k motion options, a lookahead horizon of δ, and p pruned nodes, these operations have time complexities of $O(k^\delta Nn^2)$ and $O(pNn^2)$, respectively. The overall time complexity of this step is $O(Mk^\delta Nn^2)$. The total time complexity of the algorithm is $O(NMn^3+Mk^\delta Nn^2+MlogM+N)$. From this, this work derives several key insights: 1. High-Dimensional State (n): The n^3 cost, arising from Kalman updates, dominates in this case. 2. Large Control Space or Horizon (k, δ): The k, δ factor becomes dominant in the scheduling step, especially with many AMAVs (M). Tree pruning is essential for ensuring real-time feasibility. 3. Number of Agents (N, M): The factor NM significantly affects the correction step and AMAV-BMAV pairing. Larger swarms increase computational demands depending on the component.

(vi) **Dual-role AMAVs and multi-objective trajectory optimization.** In the current TransformLoc framework, AMAVs act as dedicated mobile localization infrastructures to support the BMAV fleet. However, in more complex operational scenarios, AMAVs might be required to simultaneously fulfill their own specific mission objectives (e.g., executing high-resolution inspection of a critical target) while providing localization assistance. This dual-role paradigm introduces a highly complex multi-objective optimization problem. The trajectory generation for AMAVs would need to continuously balance the trade-off between progressing toward their individual mission waypoints and deviating to

maximize the spatial-temporal localization coverage for drifting BMAVs. Developing dynamic, multi-objective scheduling algorithms to resolve this mission-assistance trade-off represents a critical direction for the future evolution of multi-MAV systems.

(*vii*) **Value and feasibility of deep hardware-algorithm Coupling.** As currently implemented, the TransformLoc framework connects with individual perception modules primarily at a serial, state-sharing level. While the deep, feature-level integration of event cameras and mmWave radars into the swarm loop remains an ongoing endeavor, it is both highly feasible and of immense theoretical value. From a value perspective, equipping the AMAVs with event-based perception (EV-Pose) would fundamentally resolve the issue of visual tracking failure caused by motion blur during aggressive swarm maneuvers, while simultaneously enabling stable localization in HDR environments. Furthermore, utilizing the active sensing capabilities of mmWave radar (mmE-Loc) would enable ultra-low latency air-to-air relative tracking and allow the swarm to maintain peer-to-peer ranging even in visually degraded conditions. From a feasibility standpoint, the AMAVs—acting as mobile infrastructures—inherently possess the necessary payload capacity and computational resources to carry these advanced sensors, and the distributed framework validated in this chapter provides a stable algorithmic foundation for this future integration.

5.7 Summary

In this chapter, this thesis introduces TransformLoc, a novel framework that revolutionizes the role of AMAVs by transforming them into mobile localization infrastructures to support large-scale multi-MAV system localization. TransformLoc is distinguished by two key innovations: Firstly, this work introduces an error-aware joint location estimation model. This model seamlessly integrates the imprecise estimates from BMAVs with observations from AMAVs, facilitating joint location estimation of BMAVs. To achieve this, this work employs an uncertainty-aided inference method, ensuring robustness and accuracy in localization. Secondly, this work introduces a similarity-instructed adaptive grouping-scheduling strategy. This innovative approach dynamically allocates AMAVs' sensing resources through a combination of a graph-based adaptive grouping method and a search tree-based non-myopic scheduling method. Additionally, this work proposes a similarity-instructed search tree pruning method to further improve computational efficiency. Experimental evaluations demonstrate that the proposed TransformLoc framework outperforms existing solutions, showcasing its efficacy and superiority.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

With the rapid ascent of the low-altitude economy, drones have evolved into indispensable carriers for high-frequency logistics, emergency response, and autonomous inspection. However, the transition from controlled laboratory settings to unpredictable real-world environments has exposed a fundamental "Perception-Latency-Coordination" trilemma. Conventional localization methods, primarily designed for static or slow-moving entities, struggle to provide the high-fidelity state estimation required for agile drones, particularly when faced with GNSS-denied signal masking, exposure-induced motion blur, and the spatial tethering of fixed sensors.

To confront these intrinsic bottlenecks, this dissertation adopts a novel intelligent perception paradigm with the bio-inspired dynamic vision. By leveraging the microsecond-level temporal resolution of event cameras, this research systematically transforms conventional drones into autonomous low-altitude agents. The thesis executes this transformation across two core dimensions: developing Enacted agents through tightly coupled perception-action loops, and establishing Engaged coordination through intermittent swarm interactions.

(1) *Constructing the external information-based enacted perception method (mmE-Loc).* To address the latency and rigidity of individual sensing during critical maneuvers like landing, this research realizes the Enacted dimension from an exogenous perspective. This thesis proposes mmE-Loc, a method that transforms ground infrastructure into a bio-inspired "external sensory organ" by fusing mmWave radar and event cameras. By synergizing the temporal coherence and spatial complementarity of these modalities, the method achieves centimeter-level precision and millisecond-level latency, closing the drone's perception-action loop externally and ensuring physiological-grade responsiveness in GNSS-denied environments.

(2) *Developing the internal information-based enacted perception method (EV-Pose).* Shifting to endogenous onboard autonomy, this thesis further deepens the Enacted dimension by proposing EV-Pose, an event-camera-enhanced visual positioning service. To break the constraints of discrete frame-based sampling, EV-Pose utilizes asynchronous event streams to emulate the spiking mechanisms of the biolog-

ical retina. By extracting Spatio-Temporal Features to overcome semantic sparsity and implementing a motion-aware hierarchical fusion mechanism, EV-Pose directly couples onboard perception with dynamic actuation. This establishes a highly compact, self-contained perception-action loop, allowing the agent to continuously enact precise 6-DoF pose tracking even under dynamic visual degradation.

(3) *Facilitating multi-agent-oriented engaged collaboration method (TransformLoc).* Having established robust enacted agents, the thesis transcends individual physical boundaries to realize the Engaged dimension at the system level. This thesis proposes TransformLoc, a hierarchical localization framework that shifts the swarm from isolated sensing to "participatory interaction." To overcome the spatial tethering of static base stations, TransformLoc repurposes Advanced MAVs (AMAVs) as dynamic mobile infrastructures. The framework exploits physical spatial constraints to facilitate engaged, opportunistic information alignment, providing on-demand observational corrections for resource-constrained Basic MAVs (BMAVs). This heterogeneous symbiosis ensures that multi-source information dynamically flows across the swarm, significantly enhancing collective survivability and system-wide scalability without increasing overall hardware costs.

6.1.1 Holistic Adaptive Mode-switching Strategy

The methodologies presented in this thesis equip the drones with distinct operational capabilities: external enacted perception (mmE-Loc), internal enacted perception (EV-Pose), and engaged swarm collaboration (TransformLoc). To elevate these isolated capabilities into a unified intelligence system, a holistic decision-making strategy is required to guide the drones in dynamically switching between these working modes based on real-time internal states and external conditions.

This work can further incorporate a three-tiered adaptive switching framework governed by the following triggers: *(i) The Default State: Single-drone Mode (EV-Pose).* During standard low-altitude cruising or environmental exploration, the drone defaults to the Single-drone Mode. Relying on its onboard event camera and EV-Pose framework, the drone maintains autonomous navigation. This mode ensures maximum operational independence and minimizes swarm communication bandwidth overhead. *(ii) Opportunistic Switching: Multi-drone Collaboration Mode (TransformLoc)* The drone dynamically requests assistance and transitions to the Multi-drone Collaboration Mode under unexpected internal or external stressors: First, in the case of state/environmental degradation. If the

onboard EKF covariance diverges beyond a safety threshold, or if the event camera registers an extreme drop in spatio-temporal density (e.g., entering dense smoke or an absolute void), the drone proactively requests opportunistic localization correction from nearby AMAVs. Second, in the case of resource bottlenecks. Under critical battery or computational constraints, the drone can enter a power-saving state by shutting down active internal sensing, entirely offloading its state estimation to the surrounding AMAV dynamic infrastructure. *(iii) Deterministic Switching: Ground-Assisted Infrastructure Mode (mmE-Loc).* While TransformLoc provides robust relative swarm navigation, terminal flight maneuvers require absolute, centimeter-level precision. The system deterministically switches to the Ground-Assisted Infrastructure Mode based on spatial and mission triggers: First, in the case of terminal phase geofencing. When the drone enters the geofence of a designated drone airport or logistics hub, and initiates the terminal takeoff or landing phase, it seamlessly establishes a high-bandwidth link with the ground. Second, in the case of authority handover. To overcome severe ground-effect dynamics and ensure millimeter-level docking, the flight controller temporarily prioritizes the high-precision, external absolute coordinates provided by the ground-based mmE-Loc method over its internal estimators.

By implementing this dynamic, state-aware switching strategy, the agents can seamlessly transition from infrastructure-assisted liftoff (mmE-Loc), to autonomous cruising (EV-Pose), and highly resilient swarm operations (TransformLoc), achieving a fully unified perception-decision-action lifecycle.

6.1.2 Broader Applicability Beyond Terminal Landing

While the empirical evaluations of mmE-Loc and EV-Pose heavily feature the terminal landing phase—chosen as it represents the most demanding operational scenario with the strictest constraints on latency and precision—the proposed enacted perception frameworks are fundamentally agnostic to the flight phase. These techniques provide substantial utility in broader operational contexts. For instance, during the takeoff phase in dense urban delivery hubs or GPS-denied environments, drones must navigate upward through complex, obstacle-rich infrastructures. In such cases, mmE-Loc can safely guide the ascent from the ground pad, while EV-Pose ensures continuous, drift-free absolute tracking before the drone reaches an altitude sufficient for reliable satellite navigation. Furthermore, during low-altitude urban cruising, EV-Pose's high dynamic range and immunity to motion blur make it an exceptionally powerful tool for high-speed agile navigation and dynamic obstacle avoidance in complex urban corridors.

6.1.3 Summary

This thesis establishes a logically progressive "Perception-Interaction" dual-driven framework for aerial intelligence perception. By re-designing the agent's external and internal senses to realize Enacted autonomy and leveraging heterogeneous symbiosis to achieve Engaged coordination, this research provides a comprehensive solution to the perceptual ambiguity, computational latency, and coordination complexity of conventional MAVs. Ultimately, this work lays the critical theoretical and empirical foundation for the deployment of large-scale aerial swarms in the low-altitude economy.

