Event-IMU fusion strategies for faster-than-IMU estimation throughput

William Chamorro[1,2] Joan Solà[1] Juan Andrade-Cetto[1]
[1]Institut de Robòtica i Informàtica Industrial CSIC-UPC, Barcelona, Spain
[2] Universidad UTE, Quito, Ecuador
{wchamorro, jsola, cetto}@iri.upc.edu, william.chamorro@ute.edu.ec

Abstract

This study presents new methods for integrating event data and IMU readings to achieve ultra-fast camera pose estimates. The conventional predict-with-IMU-correct-with-vision approach is no longer optimal because events can be generated much more rapidly than IMU data. Therefore, two novel fusion schemes are proposed, which combine constant velocity and constant acceleration prediction models with ultra-fast (10 kHz) event-based updates and slower 1 kHz IMU updates. The first scheme uses IMU data as instantaneous measurements of acceleration and angular rate, while the second scheme considers these measurements as the average within the IMU sampling time. To provide a basis for comparison, the traditional method that predicts motion using IMU data and updates the estimates with event-feature matching over a 1ms time window is also implemented. All models are designed as Kalman filter variants, which act as the tracker module of a PTAM system in a human-made indoor scenario and are subjected to stress experiments to evaluate their capabilities. The models are also compared against an event-only estimator and a frame-based visual-inertial approach. The findings demonstrate superior performance at a throughput that is 100 times faster than the state-of-the-art.

1. Introduction

Event cameras have emerged as novel sensors that can handle high dynamic motion and challenging lighting conditions in computer vision since they were commercially available in 2008 [1, 15]. With the technological advancements in event cameras, there is a need to explore new techniques to fuse event data with other sources of information [8]. For instance, the combination of events and frames produced accurate rotational motion estimations between frames [3, 9], and event cameras and pulsed line sensors can be used to capture high-frequency light rays and reconstruct three-dimensional objects [2, 14].

Several works focus on solving the problem of visual-inertial odometry estimation using event cameras [13, 16, 18, 20, 21] and face differently the challenge of asynchronous event data arriving faster than the IMU and image frames. For instance, [21] proposes a method that uses IMU predictions and feature tracking over event windows with a multi-state constraint Kalman filter [12]. Despite achieving high-throughput rates of up to 100 Hz, this method suffers from a high computational cost. The work in [16] preintegrates inertial measurements and corrects with visual keyframes of motion-compensated event images minimizing reprojection error. The event window size used is sufficiently large to preintegrate several IMU readings. The extension in [16] also includes images in the process. Both approaches reach a top throughput of 100 Hz.

Our approach is more related to IDOL [11], which is a line-based visual-inertial system that tracks clusters of lines across multiple event windows to refine IMU preintegrated predictions through non-linear optimization. However, IDOL is known to have a high computational cost that limits its real-time performance. A more recent method, PL-EVIO [10], follows a similar approach to [16] but uses the Line Segment Detector (LSD) [19] and FAST corners [6] to detect features on motion-compensated images. Like IDOL, PL-EVIO employs the IMU preintegration scheme but achieves faster estimation rates of 30-100 Hz.

In this work, we propose novel strategies that do not follow the feature tracking and IMU preintegration schemes from most state-of-the-art approaches. Our approach does not create event images and relies only on small temporal windows of events. Our proposed camera tracking system uses the Lie Kalman filter architecture presented in [4], which uses only events and line features, and has a high throughput rate of up to 10kHz, at least one hundred times faster than the state-of-the-art. We integrate the proposed camera tracking approaches as part of a full Parallel Tracking and Mapping (PTAM) system to assess their performance in human-made environments and compare them against state-of-the-art methods. We use the mapping module presented in [5] to retrieve 3D lines from the scene using only event data, which works at a lower rate.

2. Events & IMU fusion

Our tracking system utilizes the Lie Kalman filter formulation proposed in [4]. This filter is designed to work with small event windows and can achieve a pose estimation rate that is up to 10 times faster than the DAVIS346 IMU sampling frequency. We describe here our strategy to deal with the fact that event-based poses are produced at a much higher rate than IMU readings. It uses both events and IMU data during the correction stage, each at its own sampling interval, producing estimates at the high rate of the event-based tracker. We consider two variants corresponding to two different assumptions on the nature of the IMU measurements, either instantaneous at the sampling time or averaged over a sampling period. For comparison, we also describe the classical strategy, which incorporates the IMU readings in the prediction stage and the events in the correction stage, thus requiring a reduction of the throughput to the lower rate of the IMU. In the following sections, we will provide a detailed explanation of these two strategies and their variants.

2.1. Events & IMU in correction stage

This strategy draws from the events-only correction mechanism proposed in [4], which utilizes an ultra-fast event-line data association algorithm over a small window of events. Here, we combine it with IMU corrections, which take place at central window time. When there are no IMU readings available, the correction is carried out solely with event data. The event window has a typical size of 100 μs, hence we incorporate one IMU reading every ten windows to account for the IMU rate of 1 kHz. Figure 1 summarizes the strategy.

2.1.1 Measurement models for IMU data

We consider two different assumptions on the nature of the IMU readings: instantaneous and averaged. Each of these models requires a particular parametrization of the state space, which are described in Sec. 2.2.1.

