Ultra-High-Frequency Harmony: mmWave Radar and Event Camera Orchestrate Accurate Drone Landing

Haoyang Wang[1], Jingao Xu[2], Xinyu Luo[1], Xuecheng Chen[1], Ting Zhang[1],

Ruiyang Duan[3], Yunhao Liu[4], Xinlei Chen[1]✉

[1] Shenzhen International Graduate School, Tsinghua University, China; [2] Carnegie Mellon University;
[3] Meituan Academy of Robotics Shenzhen, China; [4] School of Software, Tsinghua University, China;
Email: {haoyang-22, luo-xy23, chenxc21, yunhao}@mails.tsinghua.edu.cn, jingaox@andrew.cmu.edu,
zhangt2112@gmail.com, duanruiyang@meituan.com, chen.xinlei@sz.tsinghua.edu.cn

ABSTRACT

For precise, efficient, and safe drone landings, ground platforms should real-time, accurately locate descending drones and guide them to designated spots. While mmWave sensing combined with cameras improves localization accuracy, lower sampling frequency of traditional frame cameras compared to mmWave radar creates bottlenecks in system throughput. In this work, we replace traditional frame camera with event camera, a novel sensor that harmonizes in sampling frequency with mmWave radar within ground platform setup, and introduce mmE-Loc, a high-precision, low-latency ground localization system designed for drone landings. To fully leverage the *temporal consistency* and *spatial complementarity* between these modalities, we propose two innovative modules, *consistency-instructed collaborative tracking* and *graph-informed adaptive joint optimization*, for accurate drone measurement extraction and efficient sensor fusion. Real-world experiments in landing scenarios from a drone delivery company demonstrate that mmE-Loc outperforms SOTA methods in both accuracy and latency.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; • **Information systems → Location based services**.

KEYWORDS

Drone Ground Localization; Event Camera; mmWave Radar

✉ Corresponding author.
Project homepage: https://mmE-Loc.github.io/

(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 1: Snapshot of drone landing phase, airport, and sensors performance. (a) A delivery drone lands on the platform. (b) The real-world drone airport is equipped with multiple drones for package delivery. (c) Integrating mmWave radar with event camera combines reliable depth sensing and 2D imaging at ultra-high sampling frequencies, enabling high spatial-temporal resolution and depth sensing, while ensuring compatibility with flight controllers.

1 INTRODUCTION

Projected to soar to a $1 trillion market by 2040 [1], the drone-driven low-altitude economy is transforming sectors with revolutionary applications such as on-demand delivery [2–4], meticulous industrial inspections [5–8], and rapid relief-and-rescue [9–11]. 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 (Fig.1a) [12, 13]. Situated near populated and commercial zones, these operations emphasize safety and reliability: our research with a leading drone delivery company reveals that a landing bias of just 10*cm* will result in drones damaging delivery targets or missing their charging ports [14]. Such inaccuracies disrupt the operational efficiency of this swiftly growing economic sector, with potentially 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 [15–17]. However, traditional frame cameras' Achilles heel is capturing
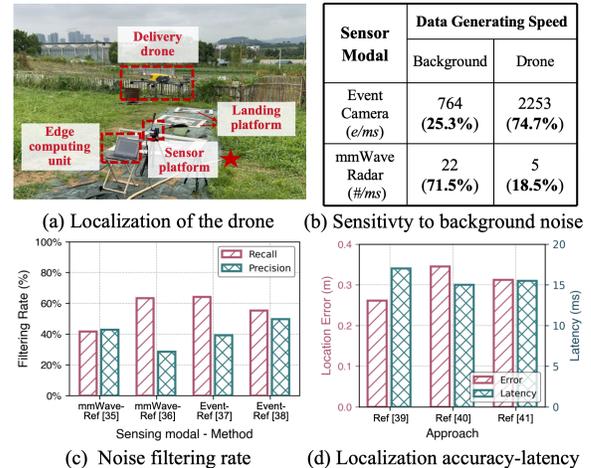
only 2D images without depth information, leading to scale uncertainty that limits the 3D localization accuracy [18–20]. To address this shortcoming, current practices have incorporated mmWave sensing to provide the lacked depth information for better localization accuracy and reliability in various conditions [21–27].

Albeit inspiring, our benchmark study with a world-leading drone delivery company in landing scenarios (Fig.1b) reveals another critical drawback (Fig.1c): the 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 below 50Hz. In contrast, drone flight controllers typically require location input rates over 150Hz to precisely adjust the drone's flight attitude for safe landing [28, 29]. The inefficiency 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 [30, 31], capturing high-speed motions without blurring [32], ideal for fast-tracking tasks [33, 34]. 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 both modalities inspire us to upgrade frame cameras with event cameras to pair with radar for accurate and fast drone localization.

Our work. Following the above insight, we present **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, our benchmark study at a real-world drone delivery airport (Fig.2a) 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 Fig.2b. Existing algorithms [35–38] are typically designed for single-modality, resulting in low noise filtering rates (recall and precision < 65% in Fig.2c). (*ii*) *How to efficiently fuse event camera and mmWave readings* that are heterogeneous in measurement precision, scale, and density? Existing EKF (extended Kalman filter) or PF (particle filter) based approaches [39–41], suffer from cumulative drift errors, making them insufficient for precise localization (Fig.2d). (*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 (Fig.2d) [39–41].

(a) Localization of the drone   (b) Sensitivty to background noise

(c) Noise filtering rate   (d) Localization accuracy-latency

Figure 2: Benchmark study on drone localization. (a) Benchmark study at a real-world drone delivery airport; (b) Both sensors are sensitive to environmental variations; (c) Existing algorithms suffer from low noise filtering rates; (d) Existing algorithms experience cumulative drift errors and delays.

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 frame camera to event camera with ms-level latency to pair the mmWave radar, mmE-Loc improves drone ground localization at data source. The system architecture tightly integrates both modalities, from early-stage noise filtering and drone detection to later-stage fusion and optimization, fully leveraging the unique advantages of both sensors (§2.1).

● *On system algorithm front.* We introduce a Consistency-Instructed Collaborative Tracking (*CCT*) algorithm, which leverages the drone's periodic micro-motion and cross-modal *temporal-consistency* to filter environment-triggered noise, achieving accurate drone detection (§3.1). We then present a Graph-Informed Adaptive Joint Optimization (*GAJO*) algorithm, which fuses *spatial-complementarity* with a novel factor graph to boost drone ground localization, resulting in a trajectory with minimal bias and low cumulative drift(§3.2).

● *On system implementation front.* We further analyze the sources of latency and propose an Adaptive Optimization method to improve the efficiency of the *GAJO* algorithm. This approach allows *GAJO* to dynamically optimize a set of locations, maintaining accuracy while reducing latency (§4).

We fully implement 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 5.15ms, surpassing baselines by >48% and >62%, respectively, and showing minimal sensitivity to drone type and environment. We also deploy mmE-Loc at a real-world drone delivery airport (Fig.2a) for 10 hours, demonstrating its practicality for commercial-level drone landing requirements.

In summary, this paper makes the following contributions. (1) We explore a fresh sensor configuration, event camera plus mmWave radar, that embraces and harmonizes ultra-high sampling frequencies and propose mmE-Loc, a ground localization system for drone landings that delivers *cm* accuracy and *ms* latency.
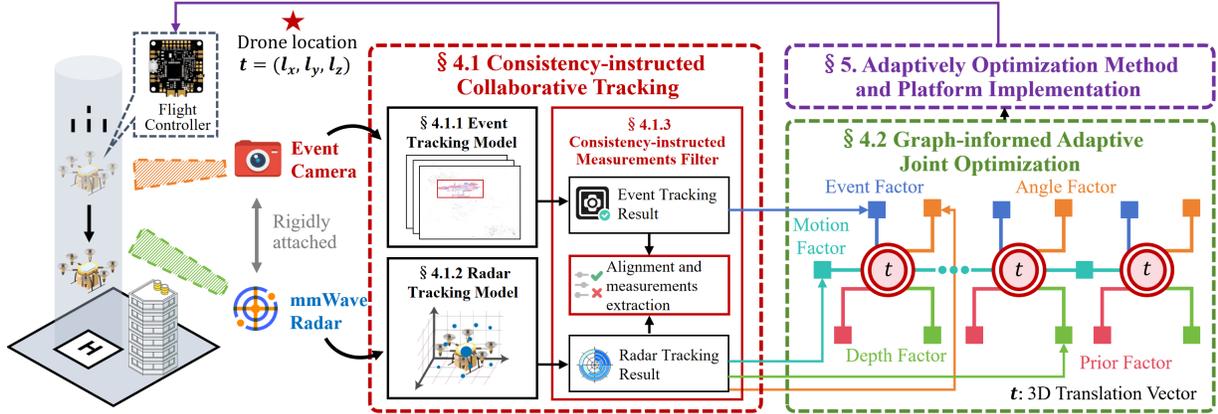
Figure 3: System architecture of mmE-Loc.

(2) We present *CCT*, which leverages *temporal consistency* and the drone's periodic micro-motions for precise drone detection; and *GAJO*, which employs *spatial complementarity* with a novel factor graph to enhance drone localization.

(3) We implement and extensively evaluate mmE-Loc by comparing it with four SOTA methods, showing its effectiveness. We also deploy mmE-Loc in a real-world drone delivery airport, demonstrating feasibility of mmE-Loc.

2 SYSTEM OVERVIEW

The mmE-Loc enhances the mmWave radar with an event camera to achieve accurate and low-latency drone ground localization, allowing the drone to rapidly adjust its location state and perform a precise landing. Given the critical importance of safety in commercial drone operations, mmE-Loc can work in conjunction with RTK or visual markers to ensure precise landing performance. In this section, we mathematically introduce the problem that mmE-Loc tries to address and provide an overview of the system design.

2.1 Problem Formulation

In this section, we illustrate key variables in mmE-Loc and introduce 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 E; (*ii*) the Radar reference system R; (*iii*) the Object reference system O; (*iv*) the Drone reference system D. Note that a drone can be considered as an object. For clarity, before an object is identified as a drone, we utilize O. Once confirmed as a drone, we use D for the drone and continue using O for other objects. Throughout the operation of system, E and R remain stationary and are rigidly attached together, while O and D undergo changes in accordance with movement of the object and the drone, respectively. The transformation from R to E can be readily obtained from calibration [42].

Goal of mmE-Loc. The goal of mmE-Loc is to determine 3D location of the drone, defined as t_{ED}, the translation from coordinate system D to E. Specifically, mmE-Loc optimizes and reports 3D location of drone (l_x, l_y, l_z) at each timestamp i with input from 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 [43]. The former representation is adopted in the paper, as it is commonly used in drone flight control systems.

