Flow-Based Visual-Inertial Odometry for Neuromorphic Vision Sensors Using non-Linear Optimization with Online Calibration

Mahmoud Z. Khairallah*[a], Abanob Soliman*[b], Fabien Bonardi†[c], David Roussel†[d]
and Samia Bouchafa†[e]

Université Paris-Saclay, Univ. Evry, IBISC Laboratory, 34 Rue du Pelvoux, Evry, 91020, Essonne, France
* *fi* _ _ *fi* _ _

Keywords: Neuromorphic Vision Sensors, Optical Flow Estimation, Visual-Inertial Odometry.

Abstract: Neuromorphic vision sensors (also known as event-based cameras) operate according to detected variations in the scene brightness intensity. Unlike conventional CCD/CMOS cameras, they provide information about the scene with a very high temporal resolution (in the order of microsecond) and high dynamic range (exceeding 120 dB). These mentioned capabilities of neuromorphic vision sensors induced their integration in various robotics applications such as visual odometry and SLAM. The way neuromorphic vision sensors trigger events is strongly coherent with the brightness constancy condition that describes optical flow. In this paper, we exploit optical flow information with the IMU readings to estimate a 6-DoF pose. Based on the proposed optical flow tracking method, we introduce an optimization scheme set up with a twist graph instead of a pose graph. Upon validation on high-quality simulated and real-world sequences, we show that our algorithm does not require any triangulation or key-frame selection and can be fine-tuned to meet real-time requirements according to the events' frequency.

1 INTRODUCTION

By providing frame-free asynchronous data, event-based cameras are designed to trigger events and react to changes in brightness in the scene whenever detected. These sensors are designed to mimic the activities of the biological retina and do not depend on any artificial clock signals. The asynchronous nature of event-based cameras enables them to suppress redundant data (compared to frame-based cameras), provide high temporal resolution and high dynamic range with low power consumption. These sensors provide a convenient replacement for frame-based vision sensors in scenarios presenting high dynamics such as drone motion.

For the past decade, many solutions have been introduced to integrate event-based cameras in robotic applications: for instance, (Kim et al., 2008), (Mueggler et al., 2014), (Rebecq et al., 2017a) and (Mueggler et al., 2018) provide accurate motion estimation. Amongst the adopted approaches to solve this problem, different probabilistic filtering methods have been introduced in (Kim et al., 2008), (Kim et al., 2016), (Weikersdorfer and Conradt, 2012) and (Weikersdorfer et al., 2013). Other methods like (Mueggler et al., 2014), (Kueng et al., 2016) and (Weikersdorfer et al., 2014) used different optimization schemes to benefit from their higher accuracy to estimate motion.

Event-based cameras' ability to provide asynchronous data with significantly high temporal resolution leads to better continuous representation compared to frame-based cameras, as well as eliminating other problems such as motion blur and low dynamic range. Furthermore, this ability provides a more stable mathematical modeling of the brightness constancy condition, which describes the apparent pixels motion known as the optical flow. In this paper, we introduce, to the extent of our knowledge, the first visual-inertial odometry algorithm that jointly optimizes the events' optical flow with the inertial measurements for neuromorphic vision sensors.

2 RELATED WORK

The change in vision sensors nature proposed by event-based cameras required a paradigm shift on how the visual odometry problem is modeled and how it can be solved. During the past decade, many at-

[a] https://orcid.org/0000-0002-0724-8450
[b] https://orcid.org/0000-0003-4956-8580
[c] https://orcid.org/0000-0002-3555-7306
[d] https://orcid.org/0000-0002-1839-0831
[e] https://orcid.org/0000-0002-2860-8128

