IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Event- and Frame-based Visual-Inertial Odometry with Adaptive Filtering based on 8-DOF Warping Uncertainty

Min Seok Lee, *Graduate Student Member, IEEE*, Jae Hyung Jung, *Graduate Student Member, IEEE*, Ye Jun Kim, and Chan Gook Park, *Member, IEEE*

***Abstract*— In this letter, we present an event- and frame-based visual-inertial odometry (VIO) algorithm that fuses frames, events, and inertial measurement in a robust and adaptive manner. Frames from standard cameras provide rich context of the scene at a fixed rate. Event cameras on the other hand asynchronously produce events at a pixel-level when changes in intensity occur, and thus are resilient to motion blur and have high dynamic range. To harness the advantages of the two sensors, our frontend fuses their outputs by creating brightness increment patches of each output and minimize the differences with an 8-DOF warping model. The warping model and the optimization process allow for robust feature tracking in the frontend of the algorithm. The minimized residual is then used in the multi-state filter-based backend where the measurement update is adaptively performed depending on the size of the residual for accurate estimation, reflecting the quality of the tracked features. Comparative evaluation on two publicly available datasets reveals that our method outperforms the state-of-the-art event-based VIO algorithms in pose estimation accuracy.**

***Index Terms*— Computer vision, event camera, Kalman filter, sensor fusion, visual-inertial odometry**

I. Introduction

Accurately estimating the ego-motion of a sensor is a crucial factor in various applications, including autonomous robotics and augmented/virtual reality. Recently, numerous vision-based algorithms have been proposed to solve such problem and have shown great progress. Yet, conventional cameras possess innate limitations of motion blur and low dynamic range. In situations of high-speed motions or environments with high dynamic range, the navigation algorithms using only the standard cameras perform underwhelmingly or even fail.

Manuscript received: August 29th, 2023; Revised: November 2nd, 2023; Accepted: November 30th, 2023.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments.

*This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT, the Republic of Korea (NRF-2022R1A2 C2012166). (*Corresponding author: Chan Gook Park.*)

M. S. Lee, J. H. Jung, and Y. J. Kim are with the Department of Aerospace Engineering, Automation and Systems Research Institute, Seoul National University, Seoul 08826, Republic of Korea (Y. J. Kim is currently with Hyundai motor group, Seoul 06182, Republic of Korea. e-mail: mslee1996@snu.ac.kr; lastflowers@snu.ac.kr, yejun@hyundai.com).

C. G. Park is with the Navigation and Electronic System Laboratory, Department of Aerospace Engineering, Institute of Advanced Aerospace Technology, Seoul National University, Seoul 08826, Republic of Korea (e-mail: chanpark@snu.ac.kr).

The novel vision sensor, the event camera, or the dynamic vision sensor (DVS) [1], brings numerous advantages that can resolve these issues elegantly. This bioinspired sensor asynchronously outputs an event whenever an intensity change has been detected at a pixel, very much in contrast with the standard cameras, which outputs absolute intensity frames at a fixed rate using global exposure. The event camera has very high temporal resolution with 1MHz clock and low latency, as it works at a pixel-level, not needing to wait for the global exposure. The neuromorphic sensor also has very high dynamic range of 140dB compared to high-quality standard cameras with 60dB. These advantages of the event camera make it a reliable choice in situations such as fast motions, low light, and high dynamic range: exactly where the standard camera lacks performance.

However, the sensor has its own drawbacks. As events are generated at a pixel-level, global context of the scene is not obtainable. Also, as the sensor reacts to the change in brightness, hardly any events are produced when the motion is limited, such as static situations. Further, different event representations are produced for the same scenery depending on the direction of the motion, making data association and correspondence a challenging task. Hence, embracing the complementary nature of the standard camera and the event camera would present an ideal solution to solving the pose estimation problem in visual navigation. With the advent of the dynamic and active pixel vision sensor (DAVIS) [2], which generates both events and frames that share a pixel array, facilitating the two sensors has become much easier. Together with built-in inertial measurement units (IMU), DAVIS also makes a suitable sensor candidate for performing visual-inertial navigation.

Although some literatures have addressed event-based visual-inertial navigation algorithms using all three modalities, there is yet a great room for improvement both in the frontend and the backend. In visual odometry, the frontend refers to the part of the pipeline that produces feature tracks or any meaningful visual information. The backend uses the visual features provided by the frontend to output pose estimates. In the frontend aspect, a robust visual feature tracking method that efficiently fuses the event camera and the standard camera is in need. As for the backend, an estimator that can reflect the information regarding the frontend, namely the uncertainty of the feature tracks, is crucial for achieving accurate and robust pose estimation. Hence, in this letter, we propose an event- and frame-based visual-inertial odometry algorithm comprised of a frontend that robustly fuses events and frames with 8-DOF warping model for accurate feature tracks and an adaptive multi-state Kalman filter-based estimator backend that

reflects the frontend uncertainty. Our contributions in this letter are:

- We propose an accurate and robust event- and frame-based visual-inertial odometry algorithm that robustly fuses outputs from the event camera and the standard camera with an 8-DOF warping model, adopted from our previous work [3].
- Our multi-state Kalman filter-based backend, modified from the framework of [4], adaptively performs measurement updates by reflecting the uncertainty of the feature tracks provided by the frontend for accurate pose estimation.
- We perform quantitative evaluation of our method on two real datasets of 28 sequences. Comparative analysis proves that our method outperforms state-of-the-art event-based VIO algorithms with accurate pose estimates and robustness to challenging scenarios.