2.2 Overview

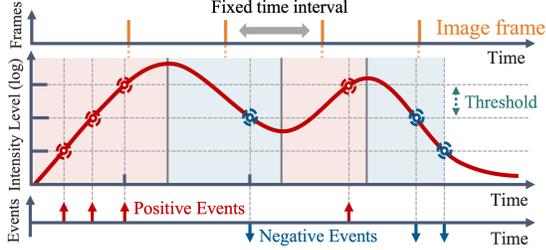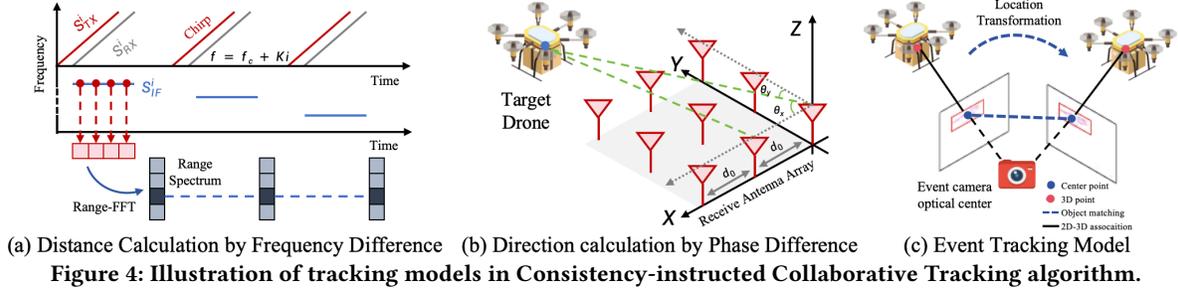As illustrated in Fig.3, mmE-Loc comprises two key modules:

• The *CCT* (**C**onsistency-instructed **C**ollaborative **T**racking) for noise filtering, drone detection, and preliminary localization of the drone. This module utilizes time-synchronized event streams and mmWave radar measurements as inputs. Subsequently, 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 periodic micro-motion to extract drone-specific measurements and achieve drone preliminary localization.

• The *GAJO* (**G**raph-informed **A**daptive **J**oint **O**ptimization) for fine 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 adaptively refines them, determining the fine location of drone with *ms*-level processing time.

3 SYSTEM DESIGN

In this section, we introduce *CCT*: Consistency-instructed Collaborative Tracking for noise filtering, detection, and preliminary localization of drone (§ 3.1). Subsequently, we delve into *GAJO*: Graph-informed Adaptive Joint Optimization for fine localization and trajectory optimization of drone (§ 3.2).

3.1 *CCT*: 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

(a) Distance Calculation by Frequency Difference (b) Direction calculation by Phase Difference (c) Event Tracking Model

Figure 4: Illustration of tracking models in Consistency-instructed Collaborative Tracking algorithm.

Figure 5: Illustration of synchronous frames and asynchronous events. Frame cameras use a global shutter to capture images at fixed intervals, while each pixel in an event camera responds independently, generating events asynchronously when intensity changes exceed a threshold.

two modalities, presents challenges for noise filtering. This results in drone detection bottlenecks, which further reduce the efficiency and accuracy of localization. Therefore, in this part, we focus on enhancing noise filtering and drone detection, while providing preliminary localization of the drone.

To address this challenge, we explore the operational principles of both sensors. Our design is based on observations: *(i) Event camera and mmWave radar demonstrate temporal consistency and distinct response mechanisms.* Event camera and mmWave radar maintain *ms*-level latency. Additionally, event cameras are unaffected by multipath effects, whereas mmWave radar remains impervious to changes in brightness. *(ii) Drone exhibits periodic micro motion features (e.g., propeller rotation),* which can serve as stable and distinctive features of drone. These facilitate efficient cross-modal noise filtering by aligning measurements from the event camera and mmWave radar and enable drone detection by extracting drone measurements through periodic micro-motions.

To realize this idea, we design *CCT*, a lightweight cross-modal drone detector and tracker. *CCT* includes several components: (*i*) a Radar Tracking Model (§3.1.1) providing sparse point cloud indicating distance and direction information of objects; (*ii*) an Event Tracking Model (§3.1.2) for event filtering, detection, and tracking of objects; (*iii*) a Consistency-instructed Measurements Filter (§3.1.3) utilizes temporal consistency between both modalities and drone's periodic micro-motion features to extract detection and point cloud of drone, facilitating the preliminary localization.

3.1.1 **Radar Tracking Model**. In this part, we calculate the distance D and direction vector $\vec{v}$ between the radar and objects, along with a preliminary estimation of the object's location, as depicted in Fig.4a and Fig.4b.

Distance calculation. As shown in Fig.4a, 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:

$$S_{TX}^i = \exp\left[j\left(2\pi f_c i + \pi K i^2\right)\right], S_{RX}^i = \alpha S_{TX}\left[i - 2D^i/c\right], \quad (1)$$

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

$$S_{IF}^i = LPF(S_{TX}^{i*} S_{RX}^i) \approx \alpha \exp\left[j2\pi\left(2KD^i/c\right)i\right]. \quad (2)$$

The frequency value f_{IF} within S_{IF}^i encapsulates distance information. After the Range-FFT operation S_{IF}^i, f_{IF} is extracted, facilitating distance calculation $D^i = cf_{IF}/2K$.

Direction calculation. Using a fixed antenna array, the mmWave radar determines the object's direction by employing two orthogonal linear arrays. As depicted in Fig.4b, 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 the object's direction at time i is given by $\vec{v}^i = [\cos\theta_x \cos\theta_y \sqrt{1 - \cos^2\theta_x - \cos^2\theta_y}]^{\mathrm{T}}$.

Using the distance and angle information obtained above, along with the spatial relationship between radar and event camera, we can determine the preliminary 3D location estimation of the object in E as $P_E = D\vec{v} + t_{ER}$. We then leverage the mmWave radar for object 3D location tracking, estimating the translation t_{EO} of the object from 0 to E at time i:

$$\begin{aligned} t_{EO}^i &= t_{EO}^{i-1} + U_E^i + w^i + w^{i-1} \\ &= t_{EO}^{i-1} + \left(P_E^i - P_E^{i-1}\right) + w^i + w^{i-1}. \end{aligned} \quad (3)$$

U_E^i is discrepancy between two radar calculation results at times i and $i-1$ in E. w_i and w_{i-1} signify the measurement noise.

While mmWave radars excel at estimating object depth along the radial direction, they struggle to accurately capture horizontal and vertical (tangential) motion [44, 45]. To address this issue, we introduce the event camera, which has similar latency but a different sensing principle. With high spatial resolution, the event camera detects objects and compensates for mmWave radars' limitations in the tangential direction.

3.1.2 **Event Tracking Model**. In this part, we demonstrate the process of noise filtering from a stream of asynchronous events, and how to detect and track objects with the filtered events, as

| (a) RGB frame | (b) Origin event | (c) Filtering event | (d) Result alignment | (e) Measurements extraction |
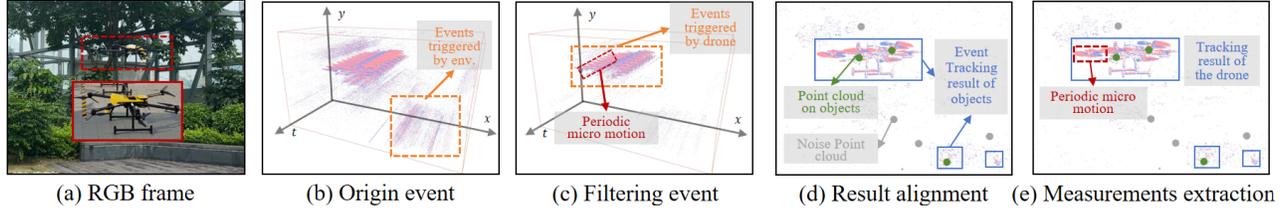
Figure 6: Step-by-step filtering performance. The *CCT* module in mmE-Loc eliminate noise events, mmWave point cloud and erroneous detection by employing *temporal-consistency* of both modalities.

depicted in Fig.4c. Compared to frame cameras that use a global shutter to capture images at fixed intervals, event cameras record pixel-wise intensity changes with *ms*-level resolution and sampling latency, enabling high-speed motion capture without blurring but adding complexity to noise filtering and object detection (Fig. 5).

Similarity-informed event filtering. Event cameras are prone to noise from transistor circuits and other non-idealities, requiring pre-processing filtering. For the ith event $e^i_{(x,y)}$ with the timestamp $t^i_{(x,y)}$, we assess 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 (Fig.6b, Fig.6c). We utilize the Surface of Active Events (SAE) [46] 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 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. We employ 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, we compare 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, we deploy Kalman filter-based trackers with a constant velocity motion model, as the Kalman filter provides low-latency estimates with minimal computational cost. A tracker predicts the state of the current object and associates it with the input cluster that has the largest Intersection Over the Union area. The input cluster corrects tracker state, generating bounding boxes, and effectively tracking moving objects, including the drone.

Using bounding box proposals and the pinhole camera model with projection function π, we estimate the preliminary 3D locations of objects. Specifically, the projection function π transforms a 3D point $\mathbf{X}_E$ in E into a 2D pixel x in the image plane as:

$$x = \pi(\mathbf{X}_E) = [f_x X_E / Z_E + c_x, f_y Y_E / Z_E + c_y]^T, \mathbf{X}_E = [X_E, Y_E, Z_E]^T, \quad (4)$$

where $[f_x, f_y]^T$ is the focal length of the event camera, and $[c_x, c_y]^T$ denotes the principal point, both being intrinsic camera parameters. Then, the object's preliminary location at time i is estimated using the center point of bounding box proposal x^i as:

$$x^i = \pi(\mathbf{X}^i_E) + v^i = \pi(\mathbf{X}^i_0 + t^i_{EO}) + v^i, \quad (5)$$

where $\mathbf{X}^i_0$ represents the corresponding 3D point of center point x^i in the object reference 0, v^i denotes the random noise. When extracting center points from bounding box proposals, we first undistort their coordinates.

3.1.3 Consistency-instructed Measurements Filter. The *Event Tracking Model* detects drones and other objects causing light changes, such as indicator lights or shadows. The system must distinguish the landing drone from these objects. Similarly, the *Radar Tracking Model* outputs a 3D point cloud containing both the drone and noise from multipath effects, requiring extraction of the drone's relevant points.

Consistency-instructed alignment. Utilizing the *temporal-consistency* from the event camera and radar, and their distinct mechanisms respond to dynamic objects, we filter event camera results affected by lighting variations on stationary objects and vice versa for radar points influenced by multipath effects. Specifically, we align synchronized radar points (*Radar Tracking Model*) to each event bounding box (*Event Tracking Model*) (Fig.6d). Using event camera's projection, we determine that object's location lies along the ray from the camera's optical center through bounding box center. The system then identifies the nearest radar points along this ray to isolate the object-associated points. If no radar point is detected, the bounding box is treated as noise and disregarded.