6.2 Future Work

6.2.1 Integration of Enacted Perception and Engaged Coordination

While this thesis presents specialized solutions for external enacted perception (mmE-Loc), internal enacted perception (EV-Pose), and engaged swarm coordination (TransformLoc), respectively, their ultimate potential lies in a deeply integrated, unified intelligent perception architecture. However, fusing these systems introduces several profound new technical challenges:

(1) *Integration of mmE-Loc and EV-Pose.* In a terminal landing scenario, the ground-based mmE-Loc could serve as a highly reliable external anchor for the onboard EV-Pose method. When the drone enters the landing zone, mmE-Loc's absolute 3D coordinates can seamlessly correct any accumulated global drift in EV-Pose's map matching. *New Technical Challenge:* The primary hurdle is cross-platform spatiotemporal synchronization. Transmitting ground-based mmE-Loc estimates to the onboard flight controller introduces inevitable wireless network latency. This latency fundamentally conflicts with the ultra-low latency (<10 ms) advantage of EV-Pose, requiring novel predictive sensor fusion algorithms that can robustly align delayed ground observations with real-time onboard event streams.

(2) *Integration of Enacted Perception into Engaged Swarms.* The TransformLoc framework utilizes Advanced MAVs (AMAVs) as mobile infrastructures. If these AMAVs are equipped with the event-based perceptual capabilities developed in mmE-Loc and EV-Pose, they could perform air-to-air tracking of resource-limited Basic MAVs (BMAVs) with microsecond resolution. *New Technical Challenge:* Unlike mmE-Loc, where the ground platform is static, air-to-air event tracking in-

volves severe ego-motion from the observing AMAV. Extracting the target BMAV's periodic micro-motions (e.g., propeller rotation) from an event stream dominated by the massive background optical flow caused by the AMAV's own flight is a highly complex non-linear filtering problem that remains an open challenge.

(3) *From Serial System Engineering to Deep Multi-modal Swarm Coupling.* A critical and highly promising direction for future research is to strengthen the in-depth coupling between the single-drone perception hardware proposed in mmE-Loc and EV-Pose with the multi-drone collaboration framework presented in TransformLoc. Currently, these modules are connected via serial system engineering. Future works should aim to break this serial boundary by implementing 'feature-level' or 'raw-data level' deep coupling. For instance, rather than simply transmitting computed coordinate states, adjacent drones could directly share compressed asynchronous event streams or sparse mmWave point clouds over high-bandwidth communication links. This deep integration would fully exploit the distinct physical advantages of both modalities: utilizing event streams to achieve blur-free, HDR resilient cross-drone feature matching during aggressive maneuvers, and leveraging the active sensing of mmWave radar to conduct distributed global state optimization even in visually degraded environments. Ultimately, this paradigm shift will realize a truly unified and resilient swarm intelligence system.

6.2.2 Toward the Evolutionary Dimension

While this dissertation has successfully established the foundational Enacted and Engaged dimensions, the ultimate vision for aerial agents in the low-altitude economy necessitates a transition toward the Evolutionary dimension. High-precision, low-latency, and highly scalable localization technologies serve as the critical bedrock for autonomous navigation. However, as operational environments become increasingly unstructured and swarm scales expand, relying solely on static algorithms or pre-defined interaction rules will become a bottleneck.

To fully realize autonomous, self-evolving swarm intelligence, future research can build upon the findings of this thesis and advance into the Evolutionary paradigm across the following key areas:

(1) *Evolutionary perception for extreme precision.* This study leveraged event cameras to enhance both external and internal information-based enacted agents, achieving robust centimeter-level localization. However, emerging drone applications, such

as navigation in highly confined spaces or collaborative object manipulation, demand sub-centimeter precision. Future research could explore evolutionary perception models that continuously learn and adapt to dynamic environmental noise. By integrating novel sensor technologies and self-calibrating algorithms, the localization infrastructure can iteratively refine its measurement models based on historical flight data, achieving continuously improving accuracy over time.

(2) *Adaptive and predictive low-latency processing.* This study has successfully utilized the microsecond-level temporal resolution of event cameras to establish a low-latency perception-action loop. Nevertheless, high-speed drones operating in dense, dynamic environments (e.g., encountering unpredictable moving obstacles) require millisecond-level predictive capabilities. Future work could investigate adaptive, learning-based processing architectures, such as Spiking Neural Networks (SNNs), that evolve their computational pathways based on flight velocity and environmental complexity. This evolutionary approach would allow the agent to dynamically compress processing latency and make proactive, rather than reactive decisions.

(3) *Context-aware evolutionary sensor fusion.* While this thesis focused on bridging infrastructure-involved localization with onboard sensing, a vast body of research exists on standalone SLAM and collaborative SLAM. To maximize environmental resilience, future low-altitude agents must evolve the capacity to dynamically evaluate and seamlessly switch between diverse localization modalities. An evolutionary heterogeneous fusion architecture could autonomously adjust the weights of external infrastructure, onboard SLAM, and multi-agent relative sensing based on real-time context and historical reliability, ensuring uninterrupted Enacted autonomy even in severely degraded scenarios.

(4) *Self-organizing evolutionary swarm coordination.* Building upon the TransformLoc framework, which established Engaged opportunistic interactions by repurposing Advanced MAVs as mobile infrastructure, the next critical frontier is true Evolutionary swarm intelligence. As the low-altitude economy scales, heuristically designed grouping and scheduling strategies will face high-dimensional computational bottlenecks. Future research must focus on developing self-organizing, evolutionary resource allocation strategies. By utilizing paradigms such as Multi-Agent Reinforcement Learning (MARL), heterogeneous swarms can continuously evolve their collaborative behaviors, learning to optimally distribute sensing tasks and mo-

bile anchor positions based on long-term environmental feedback.

(5) *Scaling Collective Coordination for High-Throughput Low-Altitude Economies.* To transition from the current evaluation of dozens of agents to the thousands required for a mature low-altitude economy, the scalability and computational agility of collective coordination must be addressed. The future research will focus on expanding the swarm capacity while simultaneously minimizing resource allocation latency. This necessitates a shift from centralized tree-based scheduling to decentralized, low-latency heuristics or predictive models capable of managing high-dimensional resource flows in real-time. Such advancements will ensure that as the density of the aerial fleet increases, the system's responsiveness remains at a physiological grade, enabling agile, non-myopic interactions across wide-area urban infrastructures.

(6) *Adaptive Enacted Perception in Complex 3D Topologies.* This work plans to transcend the 2D-centric assumptions of current flight paradigms by integrating full 3D dynamics and contextual adaptation into the intelligent perception architecture. Real-world low-altitude activities frequently involve varying altitudes and heterogeneous environmental constraints. This work intends to extend the *TransformLoc* framework to incorporate vertical motion models and 3D observation overlaps, allowing agents to dynamically adjust their perceptual focus based on terrain features and weather-specific sensor noise. This evolution will ultimately drive the transition of aerial platforms toward autonomous, self-organizing aerial swarms capable of continuous adaptation in the complex 3D airspace of future smart cities.

(7) *Evolution Toward Resilient Embodied Intelligence in Open Environments.* The current frameworks, such as *mmE-Loc* and *TransformLoc*, have demonstrated high precision under controlled or semi-structured conditions. However, a significant future trajectory involves enhancing system robustness against the inherent uncertainties of open-world deployments. This includes refining the observation noise models to account for non-zero mean errors induced by extreme weather and complex occlusions, moving beyond the current zero-mean Gaussian assumptions. Furthermore, this work aims to strengthen the system's tolerance to ego-motion errors from the infrastructure itself. By exploiting the latent spatio-temporal relationships between agents for mutual calibration, the swarm can maintain high-fidelity enacted perception even when primary localization anchors experience degradation.

Furthermore, as multi-drone systems become deeply integrated into complex soci-

etal applications, such as wide-area industrial inspections and collaborative transport, the boundary between localization and task execution will blur. Future studies could explore joint evolutionary frameworks that tightly couple accurate state estimation with autonomous task allocation. Endowing the swarm with the ability to iteratively learn and optimize these joint strategies will be the ultimate catalyst for maximizing operational efficiency and realizing the full potential of aerial embodied intelligence in the real world.

REFERENCES

[1] Duan J, Yu S, Tan H L, et al. A survey of embodied ai: From simulators to research tasks[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2022, 6(2): 230-244.

[2] Liu Y, Wang X, Liu Y, et al. Eiot: Embodied intelligence of things[J/OL]. Tsinghua Science and Technology, 2026, 31(1): 1-15. https://www.sciopen.com/article/10.26599/TST.2025.9010127.

[3] Wang Y, Young S, Komma D, et al. Global localization of energy-constrained miniature rf emitters using low earth orbit satellites[C]//Proceedings of the 21st ACM SenSys. 2023: 403-416.

[4] Dong H, Xie Y, Zhang X, et al. Gpsmirror: Expanding accurate gps positioning to shadowed and indoor regions with backscatter[C]//Proceedings of the 29th ACM MobiCom. 2023: 1-15.

[5] Wang W, Mottola L, He Y, et al. Micnest: Long-range instant acoustic localization of drones in precise landing[C]//Proceedings of the 20th ACM SenSys. 2022: 504-517.

[6] Xu J, Cao H, Li D, et al. Edge assisted mobile semantic visual slam[C]//Proceedings of the IEEE INFOCOM. 2020.

[7] Wang H, Xu J, Zhao C, et al. Transformloc: Transforming mavs into mobile localization infrastructures in heterogeneous swarms[M]//IEEE INFOCOM 2024-IEEE Conference on Computer Communications. 2024: 1101-1110.

[8] Merriaux P, Dupuis Y, Boutteau R, et al. A study of vicon system positioning performance[J]. Sensors, 2017, 17(7): 1591.

[9] Chen C, Kia S S. Optitrack-aided supervised learning for neural network-based ultra-wideband ranging bias correction[J]. IEEE Sensors Journal, 2024, 24(6): 8484-8492.

[10] Xue H, Cao Q, Ju Y, et al. M4esh: mmwave-based 3d human mesh construction for multiple subjects[C]//Proceedings of the 20th ACM SenSys. 2022: 391-406.

[11] Cui J, Shi S, He Y, et al. {VILAM}: Infrastructure-assisted 3d visual localization and mapping for autonomous driving[C]//Proceedings of the 21st USENIX NSDI. 2024: 1831-1845.

[12] Xu J, Chi G, Yang Z, et al. Followupar: Enabling follow-up effects in mobile ar applications [C]//Proceedings of the 19th ACM MobiSys. 2021: 1-13.

