

Radar and Event Camera Fusion for Agile Robot Ego-Motion Estimation

Yang Lyu, *Member, IEEE*, Zhenghao Zou, Yanfeng Li, Xiaohu Guo, Chunhui Zhao, Quan Pan, *Member, IEEE*

Abstract—Achieving reliable ego motion estimation for agile robots, e.g., aerobatic aircraft, remains challenging because most robot sensors fail to respond timely and clearly to highly dynamic robot motions, often resulting in measurement blurring, distortion, and delays. In this paper, we propose an IMU-free and feature-association-free framework to achieve aggressive ego-motion velocity estimation of a robot platform in highly dynamic scenarios by combining two types of exteroceptive sensors, an event camera and a millimeter wave radar. First, we used instantaneous raw events and Doppler measurements to derive rotational and translational velocities directly. Without a sophisticated association process between measurement frames, the proposed method is more robust in texture-less and structureless environments and is more computationally efficient for edge computing devices. Then, in the back-end, we propose a continuous-time state-space model to fuse the hybrid time-based and event-based measurements to estimate the ego-motion velocity in a fixed-lagged smoother fashion. In the end, we validate our velometer framework extensively in self-collected experiment datasets featured by aggressive motion and HDR light conditions. The results indicate that our IMU-free and association-free ego motion estimation framework can achieve reliable and efficient velocity output in challenging environments. The source code, illustrative video and dataset are available at <https://github.com/ZzhYgwh/TwistEstimator>.

Index Terms—Doppler radar, event camera, ego-motion estimation.

I. INTRODUCTION

Reliable ego-motion estimation is fundamental to autonomous robotic platforms. Early solutions rely on GNSS/INS, while more recent SLAM-based methods integrate diverse sensors such as cameras, LiDARs, and radars, making them more adaptable and widely applicable. Successful deployments of SLAM-based approaches on various platforms utilize combinations of sensors such as cameras, LiDARs and IMUs, leveraging their measurements to fully resolve all degrees of freedom in platform’s pose estimation. Nevertheless, highly agile robotic systems, such as aerobatic UAVs, racing UGVs, and jointed robots demand fast and accurate velocity-state feedback to maintain stability, enable aggressive motion control [?], and support timely decision-making under high-dynamic conditions [?]. Real-time velocity feedback can significantly reduce state update latency, for example, within a UAV flight control stack[?] or motion

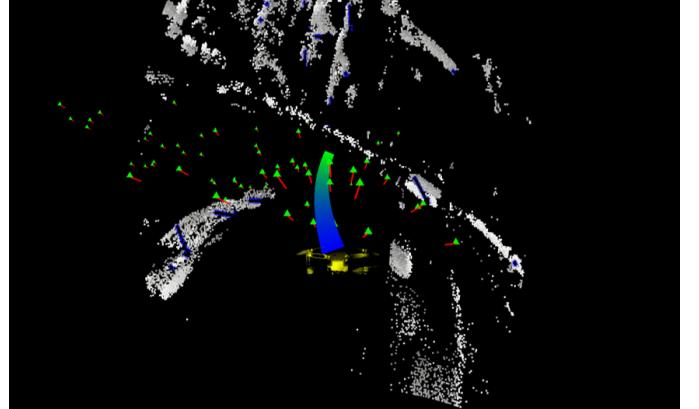


Fig. 1: Robot Ego-Motion Estimation. The proposed pipeline estimates the 6-DoF instantaneous velocity of a rapidly moving robot by fusing multi-point MMWave Radar Doppler measurements—including anchors (Green) and Doppler (Red) with sparse event-based normal optical flow (Blue) computed on the SAE (White). By integrating the estimated velocities, the 0.5-second motion trajectory is reconstructed as a Color-Gradient Line(Blue for near, Green for far).

planning [?]. Conventional perception sensors typically fail to capture instantaneous velocity, and odom-based or SLAM-based methods suffer from substantial latency [?]. Even when combined with high-frequency inertial data, these approaches cannot ensure an immediate response to rapid state changes. Therefore, real-time velocity estimation is crucial for robust closed-loop control and reactive, decision-oriented autonomy in highly dynamic robotic systems.

Cameras capture images over an exposure period, and most LiDARs perform ranging sequentially over time. For platforms with high translational and rotational velocities, motion during sensing cannot be neglected, as it causes blur and distortion in the measurements. In contrast, recently developed event cameras asynchronously output pixel measurements by generating events in response to changes in light intensity, providing motion-sensitive signals for ego-motion estimation. Moreover, event measurements inherently encode velocity information in the image domain, establishing a direct and natural link to instantaneous velocity estimation[?]. However, event cameras alone lack metric scale and cannot fully recover the 6-DoF motion of a mobile platform.

In this paper, we aim to develop a framework that combines an event camera with a complementary 4D millimeter wave radar, rather than an IMU, to estimate the ego-motion

Yang Lyu, Zhenghao Zou, Yanfeng Li, Xiaohu Guo, Chunhui Zhao, Quan Pan are with the School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi, 710129 P.R. China. e-mail: lyu.yang@nwpu.edu.cn.

This work was supported by the National Natural Science Foundation of China under Grant 62203358, Grant 62233014, and Grant 62073264. (Corresponding author: Yang Lyu)

velocity of an agile moving robot, as illustrated in Fig. 1. Specifically, we use the instantaneous Doppler measurement to provide the metric information. A detailed pipeline of our proposed method is presented in Fig. 2. To our knowledge, this work is the first ego-motion estimation attempt based on such a sensor setup. By leveraging the complementary and instantaneous measurement characteristics of the sensor setup, this framework avoids the need for computationally expensive frame-to-frame feature matching and therefore improve the efficiency and robustness in challenging sensing environments. Moreover, the velocity from radar Doppler is considered drift-free compared to that from IMU measurement, and therefore can provide more accurate velocity, even during long-term estimation.

A key consideration in our sensor setup is the combination of low-rate radar Doppler measurements with high-rate event-based camera data. While radar Doppler signals are nominally low-frequency, in practice, their incremental constraints on translational velocity are sufficient, since translational motion typically evolves more smoothly than rotational motion in most agile scenarios. High-frequency event-based measurements provide dense temporal constraints on motion, particularly for angular changes, which complements the radar's contributions. Moreover, our continuous-time state parameterization inherently accommodates asynchronous measurement streams without requiring explicit time alignment. This design mitigates potential latency or computational imbalance issues associated with naive discrete-time fusion, and ensures that the hybrid sensor setup delivers accurate 6-DOF velocity estimates at high frequency. Consequently, explicit synchronization of radar and event measurements is unnecessary, and the proposed method maintains reliable real-time performance under high-dynamic conditions. The contributions are as follows:

- We develop a lightweight 3D ego-motion estimation front-end that directly derives instantaneous and drift-free metric linear and angular velocities from event-based pixel-level dynamics and the Doppler effect of radar point clouds, and avoids frame-to-frame associations. These associations are highly susceptible to failure in scenarios where texture and structural features are insignificant, as well as during high-maneuver movements.
- We develop a continuous-time back-end to support the fusion of the high-rate asynchronous event-based measurement and the low-rate periodic radar measurements, as well as other potential ego-motion measurements (e.g. IMU measurements), to generate high-speed motion estimation that matches high-speed agile maneuvers.
- We evaluate our proposed method on various self-collected data sequences, including both aggressive motions and HDR situations, and compare it against several state-of-the-art approaches, further demonstrating the advantages of our sensor setup and method.