Periodic micro motion-aid measurements extraction. Since each platform supports one drone landing at a time, we need to identify a distinguishing feature of the landing drone, which effectively differentiates the drone from noise, and use it to extract landing drone-specific measurements from the aligned tracking results. Our finding is that drones exhibit periodic micro-motions (e.g., propeller rotation), which can serve as stable and distinctive features of the drone. We transform the spatio-temporal distribution of events into a heatmap and apply statistical metrics to isolate drone measurements leveraging this feature. Specifically, within a time window $[i, i + \delta i]$, events are binned into a 2D histogram where each bin corresponds to a spatial region (e.g.., 5×5 pixels). Bins containing propeller rotation tend to accumulate more events due to rapid light intensity changes. Meanwhile, these propeller rotations generate bipolar events within a bin, while background motion and noise typically result in unipolar events (e.g., from flying birds). Therefore, we select bins with propeller rotation based on event counts and the proportion of positive events, favoring those with higher counts and a more balanced ratio. Finally, we identify event tracking results with the most bins indicative of propeller rotation and corresponding point clouds (t_{EO}), designating them as drone tracking results (t_{ED}) for preliminary localization from two models as shown in Fig.6e. When multiple drones are scheduled to land, they descend and land sequentially. This method accurately identifies the landing drone and extracts relevant measurements.

3.2 *GAJO*: 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, we prioritize accurate drone ground localization and trajectory tracking.

Our design is founded on the insight that *the Event Tracking Model and Radar Tracking Model provide distinct features that are spatial-complementarity to each other.* As a result, the 2D imaging capability of event cameras and the depth sensing capability of mmWave radar mutually enhance each other when combined, as demonstrated in Fig.7. Since both the event stream and mmWave samples are drone-related, fully leveraging the *spatial- complementarity* of these two modalities through joint optimization offers the potential to significantly improve performance. This leads to a trajectory with reduced bias and minimized cumulative drift.

To realize this idea and push the limit of localization accuracy while minimizing latency, we introduce a *GAJO*, a factor graph-based location optimization framework designed for low-latency and accurate drone 3D localization (§3.2.1). *GAJO* includes two parallel tightly coupled modules: (*i*) short-term (inter-SAE tracking) and (*ii*) long-term (local location optimization) optimizations, collectively enhancing location tracking precision (§3.2.3). Beyond the capabilities of *Event Tracking model* and *Radar Tracking model*, *GAJO* assimilates prior knowledge of drone's flight dynamics to refine the trajectory for enhanced smoothness and accuracy (§3.2.2).

3.2.1 *Factor graph-based Optimization*.

A factor graph comprises variable nodes, indicating the states to be optimized (e.g., t_{ED}^i), 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) and Radar Tracking (RT) model (D^i, $\vec{v}^i$, and U_E^i). To estimate the values of a set of variable nodes $\mathcal{X} = \{t_{ED}^i | i \in \mathcal{T}\}$ given measurements $\mathcal{Z} = \{x^i, D^i, \vec{v}^i, U_E^i | i \in \mathcal{T}\}$, *GAJO* optimizes all connected factor nodes based on maximum a posteriori estimation:

$$\hat{\mathcal{X}} = \arg\max_{\mathcal{X}} p(\mathcal{X} \mid \mathcal{Z}) = \arg\max_{\mathcal{X}} p(\mathcal{X}) \, p(\mathcal{Z} \mid \mathcal{X})$$
$$= \arg\max_{\mathcal{X}} p(\mathcal{X}) \prod_{i \in \mathcal{T}} p\left(x^i \mid t_{ED}^i\right) p\left(D^i, \vec{v}^i, U_E^i \mid t_{ED}^i\right), \quad (6)$$

which follows the Bayes theorem. $p(\mathcal{X})$ is the prior information over $\mathcal{X} = \{t_{ED}^i | i \in \mathcal{T}\}$, which is inferred from drone flight characteristics. The $p\left(x^i \mid t_{ED}^i\right)$ is the likelihood of the ET model measurements. The $p\left(D^i \mid t_{ED}^i\right)$, $p\left(\vec{v}^i \mid t_{ED}^i\right)$ and $p\left(U_E^i \mid t_{ED}^i\right)$ are likelihood of the RT model measurements.

3.2.2 *Probabilistic Representation*.

Inferring the drone's location requires prior term and likelihood term in Eqn.(6).

Prior term. The prior term, $p(t_{ED}^i)$, represents the drone's location probability distribution at time i unaffected by current measurements. Derived from a constant velocity model, it suggests the
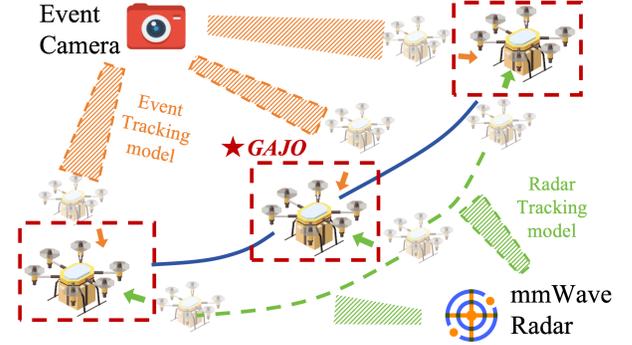
Figure 7: Illustration of relationship between *GAJO* and *CCT*. The *GAJO* module harness the *spatial-complementarity* of both modalities through a join optimization.

drone maintains steady speed over short intervals, allowing us to predict the prior location using:

$$\bar{t}_{ED}^i - t_{ED}^{i-1} = t_{ED}^{i-1} - t_{ED}^{i-2}. \quad (7)$$

ET model likelihood. The likelihood $p(x^i | t_{ED}^i)$ from ET model represents the center point distribution at a given drone location. In many tracking systems [47], center point noise v^i is assumed Gaussian, proving effective. Thus, likelihood of ET model is:

$$p(x^i | t_{ED}^i) \sim \mathcal{N}(\pi(\mathbf{X}_E^i), \sigma_{ET}), \quad (8)$$

where σ_{ET} is the center point standard deviation.

RT model likelihood. The likelihood of the RT model $p(D^i \mid t_{ED}^i)$, $p(\vec{v}^i \mid t_{ED}^i)$ and $p(U_E^i \mid t_{ED}^i)$ indicates the distribution of the measured distance, angle, and motion at a given drone location. The distance, angle, and motion from RT model likelihood are:

$$p(D^i \mid t_{ED}^i) \sim \mathcal{N}(||t_{ED}^i||, \sigma_D), \quad p(\vec{v}^i \mid t_{ED}^i) \sim \mathcal{N}(\vec{v}_{t_{ED}^i}, \sigma_{\vec{v}}),$$
$$p(U_E^i \mid t_{ED}^i) \sim \mathcal{N}(t_{ED}^i - t_{ED}^{i-1}, \sigma_{U_E}), \quad (9)$$

where σ_D, $\sigma_{\vec{v}}$ and $\sigma_{U_E^i}$ are the standard deviation of distance, angle, and motion measurements respectively.

3.2.3 *Fusion-based Tracking*.

In mmE-Loc, two fusion schemes are employed for sensor fusion and optimization, as depicted in Fig.8. The first, inter-SAE tracking, aims for instant drone location estimation by minimizing errors across different tracking models simultaneously. The second, local location optimization, enhances overall trajectory accuracy through the joint optimization of a selected set of locations.

Inter-SAE tracking. Once the measurements of ET model and RT model ($x^i, D^i, \vec{v}^i, U_E^i$) received, the prior factor, ET factor and the RT factor are formulated as follows:

$$E_{\text{Prior}}^i = -\log p\left(t_{ED}^i\right) \propto \left\|t_{ED}^i - \bar{t}_{ED}^i\right\|_{\sigma_{t_{ED}}}^2,$$
$$E_{\text{ET}}^i = -\log p\left(x^i \mid t_{ED}^i\right) \propto \rho(\left\|x^i - \pi(\mathbf{X}_E^i)\right\|_{\Sigma_E}^2),$$
$$E_{\text{RT}}^i = -\log p\left(D^i, \vec{v}^i, U_E^i \mid t_{ED}^i\right)$$
$$\propto \left\|||t_{ED}^i|| - D^i\right\|_{\sigma_D}^2 + \left\|\vec{v}_{t_{ED}^i} - \vec{v}^i\right\|_{\sigma_{\vec{v}}}^2 + \left\|(t_{ED}^i - t_{ED}^{i-1}) - U_E^i\right\|_{\sigma_{U_E}}^2, \quad (10)$$

where $\|e\|_{\Sigma_E}^2 = e^T \Sigma^{-1} e$. The symbol Σ_E represents the covariance matrix associated with the event camera measurements.
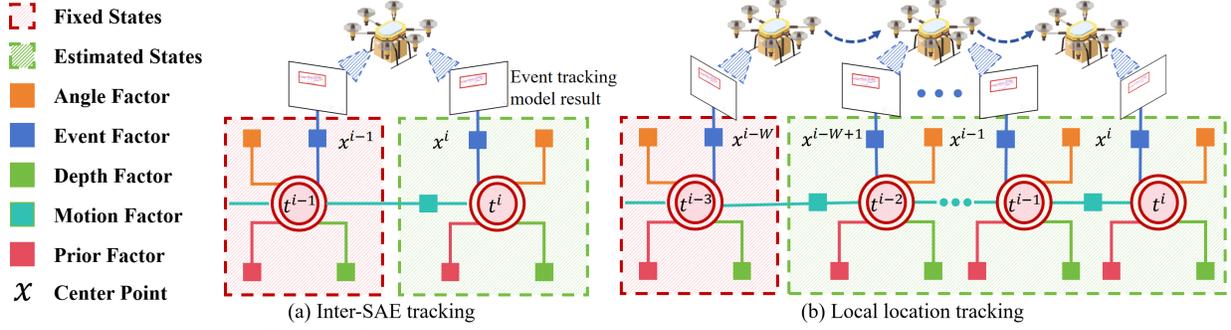
Figure 8: Long-short term optimization based on the factor graph.

On this basis, the inter-SAE tracking in Fig.8a is performed to give an instant location result based on Eqn.(6) as follows:

$$\hat{t}^i_{ED} = \arg\max_{t^i_{ED}} p(t^i_{ED} \mid t^{i-1}_{ED}, t^{i-2}_{ED}) p(x^i \mid t^i_{ED}) \, p(D^i, \vec{v}^i, U^i_E \mid t^i_{ED})$$

$$= \arg\min_{t^i_{ED}} -\log\Big(p(t^i_{ED} \mid t^{i-1}_{ED}, t^{i-2}_{ED}) p(x^i \mid t^i_{ED}) p(D^i, \vec{v}^i, U^i_E \mid t^i_{ED}) \Big)$$

$$= \arg\min_{t^i_{ED}} \Big(E^i_{\text{prior}} + E^i_{\text{ET}} + E^i_{\text{RT}} \Big).$$

(11)

Local location optimization. To mitigate cumulative drift, periodic local location optimization is conducted, correcting estimated locations based on multiple consecutive SAEs. This optimization entails jointly optimizing the locations of a SAE set denoted as $\mathcal{X} = \bigcup_{i\in\mathcal{T}} \{t^i_{ED}\}$, as shown in Fig.8b, where $W = |\mathcal{T}|$. The optimization problem formulation is as follows:

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

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

(12)

It is worth noting that (i) when the local location optimization is triggered, (ii) what is the size of $\mathcal{T}$ ($W = |\mathcal{T}|$), and (iii) how to solve the inter-SAE tracking and local location optimization problems affect the latency and accuracy of localization. Hence, we enhance the efficiency of *GAJO* through an adaptive optimization method.