Khairallah, M., Soliman, A., Bonardi, F., Roussel, D. and Bouchafa, S.
Flow-Based Visual-Inertial Odometry for Neuromorphic Vision Sensors Using non-Linear Optimization with Online Calibration.
DOI: 10.5220/0011660400003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 963-973
ISBN: 978-989-758-634-7; ISSN: 2184-4321

tempts were introduced where some adapted the acquired data from event-based cameras to suit frame-based algorithms (Gehrig et al., 2020; Muglikar et al., 2021) in order to create frames from event-based cameras, while others reformulated the problem to fully exploit event-based capabilities (Zhou et al., 2021; Rebecq et al., 2017a; Rebecq et al., 2018). A novel method was presented in (Weikersdorfer and Conradt, 2012) using a particle filter for motion tracking to estimate the camera's rotation by creating mosaic images of the scene, while an extended Kalman filter is used to refine the gradient intensity results. In (Weikersdorfer et al., 2013), a particle filter is used to estimate the 2D motion of the used rig based on the work presented in (Weikersdorfer and Conradt, 2012) and a 2D map is simultaneously reconstructed. Mueggler et al. (Mueggler et al., 2014) developed a 6-DoF motion estimation for simple, uncluttered and structured environments that contain lines where the pose is estimated by minimizing the reprojection error of each detected line in the environment. Rebecq et al. (Rebecq et al., 2017a) proposed an event-based tracking and mapping method to estimate the pose based on image alignment by warping event images using Lucas-Kanade method (Baker and Matthews, 2004) and constructed the map thanks to the event-based space-sweep presented in (Rebecq et al., 2018) to provide depth and 3D map. Kim et al. (Kim et al., 2016) pursued their work in (Kim et al., 2008) using an extended Kalman filter to estimate pose, gradient intensity and mapping implemented using a GPU.

Enhancing the robustness and accuracy of algorithms using event-based cameras can be done, similarly to frame-based cameras, by augmenting the camera with either a different kind of sensor such as frame-based RGB-D cameras, or another event-based camera for stereo-vision. Censi and Scaramuzza (Censi and Scaramuzza, 2014) provided 6-DoF visual odometry by fusing the event-based camera with a CMOS camera where only rotation was accurately estimated and translation suffered from a deteriorated accuracy. Kueng et al. (Kueng et al., 2016) tracked the features detected in a CMOS image frame using the event-based camera and used a Bayesian depth filter to estimate the depth of 2D tracked features and obtain 3D points. These 3D points are then used to minimize the reprojection error between 2D features and 3D points to estimate 6-DoF pose. Weikersdorfer et al. (Weikersdorfer et al., 2014) used an extrinsically calibrated RGB-D sensor with an event-based camera to provide an accurate transformation of each depth value in the events' frame and applied a Bayesian particle filter to estimate 6-DoF pose and a map.

Using an Inertial Measurement Unit (IMU) helps to improve estimates provided by a monocular camera to obtain accurate absolute scale. Zihao et al. (Zihao Zhu et al., 2017) track features using optical-flow-based expectation maximization to warp features and then use the tracked features with IMU measurements in a structure-less Kalman filter scheme for pose estimation. Mueggler et al. (Mueggler et al., 2018) used splines on a manifold for better representation of IMU readings and minimized the geometric reprojection and IMU error for 6-DoF pose estimation. Vidal et al. (Vidal et al., 2018) proposed a SLAM[1] system that combines an event-based camera, CMOS camera and an IMU to provide an accurate scheme based on previous work (Rebecq et al., 2017b) which mainly depends on feature tracking and non-linear key-frames optimization. (Le Gentil et al., 2020) exploited the geometric structure of the environment and developed a visual-inertial system that exploits the detected lines instead of evens in the scene to estimate ego-motion.

State-of-the-art algorithms presented in the literature of event-based cameras vary in their approach, estimated states, used sensors and performance. Despite the fact that event-based cameras adopt a mode of operation that differs from frame-based cameras, the introduced algorithms mainly depend on concepts embraced for frame-based techniques such as features extraction and key-frames optimization and triangulation. Moreover, the event-based change detection model and the high temporal resolution of event-based cameras highly improved the quality of optical flow estimation. Although optical flow incorporates 6-DoF information (Longuet-Higgins and Prazdny, 1980; Zucchelli et al., 2002), we observe that optical flow is not fully exploited in event-based 6-DoF estimation except for some image warping tasks (Zihao Zhu et al., 2017).

In this paper, we introduce a visual-inertial odometry optimization scheme that essentially depends on optical flow information corrected using the IMU measurements. The following Section presents Neuromorphic Vision and how an event is triggered. Section 4 demonstrates how visual-inertial odometry works and illustrates our optimization scheme. The experimental setup required to validate our scheme is shown in Section 5 and the obtained results in Section 6.