[13] He Y, Ma L, Cui J, et al. Automatch: Leveraging traffic camera to improve perception and localization of autonomous vehicles[C]//Proceedings of the 20th ACM SenSys. 2022: 16-30.

[14] Ben Ali A J, Kouroshli M, Semenova S, et al. Edge-slam: Edge-assisted visual simultaneous localization and mapping[J]. ACM Transactions on Embedded Computing Systems, 2022, 22 (1): 1-31.

[15] Tasneem Z, Adhivarahan C, Wang D, et al. Adaptive fovea for scanning depth sensors[J]. The International Journal of Robotics Research, 2020, 39(7): 837-855.

[16] Sie E, Liu Z, Vasisht D. Batmobility: Towards flying without seeing for autonomous drones [C]//Proceedings of the 29th ACM MobiCom. 2023: 1-16.

[17] Iizuka T, Sasatani T, Nakamura T, et al. Millisign: mmwave-based passive signs for guiding uavs in poor visibility conditions[C]//Proceedings of the 29th ACM MobiCom. 2023: 1-15.

[18] Duan L, Scargill T, Chen Y, et al. 3d object detection with vi-slam point clouds: The impact of object and environment characteristics on model performance[C]//2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024: 14014-14020.

[19] Chen Y, Inaltekin H, Gorlatova M. Adaptslam: Edge-assisted adaptive slam with resource constraints via uncertainty minimization[C]//IEEE INFOCOM 2023-IEEE Conference on Computer Communications. IEEE, 2023: 1-10.

[20] Hu T, Scargill T, Yang F, et al. Seesys: Online pose error estimation system for visual slam[C]// Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems. 2024: 322-335.

[21] Xu J, Cao H, Yang Z, et al. Swarmmap: Scaling up real-time collaborative visual slam at the edge[C]//Proceedings of the USENIX NSDI. 2022: 977-993.

[22] Zhao X, Wang G, An Z, et al. Understanding localization by a tailored gpt[C]//Proceedings of the 22nd ACM MobiSys. 2024: 318-330.

[23] Sun Y, He Y, Zhang J, et al. Bifrost: Reinventing wifi signals based on dispersion effect for accurate indoor localization[C/OL]//Proceedings of the 21th ACM SenSys. Istanbul, Turkiye: ACM, 2023: 376-389. DOI: 10.1145/3625687.3625786.

[24] Qian K, He Z, Zhang X. 3d point cloud generation with millimeter-wave radar[J]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2020, 4(4): 1-23.

[25] Zhao P, Lu C X, Wang B, et al. 3d motion capture of an unmodified drone with single-chip millimeter wave radar[C]//2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021: 5186-5192.

[26] Harlow K, Jang H, Barfoot T D, et al. A new wave in robotics: Survey on recent mmwave radar applications in robotics[J/OL]. IEEE Transactions on Robotics, 2024, 40: 4544–4560. http://dx.doi.org/10.1109/TRO.2024.3463504.

[27] Soumya A, Krishna Mohan C, Cenkeramaddi L R. Recent advances in mmwave-radar-based sensing, its applications, and machine learning techniques: A review[J/OL]. Sensors, 2023, 23 (21). https://www.mdpi.com/1424-8220/23/21/8901. DOI: 10.3390/s23218901.

[28] Lu C X, Rosa S, Zhao P, et al. See through smoke: Robust indoor mapping with low-cost mmwave radar[A/OL]. 2020. arXiv: 1911.00398. https://arxiv.org/abs/1911.00398.

[29] Solaiman S, Alsuwat E, Alharthi R. Simultaneous tracking and recognizing drone targets with millimeter-wave radar and convolutional neural network[J/OL]. Applied System Innovation, 2023, 6(4). https://www.mdpi.com/2571-5577/6/4/68. DOI: 10.3390/asi6040068.

[30] Abdelnasser H, Heggo M, Pang O, et al. Radro: Indoor drone tracking using millimeter wave radar[J/OL]. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 2024, 8(3): 1-23. https: //doi.org/10.1145/3678549.

[31] Li Y, Zhang D, Geng R, et al. A high-resolution handheld millimeter-wave imaging system with phase error estimation and compensation[J/OL]. Communications Engineering, 2024, 3. https://api.semanticscholar.org/CorpusID:266811642.

[32] Lu C X, Rosa S, Zhao P, et al. See through smoke: robust indoor mapping with low-cost mmwave radar[C]//Proceedings of the 18th MobiSys. 2020: 14-27.

[33] Zheng K, Qian K, Woodford T, et al. Neuroradar: A neuromorphic radar sensor for low-power iot systems[C]//Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems. 2023: 223-236.

[34] Al-lQubaydhi N, Alenezi A, Alanazi T, et al. Deep learning for unmanned aerial vehicles detection: A review[J/OL]. Computer Science Review, 2024, 51: 100614. https://www.sciencedirect.com/science/article/pii/S1574013723000813. DOI: https://doi.org/10.1016/j.cosrev.2023.100614.

[35] Sun Y, Wang W, Mottola L, et al. Aim: Acoustic inertial measurement for indoor drone localization and tracking[C]//Proceedings of the 20th ACM SenSys. 2022: 476-488.

[36] Chi G, Yang Z, Xu J, et al. Wi-drone: Wi-fi-based 6-dof tracking for indoor drone flight control [C]//Proceedings of the ACM MobiSys. 2022: 56-68.

[37] Cao H, Xu J, Li D, et al. Edge assisted mobile semantic visual slam[J]. IEEE Transactions on Mobile Computing, 2022: 1828-1837.

[38] Chi G, Xu J, Zhang J, et al. Locate, tell, and guide: Enabling public cameras to navigate the public[J]. IEEE Transactions on Mobile Computing, 2021, 22(2): 1010-1024.

[39] Zhao C, Ruan C, Xu J, et al. Foes or friends: Embracing ground effect for edge detection on lightweight drones[C/OL]//ACM MobiCom '24: Proceedings of the 30th Annual International Conference on Mobile Computing and Networking. New York, NY, USA, 2024: 1377–1392. DOI: 10.1145/3636534.3690699.

[40] Brossard M, Barrau A, Bonnabel S. Ai-imu dead-reckoning[J]. IEEE Transactions on Intelligent Vehicles, 2020, 5(4): 585-595.

[41] Chen C, Lu X, Markham A, et al. Ionet: Learning to cure the curse of drift in inertial odometry [C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 32. 2018.

[42] Zhou X, Wang Z, Ye H, et al. Ego-planner: An esdf-free gradient-based local planner for quadrotors[J]. IEEE Robotics and Automation Letters, 2020, 6(2): 478-485.

[43] Qin T, Shen S. Online temporal calibration for monocular visual-inertial systems[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 3662-3669.

[44] Hu J, Ren K, Xu X, et al. Efficient visual-inertial navigation with point-plane map[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023: 10659-10665.

[45] Mur-Artal R, Tardós J D. Visual-inertial monocular slam with map reuse[J]. IEEE Robotics and Automation Letters, 2017, 2(2): 796-803.

[46] Besl P J, McKay N D. Method for registration of 3-d shapes[C]//Sensor fusion IV: control paradigms and data structures: Vol. 1611. Spie, 1992: 586-606.

[47] Yang J, Li H, Jia Y. Go-icp: Solving 3d registration efficiently and globally optimally[C]// Proceedings of the IEEE International Conference on Computer Vision. 2013: 1457-1464.

[48] Biber P, Straßer W. The normal distributions transform: A new approach to laser scan matching [C]//Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453): Vol. 3. IEEE, 2003: 2743-2748.

[49] Alshawa M. lcl: Iterative closest line a novel point cloud registration algorithm based on linear features[J]. Ekscentar, 2007(10): 53-59.

[50] Zhou Q Y, Park J, Koltun V. Fast global registration[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. Springer, 2016: 766-782.

[51] Wang Y, Solomon J M. Deep closest point: Learning representations for point cloud registration [C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3523-3532.

[52] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1912-1920.

[53] Li D, Xu J, Yang Z, et al. Motion inspires notion: self-supervised visual-lidar fusion for environment depth estimation[C]//Proceedings of the 20th ACM MobiSys. 2022: 114-127.

[54] Li C, Huang Q, Zhou Y, et al. Riscan: Ris-aided multi-user indoor localization using cots wi-fi [C]//Proceedings of the 21st ACM SenSys. 2023: 445-458.

[55] Wang W, He Y, Jin M, et al. Meta-speaker: Acoustic source projection by exploiting air nonlinearity[C]//Proceedings of the 29th ACM MobiCom. 2023: 1-15.

[56] Xue H, Cao Q, Miao C, et al. Towards generalized mmwave-based human pose estimation through signal augmentation[C]//Proceedings of the 29th ACM MobiCom. 2023: 1-15.

[57] He Y, Bian C, Xia J, et al. Vi-map: Infrastructure-assisted real-time hd mapping for autonomous driving[C]//Proceedings of the 29th ACM MobiCom. 2023: 1-15.

[58] Shi S, Ling N, Jiang Z, et al. Soar: Design and deployment of a smart roadside infrastructure system for autonomous driving[C]//Proceedings of the 30th ACM MobiCom. 2024: 139-154.

[59] Safa A, Verbelen T, Ocket I, et al. Fusing event-based camera and radar for slam using spiking neural networks with continual stdp learning[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023: 2782-2788.

[60] Shuai X, Shen Y, Tang Y, et al. millieye: A lightweight mmwave radar and camera fusion system for robust object detection[C]//Proceedings of the International Conference on Internet-of-Things Design and Implementation. 2021: 145-157.

[61] Ding R, Jin H, Ding J, et al. Push the limit of single-chip mmwave radar-based egomotion estimation with moving objects in fov[C]//Proceedings of the 21st ACM SenSys. 2023: 417-430.

[62] Li W, Liu R, Wang S, et al. Egocentric human pose estimation using head-mounted mmwave radar[C]//Proceedings of the 21st ACM SenSys. 2023: 431-444.

[63] Müller L, Sifalakis M, Eissa S, et al. Aircraft marshaling signals dataset of fmcw radar and event-based camera for sensor fusion[C]//2023 IEEE Radar Conference (RadarConf23). IEEE, 2023: 01-06.

[64] Vidal A R, Rebecq H, Horstschaefer T, et al. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios[J]. IEEE Robotics and Automation Letters, 2018, 3(2): 994-1001.