4 IMPLEMENTATION

4.1 Push the Limit of Accuracy and Latency

Factor graph solving. We represent the estimation problems Eqn.(11) and Eqn.(12) using a factor graph model. To solve the nonlinear least-squares problems, we linearize the observation model and solve the least-squares formulation:

$$\hat{\mathcal{X}} = \arg\min_{\mathcal{X}} \|A\mathcal{X} - \mathbf{b}\|^2,$$

(13)

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 [48] 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 $\mathbf{d} \in \mathbb{R}^n$ [49]. Although re-linearizing and regenerating R with new measurements can help mitigate errors, applying this approach to problem (12) can

be computationally expensive, as it requires frequent updates and increased processing overhead, limiting real-time performance.

Algorithm 1: Adaptively Optimization method

Data: Original factor graph G; New measurements $D^i, \vec{v}^i, U^i_E$; square root information matrix R

Result: Updated locations $\hat{\mathcal{X}}$

1   $G \leftarrow$ AddFactorToGraph($G^i, D^i, \vec{v}, U^i_E$);

2   $R \leftarrow$ **IncrementalUpdate**(G);

3   $\hat{\mathcal{X}} \leftarrow$ Backsubstitution(R);

4   $L \leftarrow \emptyset$;   ▷*Set of nodes need to be linearized*;

5   **for** *all* $t^i_{ED} \in \mathcal{X}$ *and all* $\hat{t}^i_{ED} \in \hat{\mathcal{X}}$ **do**

6   **if** $\hat{t}^i_{ED} - t^i_{ED} \geq \delta$ **then**

7      $L \leftarrow L \cup t^i_{ED}$;

8   **end**

9   **end**

10 **if** $|L| \geq L_T$ *or* $\|\hat{\mathcal{X}} - \mathcal{X}\| \geq \Delta$ **then**

11   **for** *all* $t^i_{ED} \in \mathcal{X}$ **do**

12   UpdateLinearizationPoint(t^i_{ED});

13   **end**

14    $R \leftarrow$ **FullUpdate**(G);

15    $\hat{\mathcal{X}} \leftarrow$ Backsubstitution(R);

16 **end**

Adaptive optimization method. To tackle this, we propose an adaptive 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 [50]. By adaptively combining updated and regenerated R, this method reduces latency and enhances accuracy.

Algorithm1 shows how adaptively optimization method solves local location optimization problem. Lines 1-3 depict tracking with incrementally updated R, while lines 4-16 show tracking with re-generated R. Specifically, when receiving new measurements, function AddFactorToGraph updates the factor graph, and the function IncrementalUpdate incrementally updates R with new measurements [50]. We then solve local location tracking with this incrementally updated R. When one of two conditions is met, we solve local location tracking with re-generated R: (i) we track locations that have changed significantly in a set $L = \{t^i_{ED} : t^{\hat{i}}_{ED} - t^i_{ED} \geq \delta\}$. If enough locations have undergone significant changes (i.e., $|L| \geq L_T$), we solve location tracking with re-generated R output by function FullUpdate. (ii) The norm of total location changes exceeds

a threshold Δ (i.e., $||\hat{X} - X|| \geq \Delta$). Since the local location tracking involves repeatedly solving linear equations, this condition keeps current solution from diverging too far from optimal solution.

4.2 Platform Implementation

In this section, we will detail the platform implementation.

Sensor platform configuration. As illustrated in Fig.9, we implement our 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 deployed at the center of the landing pad. In the case study at a real-world airport, the platform is positioned at the edge of the landing pad for safety reasons, with the distance between the sensor platform and the center of the pad being less than 1*m*. After takeoff, the drone appears within FoV of the event camera. All sensors are synchronized through the Robot Operating System (ROS). mmE-Loc is deployed on a PC with Ubuntu 20.04, featuring an Intel i7-12900K CPU, 32GB of RAM, and an NVIDIA GeForce GTX 1070 GPU. For practical deployments, the sensor platform is expected to be positioned at the center of the landing pad, with its height aligned to that of that of the pad, ensuring the drone remains within the event camera's FoV at all times for continuous localization.

5 EVALUATION

Our evaluation of mmE-Loc is comprehensive and grounded entirely in real-world experimentation. We begin with experimental settings in §5.1. Then, in §5.2, we focus on key performance metrics: localization accuracy and latency. §5.3 delves into external factors impacting mmE-Loc, such as drone characteristics and environmental conditions. §5.4 assesses benefits of sensor fusion and the performance of various modules. Finally, in §5.5, we evaluate system load, including latency, CPU usage, and memory consumption.

5.1 Experimental Methodology

In this section, we will outline the experimental methodology.

Experiment setting. We use a deployment setup and drone equipment that closely mimic real-world applications. Fig.9 shows the experimental scenarios in both an *indoor* laboratory and an *outdoor* flight test site. The setup in Fig.9a includes the event camera and mmWave radar mounted together *at the ground of the experimental area*. We evaluate 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. We conducted extensive experiments over 30 hours, collecting more than 400GB of raw data.

Ground truth. In the indoor scenario, we used a motion capture system with fourteen cameras. This system covers an *8m × 8m × 8m* area, providing localization accuracy within *1 mm*, allowing precise performance evaluation under controlled conditions. In the outdoor scenario, we conducted experiments at a secluded site with

excellent GPS signal reception. We set up a Real-Time Kinematic Positioning (RTK) base station to rebroadcast the GPS signal phase, ensuring high-fidelity RTK processing. The RTK localization results served as the ground truth for our outdoor tests.

Comparative Methods. We compare mmE-Loc under various conditions with three related systems: (*i*) *Baseline-I* [40]: a SOTA point cloud-based drone localization system using single-chip mmWave radar, which is a ground localization system. (*ii*) *Baseline-II* [39]: a SOTA single event camera-based drone ground localization system applies to drones with known geometries. The original method, designed for onboard obstacle localization, lacks publicly available code. We implement its monocular 3D localization module which uses drone physical characteristics, adapting it for *ground-based* drone localization. For a fair comparison, we exclude the original method's ego-motion compensation used for noise reduction due to drone movement. (*iii*) *Baseline-III* [39]: a SOTA dual event camera-based drone localization system. The original method utilizes onboard stereo event cameras for obstacle localization, but its source code is not available. Our implementation excludes the ego-motion compensation component from the original method, implementing its stereo 3D location estimation module to enable *ground-based* drone localization. (*iii*) *Baseline-IV* [51]: a SOTA deep learning-based object localization system utilizes monocular images and mmWave point clouds as input, harnessing the complementary information from both the radar and the camera. We pre-train the neural networks and apply a Kalman Filter to adapt the original method for *ground-based* 3D drone localization.
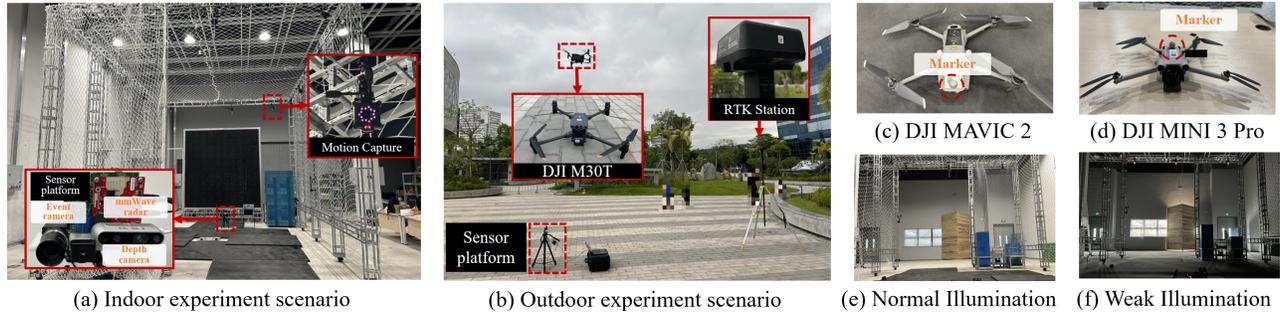
Evaluation Metrics. The mmE-Loc continuously reports the drone's location estimation with low latency, ensuring accurate and responsive localization. Similar to related works, we measure location estimation error in meters and processing latency in milliseconds, allowing for a direct comparison with existing methods in terms of both accuracy and computational efficiency.

Robustness Experiments. During the drone landing, localization may be performed under varying lighting conditions. To assess the adaptability of mmE-Loc in different scenarios, we conduct experiments across a range of conditions, including different environments, drone models, lighting intensities (Fig.9a-f), and background dynamics (controlled by the presence of other moving objects, e.g., balls). We also evaluate the impact of varying distances, occlusions (by partially obstructing the drone within the field of view of the event camera), and velocities to highlight the robustness. In this part, Baseline-IV is excluded from comparison due to its high latency stemming from frame camera's exposure time.
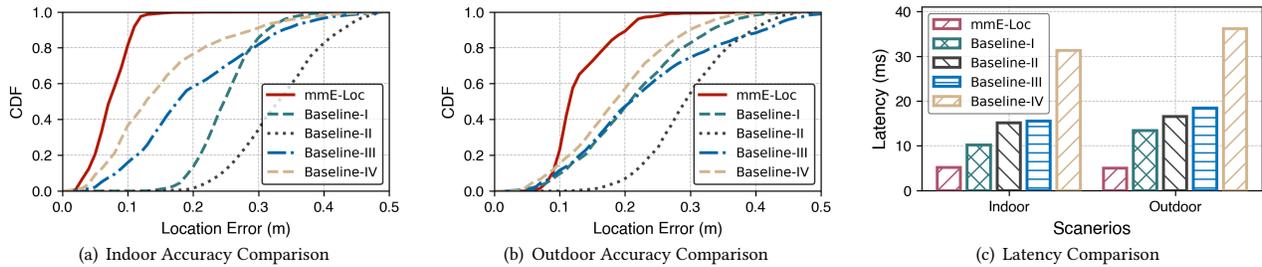
5.2 Overall Performance

In this section, we demonstrate overall performance of mmE-Loc.