[1] Simultaneous Localization And Mapping

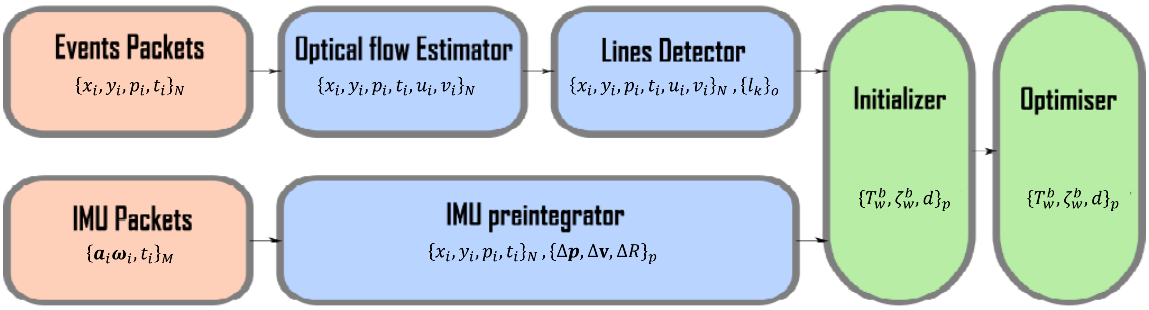

Events Packets
$\{x_i, y_i, p_i, t_i\}_N$
Optical flow Estimator
$\{x_i, y_i, p_i, t_i, u_i, v_i\}_N$
Lines Detector
$\{x_i, y_i, p_i, t_i, u_i, v_i\}_N, \{l_k\}_o$
Initializer
$\{T_w^b, \zeta_w^b, d\}_p$
Optimiser
$\{T_w^b, \zeta_w^b, d\}_p$
IMU Packets
$\{\boldsymbol{a}_i \boldsymbol{\omega}_i, t_i\}_M$
IMU preintegrator
$\{x_i, y_i, p_i, t_i\}_N, \{\Delta\mathbf{p}, \Delta\mathbf{v}, \Delta R\}_p$

Figure 1: Optical Flow (OF) based visual-inertial odometry scheme where each block shows its expected output. In red: Raw data, green: processed data required for optimization, blue: The optimization scheme and the initializer outputting 6-DoF Pose, twist and line depth.

3 NEUROMORPHIC VISION MODEL

Rather than providing complete frames at regular intervals, each pixel of an event-based camera generates an asynchronous flow of events. The generated flow is triggered whenever a change in light intensity is detected. An event $e \doteq \{x, y, p, t\}$ is described by its pixel position (x, y), its polarity $p \in \{-1, 1\}$ and the timestamp of the event t. Whenever light intensity variation on a pixel exceeds the threshold $\delta_l \in [10\%, 15\%]$, an event is triggered according to the equation:

$$\Delta L(x_i, y_i, t_i) = L(x_i, y_i, t_i) - L(x_i, y_i, t_i - \Delta t) = p_i \delta_l, \tag{1}$$

where, for each pixel (x_i, y_i), $L(x_i, y_i, t_i)$ and $L(x_i, y_i, t_i - \Delta t)$ are the light intensity log at time t_i and earlier time $t_i - \Delta t$. The triggered event e has a ± 1 polarity based on the increase or the decrease of light intensity $\pm \Delta L$.

4 FLOW-BASED VISUAL-INERTIAL ODOMETRY

4.1 Preliminaries

4.1.1 Pose

Vision-based 6-DoF state estimation algorithms incrementally estimate a 6-DoF pose $T \in SE(3)$ defined as the rigid body transformation. A rigid body transformation T_{ij} expressed as a Lie group $\mathcal{L}$ differentiable on manifold with the Lie algebra $\mathcal{A}$ as its tangent space at the identity is called a twist ζ_{ij}. The logarithmic map $Log : \mathcal{L} \rightarrow \mathcal{A}$ is used to obtain the twist ζ_{ij} of T_{ij} at the identity space and its inverse can be found using the exponential map $Exp : \mathcal{A} \rightarrow \mathcal{L}$ (Chirikjian, 2011).