The remainder of the paper is organized as follows. Section II provides a review of related literature. The association-free and IMU-free velocity estimation front-end is derived in Section III. Section IV formulates the continuous-time ego-motion estimator. Validations of the proposed framework are

provided in Section V. Section VI concludes the paper.

II. RELATED WORKS

A. Event-camera based ego-motion estimation

Event cameras provide high temporal resolution, wide dynamic range, and low power use, making them well-suited for vSLAM in challenging settings.

Event-based odometry methods can be broadly categorized by how event data is processed. Some works, such as [?] and [?], convert asynchronous events into event frames within spatio-temporal windows for feature detection and tracking. To enhance geometric constraints, line features are further explored in [?], [?], while [?] integrates line features with IMU data to achieve high-frequency motion estimation.

Some works aim to directly estimate motion from events, bypassing traditional feature detection and tracking. A simple approach generates event frames through spatio-temporal alignment [?]. More advanced methods, like [?], tightly fuse events and frames using EGM and PBA.

With deep learning's rise in computer vision, applying it to event camera SLAM is promising. Event data is usually converted into frame-like forms for CNN processing. [?] introduces an unsupervised method that encodes events as temporal volumes and estimates optical flow, ego-motion, and depth from motion blur. In a supervised approach, [?] presents a network that fuses asynchronous events and monocular images for continuous-time depth and motion estimation. DEVO[?] advances event-based odometry but depends on the quality of trained event patches for accurate motion estimation.

Building on multi-sensor fusion, [?] combines event and RGB-D data to improve pose estimation for agile-legged robots under dynamic maneuvers, and [?] demonstrates that tightly coupling LiDAR, IMU, event, and standard camera measurements enables real-time and robust odometry in challenging conditions. These methods generally leverage the dynamic nature of event data to enhance tracking robustness and accuracy. PL-EVIO[?] achieves robust, real-time state estimation by tightly fusing asynchronous event streams, standard images, and inertial measurements.

Although these methods show strong performance, their reliance on frame-to-frame association can be computationally demanding for onboard systems with limited resources. In contrast, due to its sensing mechanism, an event camera provides instantaneous motion cues. Thus, we explore deriving instantaneous velocity directly from event data.

B. Radar ego-motion estimation

A mmwave radar can provide two types of information to achieve ego-motion estimation.

First, relative transformations can be obtained by registering radar point clouds between frames or with a map. A radar SLAM system in [?] shows strong performance in all-weather conditions. Similarly, [?] proposes an unsupervised method for feature detection and tracking, followed by odometry estimation. [?] introduces a radar-inertial odometry (RIO) system with online extrinsic calibration for robust localization. [?] presents a continuous-time radar-inertial framework that

fuses IMU data with spinning radar and models radar point uncertainty to improve real-time accuracy and robustness, and [?] proposes a 4D radar odometry method leveraging Doppler and RCS with weighted scan-to-submap matching for IMU-free ego-motion estimation. However, these point cloud registration front-ends often incur high computational costs, posing challenges for resource-constrained robotic platforms.

Radar uniquely offers instantaneous radial velocity through the Doppler effect, enabling direct ego-motion estimation. Compared to the IMU, the velocity from the Doppler effect can be treated as a drift-free measurement, even in long-term sensing environments. Studies[?], [?] estimate vehicle velocity from one or multiple radars, while [?] incorporates a 3D velocity factor in pose graph optimization. These methods reduce computation by avoiding point cloud pre-processing but can face error accumulation during pose estimation and have limited angular velocity observability.

Recent radar-based SLAM frameworks combine point and velocity information for robust pose estimation. [?] formulates frame-to-map registration and Doppler velocity as pose and velocity prior factors. Similar approaches are adopted in [?] and [?], using different velocity integration strategies. A related framework, [?], fuses Doppler LiDAR, IMU, and velocity in a graph-based optimization.

In this paper, we aim to achieve ego-motion with limited computation and storage resources by leveraging Doppler velocity. Specifically, we consider removing the angular unobservability by combining an event camera. With the two instantaneous measurements, registration-free 6-DOF velocities can be recovered directly without the aid of an inertial measurement unit (IMU).

C. Continuous-time Representation

Continuous-time (CT) based localization has gained popularity, especially with multi-sensor fusion becoming standard. Furgale *et al.*[?] first introduced representing trajectories as Gaussian bases in a CT SLAM framework. Subsequent works extended this to various sensor setups. For example, [?] presents a CT SLAM with asynchronous stereo-inertial sensors, avoiding IMU pre-integration via spline-based trajectories. A LiDAR-only odometry using CT formulation is proposed in [?], enabling pose estimation during aggressive motions by processing high-frequency streaming LiDAR points without motion compensation. Closely related to our work, [?] fuses asynchronous event camera data with IMU in a CT SLAM, effectively handling high-rate measurements for dynamic platforms. Additionally, CT SLAM frameworks like [?] support multi-sensor fusion (LiDAR, camera, IMU) with online time-offset estimation, demonstrating CT's flexibility in handling complex sensor fusion tasks.

In this paper, we plan to achieve ego-motion estimation based on the fusion of an event camera and a Doppler radar, which outputs high-frequency 6-DOF velocities for an agile robot. However, the sampling rate of the radar Doppler measurement is much lower than the events, and may not match the agility of a robot's motion pattern. Thanks to the nature of handling asynchronous measurement of the CT-SLAM, we can still fuse them in a CT fashion, and the status

parameterization method allows us to output high-frequency estimates.

D. Notations

In this paper, we use lowercase and uppercase bold letters to represent vectors and matrices, respectively. Time in continuous- and discrete-time is denoted by $t \in \mathbb{R}^+ \cup \{0\}$ and $k \in \mathbb{Z}$. Specifically, we use $(\cdot)(t)$ and $(\cdot)^{(k)}$ to represent variables in the continuous-time domain and the discrete time, respectively. In addition, we use calligraphic font letters to denote variables in different frames. We use \mathcal{G} to represent the global frame, and \mathcal{R} and \mathcal{E} are the radar frame and camera frame respectively. For example, ${}^{\mathcal{G}}\mathbf{p}_r(t)$ denotes the position of the radar in the global frame at time t , and ${}^{\mathcal{R}}\boldsymbol{\omega}_r^k$ is the radar's rotation velocity at time instance k .