Drone localization. Fig.10a illustrates the localization performance of mmE-Loc compared to four baselines in an indoor environment, using a DJI Mini 3 Pro drone. mmE-Loc's average end-to-end localization error is 0.083*m*, which outperforms the other baselines with errors of 0.261*m*, 0.345*m*, and 0.209*m*, 0.160*m*, making it suitable for aiding in landing. Fig.10b shows mmE-Loc's localization performance in an outdoor setting using a DJI M30T drone. mmE-Loc achieves the lowest average error of 0.135*m* compared to the other baselines, outperforming them by 39.2%, 56.0%, 43.8%, 31.8%. Fig.11 shows error distribution along the x, y, and z dimensions

(a) Indoor experiment scenario (b) Outdoor experiment scenario (c) DJI MAVIC 2 (d) DJI MINI 3 Pro (e) Normal Illumination (f) Weak Illumination

Figure 9: Experimental setup and scenarios of mmE-Loc. (a) A laboratory scenario with motion capture system for ground-truth collection. (b) An outdoor scenario with RTK system for ground-truth collection. (c)-(f) Different drone with different size (DJI MAVIC 2 and DJI MINI 3 Pro), and different illumination of the laboratory (normal and weak), respectively.

(a) Indoor Accuracy Comparison (b) Outdoor Accuracy Comparison (c) Latency Comparison

Figure 10: Overall performance comparison of mmE-Loc and four related works.
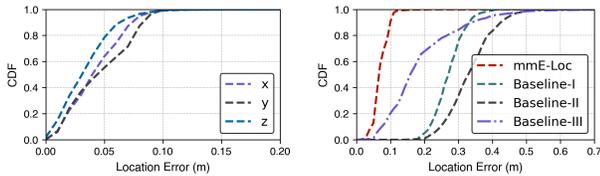
Figure 11: Error distribution along the x, y, and z. **Figure 12: Error in the near-distance setting**

during the typical landing process, and Fig.12 shows errors in the near-distance setting. Baseline-I introduces point cloud errors due to phase center offset, causing the point cloud data to deviate from the drone's geometric center. Additionally, in indoor environments, radar measurements are susceptible to specular reflections, diffraction, and multi-path effects, further contributing to measurement errors. Baseline-II uses a single event camera with known drone geometries, while Baseline-III, with dual event cameras, estimates depth through stereo vision. Both are prone to errors when drone detection is affected by outdoor environmental noise (e.g., birds), leading to depth estimation inaccuracies. Baseline IV incorporates deep learning, which can lead to increased errors when used in environments not present in the training set. The results demonstrate that mmE-Loc achieves significant improvements by combining complementary strengths of radar and event camera. mmE-Loc does not require a pre-training procedure, ensuring its applicability.

End-to-end latency. We evaluate end-to-end latency, including the *CCT* and *GAJO* phases. As shown in Fig.10c, mmE-Loc achieves 5.15ms latency indoors, outperforming the four baselines at 10.19ms, 15.12ms, 15.52ms, and 31.2 ms, 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. The

mmE-Loc tightly couples the event camera and mmWave radar, introducing the *CCT* module to leverage temporal consistency and periodic micro motion for drone detection. The *GAJO* module and the adaptive optimization method are then used to exploit spatial complementarity, further reducing latency.

5.3 System Robustness Evaluation

To demonstrate the versatility and robustness of mmE-Loc, we conduct experiments under various conditions.

Impact of Drone Type. We evaluate the impact of different drone types under controlled indoor conditions, using a DJI Mini 3 Pro (Fig.9c) and a DJI Mavic 2 (Fig.9d), the results are presented in Fig.13. The results of average errors show that mmE-Loc outperforms all baselines with the DJI Mini 3 Pro. With the DJI Mavic 2, Baseline-II and -III show larger localization errors due to changes in drone geometry affecting depth estimation. mmE-Loc's localization 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 different drone types.

Impact of Environment & Illumination. The performance of mmE-Loc in different environments with varying lighting conditions (Fig.9e and Fig.9f) is shown in Fig.14. Although event cameras have a high dynamic range, drone-generated events are still affected by illumination. Baseline-II and -III experience increased errors as illumination decreases due to deteriorating depth estimation. 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. 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. We test mmE-Loc with dynamic background motion, as shown in Fig.15. The intensity of dynamic background motion is controlled by deploying other moving
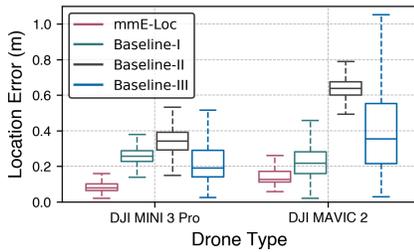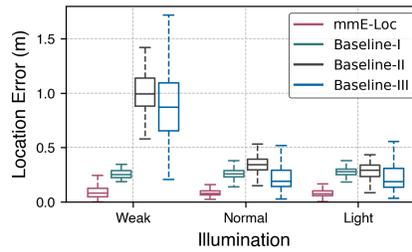
Figure 13: Impact of Drone Type
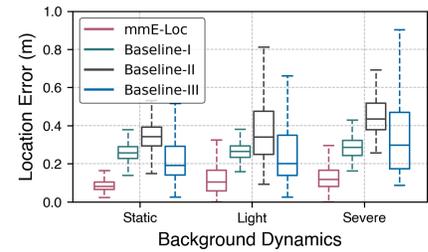

Figure 14: Impact of Env. & Illu.

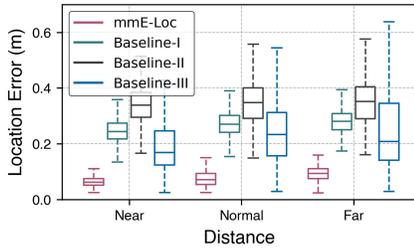**Figure 15: Impact of Background**
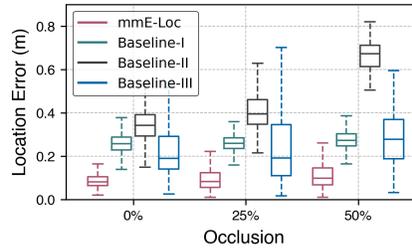

Figure 16: Impact of Distance
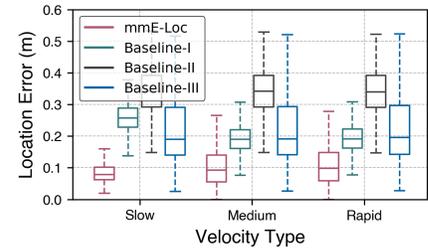

Figure 17: Impact of Occlusion

Figure 18: Impact of Velocity

objects in the scene (e.g., balls). All baselines show higher localization errors with increasing background motion due to misidentification of the radar and camera. Despite this, mmE-Loc maintains an average error of $0.129m$ in the most challenging scenarios. This is achieved through mmE-Loc's consistency-instructed measurement filter, which uses the drone's periodic micro-motions of propeller rotation to distinguish it from other objects.

Impact of Distance. We investigate the impact of drone-to-platform distance indoors with a DJI Mini 3 Pro ($0.25m \times 0.36m \times 0.07m$). Fig.16 shows the results, categorized into Near (< $3m$), Normal ($3m \sim 6m$), and Far (> $6m$) distances. 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 leverages mmWave radar for depth information and an event camera for high-resolution 2D imaging, effectively overcoming the low spatial resolution of mmWave radar and the scale uncertainty of the event camera. Acknowledging the importance of precise landings, mmE-Loc complements existing solutions. Working alongside RTK and visual markers, it enhances the reliability and accuracy of localization in real-world scenarios.

Impact of Occlusion. We validate mmE-Loc's robustness with partially occluded drones. The occlusion is controlled by partially obstructing the drone within FoV of the event camera. Fig.17 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 to $0.12m$. Baseline-II and -III 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. We further evaluate mmE-Loc's robustness under different drone velocities in Fig.18. Velocities are categorized as slow ($v < 0.5m/s$), medium ($0.5m/s \leq v < 1m/s$), and rapid ($1m/s \leq v < 1.5m/s$), corresponding to various drone landing stages. Although error 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%.

5.4 Ablation Study

We experimentally analyze core components of mmE-Loc, focusing on enhancements contributed by each component to overall system.

Effectiveness of Multi-modal Fusion. We demonstrate the superiority of fusing radar and event cameras over using each sensor individually. Fig.19a shows that the fusion-based approach significantly outperforms both radar-only and event camera-only methods in terms of location error. mmE-Loc outperforms event-only approach by 63.6% and exceeds radar-only method by 52.9%.

Contributions of each module. We investigate the contributions of *CCT* and *GAJO* to mmE-Loc by gradually integrating them with the event camera into the baseline system (i.e., the radar-only-based method) and assessing localization accuracy and end-to-end latency. Fig.19b illustrates that without these modules, the baseline method achieves a localization error of $0.229m$ and latency of $10.19ms$. Integrating the event camera with the *CCT* module reduces the localization error to $0.178m$ and decreases latency to $7.27ms$. 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. Finally, integrating both *CCT* and *GAJO* minimizes both the error and latency.

Performance of *CCT*. We tested *CCT*'s filtering performance on mmWave and event data in both indoor and outdoor scenarios. In Fig.19c, higher recall signifies more drone-triggered events preserved, while higher precision indicates more background events removed. In indoor conditions, *CCT* achieves recalls over 86% for mmWave and 89% for event, with precision above 85% and 80%, respectively. In outdoor conditions, recall and precision remain above 80% for both types, demonstrating *CCT*'s effectiveness.