The use of IMU readings as instantaneous values at the sampling time yields what we call the Instantaneous Acceleration measurement model (IA):

$$\hat{\mathbf{a}}_{\mathbf{m}k} = {\mathbf{R}_k}^\top (\mathbf{a}_k - \mathbf{g}) + \mathbf{b}_{\mathbf{a}k} + \boldsymbol{\sigma}_{\mathbf{a}} \tag{1}$$

$$\hat{\boldsymbol{\omega}}_{\mathbf{m}k} = \boldsymbol{\omega}_k + \mathbf{b}_{\boldsymbol{\omega}k} + \boldsymbol{\sigma}_{\boldsymbol{\omega}}, \tag{2}$$

where $\hat{\mathbf{a}}_{\mathbf{m}k}$ and $\hat{\boldsymbol{\omega}}_{\mathbf{m}k}$ are the estimated IMU linear acceleration and angular velocity at the IMU reference frame $\mathcal{B}$, which are computed from the current estimates of the camera linear acceleration $\mathbf{a}_k$, its angular velocity $\boldsymbol{\omega}_k$, and orientation $\mathbf{R}_k \in SO(3)$.

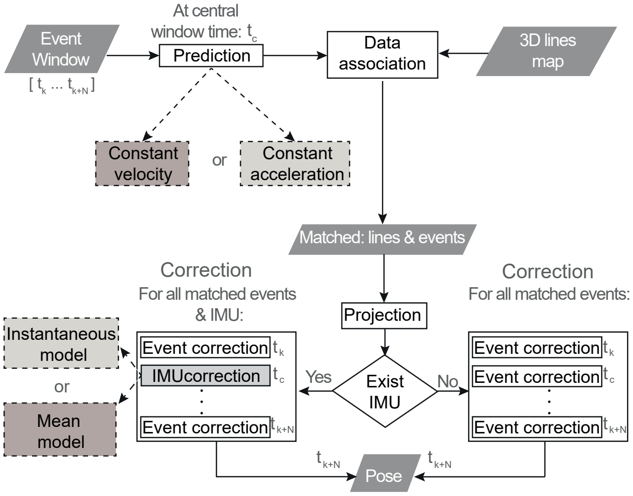

Figure 1. Strategy 1: Events and IMU in the correction stage.

The Mean Acceleration measurement model (MA) deals with IMU readings as mean values over a ΔT interval:

$$\hat{\mathbf{a}}_{\mathbf{m}k} = {\mathbf{R}_k}^\top \left(\frac{\mathbf{v}_k - \mathbf{v}_{k-j}}{\Delta T} - \mathbf{g} \right) + \mathbf{b}_{\mathbf{a}k} + \boldsymbol{\sigma}_{\mathbf{a}} \tag{3}$$

$$\hat{\boldsymbol{\omega}}_{\mathbf{m}k} = \frac{\mathbf{R}_k \ominus \mathbf{R}_{k-j}}{\Delta T} + \mathbf{b}_{\boldsymbol{\omega}k} + \boldsymbol{\sigma}_{\boldsymbol{\omega}}, \tag{4}$$

where we compute the mean acceleration from the velocity estimates at two consecutive IMU sampling times $\mathbf{v}_k$ and $\mathbf{v}_{k-j}$. In addition, the mean angular velocity is computed as an $SO(3)$ right minus operation between the current orientation and the orientation at the previous IMU sampling time $\mathbf{R}_{k-j}$. Note that the right minus operator $\ominus$ is defined with the $SO(3)$ logarithmic map as $\mathrm{Log}({\mathbf{R}_{k-j}}^{-1}\mathbf{R}_k)$, and ΔT is an IMU time step fixed at 1 ms for the DAVIS346 IMU sensor used in this work. Both measurement models include the gravity vector $\mathbf{g} = [0, 0, -9.81]^\top\ m/s^2$, the acceleration and angular velocity biases ($\mathbf{ba}_k$, $\mathbf{b}_{\boldsymbol{\omega}k}$ $\in \mathbb{R}^3$), and the sensor noise ($\boldsymbol{\sigma}_{\mathbf{a}}$, $\boldsymbol{\sigma}_{\boldsymbol{\omega}}$ $\in \mathbb{R}^3$).

The innovation related to IMU readings is computed as,

$$\mathbf{z} = \begin{bmatrix} \mathbf{a}_{\mathbf{m}} - \hat{\mathbf{a}}_{\mathbf{m}} \\ \boldsymbol{\omega}_{\mathbf{m}} - \hat{\boldsymbol{\omega}}_{\mathbf{m}} \end{bmatrix} \in \mathbb{R}^{6\times 1}, \tag{5}$$

which is the difference between the sensor readings of acceleration $\mathbf{a}_{\mathbf{m}}$, and angular velocity $\boldsymbol{\omega}_{\mathbf{m}}$, and the estimation from any of the measurement models.

The Jacobians of the innovation with respect to the state terms are derived using the $SO(3)$ Lie properties summarized in [17]. The Jacobians follow the notation $\mathbf{J}^{\mathbf{a}}_{\mathbf{b}} \triangleq \partial\mathbf{a}/\partial\mathbf{b}$, and their computations are aided with the chain rule resulting in the following expressions for each measurement model:

$$\text{IA model}\begin{cases} \mathbf{J}^{\mathbf{am}}_{\mathbf{R}} &= -\mathbf{R}^\top[\mathbf{a}-\mathbf{g}]_\times \\ \mathbf{J}^{\mathbf{am}}_{\mathbf{a}} &= -\mathbf{R}^\top \end{cases} \quad (6)$$

$$\text{MA model}\begin{cases} \mathbf{J}^{\mathbf{am}}_{\mathbf{R}} &= \left[\mathbf{R}^\top\left(\frac{\mathbf{v}_k-\mathbf{v}_{k-j}}{\Delta T}-\mathbf{g}\right)\right]_\times \\ \mathbf{J}^{\mathbf{am}}_{\mathbf{v_k}} &= \mathbf{R}^\top\Delta T^{-1} \\ \mathbf{J}^{\mathbf{am}}_{\mathbf{v_{k-j}}} &= -\mathbf{R}^\top\Delta T^{-1} \\ \mathbf{J}^{\omega\mathbf{m}}_{\mathbf{R_k}} &= \mathbf{J}_r{}^{-1}(\boldsymbol{\theta})\Delta T^{-1} \\ \mathbf{J}^{\omega\mathbf{m}}_{\mathbf{R_{k-j}}} &= \mathbf{J}_l{}^{-1}(\boldsymbol{\theta})\Delta T^{-1} \end{cases}, \quad (7)$$

where $\mathbf{J}_r{}^{-1}$ and $\mathbf{J}_l{}^{-1}$ stands for the $SO(3)$ inverse right and left Jacobians, which expressions are stated in the equations (144) and (146) in [17]. The Jacobians with respect to the biases are trivial and equal to the identity, and the operator $[\cdot]_\times$ denotes a skew-symmetric matrix.