The rest of the paper is organized as follows: In Section II, we review relevant literature on visual/visual-inertial navigation algorithms, with focus on event-based methods. In Section III, we present our event- and frame-based visual-inertial odometry algorithm comprising a robust feature tracker frontend using 8-DOF warping model and a filter-based backend with adaptive measurement updates. In Section IV, we perform quantitative evaluation on two public datasets [5], [6], and compare pose estimation accuracy with state-of-the-art event-based VIO algorithms. Finally, this paper concludes with Section V.

II. Related Work

Due to its rather recent advent, event camera has only recently been adopted as a sensor for ego-motion estimation in the fields of robotics and navigation as compared to a more maturely researched sensors, such as the standard camera. Yet, thanks to its numerous advantages, many works have already presented visual navigation methods that utilizes event cameras. Some of these methods use event cameras alone, and many others use them in conjunction with either standard cameras, IMUs, or both.

One of the first event-based visual odometry (VO) algorithms is [7], where the authors chose DAVIS as their sensor, using both frames and events. The work first detects corner features from a frame using the Harris detector, and then creates local edge-maps around the features using the Canny edge detector. Then, for incoming events between the frames, the algorithm creates 2D histograms and registers them with the frame-formulated edge-maps using the weighted iterative closest point (ICP). With the tracked features, the algorithm performs odometry using depth filter-based 3D mapping and reprojection error minimization. In [8], the authors proposed a real-time VO algorithm that uses just an event camera and tracks the camera pose using a ray-triangle intersection method [9] in the framework of extended Kalman filter (EKF). [10] proposed EVO, an event camera only algorithm that performs parallel tracking and mapping, following the principles of simultaneous localization and mapping (SLAM) systems. EVO leverages the fact that event cameras respond to edges, and thus performs geometric registration between the semi-dense map and the event image using the inverse compositional Lucas-Kanade method [11], [12]. The mapping module of EVO is based on the event-based multi-view stereo (EMVS) [13] method which discretizes space into a voxel grid and creates semi-dense depth map with the voxels that are more intersected by the rays than others. [14] takes a more probabilistic approach in utilizing a photometric depth map by using the framework of Bayesian filtering. The authors designed a robust sensor model that incorporates a likelihood function of a normal-uniform mixture model and approximates the posterior distribution using a tractable distribution. [15] proposes a spatiotemporal registration technique for event-based motion estimation, in contrast to majority of other works that adopt contrast maximization or indirect methods. The algorithm estimates relative motion, which is reduced to relative rotation, and hence the results show strength especially in purely rotational sequences. [16] also focuses on rotational motion estimation but uses contrast maximization technique. On top of aligning local events using contrast maximization, the algorithm also aligns the events globally, meaning the events are constantly aligned to the initial camera coordinate. The events are locally and globally aligned using the warping function with Rodrigues' rotation formula. More recently, some literature began to address the stereo VO methods based on event cameras, where [17] claims to be the first event-based stereo VO algorithm. [17] uses the concept of a time surface, which stores the timestamp of the latest event at each pixel, and performs inverse depth estimation to create semi-dense map that is registered with the current time surface through the forward compositional Lucas-Kanade method [12]. Being the first event-based stereo system, the algorithm was compared with frame-based stereo systems and showed comparable results.

More recently, event-based visual navigation systems started to incorporate inertial data using IMU as an additional sensing modality. These IMU-aided algorithms can be categorized into those that use event camera only and those that use both event camera and standard camera. EVIO [18] is one of the seminal works in event-based visual navigation, used in many relevant literatures as a comparison. The system is claimed to be the first event-based odometry system that incorporates inertial data. Its feature tracking scheme employs two expectation-maximization (EM) steps, one estimating the optical flow and the other estimating the template alignment, and thus achieves robust results in high speed and high dynamic range situations. Use of EM algorithm in event-based feature tracking has motivated numerous related works, including [19]. In [19], the authors proposed a constrained EKF-based system that uses all three sensing modalities: event camera, standard camera, and the IMU. The hybrid system extracts feature tracks from frames using the FAST corner detector [20] and the KLT tracker [11], and estimates optical flow from events with the EM method, and constrains a range of scene-depth in the estimator. [21] also uses all three sensors as its input, but in a continuous-time representation. The algorithm achieves this by implementing the cubic B-spline trajectory in $SE(3)$ [22], as it interpolates pose at any given time, allowing fusion of outputs with different synchrony. [23] proposed a tightly-coupled VIO that works with an event camera and an