[65] Wang H, Guo R, Ma P, et al. Event camera meets mobile embodied perception: Abstraction, algorithm, acceleration, application[J]. ACM Computing Surveys, 2025: 1-41.

[66] Zhou Y, Gallego G, Shen S. Event-based stereo visual odometry[J]. IEEE Transactions on Robotics, 2021, 37(5).

[67] Chen X, Xu S, Fu H, et al. Asc: Actuation system for city-wide crowdsensing with ride-sharing vehicular platform[C]//Proceedings of the Fourth Workshop on International Science of Smart City Operations and Platforms Engineering. 2019: 19-24.

[68] Chen X, Wang H, Li Z, et al. Deliversense: Efficient delivery drone scheduling for crowdsensing with deep reinforcement learning[C]//Proceedings of the ACM Ubicomp. 2022: 403-408.

[69] Wang Y, Su Z, Xu Q, et al. Lifesaving with rescuechain: Energy-efficient and partition-tolerant blockchain based secure information sharing for uav-aided disaster rescue[C]//Processings of the IEEE INFOCOM. 2021: 1-10.

[70] Li Z, Man F, Chen X, et al. Quest: Quality-informed multi-agent dispatching system for optimal mobile crowdsensing[C]//IEEE INFOCOM 2024-IEEE Conference on Computer Communications. IEEE, 2024: 1811-1820.

[71] Chen X, Wang H, Cheng Y, et al. Ddl: Empowering delivery drones with large-scale urban sensing capability[J]. IEEE Journal of Selected Topics in Signal Processing, 2024: 502-515.

[72] Wang L, Wang W, Dai H, et al. Joint deployment of truck-drone systems for camera-based object monitoring[J]. IEEE Transactions on Mobile Computing, 2024(01): 1-18.

[73] Zhao C, Wang H, Li J, et al. Smoothlander: A quadrotor landing control system with smooth trajectory guarantee based on reinforcement learning[C]//Proceedings of the 2023 ACM UbiComp. 2023: 682-687.

[74] Chen X, Purohit A, Dominguez C R, et al. Drunkwalk: Collaborative and adaptive planning for navigation of micro-aerial sensor swarms[C]//Proceedings of the 13th ACM Sensys. 2015: 295-308.

[75] Shan F, Zeng J, Li Z, et al. Ultra-wideband swarm ranging[C]//Processings of the IEEE INFOCOM. 2021: 1-10.

[76] Albanese A, Sciancalepore V, Costa-Pérez X. Sardo: An automated search-and-rescue drone-based solution for victims localization[J]. IEEE Transactions on Mobile Computing, 2021, 21 (9): 3312-3325.

[77] Xu S, Chen X, Pi X, et al. ilocus: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing[J]. IEEE Transactions on Mobile Computing, 2019, 19(8): 1831-1847.

[78] Chen X, Xu S, Han J, et al. Pas: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing[J]. IEEE Internet of Things Journal, 2020, 7(5): 3719-3734.

[79] McGuire K, De Wagter C, Tuyls K, et al. Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment[J]. Science Robotics, 2019, 4(35): eaaw9710.

[80] Liu Y, Wang H, Man F, et al. Mobiair: Unleashing sensor mobility for city-scale and fine-grained air-quality monitoring with airbert[C]//Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services. 2024: 223-236.

[81] Ren J, Xu Y, Li Z, et al. Scheduling uav swarm with attention-based graph reinforcement learning for ground-to-air heterogeneous data communication[C]//Proceedings of the ACM Ubicomp. 2023: 670-675.

[82] Kumar P, Singh P, Darshi S, et al. Analysis of drone assisted network coded cooperation for next generation wireless network[J]. IEEE Transactions on Mobile Computing, 2019, 20(1): 93-103.

[83] Arribas E, Mancuso V, Cholvi V. Coverage optimization with a dynamic network of drone relays [J]. IEEE Transactions on Mobile Computing, 2019, 19(10): 2278-2298.

[84] Li J, Kang H, Sun G, et al. Physical layer secure communications based on collaborative beamforming for uav networks: A multi-objective optimization approach[C]//Processings of the IEEE INFOCOM. 2021: 1-10.

[85] Xu Y, Jian Z, Zha J, et al. Emergency networking using uavs: A reinforcement learning approach with large language model[C]//In Processings of 2024 23rd ACM/IEEE IPSN. IEEE, 2024: 281-282.

[86] Alsamhi S H, Almalki F A, Ma O, et al. Predictive estimation of optimal signal strength from drones over iot frameworks in smart cities[J]. IEEE Transactions on Mobile Computing, 2021, 22(1): 402-416.

[87] Da Van L, Chang C H, Tong K L, et al. Tagging iot data in a drone view[C]//Proceedings of the ACM MobiCom. 2019: 1-3.

[88] Zhou L, Leng S, Liu Q, et al. Intelligent uav swarm cooperation for multiple targets tracking[J]. IEEE Internet of Things Journal, 2021, 9(1): 743-754.

[89] Arafat M Y, Moh S. Localization and clustering based on swarm intelligence in uav networks for emergency communications[J/OL]. IEEE Internet of Things Journal, 2019, 6(5): 8958-8976. DOI: 10.1109/JIOT.2019.2925567.

[90] Chen X, Xiao Z, Cheng Y, et al. Soscheduler: Toward proactive and adaptive wildfire suppression via multi-uav collaborative scheduling[J]. IEEE Internet of Things Journal, 2024: 24858-24871.

[91] Jian Z, Liu Z, Shao H, et al. Path generation for wheeled robots autonomous navigation on vegetated terrain[J]. IEEE Robotics and Automation Letters, 2023: 1764-1771.

[92] Tang W. Hong kong can succeed in developing a thriving low-altitude economy[EB/OL]. 2026. https://www.chinadailyhk.com/hk/article/593176.

[93] Zhang B B, Zhang D, Song R, et al. Rf-search: Searching unconscious victim in smoke scenes with rf-enabled drone[C]//Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. 2023: 1-15.

[94] González-Trejo J, Mercado-Ravell D, Becerra I, et al. On the visual-based safe landing of uavs in populated areas: a crucial aspect for urban deployment[J]. IEEE Robotics and Automation Letters, 2021, 6(4): 7901-7908.

[95] Li Y, Mund M, Hoess P, et al. Real-time 3d single-molecule localization using experimental point spread functions[J]. Nature methods, 2018, 15(5): 367-369.

[96] Xia S, Zhao M, Adhivarahan C, et al. Anemoi: A low-cost sensorless indoor drone system for automatic mapping of 3d airflow fields[C]//Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. 2023: 1-16.

[97] Zhang H, Wang G, Lei Z, et al. Eye in the sky: Drone-based object tracking and 3d localization [C]//Proceedings of the 27th ACM international conference on multimedia. 2019: 899-907.

[98] Zhang J, Yang H, Ren J, et al. Mobidepth: Real-time depth estimation using on-device dual cameras[C]//Proceedings of the 28th ACM MobiCom. 2022: 528-541.

[99] Xie Z, Ouyang X, Pan L, et al. Mozart: A mobile tof system for sensing in the dark through phase manipulation[C]//Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services. 2023: 163-176.

[100] Zhang Y, Scargill T, Vaishnav A, et al. Indepth: Real-time depth inpainting for mobile augmented reality[J/OL]. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 2022, 6(1): 528-541. https://doi.org/10.1145/3517260.

[101] Deng K, Zhao D, Han Q, et al. Geryon: Edge assisted real-time and robust object detection on drones via mmwave radar and camera fusion[J]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2022, 6(3): 1-27.

[102] Zhang J, Na X, Xi R, et al. mmhawkeye: Passive uav detection with a cots mmwave radar[C]// 2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2023: 267-275.

[103] Lu C X, Saputra M R U, Zhao P, et al. milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion[C]//Proceedings of the 18th Sensys. 2020: 109-122.

[104] Cheng Z, West R, Einstein C. End-to-end analysis and design of a drone flight controller[J/OL]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37 (11): 2404-2415. DOI: 10.1109/TCAD.2018.2857399.

[105] Gallego G, Delbrück T, Orchard G, et al. Event-based vision: A survey[J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 44(1): 154-180.

[106] He B, Wang Z, Zhou Y, et al. Microsaccade-inspired event camera for robotics[J]. Science Robotics, 2024, 9(90): eadj8124.

[107] Luo X, Wang H, Ruan C, et al. Eventtracker: 3d localization and tracking of high-speed object with event and depth fusion[C]//Proceedings of the 30th Annual International Conference on Mobile Computing and Networking. 2024: 1974-1979.

[108] Cao Z, Mei G, Guo X, et al. Virteach: mmwave radar point cloud based pose estimation with virtual data as a teacher[J]. IEEE Internet of Things Journal, 2024: 17615-17628.

[109] Liu H, Liu X, Xie X, et al. Pmtrack: Enabling personalized mmwave-based human tracking[J]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2024, 7(4): 1-30.

[110] Wang Z, Ng Y, Scheerlinck C, et al. An asynchronous kalman filter for hybrid event cameras[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 448-457.

[111] Alzugaray I, Chli M. Asynchronous corner detection and tracking for event cameras in real time [J]. IEEE Robotics and Automation Letters, 2018, 3(4): 3177-3184.

[112] Falanga D, Kleber K, Scaramuzza D. Dynamic obstacle avoidance for quadrotors with event cameras[J]. Science Robotics, 2020, 5(40): eaaz9712.

[113] Mitrokhin A, Fermüller C, Parameshwara C, et al. Event-based moving object detection and tracking[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 1-9.

[114] Wang Y, Wang Z, Zhang J A, et al. Vital sign monitoring in dynamic environment via mmwave radar and camera fusion[J]. IEEE Transactions on Mobile Computing, 2023, 23(5): 4163-4180.

[115] Min Z, Wang J, Meng M Q H. Joint rigid registration of multiple generalized point sets with anisotropic positional uncertainties in image-guided surgery[J]. IEEE Transactions on Automation Science and Engineering, 2021, 19(4): 3612-3627.

[116] Zhang G, Chi G, Zhang Y, et al. Push the limit of millimeter-wave radar localization[J]. ACM Transactions on Sensor Networks, 2023, 19(3): 1-21.

[117] Lin S, Xu F, Wang X, et al. Efficient spatial-temporal normalization of sae representation for event camera[J]. IEEE Robotics and Automation Letters, 2020, 5(3): 4265-4272.

[118] Campos C, Elvira R, Rodríguez J J G, et al. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[119] DJI. Dji d-rtk2[Z]. 2026.