The Jacobians summarized in (6) and (7) are part of an innovation Jacobian matrix $\mathbf{H} \in \mathbb{R}^{6\times m}$ with $m = 21$ for the IA model, and $m = 24$ for the MA model. The innovation covariance is computed as $\mathbf{Z} = \mathbf{HPH}^\top + \mathbf{N} \in \mathbb{R}^{6\times 6}$, where $\mathbf{P}$ stands for the state covariance of the transition models of each filter variant detailed in Sec.2.2.1. In addition, $\mathbf{N} = \text{diag}(\sigma_a{}^2\mathbf{I}, \sigma_\omega{}^2\mathbf{I}) \in \mathbb{R}^{6\times 6}$ is the noise matrix that contains the IMU sensor noise of linear acceleration $\sigma_\mathbf{a}$ $[m/s^2]$, and angular velocity $\sigma_{\boldsymbol{\omega}}$ $[rad/s]$.

2.2. Measurement model for segment observations

The three-dimensional segments' endpoints $\mathbf{p}_j\ \forall\ j \in \{1,2\}$ are projected to the image plane as follows,

$$\underline{\mathbf{u}}_j = \mathbf{K}^{\mathcal{B}}\mathbf{R}_{\mathcal{C}}{}^\top(\mathbf{R}^\top(\mathbf{p}_j-\mathbf{r})-\mathbf{r}^{\mathcal{B}}_{\mathcal{C}}) \in \mathbb{P}^2, \quad (8)$$

where $\mathbf{r}$ and $\mathbf{R}$ are the system pose computed at IMU reference frame, $\mathbf{K}$ is the intrinsic camera matrix, and $\{\mathbf{r}^{\mathcal{B}}_{\mathcal{C}}, {}^{\mathcal{B}}\mathbf{R}_{\mathcal{C}}\}$ are the IMU-to-camera extrinsic parameters.

In the absence of IMU data, the partial event innovations belong to the signed event-line distance, which is computed considering the projected line $\mathbf{l} = \underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2 = [a,b,c]^\top$ and the event $\underline{\mathbf{e}}$ associated to a line as follows,

$$z = d(\mathbf{e},\mathbf{l}) = \frac{\underline{\mathbf{e}}^\top\mathbf{l}}{\sqrt{a^2+b^2}} \qquad \in \mathbb{R}\,. \quad (9)$$

The innovation covariance is $Z = \mathbf{HPH}^\top + \sigma_d{}^2 \in \mathbb{R}$, which results in scalar term. The measurement noise is denoted with σ_d $[pix]$, and the innovation Jacobians $\mathbf{H}$ of this measurement model are fully detailed in [4].

Recall that the data association algorithm is the same used in [4] and [5] and is fully detailed in these works. This algorithm allows for discarding or validating events as fast as possible. The fast operation is the result of the application of three simple but effective steps: the visible line projection onto the image plane in an event window, Fig. 2a; the image tessellation for a fast event searching, Fig. 2b; and the event thresholding considering a minimum event-line distance α $[pix]$ and the distance with respect closer lines β $[pix]$, Fig. 2c.

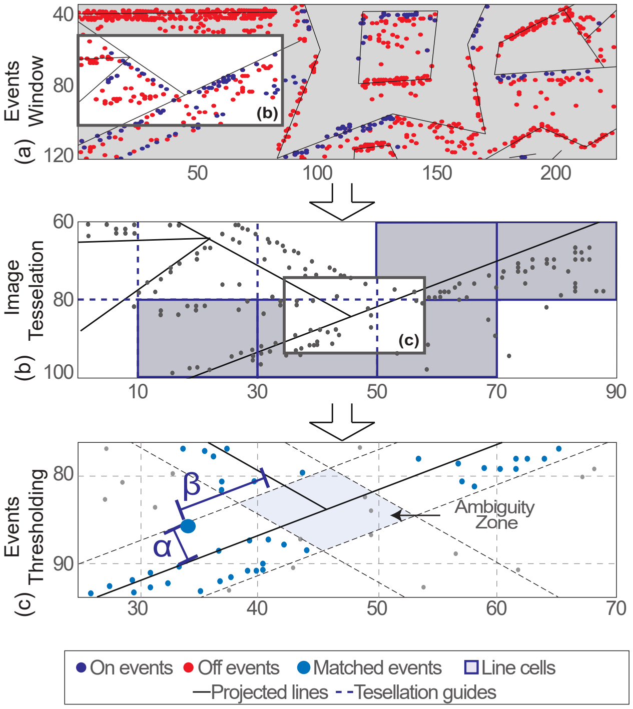

Figure 2. Data association algorithm [4]

2.2.1 State representations and prediction models

Depending on the measurement model chosen, a specific prediction model is to be used in the filter. Two variants are proposed: the constant acceleration (CA) model combined with the instantaneous measurement model (CA + IA), and the constant velocity (CV) model corrected with the mean measurement model (CV + MA). Note that both variants use event-only corrections in between IMU readings, as shown in Fig.1.