III. VELOCITY FRONT-ENDS

In this section, the ego-motion front end is provided. Specifically, we derive linear and angular velocity from a 4D MMWR and an event camera, respectively. The coordinate system and system setup are illustrated in Fig.3.

A. Linear Velocity from 4D Radar

The 3D ego-motion velocity of a 4D millimeter-wave radar in its local frame can be obtained based on the position of the point clouds and their Doppler velocities.

Given one radar point cloud frame with point set as C , the coordinate of a static 3D world point $i \in C$ in the radar local frame $\{\mathcal{R}\}$ is defined as ${}^{\mathcal{R}}\mathbf{p}_i \in \mathbb{R}^3$, the corresponding Doppler velocity measurement is defined as

$$v_i = -\frac{{}^{\mathcal{R}}\mathbf{p}_i^\top}{{}^{\mathcal{R}}\|\mathbf{p}_i\|} {}^{\mathcal{R}}\mathbf{v}_r, \quad (1)$$

where ${}^{\mathcal{R}}\mathbf{v}_r$ is the ego-motion velocity of the radar in its local frame. Given $n \geq 3$ valid points, the ego-motion velocity ${}^{\mathcal{R}}\mathbf{v}_r$ can be estimated in a least-square manner with

$$\mathbf{H}^{\mathcal{R}}\mathbf{v}_r = \mathbf{v}_d, \quad (2)$$

where $\mathbf{H} \triangleq \begin{bmatrix} {}^{\mathcal{R}}\mathbf{p}_1 & {}^{\mathcal{R}}\mathbf{p}_2 & \cdots & {}^{\mathcal{R}}\mathbf{p}_n \end{bmatrix}^\top$, and $\mathbf{v}_d \triangleq [-v_1 \ -v_2 \ \cdots \ -v_n]^\top$ is the stacked Doppler velocities of all points in a frame.

Practically, the velocity accuracy of the above estimation is often degraded due to outliers from moving objects or clutter. To mitigate the effects of Doppler measurement noise, we apply RANSAC-based outlier rejection. Furthermore, we need to evaluate the uncertainties of each dimension based on the distribution of valid points in each frame, as below:

$$\Sigma_{\mathbf{v}} = \sigma^2 (\mathbf{H}^\top \mathbf{H})^{-1} + (\mathbf{H}^\top \mathbf{H})^{-1} \left(\sum_i J_i^\top \Sigma_{p,i} J_i \right) (\mathbf{H}^\top \mathbf{H})^{-1}, \quad (3)$$

where $J_i = \partial \mathbf{h}_i / \partial \mathbf{p}_i$, and $\mathbf{h}_i = -\mathbf{p}_i / \|\mathbf{p}_i\|$. $\Sigma_{p,i}$ is the position covariance of each point calculated from the noise variances of azimuth angle, elevation angle, and range measurements of each point i .

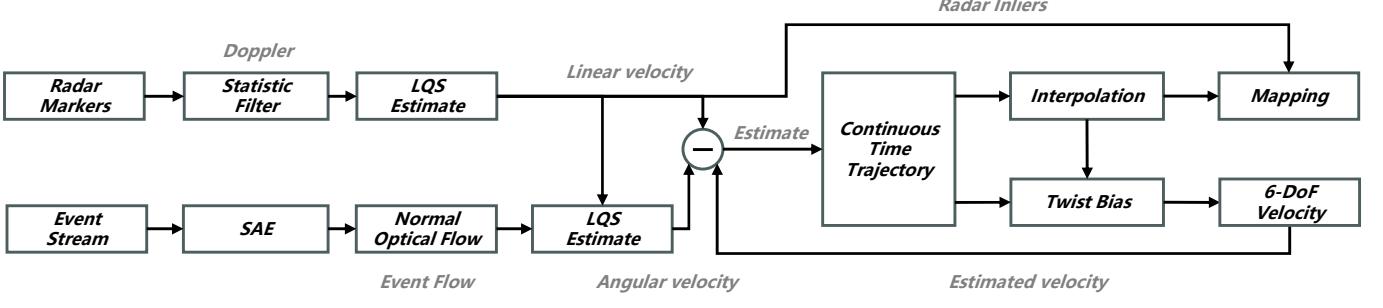


Fig. 2: The proposed ego-motion estimation pipeline.

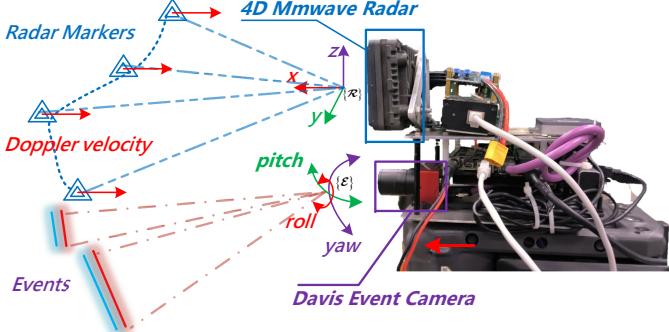


Fig. 3: Coordinate Systems.

B. Angular Velocity from Event Camera

In this section, we use instantaneous event camera data to derive the angular velocity. First, we obtain normal flow directly from raw events, and then we derive the angular velocity based on a continuous-time epipolar constraint.

1) *Surface of Active Events (SAE)*: Event cameras produce asynchronous events $\mathbf{e}_i = (x_i, y_i, t_i, p_i)$ when pixel brightness changes, where (x_i, y_i) is the pixel location, t_i is the timestamp, and $p_i \in \{-1, +1\}$ is the polarity. To compactly represent recent activity, we construct a *Surface of Active Events* (SAE), a 2D map $T \in \mathbb{R}^{H \times W}$, where each pixel stores the latest event timestamp:

$$T(x, y) = \max_{i \in \mathcal{E}_{x,y}} t_i, \quad (4)$$

with $\mathcal{E}_{x,y} = \{i \mid (x_i, y_i) = (x, y)\}$ the set of events at pixel (x, y) . Each incoming event \mathbf{e}_i updates the SAE at its location:

$$T(x_i, y_i) \leftarrow t_i, \quad (5)$$

providing a continuously updated temporal surface for motion estimation. To approximate the true velocity, SAE is constructed using events within a short temporal window of 30–50 ms.