[120] Vrba M, Walter V, Pritzl V, et al. On onboard lidar-based flying object detection[J]. IEEE Transactions on Robotics, 2024, 41: 593-611.

[121] Thrun S, Burgard W, Fox D. Probabilistic robotics (intelligent robotics and autonomous agents) [M]. The MIT Press, 2005.

[122] Wang Y, Shen Y, Xu K, et al. Towards high-speed passive visible light communication with event cameras and digital micro-mirrors[C]//Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems. 2024: 704-717.

[123] Springer J, Kyas M. Autonomous drone landing with fiducial markers and a gimbal-mounted camera for active tracking[C]//2022 Sixth IEEE international conference on robotic computing (IRC). IEEE, 2022: 243-247.

[124] Springer J, Kyas M. Evaluation of orientation ambiguity and detection rate in april tag and whycode[C]//2022 Sixth IEEE International Conference on Robotic Computing (IRC). IEEE, 2022: 281-286.

[125] Prophesee. Prophesee genx320[Z]. 2026.

[126] Wang H, Xu J, Luo X, et al. Ultra-high-frequency harmony: mmwave radar and event camera orchestrate accurate drone landing[C]//Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems. 2025: 15-29.

[127] Fact.MR. Global drones market outlook (2022-2032)[EB/OL]. https://www.factmr.com/report/62/drone-market.

[128] Sharma S, Simha A, Prasad V, et al. Beavis: Balloon enabled aerial vehicle for iot and sensing[C]//Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. 2023: 1-15.

[129] Xu W, Cai Y, He D, et al. Fast-lio2: Fast direct lidar-inertial odometry[J]. IEEE Transactions on Robotics, 2022, 38(4): 2053-2073.

[130] Seth A, James A, Kuantama E, et al. Aerobridge: Autonomous drone handoff system for emergency battery service[C]//Proceedings of the 30th Annual International Conference on Mobile Computing and Networking. 2024: 573-587.

[131] Gowda M, Manweiler J, Dhekne A, et al. Tracking drone orientation with multiple gps receivers [C]//Proceedings of the 22nd ACM MobiCom. 2016: 280-293.

[132] Pang W, Xia C, Leong B, et al. Ubipose: Towards ubiquitous outdoor ar pose tracking using aerial meshes[C]//Proceedings of the 29th ACM MobiCom. 2023: 1-16.

[133] Reinhardt T. Using global localization to improve navigation[J]. Google AI Blog, 2019.

[134] Shin C S, Pang W, Li C, et al. Recap: 3d traffic reconstruction[C]//30th Annual International Conference on Mobile Computing and Network (ACM MobiCom 2024). ACM New York, NY, USA, 2024.

[135] Khan K N, Khalid A, Turkar Y, et al. Vrf: Vehicle road-side point cloud fusion[C]//Proceedings of the 22nd ACM MobiSys. 2024: 547-560.

[136] Zhou L, Huang G, Mao Y, et al. Edplvo: Efficient direct point-line visual odometry[C/OL]// 2022 International Conference on Robotics and Automation (ICRA). 2022: 7559-7565. DOI: 10.1109/ICRA46639.2022.9812133.

[137] Argaw D M, Kim J, Rameau F, et al. Motion-blurred video interpolation and extrapolation[C]// Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 35. 2021: 901-910.

[138] Wang H, Guo R, Ma P, et al. Event camera meets mobile embodied perception: abstraction, algorithm, acceleration, application[J]. ACM Computing Surveys, 2026, 58(8): 1-41.

[139] Alzugaray I. Event-driven feature detection and tracking for visual slam[D]. ETH Zurich, 2022.

[140] Lee W, Alkouz B, Shahzaad B, et al. Package delivery using autonomous drones in skyways [C]//Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers. 2021: 48-50.

[141] Li D, Zhao Y, Xu J, et al. Reshaping edge-assisted visual slam by embracing on-chip intelligence [J/OL]. IEEE Transactions on Mobile Computing, 2024: 12983-12997. DOI: 10.1109/TMC.2024.3424452.

[142] Hu J, Lang X, Zhang F, et al. Square-root inverse filter-based gnss-visual-inertial navigation[C/OL]//2024 IEEE International Conference on Robotics and Automation (ICRA). 2024: 1646-1652. DOI: 10.1109/ICRA57147.2024.10611666.

[143] Ahmad F, Shin C S, Pang W, et al. Cooperative infrastructure perception[C]//2024 IEEE/ACM Ninth International Conference on Internet-of-Things Design and Implementation (IoTDI). IEEE, 2024: 61-72.

[144] Hu J, Hu J, Shen Y, et al. 1d-lrf aided visual-inertial odometry for high-altitude mav flight [C/OL]//2022 International Conference on Robotics and Automation (ICRA). 2022: 5858-5864. DOI: 10.1109/ICRA46639.2022.9811757.

[145] Cui J, He Y, Niu J, et al. αlidar: An adaptive high-resolution panoramic lidar system[C]// Proceedings of the 30th ACM MobiCom. 2024: 1515-1529.

[146] He Y, Ma L, Jiang Z, et al. Vi-eye: semantic-based 3d point cloud registration for infrastructure-assisted autonomous driving[C]//Proceedings of the 27th ACM MobiCom. 2021: 573-586.

[147] Wu Y, Xu J, Li D, et al. Flytracker: Motion tracking and obstacle detection for drones using event cameras[C]//Proceedings of the IEEE INFOCOM 2023. IEEE, 2023: 1-10.

[148] Cao H, Xu J, Li D, et al. Eventboost: Event-based acceleration platform for real-time drone localization and tracking[C]//Proceedings of the IEEE INFOCOM 2024. 2024: 1851-1859.

[149] Xu J, Li D, Yang Z, et al. Taming event cameras with bio-inspired architecture and algorithm: A case for drone obstacle avoidance[C]//Proceedings of the 29th ACM MobiCom. 2023: 1-16.

[150] Agarwal S, Mierle K, Team T C S. Ceres Solver[CP/OL]. 2023. https://github.com/ceres-solver/ceres-solver.

[151] Beauchemin S S, Barron J L. The computation of optical flow[J]. ACM computing surveys (CSUR), 1995, 27(3): 433-466.

[152] 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.

[153] Thrun S. Probabilistic robotics[J]. Communications of the ACM, 2002, 45(3): 52-57.

[154] Open Robotics. Ros melodic.[Z]. 2018.

[155] CHINGMU. Chingmu motion capture system[Z]. 2026.

[156] Hi-Target. Hi-target d8 rtk system[Z]. 2026.

[157] Klenk S, Chui J, Demmel N, et al. Tum-vie: The tum stereo visual-inertial event dataset[C]// 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021: 8601-8608.

[158] Gao L, Liang Y, Yang J, et al. Vector: A versatile event-centric benchmark for multi-sensor slam[J]. IEEE Robotics and Automation Letters, 2022, 7(3): 8217-8224.

[159] Guan W, Chen P, Xie Y, et al. Pl-evio: Robust monocular event-based visual inertial odometry with point and line features[J]. IEEE Transactions on Automation Science and Engineering, 2023, 21(4): 6277-6293.

[160] Zhou Y, Li H, Kneip L. Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment[J]. IEEE Transactions on Robotics, 2018, 35(1): 184-199.

[161] Rashid M T, Zhang D Y, Wang D. Socialdrone: An integrated social media and drone sensing system for reliable disaster response[C]//Proceedings of the IEEE INFOCOM. 2020: 218-227.

[162] Duisterhof B P, Li S, Burgués J, et al. Sniffy bug: A fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021: 9099-9106.

[163] Francis A, Li S, Griffiths C, et al. Gas source localization and mapping with mobile robots: A review[J]. Journal of Field Robotics, 2022, 39(8): 1341-1373.

[164] Chen X x, Huang J. Odor source localization algorithms on mobile robots: A review and future outlook[J]. Robotics and Autonomous Systems, 2019, 112: 123-136.

[165] Khochare A, Simmhan Y, Sorbelli F B, et al. Heuristic algorithms for co-scheduling of edge analytics and routes for uav fleet missions[C]//Processings of IEEE INFOCOM. 2021: 1-10.

[166] Bertizzolo L, D'oro S, Ferranti L, et al. Swarmcontrol: An automated distributed control framework for self-optimizing drone networks[C]//Proceedings of the IEEE INFOCOM. 2020: 1768-1777.

[167] 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.

[168] Schmuck P, Chli M. Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams[J]. Journal of Field Robotics, 2019, 36 (4): 763-781.

[169] Shen Y, Qu Y, Dong C, et al. Joint training and resource allocation optimization for federated learning in uav swarm[J]. IEEE Internet of Things Journal, 2022, 10(3): 2272-2284.

[170] Wang H, Chen X, Cheng Y, et al. H-swarmloc: Efficient scheduling for localization of heterogeneous mav swarm with deep reinforcement learning[C]//Proceedings of the 20th ACM Sensys. 2022: 1148-1154.

[171] Li T, Leng S, Wang Z, et al. Intelligent resource allocation schemes for uav-swarm-based cooperative sensing[J]. IEEE Internet of Things Journal, 2022, 9(21): 21570-21582.

[172] Rashid M T, Zhang D, Wang D. A computational model-driven hybrid social media and drone-based wildfire monitoring framework[C]//Proceedings of the IEEE INFOCOM WKSHPS. 2020: 1362-1363.

[173] Yang B, Wu H H, Cao X, et al. Intelli-eye: An uav tracking system with optimized machine learning tasks offloading[C]//Proceedings of the IEEE INFOCOM WKSHPS. 2019: 1-6.

[174] Trotta A, Andreagiovanni F D, Di Felice M, et al. When uavs ride a bus: Towards energy-efficient city-scale video surveillance[C]//Processings of the IEEE INFOCOM. 2018: 1043-1051.

[175] Sharp J, Wu C, Zeng Q. Authentication for drone delivery through a novel way of using face biometrics[C]//Proceedings of the 28th ACM MobiCom. 2022: 609-622.

[176] Li D, Xu J, Yang Z, et al. Train once, locate anytime for anyone: Adversarial learning-based wireless localization[J]. ACM Transactions on Sensor Networks, 2024, 20(2): 1-21.

[177] Fu Y, Zhang T, Wang L, et al. Coded hyperspectral image reconstruction using deep external and internal learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44 (7): 3404-3420.

[178] Li M, Fu Y, Zhang T, et al. Latent diffusion enhanced rectangle transformer for hyperspectral image restoration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 47(1): 549-564.