The state transitions for both variants are shown in Table 1, where in CV+MA, $\mathbf{v}_{k-j}$ and $\mathbf{R}_{k-j}$ denote the delayed-state IMU pose at the time when the previous IMU data arrived. These terms are updated with current values at time k after the correction stage, and Δt is the event window time step, which is set to 100 μs in our experiments.

The perturbation in all models is modeled as a Gaussian noise with mean zero and a covariance, which is computed as a random impulse integrated over a timestep Δt. The following expressions detail the covariance name associate the each noise term and the integration result as follows:

Table 1. State transitions for CA+IA and CV+MA variants

$\mathbf{x}_{k+1} = f($	$\mathbf{CA + IA}$	$\mathbf{CV + MA}$)	$\in \mathcal{M}$
$\mathbf{r}_{k+1} =$	$\mathbf{r}_k + \mathbf{v}_k \Delta t + \frac{1}{2}\mathbf{a}_k \Delta t^2$	$\mathbf{r}_k + \mathbf{v}_k \Delta t$	$\in \mathbb{R}^3$
$\mathbf{v}_{k+1} =$	$\mathbf{v}_k + \mathbf{a}_k \Delta t$	$\mathbf{v}_k + \mathbf{v_n}$	$\in \mathbb{R}^3$
$\mathbf{a}_{k+1} =$	$\mathbf{a}_k + \mathbf{a_n}$	$-$	$\in \mathbb{R}^3$
$\mathbf{v}_{k-j} =$	$-$	$\mathbf{v}_{k-j}$	$\in \mathbb{R}^3$
$\mathbf{R}_{k-j} =$	$-$	$\mathbf{R}_{k-j}$	$\in SO(3)$
$\mathbf{R}_{k+1} =$	$\mathbf{R}_k \oplus \{\boldsymbol{\omega}_k \Delta t\}$		$\in SO(3)$
$\boldsymbol{\omega}_{k+1} =$	$\boldsymbol{\omega}_k + \boldsymbol{\omega}_\mathbf{n}$		$\in \mathbb{R}^3$
$\mathbf{b}_{\mathbf{a}k+1} =$	$\mathbf{b}_{\mathbf{a}k} + \mathbf{b}_{\mathbf{an}}$		$\in \mathbb{R}^3$
$\mathbf{b}_{\boldsymbol{\omega}k+1} =$	$\mathbf{b}_{\boldsymbol{\omega}k} + \mathbf{b}_{\boldsymbol{\omega}\mathbf{n}}$		$\in \mathbb{R}^3$

Table 2. State transition for $\mathbf{IMU + e}$ variant

$\mathbf{x}_{k+1} = f($	$\mathbf{IMU + e}$	$\in \mathcal{M}$
$\mathbf{r}_{k+1} =$	$\mathbf{r}_k + \mathbf{v}\Delta T$	$\in \mathbb{R}^3$
$\mathbf{v}_{k+1} =$	$\mathbf{v}_k + (\mathbf{R}(\mathbf{a_m} - \mathbf{b}_{\mathbf{a}k}) + \mathbf{g} + \boldsymbol{\sigma}_\mathbf{a})\Delta T$	$\in \mathbb{R}^3$
$\mathbf{R}_{k+1} =$	$\mathbf{R}_k \oplus \{(\boldsymbol{\omega}_\mathbf{m} - \mathbf{b}_{\boldsymbol{\omega}k} + \boldsymbol{\sigma}_{\boldsymbol{\omega}})\Delta T\}$	$\in SO(3)$
$\mathbf{b}_{\mathbf{a}k+1} =$	$\mathbf{b}_{\mathbf{a}k} + \mathbf{b}_{\mathbf{an}}$	$\in \mathbb{R}^3$
$\mathbf{b}_{\boldsymbol{\omega}k+1} =$	$\mathbf{b}_{\boldsymbol{\omega}k} + \mathbf{b}_{\boldsymbol{\omega}\mathbf{n}}$	$\in \mathbb{R}^3$

$$
\begin{array}{lll}
\mathbf{v_n} \sim \mathcal{N}(0, \mathbf{V_i}) & \text{with} & \mathbf{V_i} = {\sigma_{\mathbf{vn}}}^2 \Delta t \mathbf{I} \\
\boldsymbol{\omega}_\mathbf{n} \sim \mathcal{N}(0, \boldsymbol{\Omega}_\mathbf{i}) & \text{with} & \boldsymbol{\Omega}_\mathbf{i} = {\sigma_{\boldsymbol{\omega}\mathbf{n}}}^2 \Delta t \mathbf{I} \\
\mathbf{a_n} \sim \mathcal{N}(0, \mathbf{A_i}) & \text{with} & \mathbf{A_i} = {\sigma_{\mathbf{an}}}^2 \Delta t \mathbf{I} \\
\mathbf{b}_{\mathbf{a_n}} \sim \mathcal{N}(0, \mathbf{Ba_i}) & \text{with} & \mathbf{Ba_i} = {\sigma_{\mathbf{ban}}}^2 \Delta t \mathbf{I} \\
\mathbf{b}_{\boldsymbol{\omega}_\mathbf{n}} \sim \mathcal{N}(0, \mathbf{B}\boldsymbol{\omega}_\mathbf{i}) & \text{with} & \mathbf{B}\boldsymbol{\omega}_\mathbf{i} = {\sigma_{\mathbf{b}\boldsymbol{\omega}\mathbf{n}}}^2 \Delta t \mathbf{I}
\end{array}
\quad , \qquad (10)
$$

where $\sigma_{\mathbf{vn}}[m/s\sqrt{s}]$, $\sigma_{\boldsymbol{\omega}\mathbf{n}}[rad/s\sqrt{s}]$, $\sigma_{\mathbf{an}}[m/s^2\sqrt{s}]$, $\sigma_{\mathbf{ban}}[m/s^2\sqrt{s}]$, are noise constants, and $\sigma_{\mathbf{b}\boldsymbol{\omega}\mathbf{n}}[rad/s\sqrt{s}]$ are bias drifts. These constants were tuned empirically based on characteristics of the experiments and the IMU sensor specifications.