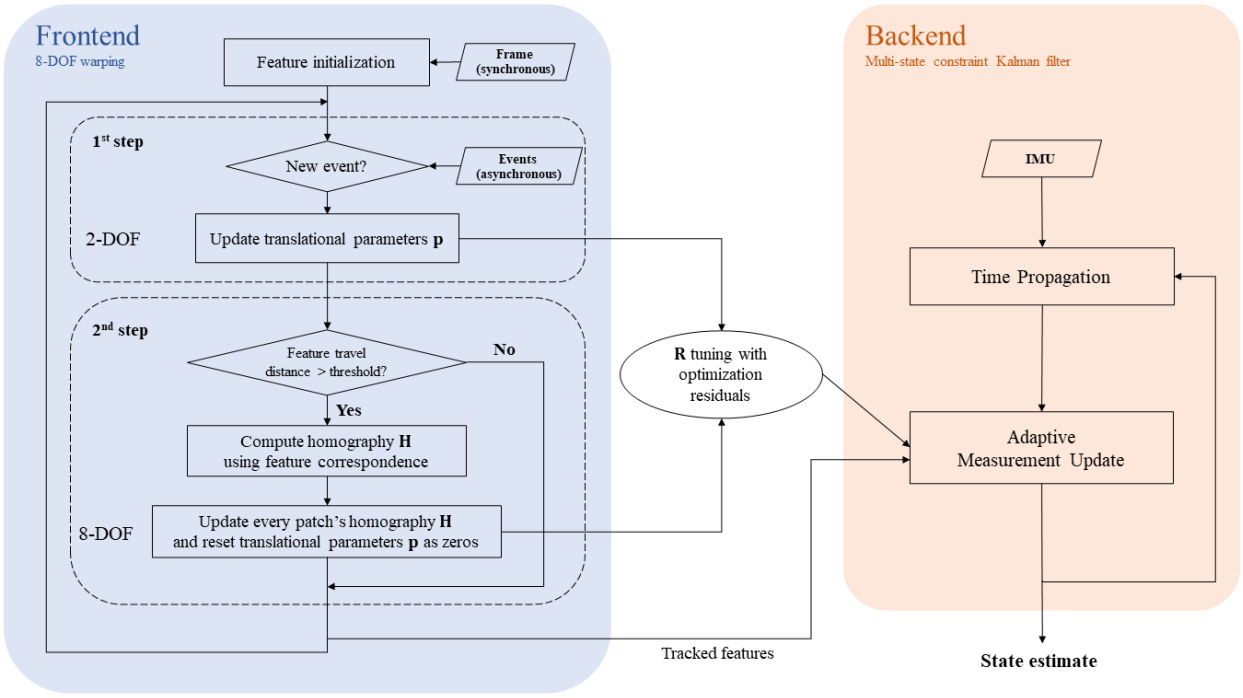

Fig. 1. System overview of the proposed event-and frame-based VIO algorithm. The blue section denotes the frontend of our pipeline, where we produce feature tracks through 8-DOF warping registration of brightness increment patches from events and frames. The orange section denotes the adaptive multi-state Kalman filter-based backend, where the warping residual is used to adaptively perform measurement update.

IMU. The algorithm uses a feature tracker that synthesizes motion-compensated event frames by fusing events in a spatio-temporal window, and combines these tracks through a keyframe-based nonlinear optimization framework. This optimization-based pipeline served as a ground work for ultimate SLAM (USLAM) [24], which extends [23] by adding frames from standard camera, proposing a state estimation pipeline that fuses all three sensor outputs. USLAM tracks features from both events and frames independently, and like [23], the feature tracks from both outputs and the inertial measurements are fused through nonlinear optimization. On the other hand, [25] achieves the integration of the three outputs with a filter-based backend, named xVIO [26]. xVIO is an extended Kalman filter with an IMU state and a visual state, where the latter is comprised of SLAM feature states and MSCKF (multi-state constraint Kalman filter) [27] feature states. As for the frontend, [25] adopts the state-of-the-art feature tracker EKLT [28], an extension of the KLT tracker [11] for event-frame fusion.

While the abovementioned state-of-the-art event-based VIO algorithms show decent results, each still poses limitations that need to be improved. Some of these limitations are: the frontends that manage visual features or flows are suboptimal, meaning they either lack robustness or are not tailor-made for events; the backends, whether optimization-based or filter-based, are mere addition of the inertial information, without considerations of the uncertainty or the quality of the frontend output. These limitations are what motivated us to develop our event- and frame-based VIO algorithm with a robust event-frame-fused frontend and an accurate adaptive filter-based backend.

More recently, data-driven methods were proposed in the field of event-based navigation. [29] proposes a data-driven feature tracker for event cameras. The method uses frame attention module to provide robust feature tracks that outperforms other methods in feature age. [30] introduces learning-based dense optical flow estimation for event cameras, where the method uses temporally recurrent architecture on two convolutional neural network (CNN) layers. EV-FlowNet [31] is one of the first event-based data-driven optical flow estimation method. The pipeline learns motion information from events through a self-supervised CNN that resembles a U-Net structure. To leverage the asynchronous nature of the event camera, [32] proposes recurrent asynchronous multimodal (RAM) networks, which are adaptations of the traditional recurrent neural networks (RNN) to handle asynchronous data. While these data-drive methods are promising in solving numerous event-based vision problems, they still possess critical limitations to overcome, such as the need for massive event datasets for training and being computationally much heavier than their classical counterparts.

III. Methodology

In this section, we present our accurate event- and frame-based VIO algorithm, illustrated in Fig. 1. The algorithm can be broken down into two parts: a robust feature tracker frontend based on our previous work [3] and an adaptive estimator backend based on the multi-state Kalman filter [4].

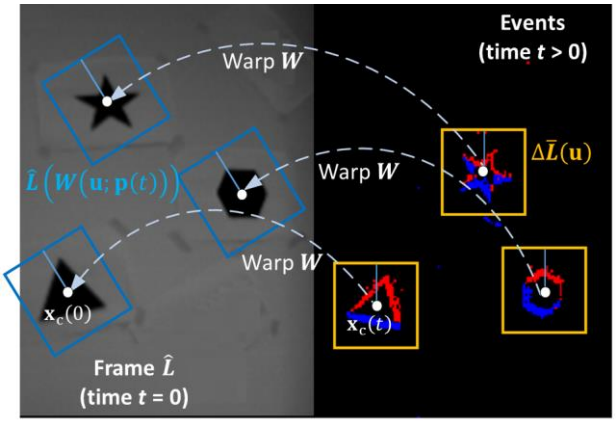

Fig. 2. Illustration of patch warping. Blue patches from frame $\hat{\boldsymbol{L}}(t = 0)$ are obtained from warping of the yellow patches from events at $t > 0$. Red and blue points represent events with $p_k = 1$ and -1, respectively.