[179] Hu J, Jiang H, Liu D, et al. Combining imu with acoustics for head motion tracking leveraging wireless earphone[J]. IEEE Transactions on Mobile Computing, 2023, 23(6): 6835-6847.

[180] Li D, Xu J, Yang Z, et al. Leovr: Motion-inspired visual-lidar fusion for environment depth estimation[J]. IEEE Transactions on Mobile Computing, 2023, 23(6): 7499-7516.

[181] Dong E, Xu J, Wu C, et al. Pair-navi: Peer-to-peer indoor navigation with mobile visual slam[C]//IEEE INFOCOM 2019-IEEE conference on computer communications. IEEE, 2019: 1189-1197.

[182] Fu Y, Wang Z, Zhang T, et al. Low-light raw video denoising with a high-quality realistic motion dataset[J]. IEEE Transactions on Multimedia, 2022, 25: 8119-8131.

[183] Zhang T, Fu Y, Zhang J, et al. Deep guided attention network for joint denoising and demosaicing in real image[J]. Chinese Journal of Electronics, 2024, 33(1): 303-312.

[184] Wei K, Fu Y, Zheng Y, et al. Physics-based noise modeling for extreme low-light photography [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(11): 8520-8537.

[185] Wang J, Olson E. Apriltag 2: Efficient and robust fiducial detection[C]//Processings of the IEEE IROS. 2016: 4193-4198.

[186] Xu J, Chi G, Yang Z, et al. Followupar: Enabling follow-up effects in mobile ar applications [C]//Proceedings of the ACM MobiSys. 2021: 1-13.

[187] Purohit A, Sun Z, Zhang P. Sugarmap: Location-less coverage for micro-aerial sensing swarms [C]//Proceedings of the ACM Sensys. 2013: 253-264.

[188] Chen X, Ruiz C, Zeng S, et al. H-drunkwalk: Collaborative and adaptive navigation for heterogeneous mav swarm[J]. ACM Transactions on Sensor Networks, 2020, 16(2): 1-27.

[189] Ucinski D. Optimal measurement methods for distributed parameter system identification[M]. CRC press, 2004.

[190] Aurenhammer F. Voronoi diagrams—a survey of a fundamental geometric data structure[J]. ACM CSUR, 1991, 23(3): 345-405.

[191] Vitus M P, Zhang W, Abate A, et al. On efficient sensor scheduling for linear dynamical systems [J]. Automatica, 2012, 48(10): 2482-2493.

[192] Atanasov N, Le Ny J, Daniilidis K, et al. Information acquisition with sensing robots: Algorithms and error bounds[C]//Processings of the IEEE ICRA. 2014: 6447-6454.

[193] Borenstein J, Everett H, Feng L. Navigating mobile robots: Systems and techniques[M]. AK Peters, Ltd., 1996.

[194] Xu J, Zhong F, Wang Y. Learning multi-agent coordination for enhancing target coverage in directional sensor networks[J]. Processings of the NeurIPS, 2020, 33: 10053-10064.

[195] Kalaitzakis M, Cain B, Carroll S, et al. Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers[J]. Journal of Intelligent & Robotic Systems, 2021, 101: 1-26.

[196] Singhal C, De S. Resource allocation in next-generation broadband wireless access networks [M]. IGI Global, 2017.

[197] Arafat M Y, Moh S. Localization and clustering based on swarm intelligence in uav networks for emergency communications[J]. IEEE Internet of Things Journal, 2019, 6(5): 8958-8976.

[198] Zhou B, Xu H, Shen S. Racer: Rapid collaborative exploration with a decentralized multi-uav system[J]. IEEE Transactions on Robotics, 2023, 39(3): 1816-1835.

[199] Wang H, Wang S, Yang Y, et al. Gcrl: Efficient delivery area assignment for last-mile logistics with group-based cooperative reinforcement learning[C]//2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE, 2023: 3522-3534.

[200] Jeong H, Schlotfeldt B, Hassani H, et al. Learning q-network for active information acquisition [C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 6822-6827.

[201] Zhou X, Wen X, Wang Z, et al. Swarm of micro flying robots in the wild[J]. Science Robotics, 2022, 7(66): eabm5954.

[202] Ashraf U. Capacity augmentation in wireless mesh networks[J]. IEEE Transactions on Mobile Computing, 2014, 14(7): 1344-1354.

[203] Soorki M N, Aghajari H, Ahmadinabi S, et al. Catch me if you can: Deep meta-rl for search-and-rescue using lora uav networks[J]. IEEE Transactions on Mobile Computing, 2024, 24(2): 763-778.

[204] Bertizzolo L, Tran T X, Buczek J, et al. Streaming from the air: Enabling drone-sourced video streaming applications on 5g open-ran architectures[J]. IEEE Transactions on Mobile Computing, 2021, 22(5): 3004-3016.

APPENDIX A SUPPLEMENTAL MATERIAL OF MME-LOC

A.1 Periodic Micro-motions in Event Camera

From a frequency-domain perspective, the rotation of drone propellers introduces periodic motion signals with frequencies typically in the range of hundreds of Hertz. These high-frequency components manifest as sharp peaks in the temporal spectrum of the observed signal. Event cameras, with their μs-level temporal resolution and asynchronous data acquisition, are capable of faithfully capturing these rapid intensity changes without aliasing or motion blur. In contrast, conventional frame-based cameras, limited by frame rate, tend to smooth out or completely miss such high-frequency patterns.

Importantly, most background disturbances in real-world landing scenarios, such as illumination variations, pedestrian movement, or wind-blown vegetation, are low-frequency in nature, often below tens of Hertz. By applying temporal frequency analysis, the system can isolate and amplify propeller-induced high-frequency components while suppressing low-frequency environmental noise. As a result, these high-frequency signatures serve as reliable and distinctive features for detecting and tracking drones, even under challenging lighting and cluttered background conditions.

A.2 Problem Formulation of Factor Graph-based Optimization

A factor graph consists of two types of nodes: variable nodes, representing the states to be estimated and optimized (e.g., t^i_{ED}), and factor nodes, encoding the probabilistic constraints imposed by measurements or prior knowledge. In *mmE-Loc*, these measurements originate from two complementary modalities: (i) the Event Tracking (ET) model, providing image-plane positions x^i; (ii) the Radar Tracking (RT) model, providing distance D^i, direction vector $\vec{v}^i$, and coarse 3D translation U^i_E. Each measurement contributes a factor that captures the likelihood of observing the measurement given the current state estimate. The objective is to estimate the optimal configuration of variable nodes $\mathcal{X} = t^i_{ED} \mid i \in \mathcal{T}$ conditioned on the set of measurements $\mathcal{Z} = x^i, D^i, \vec{v}^i, U^i_E \mid i \in \mathcal{T}$. *GAJO* formulates this as a maximum a posteriori (MAP) estimation problem, where the posterior distribution is

represented by the product of all connected factors:

$$\begin{aligned}\hat{\mathcal{X}} &= \arg\max_{\mathcal{X}} p(\mathcal{X} \mid \mathcal{Z}) \\ &= \arg\max_{\mathcal{X}} p(\mathcal{X})\, p(\mathcal{Z} \mid \mathcal{X}) \\ &= \arg\max_{\mathcal{X}} p(\mathcal{X}) \prod_{i \in \mathcal{T}} p\left(x^i, D^i, \vec{v}^i, U_E^i \mid t_{ED}^i\right).\end{aligned} \tag{A.1}$$

This formulation enables efficient inference through graph optimization techniques such as incremental smoothing and variable relinearization, ensuring low-latency and accurate drone localization.

A.3 Mathematical Derivations of Inter-SAE Tracking

Once the measurements from the ET and RT models, i.e., $(x^i, D^i, \vec{v}^i, U_E^i)$, are obtained, the corresponding energy terms of the prior, ET, and RT factors are constructed to encode their probabilistic constraints. Specifically, the prior factor models the temporal evolution of the drone's location and penalizes deviations from the predicted position based on flight dynamics. The ET factor reflects the reprojection error between the observed event center and the expected projection derived from the estimated state. The RT factor captures discrepancies between the predicted and measured distance, angle, and motion derived from the radar. These factors are formulated as:

$$\begin{aligned}E_{\text{Prior}}^i &= -\log p\left(t_{ED}^i\right) \propto \left\|t_{ED}^i - \bar{t}_{ED}^i\right\|_{\sigma_{t_{ED}}}^2, \\ E_{\text{ET}}^i &= -\log p\left(x^i \mid t_{ED}^i\right) \propto \rho(\left\|x^i - \pi(\mathbf{X}_E^i)\right\|_{\Sigma_E}^2), \\ E_{\text{RT}}^i &= -\log p\left(D^i, \vec{v}^i, U_E^i \mid t_{ED}^i\right) \\ &\propto \left\|||t_{ED}^i|| - D^i\right\|_{\sigma_D}^2 + \left\|\vec{v}_{t_{ED}^i} - \vec{v}^i\right\|_{\sigma_{\vec{v}}}^2 + \\ &\quad \left\|(t_{ED}^i - t_{ED}^{i-1}) - U_E^i\right\|_{\sigma_{U_E}}^2,\end{aligned} \tag{A.2}$$

where $|e|_{\Sigma_E}^2 = e^{\mathsf{T}}\Sigma_E^{-1}e$, and Σ_E denotes the covariance matrix associated with the event camera measurements.

Building on these factor formulations in Eqn.(A.2), inter-SAE tracking is carried out to infer the instantaneous drone location by maximizing the joint posterior probability.

This can be equivalently transformed into an energy minimization problem:

$$\begin{aligned}\hat{t}^i_{ED} &= \arg\max_{t^i_{ED}} p(t^i_{ED} \mid t^{i-1}_{ED}, t^{i-2}_{ED}) p(x^i \mid t^i_{ED})\ p(D^i, \vec{v}^i, U^i_E \mid t^i_{ED}) \\ &= \arg\min_{t^i_{ED}} -\log\left(p(t^i_{ED} \mid t^{i-1}_{ED}, t^{i-2}_{ED}) p(x^i \mid t^i_{ED}) p(D^i, \vec{v}^i, U^i_E \mid t^i_{ED})\right) \\ &= \arg\min_{t^i_{ED}} \left(E^i_{\text{prior}} + E^i_{\text{ET}} + E^i_{\text{RT}}\right).\end{aligned} \tag{A.3}$$

This formulation allows the system to jointly optimize state by integrating temporal priors, event-based observations, and radar measurements, enabling fast drone localization.