The state covariance is propagated as $\mathbf{P}_{k+1} = \mathbf{F}\mathbf{P}_k\mathbf{F}^\top + \mathbf{Q}$, where $\mathbf{F}$ is Jacobian matrix with respect to the state derived following $SO(3)$ lie group properties and the state transitions of Table 1. $\mathbf{Q}$ is the perturbation covariance conformed with the random impulses in (10), and built according to the noise terms of each variant detailed in Table 1. The covariance matrices have a size of $m \times m$ with $m = 21$ for the CA+IA variant and $m = 24$ for CV+MA.

2.3. Prediction with IMU and correction with events

For sake of completeness, we also implement a second strategy that incorporates the IMU readings in the prediction stage; as a result, each event window is now centered at IMU time and has a fixed size of 1 ms, as detailed in Fig. 3.

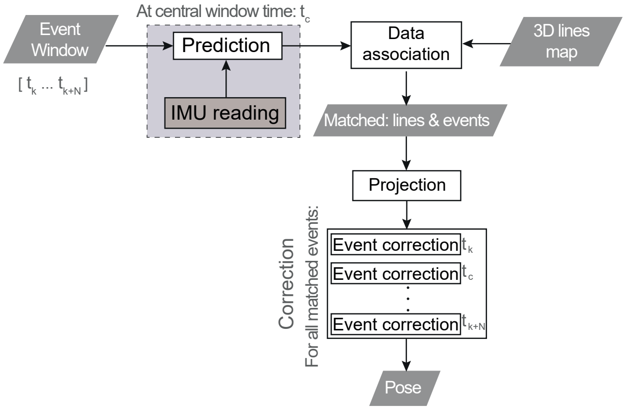

Figure 3. Strategy 2: Prediction with IMU and correction with events.

The resulting variant of the Lie Kalman filter is denoted IMU + e because it incorporates both IMU and event data. The state transition follows the classic visual-inertial formulation detailed in Table 2, where the sensor bias and noise is included in the linear velocity and orientation estimations. But in contrast to frame-based visual-inertial methods in the state of the art, our method is 10 to 30 times faster, thanks to the use of event windows of 1ms and centered at IMU sampling times. The state Jacobians are computed using the $SO(3)$ Lie properties, where the nontrivial expressions are listed as follows,

$$
\begin{array}{lll}
\mathbf{J}_\mathbf{R}^\mathbf{v} & = & -\mathbf{R}[\mathbf{a_m} - \mathbf{b_a}]_\times \\
\mathbf{J}_\mathbf{ba}^\mathbf{v} & = & -\mathbf{R}\Delta t \\
\mathbf{J}_\mathbf{R}^\mathbf{R} & = & \mathbf{R}\{(\boldsymbol{\omega}_\mathbf{m} - \mathbf{b}_\omega)\}^\top \\
\mathbf{J}_{\mathbf{b}\omega}^\mathbf{R} & = & -\mathbf{J}_r(\boldsymbol{\omega}_\mathbf{m} - \mathbf{b}_\omega)\Delta T
\end{array}
\qquad (11)
$$

The correction for the IMU+e variant follows the same approach described in Sec. 2.2, which uses the signed event-to-line function stated in (9).

2.4. Events & IMU in a PTAM system

Our proposed sensor fusion variants enable camera pose estimation in conjunction with a mapping module within a PTAM system. We leverage the mapping formulation from [5], which allows us to recover 3D lines from the scene. The mapping module builds a spatial grid at a keyframe position, which is intersected by back-projected rays from events in several event windows. The tracking module determines the pose of each event window. The grid is divided into voxels, which are populated with votes from each ray that passes through them. Voxels with a greater number of votes are more likely to contain a 3D point. We can retrieve a 3D line from a set of 3D points aligned in straight patterns located in straight edges.

Since the mapping and tracking modules operate independently, we use a local bundle adjustment module to optimize 3D lines and camera poses using a cost function that minimizes the distance between the event and line in the image plane. For more details on the mapping process, see [5]. In section 3.1, we present the results of a PTAM application that uses our proposed sensor fusion variants.

Table 3. RMSE for event-only and event-IMU PTAM configurations and for visual-inertial SVO.

Dataset		CV	CA	CA+IA	CV+MA	IMU+E	SVO
Office	Pos. [cm]	3.87	3.91	3.79	**3.68**	3.75	4.77
L_shape	Rot. [rad]	0.23	0.24	0.23	**0.22**	0.23	0.25
Office	Pos. [cm]	6.62	6.58	6.60	6.53	**6.38**	6.88
large	Rot. [rad]	0.23	0.23	0.22	0.22	**0.21**	0.22
Trihedron	Pos. [cm]	3.31	3.46	3.27	**3.12**	3.15	8.20
	Rot. [rad]	0.20	0.18	0.17	0.17	**0.16**	0.18
Office	Pos. [cm]	3.84	3.77	3.72	3.71	3.69	**3.61**
far	Rot. [rad]	0.27	0.26	0.27	0.26	0.26	**0.25**
Pose throughput	[kHz]	**10**	**10**	**10**	**10**	1	0.033*
Stream Processing time	[ms]	11.6	14.5	15.20	13.90	**10.9**	22.22