Performance of *GAJO*. We also compare the performance of different multi-modal fusion strategies. Specifically, we evaluate mmE-Loc against two widely used approaches: the extended Kalman filter (EKF) and Graph Optimization (GO). As shown in Fig.19d, 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 adaptive optimization method.
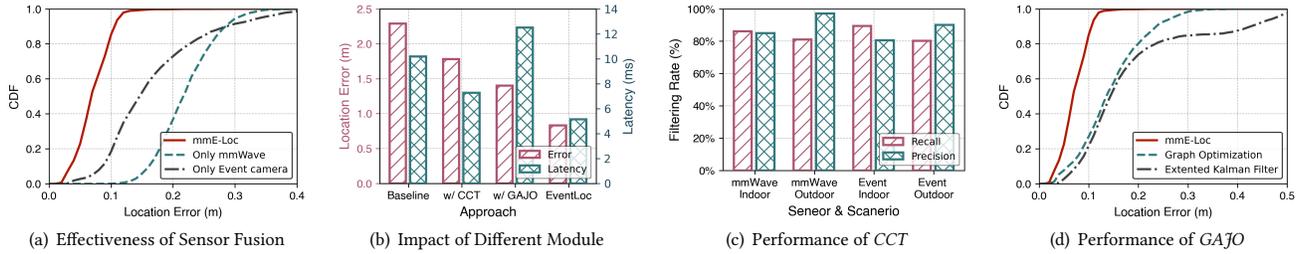
(a) Effectiveness of Sensor Fusion   (b) Impact of Different Module   (c) Performance of *CCT*   (d) Performance of *GAJO*
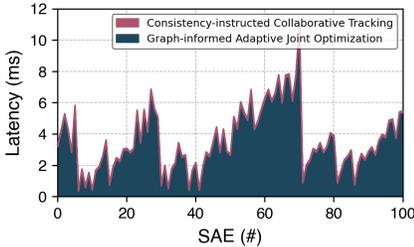
Figure 19: Ablation Study.
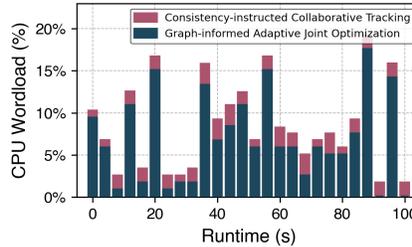

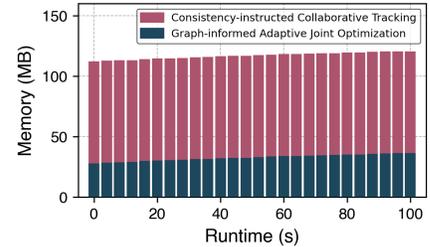**Figure 20: System Latency**   **Figure 21: CPU Workload**   **Figure 22: Memory Usage**

5.5 System Efficiency Study

The mmE-Loc distinguishes itself from existing models and learning-based methods due to its low latency and minimal resource overhead. Fig.20 illustrates the end-to-end latency (including delays from *CCT* and *GAJO* modules) throughout the localization process. The average end-to-end latency of mmE-Loc is around 3.58*ms*, with the *CCT* module contributing an average delay of 0.09*ms* and the *GAJO* module adding 3.49*ms*. During drone localization, mmE-Loc's latency may fluctuate due to the adaptive optimization method, which optimizes different sets of locations at different times. Nonetheless, mmE-Loc's latency remains suitable for use in flight control loops. Fig.21 and Fig.22 indicate that CPU usage doesn't exceed 20%, with memory usage under 120*MB*. Meanwhile, memory usage increases as the location set size grows.

6 CASE STUDY: DRONE AIRPORT

As shown in Fig.23a, to verify the system's usability, we conduct an experiment using a custom drone equipped with six propellers and managed by a PX4 flight controller. This drone is developed by a world-class delivery company exploring the feasibility of instant deliveries. The experiment takes place at a real-world delivery drone airport. To enable drone localization over a larger area, we employ an ARS548 mmWave radar, as depicted in Fig.23b. Fig.23c, Fig.23d, and Fig.23e illustrate the localization results as the drone follows a square spiral trajectory at an altitude of 30 *m*. The results show that in real-world scenarios, mmE-Loc achieves high localization precision, maintaining a maximum absolute location error error below 0.5*m* and relative location error error below 0.1*m*, and producing smooth trajectories that closely align with those of RTK. mmE-Loc has significant potential as a complementary system to RTK, aiding drone in challenging environments (e.g., urban canyons, where RTK may be compromised by signal blockage).

7 RELATED WORK

Drone ground localization. Several types of work have been proposed to assist drones in localization. (*i*) *Satellite-based systems.* The Global Positioning System (GPS) provides *m*-level accuracy outdoors [52, 53], while Real-Time Kinematic (RTK) achieves *cm*-level precision but is costly. However, these satellite-based systems struggle in urban canyons [2, 54–56]. (*ii*) *Optics-based systems.* These systems, such as motion capture, offer *cm*-level accuracy indoors but require precise calibration, making them impractical for outdoor use [57]. (*iii*) *Sensor-based systems.* To address these issues, various sensor-based techniques are proposed, including camera [58–60], radar [24, 25], LiDAR [61, 62] and acoustic [2] often combined with SLAM (Simultaneous Localization and Mapping) [63–66] or deep learning algorithms [67–71], aim to improve drone localization. However, limited spatio-temporal resolution in these sensors affects precise, low-latency landing drone localization. For example, cameras and LiDARs, with low frame rates (< 50*Hz*), may miss rapid movements during frames, reducing localization accuracy [72]. Acoustic signal-based systems are highly susceptible to environmental noise, with even nearby humans impacting their stability. Visual marker-based systems work in conjunction with downward-facing cameras mounted on drones. However, while these systems assist drones in obtaining their location, they do not help ground platforms track drones. They also are sensitive to lighting conditions due to limited dynamic range of frame cameras.

Compared to previous methods, mmE-Loc leverages a novel sensor configuration combining event camera with mmWave radar, harmonizing ultra-high sampling frequencies, to achieve superior drone ground localization accuracy with low latency. Meanwhile, mmE-Loc is resistant to lighting variations due to its sensor configuration, with event cameras offering a high dynamic range. It is important to note that mmE-Loc complements, rather than replaces, existing localization solutions. To ensure precise landing, mmE-Loc will work in conjunction with RTK and visual markers, providing a more reliable and accurate localization service.

mmWave for localization and tracking. Millimeter-wave is highly sensitive and more accurate due to its mm-level wavelength. mmWave radar offers high sensitivity and precision due to its sub-millimeter wavelength [22, 73, 74]. Several mmWave radar-based solutions for drone ground localization combine signal intensity methods, but face challenges in accurately tracking the drone's
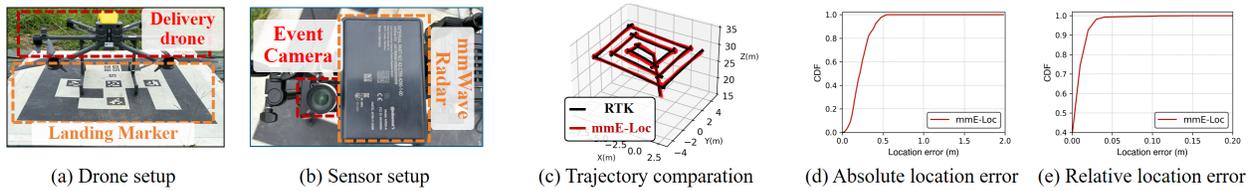
(a) Drone setup   (b) Sensor setup   (c) Trajectory comparison   (d) Absolute location error   (e) Relative location error

Figure 23: Case Study: Delivery Drone Airport.

center [44, 75, 76]. This difficulty arises from the drone's large size (e.g., 80 *cm* across), causing it to appear as a non-uniform blob in radar returns. Additionally, these methods often produce unstable results with frequent outliers, as multipath scattering can obscure the main signal and low spatial resolution of radar [77]. Other approaches leverage deep learning-based methods but require extensive pre-modeling and neural network training for each drone model [26, 78, 79]. These methods tend to struggle with tracking different drone models and perform poorly in environments not represented in training dataset. Several solutions integrate visual sensors to assist radar [51, 80, 81]. However, these approaches introduce latency due to exposure times and image processing delays.

To overcome the accuracy and latency bottlenecks, we upgrade frame cameras with event cameras to pair with mmWave radar, boosting system performance. By exploiting temporal consistency and drone's periodic micro-motion with *CCT* module, mmE-Loc achieves impressive tracking accuracy without the need for prior knowledge, (e.g., training data or 3D models). The integration of the *GAJO* module employing spatial complementarity and an adaptive optimization method further enables accurate drone ground localization and reduces latency to the *ms*-level.

Sensor Fusion Techniques. Sensor fusion techniques are widely used in localization [82–85]. *(i) Traditional pipeline fusion methods.* These approaches typically integrate dense point clouds from sensors (e.g., LiDAR) to provide depth information, alongside pixel data from frame cameras [86, 87]. However, these methods are generally limited to fusing two types of dense measurements and cannot directly accommodate the sparse data output from event cameras and mmWave radar. *(ii) Deep learning-based fusion methods.* Recent advancements have explored learning-based methods for fusing mmWave radar, frame cameras, and IMUs for localization [88, 89]. While these methods offer promising results, they often require extensive labeled training data and experience performance degradation in dynamic environments [26, 51, 90–92]. In contrast, mmE-Loc adopts a tightly coupled fusion framework, leveraging factor graphs and adaptive optimization to jointly optimize radar and event tracking models. This approach provides a clearer probabilistic interpretation compared to deep learning-based methods.

8 DISCUSSION

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 and accuracy 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 to manage simultaneous drone landings? As multiple drones land simultaneously, mmE-Loc will initialize multiple trackers within the event tracking model to track each drone, associate the results with radar tracking model, and subsequently perform localization and optimization for each drone individually.

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, airports should 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 [93].

How does strong light affect system performance? Lighting conditions primarily affect event cameras. Strong light has minimal impact on event cameras, as they detect only changes in light intensity. In contrast, weak illumination affects performance more significantly since intensity changes are less pronounced. Our experimental results on illumination effects (Fig. 12) confirm that weak illumination introduces greater errors than strong light.

9 CONCLUSION

This paper 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 performance and reduces latency by utilizing cross-modal *spatial complementarity*. Extensive evaluations conducted across various scenarios demonstrate performance of mmE-Loc.

10 ACKNOWLEDGEMENT

REFERENCES

[1] Hong kong can succeed in developing a thriving low-altitude economy. https://www.chinadailyhk.com/hk/article/593176.

[2] Weiguo Wang, Luca Mottola, Yuan He, Jinming Li, Yimiao Sun, Shuai Li, Hua Jing, and Yulei Wang. Micnest: Long-range instant acoustic localization of drones in precise landing. In *Proceedings of the 20th ACM SenSys*, pages 504–517, 2022.

[3] Xuecheng Chen, Haoyang Wang, Yuhan Cheng, Haohao Fu, Yuxuan Liu, Fan Dang, Yunhao Liu, Jinqiang Cui, and Xinlei Chen. Ddl: Empowering delivery drones with large-scale urban sensing capability. *IEEE Journal of Selected Topics in Signal Processing*, 2024.

[4] Xuecheng Chen, Haoyang Wang, Zuxin Li, Wenbo Ding, Fan Dang, Chenye Wu, and Xinlei Chen. Deliversense: Efficient delivery drone scheduling for crowdsensing with deep reinforcement learning. 2022.

[5] Jingao Xu, Hao Cao, Zheng Yang, Longfei Shangguan, Jialin Zhang, Xiaowu He, and Yunhao Liu. {SwarmMap}: Scaling up real-time collaborative visual {SLAM} at the edge. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 977–993, 2022.

[6] Zuxin Li, Fanhang Man, Xuecheng Chen, Susu Xu, Fan Dang, Xiao-Ping Zhang, and Xinlei Chen. Quest: Quality-informed multi-agent dispatching system for optimal mobile crowdsensing. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, pages 1811–1820. IEEE, 2024.