2) *Normal Flow on SAE*: According to [?], only normal flow, which is defined as the part of optical flow parallel to the image gradient, can be recovered from the SAE. To obtain the full optical flow, we assume a patch of pixel area in SAE shares a common optical flow, and there are multiple edges that can be detected within a short time period. According to the last subsection for SAE, $T(\mathbf{u}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ where $\mathbf{u} \in \mathbb{R}^2$ denotes the position of the pixel on the image plane. The function returns the time stamp of the latest report event of a specific

pixel \mathbf{u} . When an edge moves with a constant normal velocity $v_n \in \mathbb{R}^+$, a simple model of $T(\mathbf{u})$ can be defined as

$$T(\mathbf{u}) = t - \frac{d(\mathbf{u})}{v_n}, \quad (6)$$

where t is the current time, and $d(\mathbf{u}) = \mathbf{u}^\top \mathbf{n}$ is the signed distance of a point \mathbf{u} from the edge. \mathbf{n} is a unit vector normal to the edge and pointing in the direction of increasing time.

Taking the gradient of $T(\mathbf{u})$, we have

$$\nabla T(\mathbf{u}) = -\frac{1}{v_n} \nabla d(\mathbf{u}) = -\frac{\mathbf{n}}{v_n}. \quad (7)$$

Then taking the norm of both sides,

$$\|\nabla T\| = \frac{1}{v_n} \implies \|\dot{\mathbf{u}}\| = v_n = \frac{1}{\|\nabla T\|}. \quad (8)$$

Given the raw events, ∇T can be obtained by fitting a local spatio-temporal plane on SAE and then calculating its gradient [?]. Based on the definition of normal flow, we have

$$\frac{\nabla T_1}{\|\nabla T_1\|} \dot{\mathbf{u}} = \frac{1}{\|\nabla T\|}. \quad (9)$$

Given more than two edges within a patch, a local optical flow can be obtained in a least-square fashion based on the following equations

$$\begin{bmatrix} \frac{\nabla T_1}{\|\nabla T_1\|} \\ \frac{\nabla T_2}{\|\nabla T_2\|} \\ \vdots \\ \frac{\nabla T_m}{\|\nabla T_m\|} \end{bmatrix} \dot{\mathbf{u}} = \begin{bmatrix} \frac{1}{\|\nabla T_1\|} \\ \frac{1}{\|\nabla T_2\|} \\ \vdots \\ \frac{1}{\|\nabla T_m\|} \end{bmatrix}. \quad (10)$$

3) *Angular Velocity from Normal Flow*: The linear and angular velocity of the event camera in its local frame $\{\mathbf{E}\}$ are defined as ${}^E\mathbf{v}_e \in \mathbb{R}^3$ and ${}^E\omega_e \in \mathbb{R}^3$ respectively. Given a static landmark point in the camera frame ${}^E\mathbf{p}_l \in \mathbb{R}^3$, the following equation holds:

$${}^E\dot{\mathbf{p}}_l = [{}^E\omega_e]_x {}^E\mathbf{p}_l + {}^E\mathbf{v}_e, \quad (11)$$

where $[{}^E\omega_e]_x$ is the skew-symmetric matrix associated with angular velocity vector ${}^E\omega_e \in \mathbb{R}^3$, and ${}^E\mathbf{v}_e(t) \in \mathbb{R}^3$ denotes the translational velocity. We further define ${}^E\mathbf{p}_l = \lambda \mathbf{x}$, where $\mathbf{x} = [\mathbf{u}^\top \ 1]^\top$ is the homogeneous coordinate form of \mathbf{u} , then ${}^E\dot{\mathbf{p}}_l = \lambda \dot{\mathbf{x}} + \lambda \ddot{\mathbf{x}}$. Substitute it into (11), we have

$$\dot{\mathbf{x}} = [{}^E\omega_e]_x \mathbf{x} + \frac{1}{\lambda} {}^E\mathbf{v}_e - \frac{\lambda}{\lambda} \ddot{\mathbf{x}}. \quad (12)$$

Now, multiplying both sides of (12) by $([\mathbf{x}]_{\times}^{\mathcal{E}} \mathbf{v}_e)^{\top}$, we have

$$([\mathbf{x}]_{\times}^{\mathcal{E}} \mathbf{v}_e)^{\top} \dot{\mathbf{x}} + ([\mathbf{x}]_{\times}^{\mathcal{E}} \mathbf{v}_e)^{\top} [\mathbf{x}]_{\times}^{\mathcal{E}} \boldsymbol{\omega}_e = 0. \quad (13)$$

The above equation defines the continuous epipolar constraint. Apparently, (13) does not depend on the 3D position of any world point, and only on the 2D observations of the world point. Under the assumption of brightness constancy, $\dot{\mathbf{x}}(t)$ can be approximated by the optical flow $\dot{\mathbf{u}}(t)$ obtained from the event camera.

The velocity of the camera in its own frame can be calculated from the radar velocity as

$$\mathcal{E}\mathbf{v}_e = \mathcal{E}\mathbf{R}_r \mathcal{R}\mathbf{v}_r + \mathcal{E}\mathbf{R}_r (\mathcal{R}\boldsymbol{\omega}_r \times \mathcal{R}\mathbf{l}_{er}), \quad (14)$$

where $\mathcal{E}\mathbf{R}_r$ is the rotation matrix from the radar frame to the event camera frame, respectively. $\mathcal{R}\mathbf{l}_{er}$ denotes the displacement from radar to the camera. When $\mathcal{R}\mathbf{l}_{er}$ is small enough, the velocity can be approximately obtained as $\mathcal{E}\mathbf{v}_e = \mathbf{C}_{\mathcal{R}}^{\mathcal{E}} \mathcal{R}\mathbf{v}_r$, where $\mathcal{R}\mathbf{v}_r$ can be replaced with the closest radar ego-motion estimates in discrete time, assuming slow velocity variation.

Defining $\mathbf{a} = ([\mathbf{x}]_{\times} \mathbf{C}_{\mathcal{R}}^{\mathcal{E}} \mathcal{R}\mathbf{v}_r)^{\top} [\mathbf{x}]_{\times}$, and $b = -([\mathbf{x}(t)]_{\times} \mathbf{C}_{\mathcal{R}}^{\mathcal{E}} \mathcal{R}\mathbf{v}_r(t))^{\top} \mathbf{u}(t)$, equation (13) is simplified as

$$\mathbf{a}^{\mathcal{E}} \boldsymbol{\omega}_e(t) = \eta. \quad (15)$$

Given $n \geq 3$ points satisfying constraint (13), and under the assumption that the angular velocity remains constant during a very short time period, we have the angular velocity equations as

$$\mathbf{A}^{\mathcal{E}} \boldsymbol{\omega}_e(t) = \eta, \quad (16)$$

where $\mathbf{A} = [\mathbf{a}_1^{\top} \ \mathbf{a}_2^{\top} \ \cdots \ \mathbf{a}_n^{\top}]^{\top}$, $\mathbf{b} = [\eta_1 \ \eta_2 \ \cdots \ \eta_n]$. Then $\mathcal{E}\boldsymbol{\omega}_e(t)$ can be obtained in a least square fashion.