$$T_{ij} = \begin{bmatrix} R_{ij} & t_{ij} \\ \mathbf{0} & 1 \end{bmatrix}, \ \zeta_{ij} = \begin{bmatrix} \lfloor \Omega_{ij} \rfloor_\times & V_{ij} \\ 0 & 1 \end{bmatrix}, \tag{2}$$

where $R_{ij} \in SO(3)$ is the rotational matrix, $\mathbf{t}_{ij} \in \mathbb{R}^3$ is the translation vector, $\lfloor \Omega_{ij} \rfloor_\times \in \mathfrak{so}(3)$ is the skew symmetric matrix of the angular velocity vector and $V_{ij} \in \mathbb{R}^3$ is the linear velocity vector. The vector space representing the rigid body transformation (group and algebra) is represented by the *vee* operator $(.)^\vee : \mathcal{L}, \mathcal{A} \rightarrow \mathbb{R}^d$ and is reversed by the hat operator $(.)^\wedge$.

4.1.2 Pinhole Model

Event-based cameras uses the pinhole model (Cyganek and Siebert, 2011) (or any reprojection model according to the used lens) to describe the 3D/2D projection $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ of any 3D point $X_c = [X_c, Y_c, Z_c]^T \in \mathbb{R}^3$ in the camera frame to a 2D point $x_c = [x_c, y_c]^T \in \mathbb{R}^2$ on the image plane as:

$$\pi\left([X_c, Y_c, Z_c]^T\right) = \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} f_u \frac{X_c}{Z_c} + c_u \\ f_v \frac{Y_c}{Z_c} + c_v \end{bmatrix} \tag{3}$$

where (f_u, f_v) are the lens focal length values and (c_u, c_v) are the the principal point coordinates in x and y directions, respectively. The pinhole model is a planar model which requires each pixel (event in our case) to be undistorted for accurate 3D/2D projection.

4.1.3 Optical Flow Representation

Optical flow describes the pixels apparent motion (Longuet-Higgins and Prazdny, 1980) which can be approximated as the perspective projection of a 3D point X_c moving freely with linear velocity $V_c = [v_{xc}, v_{yc}, v_{zc}]^T$ and angular velocity $\Omega_c =$

$[\omega_{xc}, \omega_{yc}, \omega_{zc}]^T$ so that the point's 3D velocity is described as:

$$\begin{aligned} \dot{X}_c = -(\Omega_c \times X_c + V_c) &= \begin{bmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{Z}_c \end{bmatrix} \\ &= -\left(\begin{bmatrix} \omega_{yc} Z_c - Y_c \omega_{zc} \\ \omega_{zc} X_c - Z_c \omega_{xc} \\ \omega_{xc} Y_c - X_c \omega_{yc} \end{bmatrix} + \begin{bmatrix} v_{xc} \\ v_{yc} \\ v_{zc} \end{bmatrix} \right) \end{aligned} \tag{4}$$

2D point velocities (optical flow approximation) corresponding to the optical flow can be obtained by the derivative of Equation (3) incorporating (4):

$$\dot{x}_c = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{\dot{X}_c}{Z_c} - \frac{\dot{Z}_c}{Z_c} x_c = \frac{1}{Z_c} A(x_c, y_c) \mathbf{V}_c + B(x_c, y_c) \Omega_c \tag{5}$$

where the matrices **A** and **B** are function of image plane coordinates:

$$\begin{aligned} A &= \begin{bmatrix} -f & 0 & (x_c - c_u) \\ 0 & -f & (y_c - c_v) \end{bmatrix} \\ B &= \begin{bmatrix} \frac{(x_c - c_u)(y_c - c_v)}{f} & -\left(f + \frac{(x_c - c_u)^2}{f}\right) & (y_c - c_v) \\ \left(f + \frac{(y_c - c_v)^2}{f}\right) & -\frac{(x_c - c_u)(y_c - c_v)}{f} & -(x_c - c_u) \end{bmatrix} \end{aligned} \tag{6}$$

Hence, estimating the optical flow, if the twist vector $\zeta_c^\vee$ is known, would require also knowledge about the depth Z_c of each point.