A. Robust Event- and Frame-based Feature Tracker

We first demonstrate the event representation with the event generative model [14]. Events $e_k = (x_k, y_k, t_k, p_k)$ occur when the logarithmic brightness, $\boldsymbol{L}$, changes more than a threshold, C. t_k is the time at the event production, p_k is the event polarity, where +1 indicates an increase and -1 indicates a decrease in brightness, and $\mathbf{u}_k = (x_k, y_k)$ is the pixel coordinate of the event generated pixel. An ideal event camera would have a constant threshold C. The generative model then can be deduced down to the following equation that all e_k satisfy.

$$\Delta\boldsymbol{L}(\mathbf{u}_k, t_k) \triangleq \boldsymbol{L}(\mathbf{u}_k, t_k) - \boldsymbol{L}(\mathbf{u}_k, t_k - \Delta t_k) = p_k C, \quad (1)$$

where Δt_k is the time elapsed from the latest event in pixel $\mathbf{u}_k$.

We provide the summary of our frontend, where its details can be found in [3]. The feature tracker frontend starts by creating separate brightness increment patches from events and frames. As for the former, for a time window $\Delta\tau$, the polarities of the incoming events are accumulated to create a brightness increment patch $\Delta\bar{\boldsymbol{L}}(\mathbf{u})$ of events. The event-formulated patch can thus be expressed as,

$$\Delta\bar{\boldsymbol{L}}(\mathbf{u}) = \sum_{t_k \in \Delta\tau} p_k C \delta(\mathbf{u} - \mathbf{u}_k), \quad (2)$$

where δ is the Kronecker delta.

As for the frames, Harris corners are extracted from a frame $\hat{\boldsymbol{L}}$, and the intensity gradient $\nabla\boldsymbol{L}(\mathbf{u})$ is computed from the intensity patches of the extracted corners. This step refers to the feature initialization step shown in Fig. 1. With the gradient, the objective is to find the estimate of the event-formulated patch $\Delta\bar{\boldsymbol{L}}(\mathbf{u})$ that minimizes the difference between the estimate and $\Delta\bar{\boldsymbol{L}}(\mathbf{u})$ through optimization of the warping parameter $\mathbf{p}$ and the optical flow $\mathbf{v}$. The estimate from the intensity gradient can be represented as follows,

$$\Delta\hat{\boldsymbol{L}}(\mathbf{u};\mathbf{p},\mathbf{v}) \doteq -\nabla\hat{\boldsymbol{L}}\big(\boldsymbol{W}(\mathbf{u};\mathbf{p})\big) \cdot \mathbf{v}\Delta\tau, \quad (3)$$

where $\boldsymbol{W}$ is a warping model. This process is illustrated in Fig. 2.

Choosing an apt warping model is crucial as it directly affects the accuracy and the robustness of the feature tracker. In [3], the authors proposed an 8-DOF warping model that is expressed as,

$$\boldsymbol{W}(\mathbf{u};\mathbf{p},\mathbf{H}) = \boldsymbol{H}(\mathbf{u};\mathbf{H}) + \boldsymbol{t}(\mathbf{p}),$$
$$\mathbf{p} = \{t_x, t_y\}, \quad (4)$$

where $\boldsymbol{H}$ (with italics) is the homography warping function, $\mathbf{H}$ is the homography matrix, and $\boldsymbol{t}$ is translation. Previous works including [7] and [28] implement rigid warping of 3-DOF, which cannot describe all transformation in 2D space. By adopting 8-DOF warping model, the algorithm can explain any arbitrary warping between the event-patch and the frame-patch, hence achieving a more robust feature tracking performance [3]. The warping model is a modification of the homography warping model with the difference being the warping parameters containing only the 2-DOF translation parameters $\{t_x, t_y\}$ instead of eight parameters $\{h_i | 1 \le i \le 8\}$.

The separation of translation from the homography is deliberate, as the algorithm adopts a two-step optimization process that separately optimizes $\mathbf{H}$ and $\mathbf{p}$. The two-step process prevents loss of both the computational efficiency and the stability of optimization that occur when attempting to naively optimize eight warping parameters. In the first step, we update $\mathbf{p} = \{t_x, t_y\}$ by solving the following optimization problem:

$$\min_{\mathbf{p},\mathbf{v}} \left\| \frac{\Delta\bar{\boldsymbol{L}}(\mathbf{u})}{\|\Delta\bar{\boldsymbol{L}}(\mathbf{u})\|} - \frac{\Delta\hat{\boldsymbol{L}}(\mathbf{u};\mathbf{p},\mathbf{H},\mathbf{v})}{\|\Delta\hat{\boldsymbol{L}}(\mathbf{u};\mathbf{p},\mathbf{H},\mathbf{v})\|} \right\|^2. \quad (5)$$

The aim is to minimize the difference between $\Delta\bar{\boldsymbol{L}}(\mathbf{u})$ and $\Delta\hat{\boldsymbol{L}}(\mathbf{u})$ by optimizing $\mathbf{p}$ and $\mathbf{v}$ with fixed $\mathbf{H}$.

The second step updates $\mathbf{H}$ by solving:

$$\mathbf{x}_c^i(t) = \boldsymbol{W}^{-1}(\mathbf{x}_c^i(0);\mathbf{p}(t),\mathbf{H}) \ \ (i = 1, \dots, N), \quad (6)$$

$$\min_{\mathbf{H}} \textstyle\sum_i \left\| \mathbf{x}_c^i(0) - \boldsymbol{H}(\mathbf{x}_c^i(t);\mathbf{H}) \right\|^2, \quad (7)$$

where $\mathbf{x}_c$ is the center of features. From (6), the new locations of the features are computed, and with the correspondence between the initial and current feature locations, we can obtain $\mathbf{H}$ that minimizes (7). This process is referred to as computing homography parameters using feature correspondence in Fig. 1. When performing the above, RANSAC is used to exclude the outliers rather than utilizing all matching feature pairs. The last step of our frontend would be to update $\mathbf{H}$ of the features and reset $\mathbf{p}$ as zeros.