IV. CONTINUOUS-TIME ESTIMATION

In this section, a continuous-time ego-motion estimator is constructed to fuse the two types of instantaneous motion measurements, as shown in Fig.4. We formulate the estimation problem as a nonlinear sliding-windowed estimation problem and solve it in a discrete-time manner.

A. B-splines based trajectory

To begin with, we represent the continuous-time velocities as cumulative B-splines. Specifically, we parameterize the translational and rotational velocities in two separate splines in the body local frame, which is represented by the continuous-time functions $\mathcal{B}\mathbf{v}_b(t) \in \mathbb{R}^3$ and $\mathcal{B}\boldsymbol{\omega}_b(t) \in \mathbb{R}^3$. To simplify the problem, we assume that the body frame is aligned with the radar frame. The superscripts and subscripts are ignored, and we use $\mathbf{v}(t)$ and $\boldsymbol{\omega}(t)$ for simplification.

Consider the translational velocity function $\mathbf{v}(t)$ over a time period is of order k , and controlled by points \mathbf{v}_i , then it can be represented as

$$\mathbf{v}(t) = \mathbf{v}_i + \sum_{j=1}^{k-1} \lambda_j^v(t) \cdot \Delta\mathbf{v}_{ij}, \quad (17)$$

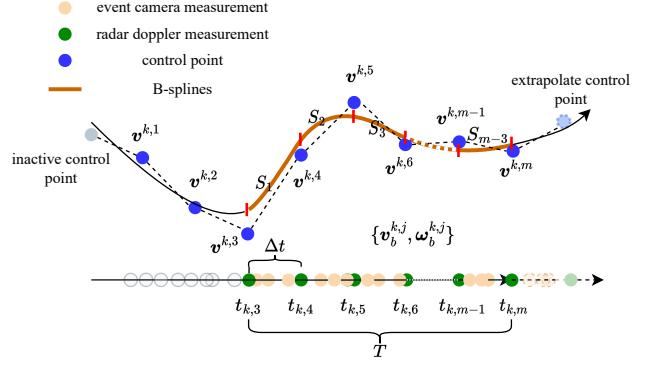


Fig. 4: The B-splines control points distribution and measurements.

where $\lambda_j^v(t)$ is constant coefficient which depends on the order k , and $\Delta\mathbf{v}_{ij} \triangleq \mathbf{v}_{i+j} - \mathbf{v}_{i+j-1} \in \mathbb{R}^3$. Similarly, we define the rotational velocity as

$$\boldsymbol{\omega}(t) = \boldsymbol{\omega}_i + \sum_{j=1}^{k-1} \lambda_j^{\omega}(t) \cdot \Delta\boldsymbol{\omega}_{ij}, \quad (18)$$

with $\Delta\boldsymbol{\omega}_{ij} \triangleq \boldsymbol{\omega}_{i+j} - \boldsymbol{\omega}_{i+j-1} \in \mathbb{R}^3$.

B. Sliding-windowed Optimization

The objective of our estimator is to simultaneously minimize two factors: 1) The predicted translational velocities should be consistent with those measured from radar Doppler, and 2) the predicted angular velocity should be consistent with the optical flow obtained from the event camera based on Eq. (13).

To begin with, we define the sliding-window state to be estimated at time instance k as follows:

$$\mathcal{X}_k \triangleq \{\mathbf{v}_j^k, \boldsymbol{\omega}_j^k\}_{j=1:m} \quad (19)$$

where $\mathbf{v}_j^k, \boldsymbol{\omega}_j^k$ denote the control points of $m-3$ B-splines at time instance k .

Our sliding window-based optimization is illustrated in Fig.4. We consider using all measurements that fall into the period $[t_{k,3}, t_{k,m}]$. The objective function is formulated as

$$\arg \min_{\mathcal{X}_k} \alpha_r(\mathcal{X}^k) + \alpha_e(\mathcal{X}^k) + \alpha_{\text{prior}}(\mathcal{X}^k), \quad (20)$$

where $\alpha_r(\cdot), \alpha_e(\cdot), \alpha_{\text{prior}}(\cdot)$ are the radar, event camera, and prior information terms, respectively.

C. Measurement Models

In this part, we formulate the measurement residuals related to the radar and the event camera.

1) *Doppler Velocity Measurement:* We consider the radar measurement obtained from III-A, $\mathcal{R}\hat{\mathbf{v}}_r$, which includes the true linear velocity corrupted by a constant bias and a time-varying noise, namely

$$\mathcal{R}\hat{\mathbf{v}}_r^k = \mathcal{R}\mathbf{v}_{r,\text{true}}^k + \boldsymbol{\xi}_v^k, \quad (21)$$

where $\boldsymbol{\xi}_v^k$ is a Gaussian white noise $\boldsymbol{\xi}_v^k \sim \mathcal{N}(\mathbf{0}, \Sigma_v)$. Specifically, in this part we directly integrate the velocity obtained from Doppler measurements in a loosely-coupled manner to

simplify the problem. All points within a scan are used to first calculate the ego-motion velocity, then the velocity error term can be calculated as

$$\mathbf{r}_v^k = \mathcal{R}\hat{\mathbf{v}}_r^k - \mathcal{R}\bar{\mathbf{v}}_r^k, \quad (22)$$

where $\mathcal{R}\hat{\mathbf{v}}_r^k$ is the interpolated/extrapolated pseudo measurement based on current estimates.

Then the cost term w.r.t. radar $\alpha_r(\cdot)$ is defined as

$$\alpha_r = \sum_{j \in M_r^k} \|\mathbf{r}_v^j\|_{\Sigma_v^{-1}}^2, \quad (23)$$

where M_r^k is the set of active radar velocity measurements within the sliding window, and Σ_v is the covariance matrix of the radar velocity measurement.

2) *Optical Flow Measurement:* In the loosely coupled fashion, an angular velocity measurement is first obtained based on Eq.16. Without loss of generality, we consider the measurement to be corrupted by a noise term $\xi_e \sim \mathcal{N}(\mathbf{0}, \Sigma_\omega)$. Then the angular velocity measurement can be modeled as

$$\mathcal{E}\omega_e^{k'} = \mathcal{E}\omega_{\text{true}}^{k'} + \eta_\omega^{k'}, \quad (24)$$

The superscript k' is used to denote the time instance when an angular velocity measurement is obtained. Note that we use a time instance variable k' to denote the time instance for the event camera due to its asynchronous sensing mechanism.