[7] Susu Xu, Xinlei Chen, Xidong Pi, Carlee Joe-Wong, Pei Zhang, and Hae Young Noh. Vehicle dispatching for sensing coverage optimization in mobile crowdsensing systems. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 311–312, 2019.

[8] Yuxuan Liu, Haoyang Wang, Fanhang Man, Jingao Xu, Fan Dang, Yunhao Liu, Xiao-Ping Zhang, and Xinlei Chen. Mobiair: Unleashing sensor mobility for city-scale and fine-grained air-quality monitoring with airbert. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, pages 223–236, 2024.

[9] Bin-Bin Zhang, Dongheng Zhang, Ruiyuan Song, Binquan Wang, Yang Hu, and Yan Chen. Rf-search: Searching unconscious victim in smoke scenes with rf-enabled drone. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

[10] Guoxuan Chi, Zheng Yang, Jingao Xu, Chenshu Wu, Jialin Zhang, Jianzhe Liang, and Yunhao Liu. Wi-drone: wi-fi-based 6-dof tracking for indoor drone flight control. In *Proceedings of the 20th ACM MobiSys*, pages 56–68, 2022.

[11] Xuecheng Chen, Zijian Xiao, Yuhan Cheng, ChenChun Hsia, Haoyang Wang, Jingao Xu, Susu Xu, Fan Dang, Xiao-Ping Zhang, Yunhao Liu, et al. Soscheduler: Toward proactive and adaptive wildfire suppression via multi-uav collaborative scheduling. *IEEE Internet of Things Journal*, 2024.

[12] Yuan He, Weiguo Wang, Luca Mottola, Shuai Li, Yimiao Sun, Jinming Li, Hua Jing, Ting Wang, and Yulei Wang. Acoustic localization system for precise drone landing. *IEEE Transactions on Mobile Computing*, 23(5):4126–4144, 2023.

[13] Yimiao Sun, Weiguo Wang, Luca Mottola, Ruijin Wang, and Yuan He. Aim: Acoustic inertial measurement for indoor drone localization and tracking. In *Proceedings of the 20th ACM SenSys*, pages 476–488, 2022.

[14] Javier González-Trejo, Diego Mercado-Ravell, Israel Becerra, and Rafael Murrieta-Cid. On the visual-based safe landing of uavs in populated areas: a crucial aspect for urban deployment. *IEEE Robotics and Automation Letters*, 6(4):7901–7908, 2021.

[15] Yiming Li, Markus Mund, Philipp Hoess, Joran Deschamps, Ulf Matti, Bianca Nijmeijer, Vilma Jimenez Sabinina, Jan Ellenberg, Ingmar Schoen, and Jonas Ries. Real-time 3d single-molecule localization using experimental point spread functions. *Nature methods*, 15(5):367–369, 2018.

[16] Stephen Xia, Minghui Zhao, Charuvahan Adhivarahan, Kaiyuan Hou, Yuyang Chen, Jingping Nie, Eugene Wu, Karthik Dantu, and Xiaofan Jiang. Anemoi: A low-cost sensorless indoor drone system for automatic mapping of 3d airflow fields. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2023.

[17] Haotian Zhang, Gaoang Wang, Zhichao Lei, and Jenq-Neng Hwang. Eye in the sky: Drone-based object tracking and 3d localization. In *Proceedings of the 27th ACM international conference on multimedia*, pages 899–907, 2019.

[18] Jinrui Zhang, Huan Yang, Ju Ren, Deyu Zhang, Bangwen He, Ting Cao, Yuanchun Li, Yaoxue Zhang, and Yunxin Liu. Mobidepth: Real-time depth estimation using on-device dual cameras. In *Proceedings of the 28th ACM MobiCom*, pages 528–541, 2022.

[19] Zhiyuan Xie, Xiaomin Ouyang, Li Pan, Wenrui Lu, Guoliang Xing, and Xiaoming Liu. Mozart: A mobile tof system for sensing in the dark through phase manipulation. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 163–176, 2023.

[20] Yunfan Zhang, Tim Scargill, Ashutosh Vaishnav, Gopika Premsankar, Mario Di Francesco, and Maria Gorlatova. Indepth: Real-time depth inpainting for mobile augmented reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(1), March 2022.

[21] Kaikai Deng, Dong Zhao, Qiaoyue Han, Shuyue Wang, Zihan Zhang, Anfu Zhou, and Huadong Ma. Geryon: Edge assisted real-time and robust object detection on drones via mmwave radar and camera fusion. *Proceedings of the ACM on*

[22] *Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–27, 2022.

[23] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A. Stankovic, Niki Trigoni, and Andrew Markham. See through smoke: Robust indoor mapping with low-cost mmwave radar, 2020.

[24] Jia Zhang, Xin Na, Rui Xi, Yimiao Sun, and Yuan He. mmhawkeye: Passive uav detection with a cots mmwave radar. In *2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 267–275. IEEE, 2023.

[25] Emerson Sie, Zikun Liu, and Deepak Vasisht. Batmobility: Towards flying without seeing for autonomous drones. In *Proceedings of the 29th ACM MobiCom*, pages 1–16, 2023.

[26] Tatsuya Iizuka, Takuya Sasatani, Toru Nakamura, Naoko Kosaka, Masaki Hisada, and Yoshihiro Kawahara. Millisign: mmwave-based passive signs for guiding uavs in poor visibility conditions. In *Proceedings of the 29th ACM MobiCom*, pages 1–15, 2023.

[27] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A Stankovic, Niki Trigoni, and Andrew Markham. See through smoke: robust indoor mapping with low-cost mmwave radar. In *Proceedings of the 18th ACM MobiSys*, pages 14–27, 2020.

[27] Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, Peijun Zhao, Yasin Almalioglu, Pedro PB De Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion. In *Proceedings of the 18th ACM SenSys*, pages 109–122, 2020.

[28] Zhuoqun Cheng, Richard West, and Craig Einstein. End-to-end analysis and design of a drone flight controller. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2404–2415, 2018.

[29] Emerson Sie, Zikun Liu, and Deepak Vasisht. Batmobility: Towards flying without seeing for autonomous drones. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '23, New York, NY, USA, 2023. Association for Computing Machinery.

[30] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

[31] Ciyu Ruan, Chenyu Zhao, Chenxin Liang, Xinyu Luo, Jingao Xu, and Xinlei Chen. Distill drops into data: Event-based rain-background decomposition network. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 2072–2077, 2024.

[32] Botao He, Ze Wang, Yuan Zhou, Jingxi Chen, Chahat Deep Singh, Haojia Li, Yuman Gao, Shaojie Shen, Kaiwei Wang, Yanjun Cao, et al. Microsaccade-inspired event camera for robotics. *Science Robotics*, 9(90):eadj8124, 2024.

[33] Jingao Xu, Danyang Li, Zheng Yang, Yishujie Zhao, Hao Cao, Yunhao Liu, and Longfei Shangguan. Taming event cameras with bio-inspired architecture and algorithm: A case for drone obstacle avoidance. In *Proceedings of the 29th ACM MobiCom*, pages 1–16, 2023.

[34] Xinyu Luo, Haoyang Wang, Ciyu Ruan, Chenxin Liang, Jingao Xu, and Xinlei Chen. Eventtracker: 3d localization and tracking of high-speed object with event and depth fusion. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 1974–1979, 2024.

[35] Zhongping Cao, Guangyu Mei, Xuemei Guo, and Guoli Wang. Virteach: mmwave radar point cloud based pose estimation with virtual data as a teacher. *IEEE Internet of Things Journal*, 2024.

[36] Hankai Liu, Xiulong Liu, Xin Xie, Xinyu Tong, and Keqiu Li. Pmtrack: Enabling personalized mmwave-based human tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4):1–30, 2024.

[37] Ziwei Wang, Yonhon Ng, Cedric Scheerlinck, and Robert Mahony. An asynchronous kalman filter for hybrid event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 448–457, 2021.

[38] Ignacio Alzugaray and Margarita Chli. Asynchronous corner detection and tracking for event cameras in real time. *IEEE Robotics and Automation Letters*, 3(4):3177–3184, 2018.

[39] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020.

[40] Peijun Zhao, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, and Andrew Markham. 3d motion capture of an unmodified drone with single-chip millimeter wave radar. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5186–5192. IEEE, 2021.

[41] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.

[42] Yingqi Wang, Zhongqin Wang, J Andrew Zhang, Haimin Zhang, and Min Xu. Vital sign monitoring in dynamic environment via mmwave radar and camera fusion. *IEEE Transactions on Mobile Computing*, 2023.

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

[44] Kun Qian, Zhaoyuan He, and Xinyu Zhang. 3d point cloud generation with millimeter-wave radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4):1–23, 2020.

[45] Guidong Zhang, Guoxuan Chi, Yi Zhang, Xuan Ding, and Zheng Yang. Push the limit of millimeter-wave radar localization. *ACM Transactions on Sensor Networks*, 19(3):1–21, 2023.

[46] Shijie Lin, Fang Xu, Xuhong Wang, Wen Yang, and Lei Yu. Efficient spatial-temporal normalization of sae representation for event camera. *IEEE Robotics and Automation Letters*, 5(3):4265–4272, 2020.

[47] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[48] Christian H Bischof and Gregorio Quintana-Ortí. Computing rank-revealing qr factorizations of dense matrices. *ACM Transactions on Mathematical Software (TOMS)*, 24(2):226–253, 1998.

[49] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[50] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[51] Xian Shuai, Yulin Shen, Yi Tang, Shuyao Shi, Luping Ji, and Guoliang Xing. millieye: A lightweight mmwave radar and camera fusion system for robust object detection. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, pages 145–157, 2021.

[52] Yunfan Wang, Steve Young, Demba Komma, Jaechan Lim, Zhen Feng, Zichen Fan, Chien-wei Tseng, Hun-Seok Kim, and David Blaauw. Global localization of energy-constrained miniature rf emitters using low earth orbit satellites. In *Proceedings of the 21st ACM SenSys*, pages 403–416, 2023.

[53] Huixin Dong, Yirong Xie, Xianan Zhang, Wei Wang, Xinyu Zhang, and Jianhua He. Gpsmirror: Expanding accurate gps positioning to shadowed and indoor regions with backscatter. In *Proceedings of the 29th ACM MobiCom*, pages 1–15, 2023.

[54] Jingao Xu, Hao Cao, Danyang Li, Kehong Huang, Chen Qian, Longfei Shangguan, and Zheng Yang. Edge assisted mobile semantic visual slam. In *Proceedings of the IEEE INFOCOM*, pages 1828–1837. IEEE, 2020.

[55] Haoyang Wang, Jingao Xu, Chenyu Zhao, Zihong Lu, Yuhan Cheng, Xuecheng Chen, Xiao-Ping Zhang, Yunhao Liu, and Xinlei Chen. Transformloc: Transforming mavs into mobile localization infrastructures in heterogeneous swarms. *Proceedings of the IEEE INFOCOM*, 2024.