*Throughput limited by the DAVIS346 camera

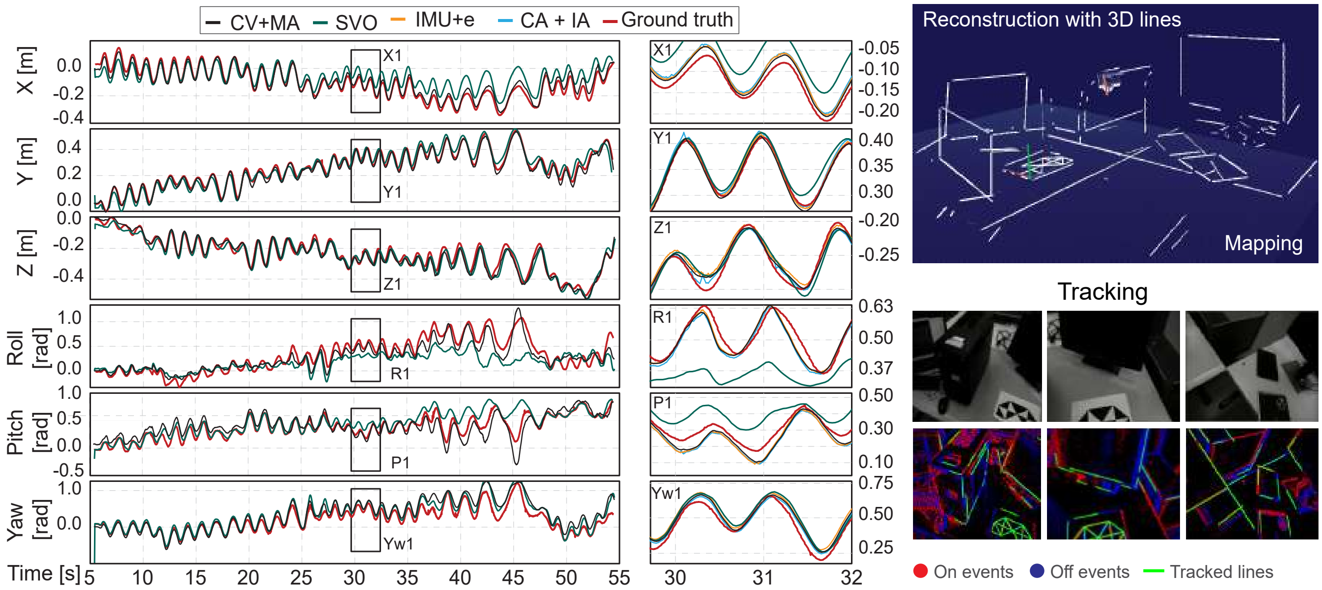

Figure 4. Trajectory and 3D reconstruction with PTAM system using CV + MA variant in *Office L shape* scenario. The tracked lines are highlighted in green in the tracking snapshots, and the line map was captured from Rviz at the end of the sequence.

3. Experiments and results

To evaluate the performance of all filter variants, we propose two experiments. Firstly, we integrate the tracking system into a PTAM (Parallel Tracking and Mapping) system to assess the pose estimation accuracy while creating 3D line-based maps. Secondly, we conduct a stress experiment to assess the estimation limits in high dynamics. We use the DAVIS346 event camera in all experiments, which includes an MPU-6500 IMU.

The sensor noise and bias drifts are tuned to IMU specifications. The used values for all variants are $\sigma_{\mathbf{a}} = 0.93\ [m/s^2]$, $\sigma_{\boldsymbol{\omega}} = 0.017\ [rad/s]$, $\sigma_{\mathbf{ban}} = 2.94\ [m/s^2\sqrt{s}]$, $\sigma_{\mathbf{b\omega n}} = 0.17\ [rad/s\sqrt{s}]$. The reminding noise constants were set experimentally in order to get a response with the lowest possible error, these parameters are $\sigma_{\mathbf{vn}} = 3\ [m/s\sqrt{s}]$, $\sigma_{\boldsymbol{\omega}\mathbf{n}} = 7\ [rad/s\sqrt{s}]$, and, $\sigma_{\mathbf{an}} = 150\ [m/s^2\sqrt{s}]$. The PTAM system runs in a standard multicore CPU with Ubuntu 18.04.6 and ROS Melodic.

3.1. PTAM evaluation

The mapping node of [5] was used to recover 3D lines for all tracking variants, and it runs at a lower asynchronous rate of about 10 Hz. The PTAM's tracking module was set up for the event-only variants of CV and CA, as extensively described in [4], and for the three proposed sensor fusion