We define the residual of the angular velocity as

$$\mathbf{r}_e^{k'} = \mathcal{E}\hat{\omega}_e - \mathcal{E}\bar{\omega}_e. \quad (25)$$

Then we can arrived at the cost function w.r.t. event camera α_e as

$$\alpha_e = \sum_{j \in M_e^k} \|\mathbf{r}_e^j\|_{\Sigma_e^{-1}}^2, \quad (26)$$

where M_e^k is the set of all active event measurements within the sliding window.

3) *Prior information:* Besides the above two factors, we also integrate prior information during each sliding-windowed optimization. Specifically, the prior factors constrain the common control points between two consecutive sliding windows. Additionally, we consider the biases and time offset as slow-varying parameters. Therefore, the prior constraints on the following states:

$$\mathcal{X}_p^k \triangleq \{\Phi^{k-1} \cap \Phi^k\}. \quad (27)$$

Then the prior factor residual can be defined as

$$\mathbf{r}_p^k = \mathbf{H}_p \mathcal{X} - \mathcal{X}_p^k, \quad (28)$$

where \mathbf{H}_p is the matrix to select overlapped terms of current states w.r.t. prior information. Then the prior constraint term α_{prior} can be defined as

$$\alpha_{\text{prior}} = \|\mathbf{r}_p^k\|^2. \quad (29)$$

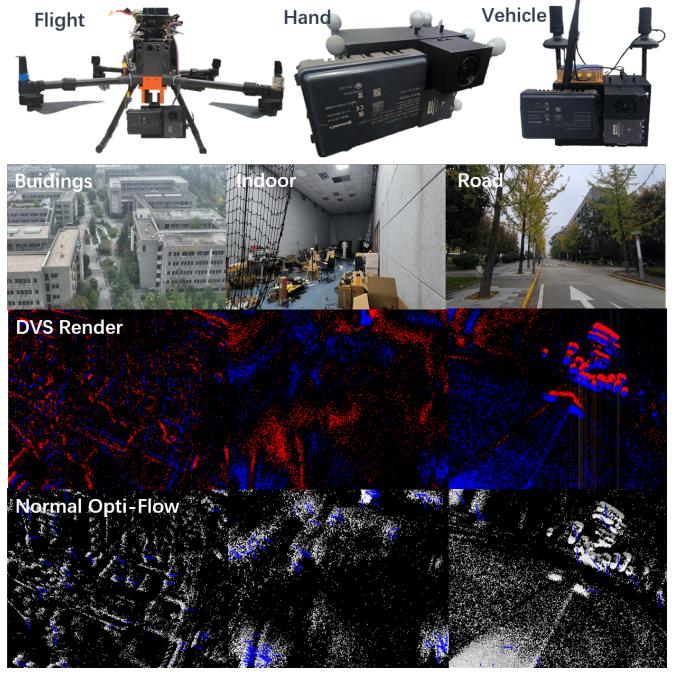


Fig. 5: Experimental platform, scenery, dataset, and detection.

V. EXPERIMENT

A. Platforms & Datasets

To validate the proposed method, we integrate multiple sensors, including a DAVIS346 event camera, an ARS548 4D millimeter-wave radar, a Parker IMU, and an Intel NUC13, forming the core sensing module. Three experiments are conducted:

- Flight The sensor suite is mounted on a DJI M300 for real-flight tests in semi-urban areas. Low-altitude, forward-view trajectories navigate through building clusters and narrow passages, to evaluate robustness under complex structural conditions. Ground truth is provided by RTK with a high-precision positioning network.
- Handheld A handheld setup captures indoor data under aggressive motion and HDR lighting. The cluttered environments, featuring corridors and reflective surfaces, serve to evaluate the robustness of perception and motion estimation. Ground truth is obtained from the OptiTrack system.
- Vehicle An electric vehicle conducts high-speed road experiments under day and night HDR conditions. Dynamic traffic scenes with moving vehicles, pedestrians, and illumination changes are used to assess system robustness and real-time performance. Ground truth is provided by RTK integrated with the positioning network.

We collected a total of 9 sequences and performed a statistical analysis of the ground-truth velocities. Table I summarizes the linear and angular velocity statistics for the nine recorded sequences.

For the dji flight sequences, linear velocities range from 3.47 to 4.01 m/s and angular velocities from 0.16 to 0.56 rad/s, reflecting moderate low-altitude flights over building clusters and narrow passages with a top-down view. These trajectories provide structural and visual complexity for validating the method under real-flight conditions. The HDR

TABLE I: Twist and HDR Statistics for Sequences.

File	Lin.Max	Lin.Avg	Ang.Max	Ang.Avg	HDR
dji1	4.49	4.01	0.34	0.16	62
dji2	4.51	3.47	0.54	0.16	63
dji3	6.30	3.74	0.56	0.18	62
hand1	4.20	0.92	0.42	0.16	89
hand2	2.32	0.87	0.45	0.17	92
hand3	4.39	1.17	0.41	0.16	97
road1	7.64	6.32	<u>0.55</u>	0.28	96
road2	7.75	<u>5.83</u>	0.54	<u>0.27</u>	101
road3	<u>7.51</u>	5.40	0.65	0.17	109

Notation: Linear velocity in m/s, angular velocity in rad/s, HDR in dB.
Bold indicates the best value
Underline indicates the second-best value.

TABLE II: Characteristic of Algorithms

Algorithm	Input				Output		
	E	R	V	I	L	A	P
CMAX[?]	✓					✓	
RIO[?]		✓	✓	✓	✓		✓
River[?]		✓	✓	✓	✓		✓
PLEVIO[?]	✓		✓	✓	✓		✓
DEVO[?]	✓					✓	
TE(front, proposed)	✓		✓	✓	✓	✓	
TE(back, proposed)	✓		✓	✓	✓	✓	

Notation: E: Event; R: Radar; V: Camera; I: IMU; L: Linear velocity; A: Angular velocity; P: Pose.

in these sequences ranges around 62 to 63 dB, representing typical outdoor lighting conditions for standard event camera perception.

Handheld sequences exhibit lower motion, with linear velocities below 1.2 m/s and angular velocities around 0.16 to 0.17 rad/s, corresponding to aggressive indoor hand motion within cluttered scenes. The HDR is estimated around 89 to 97 dB, indicating darker indoor lighting and emphasizing robustness to rapid viewpoint changes and low-light conditions.

Road sequences, collected using an electric vehicle, reach up to 7.75 m/s and 0.65 rad/s, involving high-speed motion in dynamic traffic and varying illumination, designed to test system stability and real-time performance under fast outdoor conditions. HDR in these sequences is higher, around 96 to 109 dB, reflecting strong sunlight and overexposed regions, which challenge event-based perception under high dynamic range scenarios.