As is depicted in Fig. 1, the first step is initiated whenever an event is produced, whereas the second step is only executed when the travelled distance of the features exceeds a threshold. Hence, as the whole 8-DOF, $\mathbf{H}$, is updated less frequently than $\mathbf{p}$, the optimization process ensures stability

and efficiency with the reduced 2-DOF optimization of the first step. The residuals from the two optimization steps are then used in the filter backend for adaptive measurement update, which we explain in the next subsection.

B. Adaptive Filter-based Estimator

Our multi-state Kalman filter-based backend adopts the framework of [4]. A multi-state filter for VIO was first proposed in [27] as MSCKF, and since then, various variants have been proposed. [4] applies the keyframe selection technique to the original MSCKF, and thus information from longer feature tracks is used for measurement updates with fewer filter states. Our proposed backend estimator is an adaptive modification of [4], where we utilize the residual computed from the frontend to adaptively perform measurement updates to ensure a more accurate estimation by reflecting the quality of the feature tracking process performed by the frontend.

The filter state is comprised of the IMU state and the visual state, where the former is defined as,

$$\mathbf{x}_I = \begin{pmatrix} \mathbf{q}_G^{I\,T} & \mathbf{b}_g^T & \mathbf{v}_I^{G\,T} & \mathbf{b}_a^T & \mathbf{p}_I^{G\,T} & \mathbf{q}_C^{I\,T} & \mathbf{p}_C^{I\,T} \end{pmatrix}^T, \quad (8)$$

where $\mathbf{q}_G^I$ is the rotation from the inertial frame to the body frame, the IMU frame, in quaternion, $\mathbf{b}_g$ is the bias of the gyroscope, $\mathbf{v}_I^G$ is the velocity of the body frame in the inertial frame, $\mathbf{b}_a$ is the bias of the accelerometer, $\mathbf{p}_I^G$ is the position of the body frame in the inertial frame, and the quaternion $\mathbf{q}_C^I$ and $\mathbf{p}_C^I$ are the relative transformation between the camera frame and the body frame. In the actual filter implementation, the error IMU state is used instead of the nominal state, and is defined as,

$$\tilde{\mathbf{x}}_I = \begin{pmatrix} \tilde{\boldsymbol{\theta}}_G^{I\,T} & \tilde{\mathbf{b}}_g^T & \tilde{\mathbf{v}}_I^{G\,T} & \tilde{\mathbf{b}}_a^T & \tilde{\mathbf{p}}_I^{G\,T} & \tilde{\boldsymbol{\theta}}_C^{I\,T} & \tilde{\mathbf{p}}_C^{I\,T} \end{pmatrix}^T, \quad (9)$$

where the relationship between the error IMU state and the nominal state is defined as,

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \approx \begin{pmatrix} \frac{1}{2}\tilde{\boldsymbol{\theta}}_I^{G\,T} & 1 \end{pmatrix}^T, \quad (10)$$

for the quaternions, and

$$\tilde{\mathbf{x}}_I = \mathbf{x}_I - \hat{\mathbf{x}}_I, \quad (11)$$

for the rest of the state, where $\tilde{\boldsymbol{\theta}}_I^G$ is a small angle rotation, and $\hat{\mathbf{x}}$ is the estimate of the state. As for the visual state, the error visual state is defined as,

$$\tilde{\mathbf{x}}_V = \begin{pmatrix} \tilde{\boldsymbol{\theta}}_G^{V\,T} & \tilde{\mathbf{p}}_V^G \end{pmatrix}^T. \quad (12)$$

With N number of visual states together with the error IMU state, the total error state vector can be represented as,

$$\tilde{\mathbf{x}} = \begin{pmatrix} \tilde{\mathbf{x}}_I^T & \tilde{\mathbf{x}}_{V_1}^T & \cdots & \tilde{\mathbf{x}}_{V_N}^T \end{pmatrix}^T. \quad (13)$$

The continuous filter propagation model of the IMU state is as follows,

$$\dot{\hat{\mathbf{q}}}_G^I = \frac{1}{2}\Omega(\hat{\boldsymbol{\omega}})\hat{\mathbf{q}}_G^I,$$

$$\dot{\hat{\mathbf{b}}}_g = \mathbf{0}_{3\times1},$$

$$\dot{\hat{\mathbf{v}}}^G = C(\hat{\mathbf{q}}_G^I)^T\hat{\mathbf{a}} + \mathbf{g}^G,$$

$$\dot{\hat{\mathbf{b}}}_a = \mathbf{0}_{3\times1},\ \dot{\hat{\mathbf{p}}}_I^G = \hat{\mathbf{v}}^G,$$

$$\dot{\hat{\mathbf{q}}}_V^I = \mathbf{0}_{3\times1},\ \dot{\hat{\mathbf{p}}}_V^I = \mathbf{0}_{3\times1}, \quad (14)$$

where $\hat{\boldsymbol{\omega}}$ is the angular velocity measurements, $\hat{\mathbf{a}}$ is the acceleration measurements, $C(\cdot)$ is the function converting quaternion to rotation matrix, and $\Omega(\hat{\boldsymbol{\omega}})$ is $\begin{pmatrix} -\hat{\boldsymbol{\omega}}_\times & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{pmatrix}$ with $\times$ denoting a skew symmetric matrix. With (14), we can derive the linearized continuous model for the error IMU state:

$$\dot{\tilde{\mathbf{x}}}_I = \mathbf{F}\tilde{\mathbf{x}}_I + \mathbf{G}\mathbf{n}_I, \quad (15)$$

where $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{n}_I$ are from [4]. The discretization of this model is done with the 4th order Runge-Kutta integration, where the discretized state transition matrix and process noise covariance matrix are as follows,

$$\boldsymbol{\Phi}_\mathrm{k} = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}(\tau)d\tau\right), \quad (16)$$