4.1.4 IMU Preintegration Measurements

An inertial measurement unit provides proprioceptive information as the linear acceleration $\tilde{\mathbf{a}}_b(t)$ and angular velocity $\tilde{\Omega}_b(t)$ expressed in the body frame and influenced by different noise sources described as:

$$\tilde{\Omega}_b(t) = \Omega_b(t) + \mathbf{b}_g(t) + \eta_g(t) \tag{7}$$

$$\tilde{\mathbf{a}}_b(t) = \mathbf{a}_b(t) + R_{wb}^T \mathbf{g} + \mathbf{b}_a(t) + \eta_a(t) \tag{8}$$

where $\eta_g(t)$ and $\eta_a(t)$ are the Gaussian white noise of the IMU random walk characterised as $\mathcal{N}(0, \sigma_g)$ and $\mathcal{N}(0, \sigma_a)$, respectively. $\Omega_b(t)$ and $\mathbf{a}_b(t)$ are the actual angular velocity and linear acceleration of the IMU, R_{wb} is the rotation matrix between the body frame and the world frame and $\mathbf{g}$ is the gravity vector. $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$ are the slowly varying random walk noise of the sensors with their rates defined by:

$$\dot{\mathbf{b}}_g(t) = \eta_{bg} \ , \ \dot{\mathbf{b}}_a(t) = \eta_{ba} \tag{9}$$

where η_{bg} and η_{ba} are the Gaussian white noise of the IMU biases characterised as $\mathcal{N}(0, \sigma_{bg})$ and $\mathcal{N}(0, \sigma_{ba})$, respectively.

Estimating the states of motion from an instant i to the instant j is done by integrating the linear acceleration and angular velocity:

$$R_{wb}(t_j) = R_{wb}(t_i) Exp\left(\int_{t_i}^{t_j} (\tilde{\Omega}(\tau) - \mathbf{b}_g(\tau) - \eta_g(\tau)) d\tau \right) \tag{10}$$

$$V_b(t_j) = V_b(t_i) + \int_{t_i}^{t_j} (R_{wb}(\tilde{\mathbf{a}}(\tau) - \mathbf{b}_a(\tau) - \eta_a(\tau)) - \mathbf{g})\, d\tau \tag{11}$$

$$\begin{aligned} P_b(t_j) = &P_b(t_i) + V_b(t_{ij}) \Delta t \\ &+ \int_{t_i}^{t_j} (R_{wb}(\tilde{\mathbf{a}}(\tau) - \mathbf{b}_a(\tau) - \eta_a(\tau)) - \mathbf{g})\, d\tau^2 \end{aligned} \tag{12}$$

where R_{wb} is the rotation matrix, V_b is the velocity vector and P_b is the position vector. Instead of using equations 10, 11 and 12 which would slow down optimization and increase estimation errors, we adopt a preintegration representation of IMU measurements introduced in (Lupton and Sukkarieh, 2011) and modified for representation on manifolds in (Forster et al., 2016) to avoid recomputation of parameters. Preintegration provides the increments of the state $\{R_{wb}, V_b, P_b\}$ between two time steps i and j expressed as:

$$\Delta R_{wb}(t_{ij}) = \Delta \tilde{R}_{wb}(t_{ij}) Exp(-\delta\phi_{ij}), \tag{13}$$

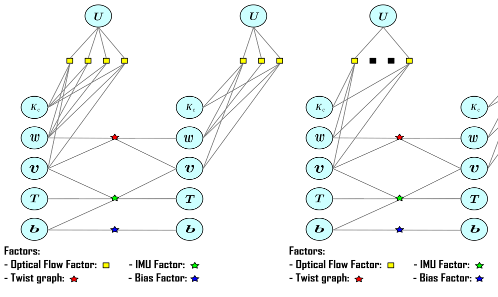
$$\Delta V_b(t_{ij}) = \Delta \tilde{V}_b(t_{ij}) - \delta V_b(t_{ij}), \tag{14}$$

$$\Delta P_b(t_{ij}) = \Delta \tilde{\mathbf{P}}_b(t_{ij}) - \delta \mathbf{P}_b(t_{ij}), \tag{15}$$

where $\Delta(.)$ represent the difference of the state between the two time steps i and j, $\tilde{(.)}$ means that the states are estimated directly from measurements with no noise estimation, $\delta(.)$ denotes the preintegration values of the rotation, velocity and position states incorporating the IMU noise propagation and defined in the method given in (Forster et al., 2016).