TABLE III: LINEAR VELOCITY ERROR (m/s)

Seq	RIO	River	PLEVIO	DEVO	TE(front)	TE(back)
dji1	1.22	1.99	×	1.07	<u>0.27</u>	0.22
dji2	3.18	2.77	×	1.77	<u>0.18</u>	0.16
dji3	×	1.29	×	0.98	<u>0.09</u>	0.05
hand1	×	×	×	×	0.67	0.40
hand2	×	2.91	×	1.38	<u>0.10</u>	0.06
hand3	×	1.63	×	0.88	<u>0.75</u>	0.44
road1	×	×	×	×	0.17	0.11
road2	×	×	0.40	1.46	<u>0.17</u>	0.12
road3	×	2.83	0.24	×	0.14	0.11

Notation: **Bold** indicates the best value
Underline indicates the second-best value.

B. Algorithms

We selected several representative algorithms for comparison, covering state-of-the-art methods across different sensor modalities. CMAX-SLAM (CMAX) [?] focuses on angular velocity estimation, RIO[?] is a radar-inertial odometry approach, River[?] estimates linear velocity using radar-inertial fusion, PL-EVIO (PLEVIO)[?] is an event-based visual-inertial odometry method, and DEVO[?] is a state-of-the-art event-only odometry approach. Collectively, these methods represent representative sensor and algorithmic combinations for high-dynamic pose and velocity estimation. Combined with our proposed Twist-Estimator (TE), including front-end (front) and back-end (back) results, we evaluate all methods on our self-recorded datasets. Input and primary output types are summarized in Table II. Compared to adaptive tuning methods such as NeuroMHE [?], we set fixed residual weights for linear and angular velocities to 400 and 800 for system simplification. The relatively large weights help prolong the optimization process and improve the resolution of small residuals. For the spline representation, we use cubic B-splines ($k = 3$), the control points in this formulation are fully observable in the velocity state, making the spline representation effectively as observable as sensors measurements.

Velocity estimation is measured by the Absolute Velocity Error (AVE), capturing both magnitude and temporal trends. Let \mathbf{x}_{est} and \mathbf{x}_{gt} denote the estimated and ground-truth velocities (linear or angular), with instantaneous error

$$\text{AVE} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_{\text{est},i} - \mathbf{x}_{\text{gt},i}\|_2,$$

where N denotes the length of the evaluated sequence.

C. Linear Velocity Evaluation

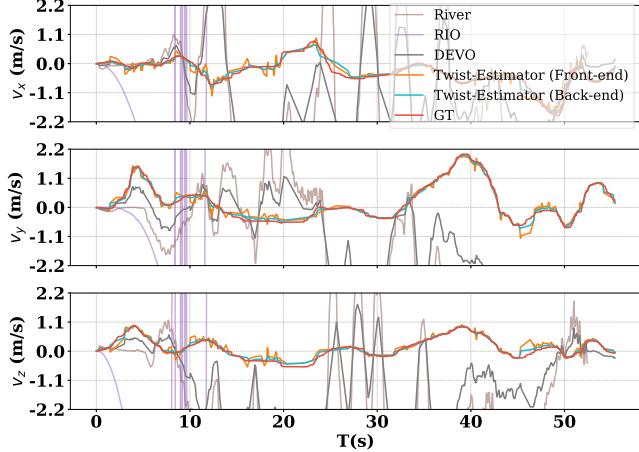
Table III summarizes linear velocity estimation across all sequences, and illustrative examples of the error curves based on dataset hand2 and road2 are provided in Fig. 6a and 7a, respectively.

- RIO relies on IMU acceleration integration and radar scan matching. It exhibits large errors in most sequences due to drift and unreliable radar constraints, achieving moderate accuracy only in structured scenes like dji1 and dji2. In unobvious structure or water surface reflective scenes

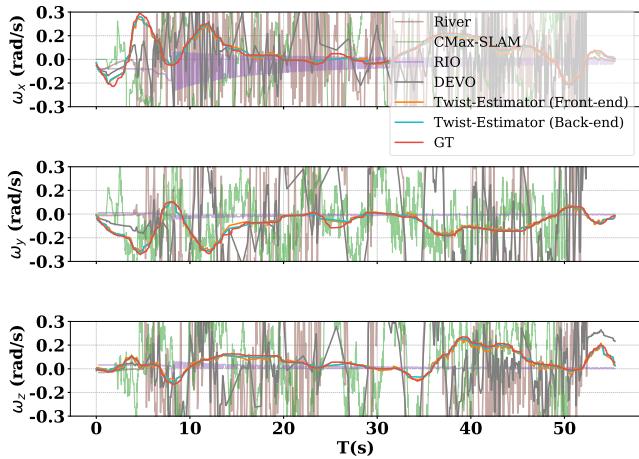
TABLE IV: ANGULAR VELOCITY ERROR (rad/s)

Seq	CMAX	RIO	River	PLEVIO	DEVO	TE(front)	TE(back)
dji1	×	0.14	0.09	×	0.21	<u>0.07</u>	0.06
dji2	0.08	0.10	0.18	×	0.34	0.10	0.07
dji3	0.07	0.08	0.11	×	0.14	<u>0.05</u>	0.03
hand1	0.17	×	0.15	×	×	0.11	0.08
hand2	0.15	0.10	0.15	×	0.16	<u>0.04</u>	0.01
hand3	0.17	0.15	0.15	×	0.26	<u>0.11</u>	0.06
road1	0.18	×	0.17	0.16	×	0.19	0.17
road2	0.17	×	0.17	0.17	<u>0.16</u>	0.17	0.15
road3	0.12	0.09	0.12	0.11	×	0.09	0.08

Notation: **Bold** indicates the best value
Underline indicates the second-best value.



(a) hand2 Linear velocity.

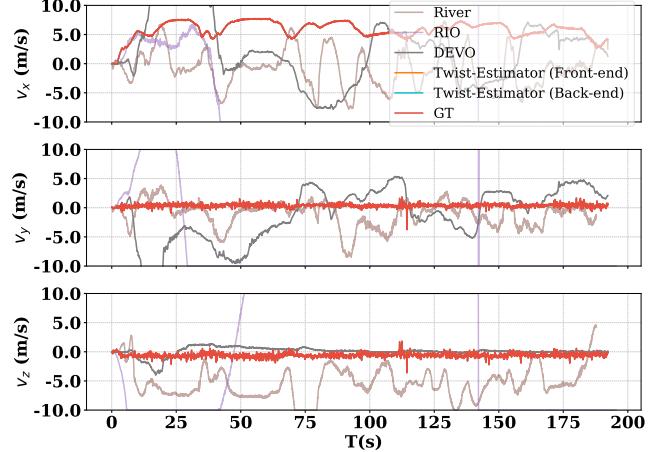


(b) hand2 Angular velocity.