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1},\tau)\mathbf{G}\mathbf{Q}\mathbf{G}\boldsymbol{\Phi}(t_{k+1},\tau)^T d\tau, \quad (17)$$

where $\mathbf{Q}$ is the continuous process noise covariance matrix.

As for the filter update, we follow the same procedure as [4], similar to [27]. In summary, measurement update is performed 1) when a feature is lost or 2) when the maximum number of camera pose estimates is reached. In the former case, all measurements of the lost feature are used to update. In the latter case, using a keyframe selection method, two camera states are chosen to be removed, where features observed from these two states are used to update. The original measurement model of [4] is a stereo version, hence we use the monocular model shown below:

$$\mathbf{z}_i^j = \frac{1}{z_j^{V_i}}\begin{bmatrix} X_j^{V_i} \\ Y_j^{V_i} \end{bmatrix} + \mathbf{n}_i^j, \quad (18)$$

where $\mathbf{n}_i^j$ is the measurement noise vector. From (18), the measurement residual can be approximated as,

$$r_i^j = \mathbf{z}_i^j - \hat{\mathbf{z}}_i^j \simeq \mathbf{H}_{V_i}^j\tilde{\mathbf{x}}_{V_i} + \mathbf{H}_{f_i}^j\tilde{\mathbf{p}}_j^G + \mathbf{n}_i^j, \quad (19)$$

where the definitions of the measurement Jacobians $\mathbf{H}_{V_i}^j$ and $\mathbf{H}_{f_i}^j$ can be found in [4]. After stacking multiple observations of the same feature and projecting the residuals on the left null space of $\mathbf{H}_f^j$ to ensure no correlation between the residual and the uncertainty of $\mathbf{p}_j^G$, we can derive the final residual as,

$$r_o^j = \mathbf{H}_o^j\tilde{\mathbf{x}}^j + \mathbf{n}_o^j. \quad (20)$$

TABLE I
Quantitative Comparison of Pose Estimate Accuracy on the Event-Camera Dataset [5]

Dataset	Mean Position Error (%)				Mean Yaw Error (deg/m)			
	EVIO (E+I) [18]	USLAM (E+F+I) [24]	EKLT-VIO (E+F+I) [25]	Ours (E+F+I)	EVIO (E+I) [18]	USLAM (E+F+I) [24]	EKLT-VIO (E+F+I) [25]	Ours (E+F+I)
Boxes 6DOF	3.61	0.89	0.90	**0.77**	0.34	**0.04**	0.12	0.10
Boxes Translation	2.69	1.32	**0.51**	0.74	**0.09**	0.74	0.27	0.13
Poster 6DOF	3.56	0.72	**0.53**	0.59	0.56	0.08	**0.02**	**0.02**
Poster Translation	0.94	0.30	0.39	**0.28**	**0.02**	0.03	0.03	**0.02**
Shapes 6DOF	2.69	1.25	0.66	**0.42**	0.40	**0.03**	**0.03**	**0.03**
Shapes Translation	2.42	1.38	**0.58**	0.83	0.52	**0.02**	0.04	0.03
Dynamic 6DOF	4.07	0.93	0.98	**0.86**	0.56	0.11	0.09	**0.07**
Dynamic Translation	1.90	0.79	**0.45**	0.71	**0.02**	0.29	0.05	0.05
HDR Boxes	1.23	1.11	**0.57**	0.69	0.05	0.58	0.06	**0.04**
HDR Poster	2.63	1.38	0.74	**0.52**	0.11	0.20	0.06	**0.04**
Checkerboard	*unprovided*	1.44	1.20	**0.89**	*unprovided*	0.09	0.07	**0.02**
Boxes Rotation		*unfeasible*	**9.45**	9.82		*unfeasible*	1.71	**0.82**
Poster Rotation			3.05	**1.55**			0.24	**0.08**
Shapes Rotation			7.40	**5.38**			6.72	**4.30**
Dynamic Rotation			8.13	**7.37**			1.64	**1.20**
Average	2.57	1.05	0.68	**0.66**	0.27	0.20	0.08	**0.05**

Most VIO systems that adopt [27], including [4], uses a fixed measurement noise covariance matrix of the noise vector $\mathbf{n}_o^j$ when performing measurement update, where the matrix is calculated as,

$$\mathbf{R}_{\text{fixed}} = \sigma_V^2 \mathbf{I}_{2j-3}. \tag{21}$$

As mentioned previously, our proposed algorithm implements adaptive filtering through modulating $\mathbf{R}$ depending on the sizes of the warping residuals of the frontend, thus reflecting the uncertainty of the event-frame feature tracks. This way, the filter intelligently fuses the visual information and the propagated state by putting more weight on the measurement when the warping residual is relatively small, resulting in a more accurate pose estimation. As shown in Fig. 1, the first step of the frontend produces warping residual when optimizing a 2-DOF parameter, while the second step produces a residual when optimizing an 8-DOF parameter. Determining an appropriate size of the warping uncertainty with an absolute value is almost impossible as there is no clear standard to how small the residual should be for a good quality feature track. Thus, we relatively evaluate the current warping residual with the previous residuals. The minimized frontend residuals from (5) and (6) are stored for 20 steps each, and when a new residual is calculated, it is compared with the average of the previous 20 residuals. The ratio between the two values is then multiplied with $\mathbf{R}_{\text{fixed}}$ to adaptively tune the matrix corresponding to the uncertainty of the frontend. For instance, when the warping uncertainty is large, $\mathbf{R}$ is inflated to downplay the influence of the visual measurement in the filter state update, ultimately producing a more accurate pose estimate.