4.2 Optimization Scheme

Using the optical flow for accurate motion estimation is a complex problem which requires, in some cases, decoupling the translational and rotational motion and a prior knowledge of depth (Zucchelli, 2002; Liu et al., 2017). In a different approach, we exploit the geometric characteristics of the environment besides augmenting the optical flow with IMU measurements to obtain accurate ego-motion and depth estimation (see Figure 1). In order to have reliable event-based optical flow estimation with acceptable computational time, we used a PCA event-based optical



(a) (b)

Figure 2: A) Factor graph with no dropped events between two optimization time steps, b)Factor graph where some events are dropped. In case of dropping events, the number of Optical Flow edges decrease, and accordingly the total optimization time is reduced significantly.

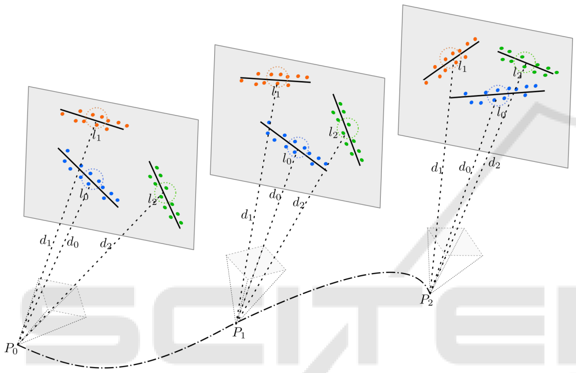


Figure 3: A conceptional drawing of different detected lines at different time steps with their assigned events with a small radius around the center point. We choose only events around the center with their optical flow to participate in the initial depth estimation assuming small depth variation.

flow approach (Khairallah et al., 2022b) where each event's information becomes $\{x,y,p,t,u,v\}$ with the optical flow u and v in x and y directions.

Event-Based cameras provide signals due to changes in the environment which would occur on contours of objects. This makes Event-Based sensors suitable for semi-dense SLAM and visual odometry algorithms. We benefit from the richness of events creating contours in structured environments to detect and track lines using a flow-based line detector (Khairallah et al., 2022a).

In our scheme, we follow a probabilistic approach exploiting optical flow and detected lines (Furgale et al., 2012). We obtain optimal state estimates $\mathcal{X}(t)$ within a time interval of $[t_0,t_f]$ using a set of measurements $\mathcal{Z}(t)$ where the environment has the structure $\mathcal{S}$ in a joint posterior estimate $p(\mathcal{X}(t)|\mathcal{Z}(t))$ with no map or prior belief. The set of measurements consist of measured optical flow given the position of each event $\mathcal{U}_m(t)$, the accelerometer measurements $\mathcal{A}(t)$ and gyroscope measurements $\mathcal{W}(t)$. With no prior belief, we try to find a maximum likelihood of measurements using the estimated states as:

$$\begin{aligned} &p(\mathcal{X}(t)|\mathcal{U}_m(t),\mathcal{A}(t),\mathcal{W}(t)) \\ &\quad = p(\mathcal{X}(t)|\mathcal{U}_m(t))p(\mathcal{X}(t)|\mathcal{A}(t))p(\mathcal{X}(t)|\mathcal{W}(t)) \end{aligned} \quad (16)$$

where the conditional probability of Equation (16) consists of the multiplication of conditional probabilities of measurements given that each set of measurements is independent of the others. We assume that each conditional probability is described as a Gaussian probability distribution with zero mean and a variance σ. Obtaining the maximum likelihood is equivalent to estimating the minimum of the log function which is expressed as the following cost function:

$$F = \frac{1}{N}\sum_{i=1}^{N}\Delta_{iu} + \frac{1}{M}\sum_{i=1}^{M}\Delta_{jimu} + \frac{1}{M}\sum_{i=1}^{M}\Delta_{jba} + \frac{1}{M}\sum_{i=1}^{M}\Delta_{jbw} \quad (17)$$