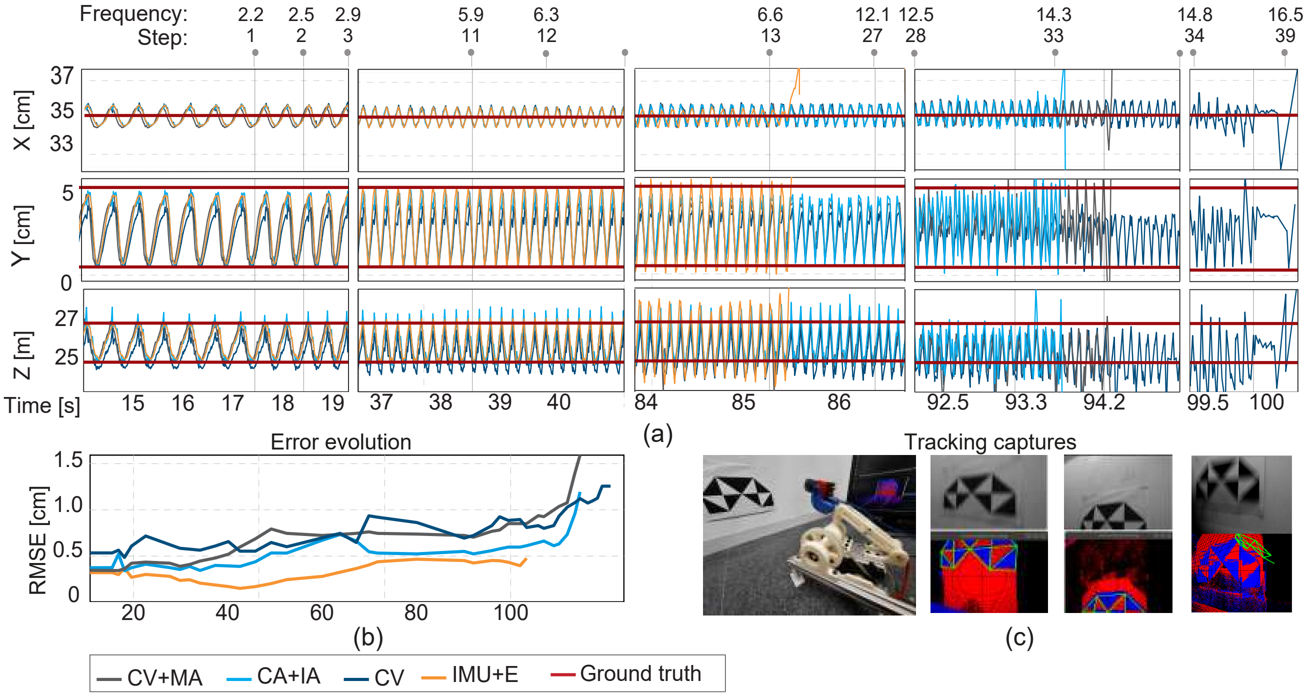

Figure 5. Position estimation in stress test from 240 rpm to 990 rpm with CV+MA, CV+IA, and IMU+e variants: (a) position estimation, (b) error evolution, (c) four-bar mechanism with tracking snapshots until the system failure.

variants: CV+MA, CA+IA, and IMU+e. Additionally, for comparison purposes, the frame-based visual-inertial approach SVO [7] was also tested on our sequences. Table 3 summarizes the scenarios and the mean pose estimation errors, compared to Optitrack ground truth.

The tracking variants and the mapping node run in parallel threads. Our efficient implementation of CV+MA and CA+IA allows reaching a no-lag real-time throughput of up to 10kHz, with event windows of 100μs. Event windows smaller than 100μs diminished the mapping accuracy due to the reduced amount of events per window. SVO works at a frame rate of 30 Hz for the DAVIS346 camera. Note that according to [7] SVO is capable of running at 300 fps showing real-time performance. In our experiments the throughput of SVO is limited by the DAVIS camera.

A PTAM performance example in the office scenario is shown in Fig. 4. Notice that whereas SVO has large drifts in some short intervals, all the proposed methods better match the ground truth trajectory. The mean errors in Table 3 show an overall accuracy improvement of about 5% in all sensor fusion variants with respect to event-only ones. The best-achieving variants were CV+MA and IMU+e, which present better results than the frame-based approach, probably because the scenarios are richer in straight edges than texture. IMU+e was the most accurate approach but limited to a throughput of 1kHz. Its accuracy is about 5% higher than CV+MA or CA+IA. While our approach works well in human-made environments where straight lines are prevalent and provide stable estimations of camera pose, it may face challenges in natural scenarios where straight lines are absent. Retrieving small, noisy lines in such environments can compromise the accuracy of our estimations. Therefore, further research is needed to improve the robustness of the approach in natural scenarios. Nevertheless, our approach remains a promising solution for fast and accurate camera pose estimations in human-made environments.

The time required to process all event windows within a 33ms event stream is named *stream processing time*, and it is displayed in the last row in Table 3. The sensor fusion variants show an increase in computational time of about 6% with respect to the event-only CV and CA methods. IMU+e was the cheapest option given its smaller state size. The computational time of the mapping thread is the same reported in [5] due to there were no modifications proposed in our approach.

3.2. Stress evaluation

In this experiment, the sensor fusion variants were tested to determine their tracking limits under high dynamics. The experiment involved an event camera placed on top of a four-bar mechanism that observes a statically known marker, as shown in Fig.5c. The marker is used as a line map to estimate the camera pose in high dynamics conditions The device's kinematics and dimensions are well known, providing ground truth for the experiment.

The DC motor powering the device was discretely increased from 132 rpm (2.2Hz) until tracking failed. Fig. 5a shows that the IMU+e (dark yellow) failed first at 750 rpm (12.5 Hz), followed by CA+IA (cyan) at 858 rpm (14.3 Hz) and CV+MA (gray) at 870 rpm (14.5 Hz), while the event-only variant CV (dark blue) reached the maximum speed of 990 rpm (16 Hz).

Although the sensor fusion variants failed before the event-only variant, they provided better estimations than CV, as illustrated in the error evolution plot in Figure 5b. Specifically, Table 3 shows that IMU+e produced the most accurate pose estimations, followed by CV+MA and CA+IA, whose error ranges were similar.

However, the main cause of failure for the sensor fusion variants was the acceleration exceeding the MPU-6500 acceleration limit of 16g at 750 rpm, leading to saturation. This saturation issue was verified by comparing the estimated and measured linear acceleration signals, as shown in Figure 6. The estimated acceleration (light blue in Figure 6) exceeded 16g, but the IMU sensor (dark blue in Figure 6) can not produce measurements beyond that value, which introduced errors in the estimations. Towards the end of the experiment, CV experienced strong vibrations over 25g, which caused the tracking estimation to degrade. Due to the strong vibrations in the mechanism, experiments with accelerations over 25g were not conducted to avoid damaging the event camera.