Although the relative size of the residual provides appropriate guidance in determining the reliability of the measurement, there are cases where it alone can produce inaccurate estimates. For example, if the residual is kept high for a long period, the moving average ratio would become low, although the measurement is in fact less reliable. Hence to prevent this, our algorithm incorporates an absolute threshold for the size of the residual to be considered too large. When above this threshold, no matter how small the moving average ratio is, the measurement is considered unreliable, and the filter is ultimately updated with $20\mathbf{R}_{\text{fixed}}$. The absolute threshold was set as 2px, which we derived from monitoring the warping residual when undergoing a somewhat inaccurate warping.

IV. Experiments

In this section, we quantitatively evaluate our proposed algorithm in two publicly available datasets: the Event-Camera Dataset [5] and the UZH-FPV Drone Racing Dataset [6]. The pose estimation performance of our algorithm is compared against state-of-the-art event-based VIO algorithms: EVIO [18], USLAM [24], and EKLT-VIO [25]. As mentioned in Section II, EVIO is an

TABLE II
Quantitative Comparison of Pose Estimate Accuracy on the UZH-FPV Drone Racing Dataset [6]

Dataset	Mean Position Error (%)		
	USLAM (E+F+I) [24]	EKLT-VIO (E+F+I) [25]	Ours (E+F+I)
Indoor Forward 3	5.05	**3.74**	3.90
Indoor Forward 5	6.24	4.14	**3.83**
Indoor Forward 6	5.58	4.35	**4.11**
Indoor Forward 7	9.71	5.88	**4.78**
Indoor Forward 9	4.32	**3.64**	3.95
Indoor Forward 10	3.96	3.20	**2.95**
Indoor 45 2	6.99	4.21	**3.68**
Indoor 45 4	6.10	4.76	**3.49**
Indoor 45 9	9.29	5.50	**4.10**
Indoor 45 12	7.39	3.93	**3.55**
Outdoor Forward 1	5.87	3.96	**3.52**
Outdoor Forward 3	9.45	5.78	**4.63**
Outdoor Forward 5	9.23	6.01	**4.74**
Average	6.89	4.55	**3.96**

event-inertial-fused VIO algorithm whereas the other two are event-frame-inertial-fused VIO algorithms.

A. *The Event-Camera Dataset [5]*

[5] provides event, frame, and inertial outputs recorded with DAVIS 240C [2] on variety of sequences with different conditions, including dynamic, 6-DOF, rotation-only, and HDR. Of those sequences, we chose 15 of the most frequently benchmarked to evaluate our VIO algorithm.

Table I shows comparative results of pose estimation accuracy. The notations E, F, and I next to the name of the algorithms stand for the use of event, frame, and inertial, respectively. Metrics used to compare follow those of [24] and [25]. For EVIO, the bottom 5 sequences are marked *unprovided*, as we were not able to find an open-source code and could only gather results that are reported in [18]. The results demonstrate that our proposed method outperforms state-of-the-art algorithms in 9 out of 15 sequences in terms of mean position error (MPE), and in 11 out of 15 sequences in terms of mean yaw error (MYE). We must pay attention to results on the bottom 4 rotation-only sequences. As stated in the table, results of USLAM are marked *unfeasible* as the algorithm is based on the initialization of the keyframe, and the rotation-only sequences did not provide enough translation to do so. On the contrary, EKLT-VIO and our algorithm was able to successfully initiate, where our method outperforms the former in both MPE and MYE, proving robustness to challenging situations. We claim that such improvement in accuracy, especially in challenging sequences, is mostly owed to the robust feature tracker,

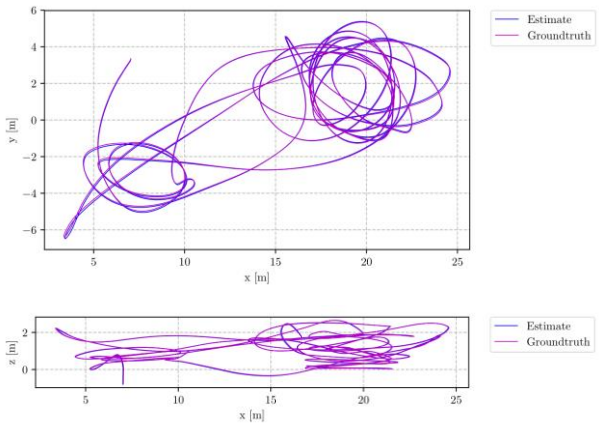

Fig. 3. Trajectory of the proposed method in Indoor Forward 7 [6].

which showed prolonged feature age [3] compared to the frontend of EKLT-VIO on the same dataset. The improvement is partly attributed to the adaptive nature of our backend. In such scenarios that are deemed challenging for the visual frontend, the measurement noise covariance matrix is inflated, reflecting the increased uncertainty in the frontend, hence the measurement update is less tainted.