Where N is the number of events providing optical flow during optimization span and M is the number of IMU measurements used. Δ_{iu}, Δ_{jimu} are the error terms corresponding to optical flow estimation, IMU measurements, respectively. Δ_{jba} and Δ_{jbw} are the error terms corresponding to the accelerometer and gyroscope bias. In order to enhance the optimization process we added a twist error term $\Delta_{j\zeta}$ responsible for refining the twist used for optical flow estimation (see Figure 2). The new enhanced robust objective function is defined as:

$$\begin{aligned} F = &\frac{1}{N}\sum_{i=1}^{N}\Delta^{\rho}_{iu} + \frac{1}{M}\sum_{i=1}^{M}\Delta^{\rho}_{jimu} + \frac{1}{M}\sum_{i=1}^{M}\Delta^{\rho}_{jba} + \\ &+ \frac{1}{M}\sum_{i=1}^{M}\Delta^{\rho}_{jbw} + \frac{1}{M}\sum_{i=1}^{M}\Delta^{\rho}_{j\zeta}, \end{aligned} \quad (18)$$

Where ρ denotes the Huber norm (Huber, 1992).

The optical flow error Δ_{iu} is defined as:

$$\Delta_u = (\mathbf{u}_e(t) - \mathbf{u}_m(d(t)))^T\,\Sigma_u(\mathbf{u}_e(t) - \mathbf{u}_m(d(t))) \quad (19)$$

where Σ_u is the covariance matrix associated with the optical flow. $\mathbf{u}_e(t)$ is the estimated optical flow, $\mathbf{u}_m(d(t))$ is the measured optical flow using the IMU measurements (see Equation (5)) where the depth Z_c initial estimate is shown in the initialization step (see Section 4.3.1). To alleviate the problem of estimating the depth of each event independently – which would require heavier computations – and since the provided events are created due to the motion of contours of objects, we assumed that the environment contains a sufficient amount of contour lines that can be used to estimate the depth.

The IMU measurements error term Δ_{imu} is defined as:

$$\Delta_{imu} = [\Delta^T_{Rij},\Delta^T_{vij},\Delta^T_{pij}]^T\,\Sigma_{imu}[\Delta^T_{Rij},\Delta^T_{vij},\Delta^T_{pij}] \quad (20)$$

where Σ_{imu} is the IMU covariance matrix. The preintegration error terms are:

$$\begin{aligned}
\Delta_{Rij} &= Log\left(\left(\Delta\tilde{R}_{wb_{ij}} Exp\left(\frac{\partial\tilde{R}_{wb}}{\partial\mathbf{b}_g}\partial\mathbf{b}_g\right)\right)^T R_{wb_i}(t_i)^T R_{wb_j}\right)\\
\Delta_{vij} &= R_{wb_i}\left(V_{b_j} - V_{b_i} - \mathbf{g}\Delta t_{ij}\right) -\\
&\quad - \left(\Delta\tilde{V}_{b_{ij}}\frac{\partial\tilde{V}_b}{\partial\mathbf{b}_a}\partial\mathbf{b}_a + \frac{\partial\tilde{V}_b}{\partial\mathbf{b}_g}\partial\mathbf{b}_g\right)\\
\Delta_{pij} &= R_{wb_i}\left(P_{b_j} - P_{b_i} - V_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) -\\
&\quad - \left(\Delta\tilde{P}_{b_{ij}} + \frac{\partial\tilde{P}_b}{\partial\mathbf{b}_a}\partial\mathbf{b}_a + \frac{\partial\tilde{P}_b}{\partial\mathbf{b}_g}\partial\mathbf{b}_g\right)
\end{aligned} \tag{21}$$

where the partial derivatives $[\frac{\partial\tilde{R}_{wb}}{\partial\mathbf{b}_g}, \frac{\partial\tilde{\mathbf{V}}_b}{\partial\mathbf{b}_a}, \frac{\partial\tilde{\mathbf{V}}_b}{\partial\mathbf{b}_g}, \frac{\partial\tilde{\mathbf{P}}_b}{\partial\mathbf{b}_a}, \frac{\partial\tilde{\mathbf{P}}_b}{\partial\mathbf{b}_a}]$ are calculated as explained in the supplementary materials of (Forster et al., 2016).
The error terms for the biases Δ_{ba} and Δ_{bw} are defined as:

$$\Delta_{ba} = (\mathbf{b}_{aj} - \mathbf{b}_{ai})^T \Sigma_{ba} (\mathbf{b}_{aj} - \mathbf{b}_{ai}), \tag{22}$$

$$\Delta_{bw} = (\mathbf{b}_{wj} - \mathbf{b}_{wi})^T \Sigma_{bw} (\mathbf{b}_{wj} - \mathbf{b}_{wi}). \tag{23}$$