[56] Haoyang Wang, Xuecheng Chen, Yuhan Cheng, Chenye Wu, Fan Dang, and Xinlei Chen. H-swarmloc: efficient scheduling for localization of heterogeneous mav swarm with deep reinforcement learning. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 1148–1154, 2022.

[57] Hongfei Xue, Qiming Cao, Yan Ju, Haochen Hu, Haoyu Wang, Aidong Zhang, and Lu Su. M4esh: mmwave-based 3d human mesh construction for multiple subjects. In *Proceedings of the 20th ACM SenSys*, pages 391–406, 2022.

[58] Yuze He, Li Ma, Jiahe Cui, Zhenyu Yan, Guoliang Xing, Sen Wang, Qintao Hu, and Chen Pan. Automatch: Leveraging traffic camera to improve perception and localization of autonomous vehicles. In *Proceedings of the 20th ACM SenSys*, pages 16–30, 2022.

[59] Ali J Ben Ali, Marziye Kouroshli, Sofiya Semenova, Zakieh Sadat Hashemifar, Steven Y Ko, and Karthik Dantu. Edge-slam: Edge-assisted visual simultaneous localization and mapping. *ACM Transactions on Embedded Computing Systems*, 22(1):1–31, 2022.

[60] Zaid Tasneem, Charuvahan Adhivarahan, Dingkang Wang, Huikai Xie, Karthik Dantu, and Sanjeev J Koppal. Adaptive fovea for scanning depth sensors. *The International Journal of Robotics Research*, 39(7):837–855, 2020.

[61] Jiahe Cui, Shuyao Shi, Yuze He, Jianwei Niu, Guoliang Xing, and Zhenchao Ouyang. {VILAM}: Infrastructure-assisted 3d visual localization and mapping for autonomous driving. In *21st USENIX NSDI 24*, pages 1831–1845, 2024.

[62] Zhuozhu Jian, Qixuan Li, Shengtao Zheng, Xueqian Wang, and Xinlei Chen. Lvcp: Lidar-vision tightly coupled collaborative real-time relative positioning. *arXiv preprint arXiv:2407.10782*, 2024.

[63] Xinlei Chen, Carlos Ruiz, Sihan Zeng, Liyao Gao, Aveek Purohit, Stefano Carpin, and Pei Zhang. H-drunkwalk: Collaborative and adaptive navigation for heterogeneous mav swarm. *ACM Transactions on Sensor Networks (TOSN)*, 16(2):1–27, 2020.

[64] Xinlei Chen, Aveek Purohit, Shijia Pan, Carlos Ruiz, Jun Han, Zheng Sun, Frank Mokaya, Patric Tague, and Pei Zhang. Design experiences in minimalistic flying sensor node platform through sensorfly. *ACM Transactions on Sensor Networks (TOSN)*, 13(4):1–37, 2017.

[65] Ying Chen, Hazer Inaltekin, and Maria Gorlatova. Adaptslam: Edge-assisted adaptive slam with resource constraints via uncertainty minimization. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.

[66] Xinlei Chen, Aveek Purohit, Carlos Ruiz Dominguez, Stefano Carpin, and Pei Zhang. Drunkwalk: Collaborative and adaptive planning for navigation of micro-aerial sensor swarms. In *Proceedings of the 13th ACM SenSys*, pages 295–308, 2015.

[67] Chenyu Zhao, Ciyu Ruan, Jingao Xu, Haoyang Wang, Shengbo Wang, Jiaqi Li, Jirong Zha, Zheng Yang, Yunhao Liu, Xiao-Ping Zhang, et al. Foes or friends: Embracing ground effect for edge detection on lightweight drones. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 1377–1392, 2024.

[68] Chenyu Zhao, Haoyang Wang, Jiaqi Li, Fanhang Man, Shilong Mu, Wenbo Ding, Xiao-Ping Zhang, and Xinlei Chen. Smoothlander: A quadrotor landing control system with smooth trajectory guarantee based on reinforcement learning. In *Proceedings of the 2023 ACM UbiComp*, pages 682–687, 2023.

[69] Tianyi Hu, Tim Scargill, Fan Yang, Ying Chen, Guohao Lan, and Maria Gorlatova. Seesys: Online pose error estimation system for visual slam. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, pages 322–335, 2024.

[70] Xiaopeng Zhao, Guosheng Wang, Zhenlin An, Qingrui Pan, and Lei Yang. Understanding localization by a tailored gpt. In *Proceedings of the 22nd ACM MobiSys*, pages 318–330, 2024.

[71] Yimiao Sun, Yuan He, Jiacheng Zhang, Xin Na, Yande Chen, Weiguo Wang, and Xiuzhen Guo. Bifrost: Reinventing wifi signals based on dispersion effect for accurate indoor localization. In *Proceedings of the 21th ACM SenSys*. ACM, 2023.

[72] Zhuozhu Jian, Zejia Liu, Haoyu Shao, Xueqian Wang, Xinlei Chen, and Bin Liang. Path generation for wheeled robots autonomous navigation on vegetated terrain. *IEEE Robotics and Automation Letters*, 2023.

[73] Kyle Harlow, Hyesu Jang, Timothy D. Barfoot, Ayoung Kim, and Christoffer Heckman. A new wave in robotics: Survey on recent mmwave radar applications in robotics. *IEEE Transactions on Robotics*, 40:4544–4560, 2024.

[74] A. Soumya, C. Krishna Mohan, and Linga Reddy Cenkeramaddi. Recent advances in mmwave-radar-based sensing, its applications, and machine learning techniques: A review. *Sensors*, 23(21), 2023.

[75] Suhare Solaiman, Emad Alsuwat, and Rajwa Alharthi. Simultaneous tracking and recognizing drone targets with millimeter-wave radar and convolutional neural network. *Applied System Innovation*, 6(4), 2023.

[76] Heba Abdelnasser, Mohammad Heggo, Oscar Pang, Mirko Kovac, and Julie A. McCann. Radro: Indoor drone tracking using millimeter wave radar. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8(3), September 2024.

[77] Yadong Li, Dongheng Zhang, Ruixu Geng, Zhi Lu, Zhi Wu, Yang Hu, Qibin Sun, and Yan Chen. A high-resolution handheld millimeter-wave imaging system with phase error estimation and compensation. *Communications Engineering*, 3, 2024.

[78] Kai Zheng, Kun Qian, Timothy Woodford, and Xinyu Zhang. Neuroradar: A neuromorphic radar sensor for low-power iot systems. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, pages 223–236, 2023.

[79] Nader Al-lQubaydhi, Abdulrahman Alenezi, Turki Alanazi, Abdulrahman Senyor, Naif Alanezi, Bandar Alotaibi, Munif Alotaibi, Abdul Razaque, and Salim Hariri. Deep learning for unmanned aerial vehicles detection: A review. *Computer Science Review*, 51:100614, 2024.

[80] Simon Chadwick, Will Maddern, and Paul Newman. Distant vehicle detection using radar and vision. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8311–8317. IEEE, 2019.

[81] Hyunggi Cho, Young-Woo Seo, BVK Vijaya Kumar, and Ragunathan Raj Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 1836–1843. IEEE, 2014.

[82] Danyang Li, Jingao Xu, Zheng Yang, Qian Zhang, Qiang Ma, Li Zhang, and Pengpeng Chen. Motion inspires notion: self-supervised visual-lidar fusion for environment depth estimation. In *Proceedings of the 20th ACM MobiSys*, pages 114–127, 2022.

[83] Chenggao Li, Qianyi Huang, Yuxuan Zhou, Yandao Huang, Qingyong Hu, Huangxun Chen, and Qian Zhang. Riscan: Ris-aided multi-user indoor localization using cots wi-fi. In *Proceedings of the 21st ACM SenSys*, pages 445–458, 2023.

[84] Weiguo Wang, Yuan He, Meng Jin, Yimiao Sun, and Xiuzhen Guo. Meta-speaker: Acoustic source projection by exploiting air nonlinearity. In *Proceedings of the 29th ACM MobiCom*, pages 1–15, 2023.

[85] Hongfei Xue, Qiming Cao, Chenglin Miao, Yan Ju, Haochen Hu, Aidong Zhang, and Lu Su. Towards generalized mmwave-based human pose estimation through signal augmentation. In *Proceedings of the 29th ACM MobiCom*, pages 1–15, 2023.

[86] Yuze He, Chen Bian, Jingfei Xia, Shuyao Shi, Zhenyu Yan, Qun Song, and Guoliang Xing. Vi-map: Infrastructure-assisted real-time hd mapping for autonomous driving. In *Proceedings of the 29th ACM MobiCom*, pages 1–15, 2023.

[87] Shuyao Shi, Neiwen Ling, Zhehao Jiang, Xuan Huang, Yuze He, Xiaoguang Zhao, Bufang Yang, Chen Bian, Jingfei Xia, Zhenyu Yan, et al. Soar: Design and deployment of a smart roadside infrastructure system for autonomous driving. In *Proceedings of the 30th ACM MobiCom*, pages 139–154, 2024.

[88] Jingao Xu, Guoxuan Chi, Zheng Yang, Danyang Li, Qian Zhang, Qiang Ma, and Xin Miao. Followupar: Enabling follow-up effects in mobile ar applications. In *Proceedings of the 19th ACM MobiSys*, pages 1–13, 2021.

[89] Ali Safa, Tim Verbelen, Ilja Ocket, André Bourdoux, Hichem Sahli, Francky Catthoor, and Georges Gielen. Fusing event-based camera and radar for slam using spiking neural networks with continual stdp learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2782–2788. IEEE, 2023.

[90] Rong Ding, Haiming Jin, Jianrong Ding, Xiaocheng Wang, Guiyun Fan, Fengyuan Zhu, Xiaohua Tian, and Linghe Kong. Push the limit of single-chip mmwave radar-based egomotion estimation with moving objects in fov. In *Proceedings of the 21st ACM SenSys*, pages 417–430, 2023.

[91] Wenwei Li, Ruofeng Liu, Shuai Wang, Dongjiang Cao, and Wenchao Jiang. Egocentric human pose estimation using head-mounted mmwave radar. In *Proceedings of the 21st ACM SenSys*, pages 431–444, 2023.

[92] Leon Müller, Manolis Sifalakis, Sherif Eissa, Amirreza Yousefzadeh, Paul Detterer, Sander Stuijk, and Federico Corradi. Aircraft marshaling signals dataset of fmcw radar and event-based camera for sensor fusion. In *2023 IEEE Radar Conference (RadarConf23)*, pages 01–06. IEEE, 2023.

[93] Yanxiang Wang, Yiran Shen, Kenuo Xu, Mahbub Hassan, Guangrong Zhao, Chenren Xu, and Wen Hu. Towards high-speed passive visible light communication with event cameras and digital micro-mirrors. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, pages 704–717, 2024.