B. *The UZH-FPV Drone Racing Dataset [6]*

[6] is comprised of data collected by miniDAVIS346 equipped on a drone racing quadrotor. The dataset is recorded in both indoor and outdoor environments, with the sensor facing forward and 45-degree downward, hence the namesake sequences. As the dataset serves as a benchmark for drone racing competitions, it is known to be the most aggressive and challenging event camera dataset. We believed such difficult nature of the dataset is what makes it felicitous to test accuracy and robustness of our proposed VIO algorithm. We evaluate state-of-the-art algorithms and ours on 13 sequences that vary in environments and difficulty.

Table II presents the comparative pose estimate accuracy results on the dataset. EVIO was deducted for this dataset for the same reason as the *unprovided* sequences of Table I. MYE is not evaluated as an issue with the ground truth was reported [33]. The results demonstrate that the proposed algorithm outperforms state-of-the-art methods in 11 out of 13 sequences in terms of MPE. In addition, not only does our method hold the lowest MPE in average, but also has the smallest discrepancies across the sequences. This implies that our algorithm provides a stable, yet accurate, pose estimates. Considering the difficulty of the dataset, the superior results further highlight the contributions of this work, proving robustness and accuracy in challenging situations. Fig. 3 shows the trajectory of the proposed algorithm against the ground truth on Indoor Forward 7, marked as *Hard* difficulty within the dataset. The trajectory was drawn using [34]. The trajectory once again validates that the proposed VIO system is capable of enduring highly dynamic situations with sustained accuracy.

V. Conclusion

In this letter, we presented an event- and frame-based visual-inertial odometry algorithm that robustly fuses events

and frames with 8-DOF warping, and accurately estimates pose through adaptive multi-state Kalman filter. We demonstrated the accuracy and robustness of the proposed algorithm using two datasets: the most benchmarked [5] and the most challenging [6]. The robustness of our frontend and the ability to cope with visually-challenging situations with the adaptive filter have proven to be effective, as our algorithm outperformed state-of-the-art methods in majority of the evaluated sequences. Our future work includes tightly-coupled framework that exploits inertial data in the process of fusing events and frames for a more accurate feature track.

REFERENCES

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-state Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[2] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE J. of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[3] M. S. Lee, Y. J. Kim, J. H. Jung, and C. G. Park, "Fusion of events and frames using 8-dof warping model for robust feature tracking," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 834-840.

[4] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, 2018.

[5] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, 2017.

[6] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? The UZH-FPV drone racing dataset," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 6713-6719.

[7] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2016, pp. 16–23.

[8] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 349–364.

[9] T. Moller and B. Trumbore, "Fast minimum storage ray-triangle intersection", *J. Graph. Tools*, vol. 2, no. 1, pp. 21-28, 1997.

[10] H. Rebecq, T. Horstschafer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real-time," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 593–600, 2017.

[11] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 121–130.

[12] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221– 255, 2004.

[13] H. Rebecq, G. Gallego, and D. Scaramuzza, "EMVS: Event-based multi-view stereo," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–11.

[14] G. Gallego, J. E. A. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, "Event-based, 6-dof camera tracking from photometric depth maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2402–2412, 2018.

[15] D. Liu, A. Parra, and T. -J. Chin, "Spatiotemporal registration for event-based visual odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 4937-4946.

[16] H. Kim and H. J. Kim, "Real-time rotational motion estimation with contrast maximization over globally aligned events", *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 6016-6023, 2021.

[17] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433-1450, 2021.

[18] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 5391–5399.

[19] J. H. Jung and C. G. Park, "Constrained filtering-based fusion of images events and inertial measurements for pose estimation", in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 644-650.

[20] M. Trajković and M. Hedley, "Fast corner detection", *Imag. Vis. Comput.*, vol. 16, no. 2, pp. 75-87, 1998.

[21] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, 2018.

[22] A. Patron-Perez, S. Lovegrove and G. Sibley, "A spline-based trajectory representation for sensor fusion and rolling shutter cameras", *Intl. J. Comput. Vis.*, vol. 113, no. 3, pp. 208-219, 2015.

[23] H. Rebecq, T. Horstschafer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1-8.

[24] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, 2018.

[25] F. Mahlknecht, D. Gehrig, J. Nash, F. M. Rockenbauer, B. Morrell, J. Delaune, and D. Scaramuzza, "Exploring event camera-based odometry for planetary robots", *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 8651-8658, 2022.

[26] J. Delaune, D. S. Bayard, and R. Brockers, "Range-visual-inertial odometry: Scale observability without excitation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2421–2428, 2021.

[27] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2007, pp. 3565–3572.

[28] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "EKLT: asynchronous photometric feature tracking using events and frames," *Int. J. Comput. Vis.*, vol. 128, pp. 601–618, 2020.

[29] N. Messikommer, C. Fang, M. Gehrig, and D. Scaramuzza, "Data-driven feature tracking for event cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 5642-5651.

[30] M. Gehrig, M. Millhäusler, D. Gehrig, and D. Scaramuzza, "E-RAFT: dense optical flow from event cameras," in *Proc. Int. Conf. 3D Vis.*, 2021, pp. 197-206.

[31] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: self-supervised optical flow estimation for event-based cameras," in *Proc. Robot.: Sci. Syst.*, 2018, pp. 1-9.

[32] D. Gehrig, M. Rüegg, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2822-2829, 2021.

[33] T. Cieslewski, G. Cioffi, and D. Scaramuzza, "Report on a rotation issue with the original uzh fpv dataset ground truth," Nov. 10 2020. [Online]. Available: https://fpv.ifi.uzh.ch/wp-content/uploads/2020/11/Ground-Truth-Rotation-Issue-Report.pdf

[34] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry", in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2018, pp. 7244-7251.