A.4 Mathematical Derivations of Local Location Optimization

To mitigate the accumulation of localization drift that inevitably arises from sequential estimation, we perform a local location optimization step, which refines the initially estimated positions by exploiting the temporal consistency of multiple consecutive SAEs. Rather than treating each SAE independently, this process leverages their joint spatial-temporal correlations to achieve a more stable and accurate trajectory.

Specifically, the optimization is performed over a local window of W consecutive SAEs, and the associated location set is denoted as $\mathcal{X} = \bigcup_{i \in \mathcal{T}} \left\{t^i_{ED}\right\}$, where $\mathcal{T}$ represents the set of indices of the SAEs within the optimization window. These variables are organized in a factor graph structure, where each node corresponds to a location state and the edges encode measurement or prior constraints.

Based on the probabilistic formulation in Eq. (A.1), the joint optimization problem can be written as

$$\begin{aligned}\hat{\mathcal{X}} &= \arg\max_{\mathcal{X}} p(\mathcal{X}) \prod_{i \in \mathcal{T}} p\left(x^i \mid t^i_{ED}\right) p\left(D^i, \vec{v}^i, U^i_E \mid t^i_{ED}\right), \\ &= \arg\min_{\mathcal{X}} \sum_{i \in \mathcal{T}} \left(E^{\text{prior}}_i + E^{\text{ET}}_i + E^{\text{RT}}_i\right).\end{aligned} \tag{A.4}$$

Here, E^{prior}_i encodes the prior information or motion model constraints to ensure temporal smoothness, E^{ET}_i corresponds to the event-to-target measurement residual derived from SAE alignment, and E^{RT}_i incorporates radar point cloud-based residual terms. By minimizing this cost function, the system effectively reduces accumulated errors and refines the location estimates, providing a drift-resilient local trajectory that enhances the overall accuracy of the localization pipeline.

Algorithm A.1 Incremental optimization method.

Data: Original factor graph G; New measurements $D^i, \vec{v}^i, U_E^i$; square root information matrix R
Result: Updated locations $\hat{\mathcal{X}}$

```
G ← AddFactorToGraph(G^i, D^i, v⃗, U_E^i);
R ← IncrementalUpdate(G);
X̂ ← Backsubstitution(R);
L ← ∅;        ▷Set of nodes need to be linearized;
for all t_ED^i ∈ X and all t̂_ED^i ∈ X̂ do
    if t̂_ED^i − t_ED^i ≥ δ then
        L ← L ∪ t_ED^i;
    end
end
if |L| ≥ L_T or ||X̂ − X|| ≥ Δ then
    for all t_ED^i ∈ X do
        UpdateLinearizationPoint(t_ED^i);
    end
    R ← FullUpdate(G);
    X̂ ← Backsubstitution(R);
end
```

A.5 Incremental Optimization Method

Factor graph solving. We represent the estimation problems Eqn.(A.3) and Eqn.(A.4) using a factor graph model. To solve the nonlinear least-squares problems, we linearize the observation model and solve the least-squares formulation:

$$\hat{\mathcal{X}} = \arg\min_{\mathcal{X}} \|A\mathcal{X} - \mathbf{b}\|^2, \tag{A.5}$$

where $A \in \mathbb{R}^{m\times n}$ is the measurement Jacobian and $\mathbf{b} \in \mathbb{R}^m$ is the right-hand side vector. We then utilize QR matrix factorization as $A = Q[R, 0]^T$ and solve the least squares problem $R\hat{\mathcal{X}} = \mathbf{d}$ through backsubstitution to obtain optimized locations $\hat{\mathcal{X}}$, where $R \in \mathbb{R}^{n\times n}$ is the upper triangular square root information matrix, $Q \in \mathbb{R}^{m\times m}$ is an orthogonal matrix, and $\mathbf{d} \in \mathbb{R}^n$. Although re-linearizing and regenerating R with new measurements can help mitigate errors, applying this approach to problem (A.4) can be computationally expensive, as it requires frequent updates and increased processing overhead, limiting real-time performance.

Incremental Optimization Method. To tackle this, we propose an incremental optimization method, leveraging the insight that *new measurements mainly impact localized areas, leaving distant parts unchanged*. This allows us to incrementally update R during local optimization. By adaptively combining updated and regenerated R, this method

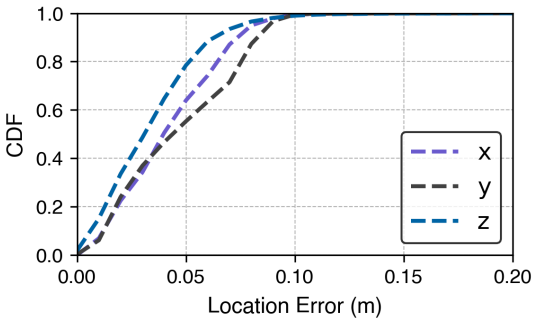


Figure A.1 Error along the x-, y-, and z-axes during standard flight.

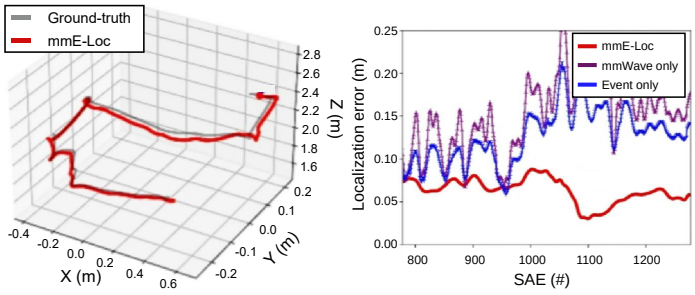
reduces latency and enhances accuracy.

Figure A.1 shows how the incremental optimization method solves the local location optimization problem. Lines 1-3 depict tracking with incrementally updated $\boldsymbol{R}$, while lines 4-16 show tracking with re-generated $\boldsymbol{R}$. Specifically, when receiving new measurements, function `AddFactorToGraph` updates the factor graph, and function `IncrementalUpdate` incrementally updates $\boldsymbol{R}$ with new measurements. We then solve local location tracking with this incrementally updated $\boldsymbol{R}$.

When one of two conditions is met, we solve local location tracking with re-generated $\boldsymbol{R}$: (i) we track locations that have changed significantly in a set $L = \{t^i_{ED} : \hat{t^i_{ED}} - t^i_{ED} \geq \delta\}$. If enough locations have undergone significant changes (i.e., $|L| \geq L_T$), we solve location tracking with re-generated $\boldsymbol{R}$ output by function `FullUpdate`. (ii) The norm of total location changes exceeds a threshold Δ (i.e., $||\hat{\mathcal{X}} - \mathcal{X}|| \geq \Delta$). Since the local location tracking involves repeatedly solving linear equations, this condition keeps the current solution from diverging too far from the optimal solution.

A.6 Flight Performance: Case Study

Error distribution along the x-, y-, and z-axes during a typical flight. To analyze the localization errors along different axes during drone flight, we examined the error distribution along the x-, y-, and z-axes in a representative experiment. Figure A.1 illustrates the variations of localization error across these three axes throughout a typical flight. As shown in the figure, the errors along the x- and y-axes are generally comparable, whereas the z-axis exhibits slightly larger deviations. This increase in vertical error arises because, in the tangential directions, the event camera effectively provides calibration cues for the mmWave radar; however, along the radial (depth) direction, the radar measurements remain less accurate without calibration cues from the event camera.



(a) Flight trajectory (b) Localization error

Figure A.2 Flight trajectory and localization error.

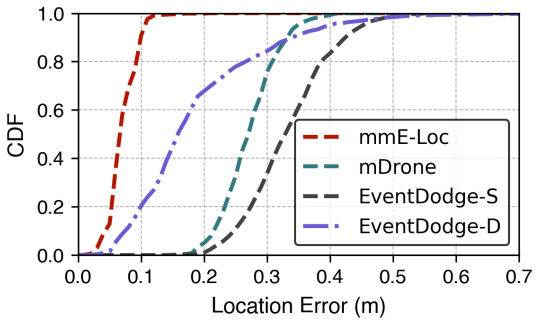


Figure A.3 Error in the near distance setting.

Flight trajectory of a typical flight. To further analyze the localization performance of mmE-Loc, we visualize the drone's trajectory during a typical flight. Figure A.2(a) shows the estimated flight trajectory compared with the ground-truth path. It can be observed that mmE-Loc's estimated trajectory closely aligns with the ground truth, demonstrating its ability to provide stable and accurate drone localization. Figure A.2(b) illustrates the localization error over time, showing that mmE-Loc maintains consistently low errors throughout the entire flight.

In contrast, the errors of the mmWave-only and Event-only methods exhibit strong fluctuations. For the mmWave-only case, the radar reflections are generated from different parts of the drone body at different times, causing the estimated positions to vary around the actual ground-truth location. For the Event-only method, the presence of the background safety net introduces spurious noise, leading to variations in the detected bounding box and, consequently, in the estimated depth, which results in unstable localization.

Since the two sensing modalities produce noise in complementary ways, mmE-Loc leverages their strengths effectively: in the tangential directions, the event camera provides strong calibration cues for the mmWave radar, while along the radial (depth) direction, the radar offers more reliable distance measurements to compensate for event-based depth uncertainty. Moreover, mmE-Loc integrates the drone's motion information to further

stabilize the fused localization output. As a result, the fusion error decreases steadily, even when the individual modality errors fluctuate.

Errors in a near-distance scenario. Since the primary goal of mmE-Loc is to provide accurate and low-latency localization when the drone approaches the landing pad, we further analyze its performance in the near-distance scenario and compare it against the baseline methods. Figure A.3 presents the localization error distribution in this close-range setting. As shown in the figure, mmE-Loc achieves consistently lower errors and outperforms all baselines across the entire error range. These results demonstrate that when the drone nears the landing pad, mmE-Loc can deliver more precise and stable localization estimates, effectively guiding drone to perform a smooth and reliable landing.

RESUME

Haoyang Wang was born on April 28, 2000, in Korla, Xinjiang Province, China.

In June 2022, he received the B.E. degree with honors from the School of Computer Science and Engineering, Central South University, China.

Since September 2022, he has been pursuing his Ph.D. degree at Tsinghua Shenzhen International Graduate School. He was a visiting scholar at the University of California, Berkeley in 2025. During this period, he has made academic publications as follows:

Publications

Note: ∗ indicates co-first authors and † indicates corresponding author.