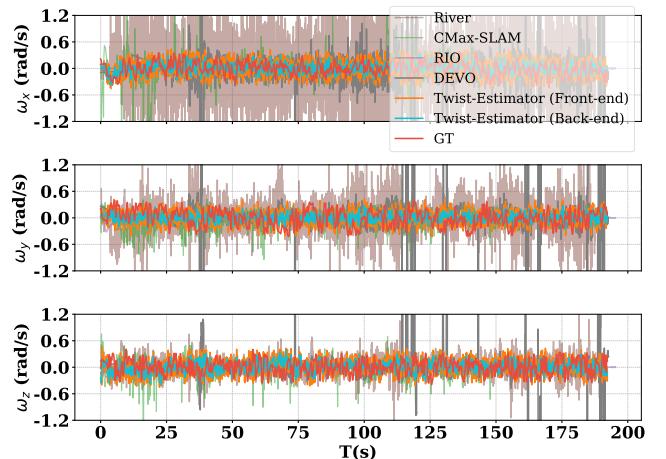
Fig. 6: Twist Evaluation of Algorithms on Seq. hand2.

such as dji3, radar matching is unreliable, causing unstable estimates.

- River fuses radar Doppler with IMU for direct velocity estimation. Inconsistencies between Doppler and inertial measurements reduce fusion effectiveness, and performance degrades under significant IMU bias or vibrations, especially in high-speed road sequences.
- PLEVIO uses event-based point-line feature tracking initialized by intensity gradients. Sparse, low-contrast, or spatially sparse events in handheld and aerial sequences often cause tracking failures. Indoor clutter and transient corridor transitions further limit reliable estimation.
- DEVO is an event-only approach, relies on reconstructed event patches. While avoiding IMU drift allows better performance in limited-motion indoor scenes, accuracy degrades in aerial or road sequences with larger motion ranges or depth variation, where the pre-trained model fails to generalize.
- Proposed TE estimates velocity directly from external observations and avoids IMU integration. TE(front) uses radar Doppler for high accuracy, and the addition of event optical flow constrains linear motion, which reduces drift. Compared to PLEVIO, our method computes optical flow over small 3×3 neighborhoods, providing robust constraints even



(a) road2 Linear velocity.



(b) road2 Angular velocity.

Fig. 7: Twist Evaluation of Algorithms on Seq. road2.

under sparse or low-contrast events. TE(back) incorporates marginalization-based smoothing to filter spikes, and yields conservative yet stable estimates. As a result, TE(back) achieves the lowest errors in sequences such as dji1, 2, 3 and road1, 2, 3, consistently outperforming existing baselines across aerial, handheld, and vehicle platforms under moderate and aggressive motion.

D. Angular Velocity Evaluation

Table IV presents angular velocity estimation results across all sequences, and illustrative examples of the error curves based on dataset hand2 and road2 are provided in Fig. 6b and 7b, respectively.

- RIO and River rely on IMU high-frequency angular velocity measurements, providing generally accurate references across sequences. Their performance benefits from inertial sensing, particularly in structured aerial or road scenes, but can degrade under high vibrations or inconsistent radar constraints.
- CMAX depends primarily on structured event-based features for rotational estimation. It performs well in aerial sequences such as dji2 and dji3, where events exhibit clear structure and flight is dominated by near-pure rotational motion. In dji1, the low flight speed and angular motion reduce the

effectiveness of CMAX. Indoor sequences with cluttered, spatially dispersed edges, or road sequences with large depth variations, violate the pure-rotation assumption, leading to larger errors.

- PLEVIO also depends on event-based point-line feature tracking. While event structures are relatively clear in aerial sequences, sparse or unstable features limit reliable angular estimation. Handheld and indoor sequences with low-contrast or dispersed events further reduce tracking accuracy.
- DEVO, an event-only method, avoids IMU-induced drift and performs well in indoor sequences with limited motion. However, in aerial and high-speed road sequences, large motion ranges, depth changes, and mismatches with training data lead to degraded performance.
- Proposed TE estimates angular velocity directly from external observations, avoiding IMU integration. TE(front) achieves near-optimal performance in most sequences, and TE(back), with marginalization-based smoothing, consistently attains the lowest errors. Across aerial, handheld, and vehicle sequences, TE surpasses the inertial angular velocity reference in many cases, though some high-vibration scenes (e.g., dji2, road1, 2, 3) are affected by environmental disturbances. Overall, TE demonstrates robust, accurate angular estimation across diverse platforms and motion dynamics.

E. Supplemental trajectory evaluation

Another promising application of instantaneous velocity estimation is pose estimation. Integrating instantaneous velocities over time mitigates the accumulation of dynamic errors in higher-dimensional motion, making it an effective approach for short-term localization. To evaluate this capability, we conducted an extended experiment in which poses obtained by integrating the velocity outputs of our proposed method over a 5-second interval were compared against the corresponding poses from baseline algorithms, with all trajectories first aligned via SE(3) transformation, as illustrated in the figure 8.

The experimental results demonstrate that the proposed method, particularly when the velocity estimates are optimized before integration, produces poses that most closely align with the ground truth. This indicates the significant potential of our approach for pose estimation, highlighting its suitability for highly dynamic scenarios where short-term accurate localization is critical.

VI. CONCLUSION

We propose a robust and efficient ego-motion estimation framework for dynamic, visually challenging environments. It avoids inertial data and complex feature matching, offering a lightweight solution for agile robots. Experiments show stable motion estimates where traditional methods struggle. Designed as a minimal 6-DoF velocity estimator, it can serve as an alternative to IMU-based approaches. In future work, we plan to integrate IMU measurements to provide system redundancy to more challenging environments.

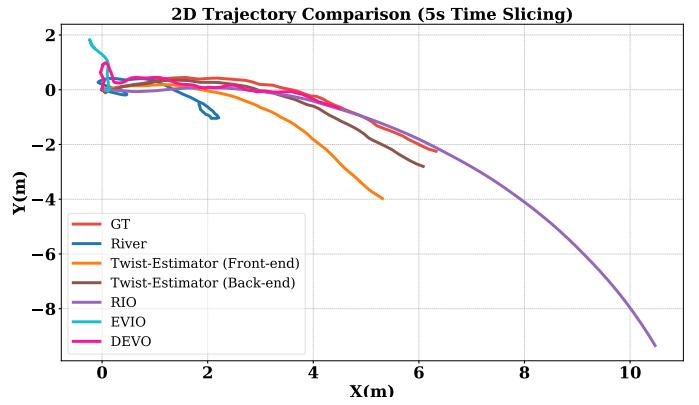


Fig. 8: Velocity-Integrated Trajectory Comparison over 5 s in road2