4. Conclusions

This work presents a comprehensive evaluation of different event-camera tracking variants, including event-only and sensor fusion approaches, for ultra-fast response in tracking and pose estimation.

The proposed variants, CV+MA and CA+IA, as well as IMU+e, showed an accuracy improvement of approximately 6% over event-only approaches. IMU+e was found to be the most accurate, albeit with limited throughput. The proposed tracking architecture achieved a throughput of about 10 kHz with CA+IA and CV+MA, and 1 kHz with IMU+e, which is significantly faster than frame-based approaches like SVO (limited to 30Hz due to DAVIS model design). Although the computational burden of sensor fusion variants is slightly higher than that of event-only methods, real-time performance is maintained.

The stress experiment showed that the sensor fusion variants failed before event-only ones due to the acceleration exceeding the sensor acceleration limit of 16g. Overall, the results demonstrate the potential of sensor fusion approaches to achieve event-based pose estimation with a throughput of 1kHZ to 10kHz and accuracy superior to event-only or image-based methods in human-made environments.

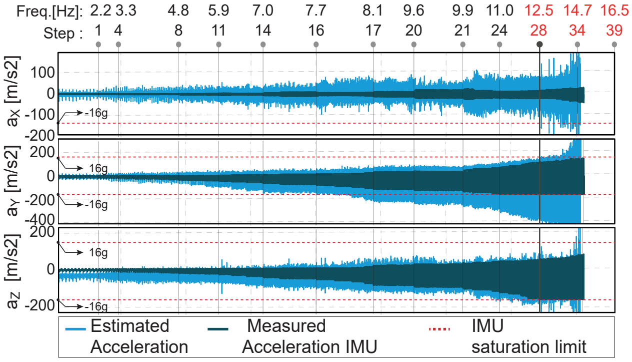

Figure 6. Linear acceleration measured and estimated during the stress experiment. Note: IMU acceleration is saturated after 750 rpm (12.5 Hz) where the camera experiences an acceleration higher than 16g

Acknowledgments

This work received support from projects EBCON (PID2020-119244GB-I00) and AUDEL (TED2021-131759A-I00) funded by MCIN/ AEI/ 10.13039/501100011033 and by the ”European Union NextGenerationEU/PRTR”, and the Consolidated Research Group RAIG (2021 SGR 00510) of the Departament de Recerca i Universitats de la Generalitat de Catalunya.

References

[1] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits*, 49(10):2333–2341, 2014. 1

[2] Christian Brandli, Thomas A. Mantel, Marco Hutter, Markus A. Höpflinger, Raphael Berner, Roland Siegwart, and Tobi Delbruck. Adaptive pulsed laser line extraction for terrain reconstruction using a dynamic vision sensor. *Frontiers in Neuroscience*, 7(8):275, 2014. 1

[3] Andrea Censi and Davide Scaramuzza. Low-latency event-based visual odometry. In *IEEE Int. Conf. Robotics and Automation*, pages 703–710, 2014. 1

[4] William Chamorro, Juan Andrade-Cetto, and Joan Solà. High-speed event camera tracking. In *British Machine Vision Conf.*, pages 1–12, Jan 2020. 1, 2, 3, 5

[5] William Chamorro, Joan Solà, and Juan Andrade-Cetto. Event-based line SLAM in real-time. *IEEE Robotics and Automation Lett.*, 7(3):8146 – 8153, 2022. 1, 3, 4, 5, 6

[6] Rosten Edward, Porter Reid, and Drummond Tom. Faster and better: A machine learning approach to cor-

ner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(1):105–119, 2010. 1

[7] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robotics*, 33(2):249–265, 2016. 6

[8] 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 Trans. Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2020. 1

[9] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Asynchronous, photometric feature tracking using events and frames. In *Eur. Conf. Computer Vision*, pages 750–765, 2018. 1

[10] Weipeng Guan, Peiyu Chen, Yuhan Xie, and Peng Lu. PL-EVIO: Robust Monocular Event-based Visual Inertial Odometry with Point and Line Features. *arXiv preprint arXiv:2209.12160*, 2022. 1

[11] Cedric Le Gentil, Florian Tschopp, Ignacio Alzugaray, Teresa Vidal-Calleja, Roland Siegwart, and Juan Nieto. IDOL: A framework for imu-dvs odometry using lines. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 5863–5870, 2020. 1

[12] Anastasios I Mourikis, Stergios I Roumeliotis, et al. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Int. Conf. Robotics and Automation*, volume 2, pages 3565–3572, 2007. 1

[13] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Trans. Robotics*, 34(6):1425–1440, 2018. 1

[14] Manasi Muglikar, Guillermo Gallego, and Davide Scaramuzza. ESL: Event-based structured light. In *IEEE Int. Conf. 3D Vision*, pages 1165–1174, 2021. 1

[15] Lichtsteiner Patrick, Christoph Posch, and Tobi Delbruck. A 128 x 128 120 dB 15μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits*, 43:566–576, 2008. 1

[16] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. *British Machine Vision Conf.*, 2017. 1

[17] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018. 2, 3

[18] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. *IEEE Robotics and Automation Lett.*, 3(2):994–1001, 2018. 1

[19] Rafael Grompone Von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: a line segment detector. *J. Image Processing Online*, 2:35–55, 2012. 1

[20] Yingxun Wang, Bo Shao, Chongchong Zhang, Jiang Zhao, and Zhihao Cai. REVIO: Range-and Event-Based Visual-Inertial Odometry for Bio-Inspired Sensors. *Biomimetics*, 7(4):169, 2022. 1

[21] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 5391–5399, 2017. 1