First author papers

[1] **Wang H**, Xu J, Luo X, Zhang T, Chen X, Duan R, Liu Y, Zheng J, Hong W, Ji X, Tang Y, Chen X†. mmE-Loc: Facilitating Accurate Drone Landing with Ultra-high-frequency Localization [J]. IEEE Transactions on Mobile Computing (IEEE TMC), 2026: 1-18. (Early access, CCF-A, THU-A, IF: 9.2)

[2] **Wang H**, Guo R, Ma P, Ruan C, Luo X, Ding W, Zhong T, Xu J, Liu Y, Chen X†. Event Camera Meets Mobile Embodied Perception: Abstraction, Algorithm, Acceleration, Application[J]. ACM Computing Surveys (ACM CSUR), 2026, 58(8): 1-41. (JCR Q1, IF: 28)

[3] **Wang H**, Xu J, Zhao C, Cheng Y, Chen X, Hong C, Zhang XP, Liu Y, Chen X†. Aerial Shepherds: Enabling Hierarchical Localization in Heterogeneous MAV Swarms[J]. IEEE Transactions on Mobile Computing (IEEE TMC), 2025, 25(3): 3150-3167. (CCF-A, THU-A, IF: 9.2)

[4] **Wang H**, Xu J, Luo X, Chen X, Zhang T, Duan R, Liu Y, Chen X†. Ultra-High-Frequency Harmony: mmWave Radar and Event Camera Orchestrate Accurate Drone Landing[C]. Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems (ACM SenSys), ACM, 2025: 15-29. (THU-A, CCF-B)

[5] **Wang H**, Xu J, Zhao C, Lu Z, Cheng Y, Chen X, Zhang XP, Liu Y, Chen X†. TransformLoc: Transforming MAVs into Mobile Localization Infrastructures in Heterogeneous Swarms[C]. Proceedings of the IEEE International Conference on Computer

Communications (INFOCOM), IEEE, 2024: 1101-1110. (CCF-A, THU-A)

[6] **Wang H**, Liu Y, Zhao C, He J, Ding W, Chen X†. Califormer: Leveraging Unlabeled Measurements to Calibrate Sensors with Self-supervised Learning[C]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (ACM Ubicomp CPD workshop), ACM, 2023: 743-748. (Best Presentation Award)

[7] **Wang H**, Chen X, Cheng Y, Wu C, Dang F, Chen X†. H-SwarmLoc: Efficient Scheduling for Localization of Heterogeneous MAV Swarm with Deep Reinforcement Learning[C]. Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems (ACM SenSys CML-IoT workshop). 2022: 1148-1154.

Co-first author papers

[8] Chen X*, **Wang H***, Cheng Y, Fu H, Liu Y, Dang F, Liu Y, Cui J, Chen X†. DDL: Empowering Delivery Drones with Large-scale Urban Sensing Capability[J]. IEEE Journal of Selected Topics in Signal Processing (IEEE J-STSP), 2024, 18(3): 502-515. (Co-first author, JCR Q1, IF: 13.7)

[9] Liu Y*, **Wang H***, Man F, Xu J, Dang F, Liu Y, Zhang XP, Chen X†. MobiAir: Unleashing Sensor Mobility for City-scale and Fine-grained Air-Quality Monitoring with AirBERT[C]. Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services (ACM MobiSys), ACM, 2024: 223-236. (Co-first author, THU-A, CCF-B)

[10] Guo R*, Ruan C*, **Wang H***, Gong Z, Xu J, Chen X†. x^2-Fusion: Cross-Modality and Cross-Dimension Flow Estimation in Event Edge Space[C]. The IEEE/CVF Conference on Computer Vision and Pattern Recognition (IEEE/CVF CVPR), 2026. (Accepted, co-first author, CCF-A, THU-A)

[11] Zhou N*, Liu Y*, **Wang H***, Man F, Xu J, Dang F, Hong C, Liu Y, Zhang XP, Song Y, Wang Q, Chen X†. CatUA: Catalyzing Urban Air Quality Intelligence through Mobile Crowd-sensing[J]. IEEE Transactions on Mobile Computing (IEEE TMC), 2025, 24(9): 9184 - 9201. (Co-first author, CCF-A, THU-A, IF: 9.2)

[12] Zhao C*, **Wang H***, Li J, Man F, Mu S, Ding W, Zhang XP, Chen X†. SmoothLander: A Quadrotor Landing Control System with Smooth Trajectory Guarantee Based on Reinforcement Learning[C]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (ACM Ubicomp CPD workshop), ACM, 2023: 682-687. (Co-first author)

Co-author papers

[13] Zhao C*, Xu J*, Ruan C, **Wang H**, Wang S, Li J, Zha J, Hong W, Yang Z, Liu Y, Zhang XP, Chen X†. Flight Dynamics to Sensing Modalities: Exploiting Drone Ground Effect for Accurate Edge Detection[J]. IEEE Transactions on Mobile Computing (IEEE TMC). (Accepted, CCF-A, THU-A, IF: 9.2)

[14] Cheng Y*, Chen X*, Yang Y, **Wang H**, Xu J, Hong C, Zhang XP, Liu Y, Chen X†. SniffySquad: Patchiness-Aware Gas Source Localization with Multi-Robot Collaboration[J]. ACM Transactions on Sensor Networks (ACM TOSN), 2026, 22(2): 1-27. (CCF-B, THU-B)

[15] Chen X*, Xu J*, Ding W, **Wang H**, Luo X, Duan R, Chen J, Wang X, Liu Y, Chen X†. Count every rotation and every rotation counts: Exploring drone dynamics via propeller sensing[C]. ACM/IEEE International Conference on Embedded Artificial Intelligence and Sensing Systems (ACM SenSys), 2026. (THU-A, CCF-B)

[16] Zhao C*, Ruan C*, Xu J, **Wang H**, Wang S, Li J, Zha J, Yang Z, Liu Y, Chen X†. Foes or Friends: Embracing Ground Effect for Edge Detection on Lightweight Drones[C]. The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom), ACM, 2024: 1377-1392. (CCF-A, THU-A)

[17] Luo X, **Wang H**, Ruan C, Liang C, Xu J, Chen X†. EventTracker: 3D Localization and Tracking of High-Speed Object with Event and Depth Fusion[C]. Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom Picasso workshop), ACM, 2024: 1974-1979.

[18] Chen X, Xiao Z, Cheng Y, Hsia CC, **Wang H**, Xu J, Xu S, Dang F, Zhang XP, Liu Y, Chen X†. SOScheduler: Toward Proactive and Adaptive Wildfire Suppression via Multi-UAV Collaborative Scheduling[J]. IEEE Internet of Things Journal (IEEE IoT-J), 2024, 11(14): 24858-24871. (JCR Q1, IF: 8.9)

[19] Chen X, **Wang H**, Li Z, Ding W, Dang F, Wu C, Chen X†. DeliverSense: Efficient delivery drone scheduling for crowdsensing with deep reinforcement learning[C]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (ACM Ubicomp CPD workshop). 2022: 403-408. (Best Paper Award)

Posters/Demos

[20] Ding W, Zhang Z, Xu X, **Wang H**, Zhu Y, Ji S, Zhou X, Xu J, Huang D, Chen X†. Demo: Hawkeye: Practical in-flight obstacle avoidance with event camera and

lidar fusion[C]. Proceedings of the 31st Annual International Conference on Mobile Computing and Networking (ACM MobiCom Demo), ACM 2025: 1213-1215.

[21] Chen X, Xiao Z, Cheng Y, Hsia CC, **Wang H**, Dang F, Xu J, Zhang XP, Liu Y, Chen X†. Poster: FireHunter: Toward proactive and adaptive wildfire suppression via multi-UAV collaborative scheduling[C]. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS Poster), IEEE, 2024: 1-2.

[22] Xiao Z, Luo J, Chen X, Cheng Y, **Wang H**, Chen X†. Poster: Sprinkler-UAV Cooperative Active Scheduling System[C]. 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (ACM/IEEE IPSN Poster), ACM/IEEE, 2024: 279-280.

[23] Cheng Y, Chen X, Yang Y, **Wang H**, Liu Y, Chen X†. Poster: Olfactory Sensing in Turbulent Airflow via Collaborative Robots[C]. Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications(ACM HotMobile Poster), ACM, 2024: 135-135.

[24] **Wang H**, Luo X, Ruan C, Chen X, Ding W, Liu Y, Chen X†. Poster: Fusing Event and Depth Sensing for Dynamic Objects Localization and Tracking[C]. Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications(ACM HotMobile Poster), ACM, 2024: 141-141.

[25] **Wang H**, Man F, Wang Z, Liu Y, Chen X, Ding W†. Poster: TENG-enabled Self-powered Human-machine Interfaces for the Metaverse[C]. Proceedings of the 22nd International Conference on Information Processing in Sensor Networks (ACM/IEEE IPSN Poster), IEEE, 2023: 306-307.

Patent

[26] 陈鑫磊, **王灏洋**, 陈学诚, 赵柏宁, 黄绍伦. 异构智能无人集群协同定位方法、设备及存储介质。专利号：ZL 2022 1 1590859.3，已授权。

[27] 陈鑫磊, **王灏洋**, 罗心雨, 徐京傲. 一种基于事件相机与毫米波雷达高频融合的地基无人机定位方案。申请号：CN202510166615.X，已受理。

[28] 陈鑫磊, 罗心雨, **王灏洋**. 一种高速动态物体三维定位与追踪的跨模态感知系统和方法。申请号：CN202511345453.2，已受理。

[29] 陈鑫磊, 罗心雨, **王灏洋**, 徐京傲. 一种基于事件相机的无人机视觉定位服务系统。申请号：CN202510166617.9，已受理。

[30] 陈鑫磊, 郭芮杉, **王灏洋**, 马捧涛. 一种中央凹仿生混合像素动态视觉传感器设

计及成像方法。申请号：CN202511682187.2，已受理。

Academic honors

[31] Silver Prize, The 50th International Exhibition of Inventions Geneva, 2025.

[32] Best Presentation Award, ACM Ubicomp CPD workshop, 2023.

Competition award

[33] The 1^{st} in Low-Altitude Economy Flight Management Challenge Creative Competition. Meituan Inc. 2023.

Honor

[34] National Scholarship, Ministry of Education, China, 2024.

[35] Leaders of Tomorrow Scholarship, Excellence Scholarship, Tsinghua SIGS, 2025.