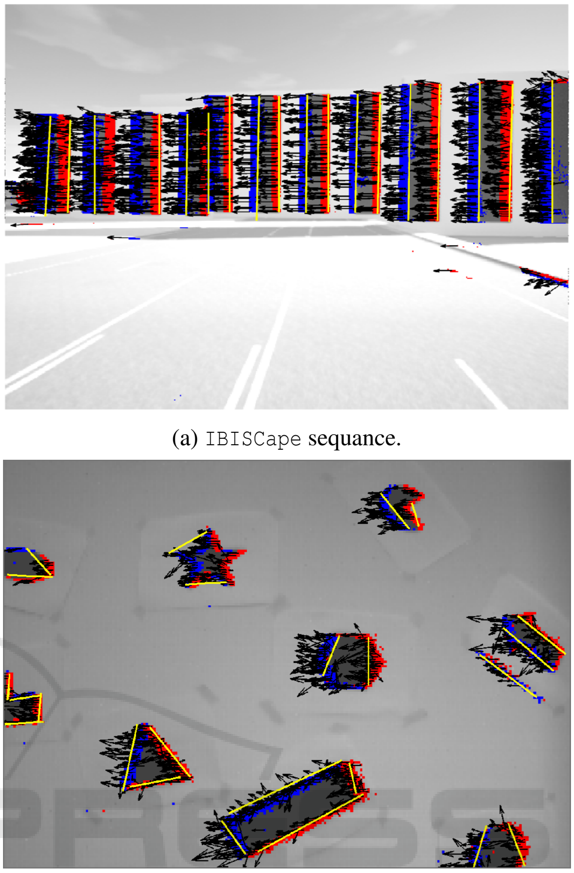
The twist error term as:

$$\Delta_{\zeta ij} = \left(\left(\frac{1}{\Delta t}\hat{T}_i^{-1}\hat{T}_j\right) \ominus \zeta_{ij}\right)^T \Sigma_\zeta \left(\left(\frac{1}{\Delta t}\hat{T}_i^{-1}\hat{T}_j\right) \ominus \zeta_{ij}\right) \tag{24}$$

Frame-based optimization schemes using features choose certain key-frames to achieve triangulation with low uncertainty. Conversely, using optical flow allows to ignore key-frames and freely choose the time steps for optimization depending on either the number of events N or the number of IMU readings M. Moreover, having rich events optical flow and lines ensure we can drop events whenever events frequency exceeds a threshold in order to maintain real-time processing. The state vector we optimize contains position, rotation quaternion, velocity, IMU biases and the camera intrinsic parameters $\{P, Q, V, d, b_a, b_g, K_c\}$, where Q is the rotation quaternions, d is the depth and K_c is the camera matrix to calibrate the camera parameters online. Our cost function is solved as a non-linear unconstrained least squares problem using Levenberg-Marquardt method. of walls (illustrative examples in Figure 4).

4.3 Optimization Conditioning

The conditioning process of our nonlinear unconstrained optimization scheme requires a reliable initialization for all the parameters undergoing optimization, i.e. the camera trajectory and the scene constituents (see Figure 1).



(a) `IBISCape` sequance.

(b) `shapes_6dof`.

Figure 4: Grayscale images of the sequences used to test our algorithm with the triggered events (red for positive polarity and blue for negative polarity). The estimated optical flow arrows in black and the detected lines in yellows.

Event-based cameras provide information about contours and that the lines are one of the repetitive geometric patterns in the environment. We exploit the detected lines (Khairallah et al., 2022a) to augment the prior information we know about the environment. We assume that the IMU and the camera are calibrated with initial camera intrinsic parameters values and the extrinsic transformation $\mathcal{T}_{ic}$ between DVS sensor and IMU (illustrated in Figure 5) is known. To ensure a reliable online calibration of the camera-

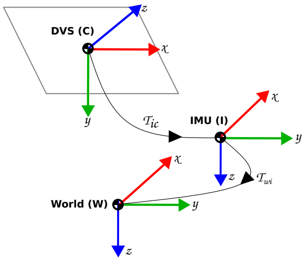


Figure 5: Event camera reference coordinate frames.

IMU setup, all the intrinsic and extrinsic parameters of both the DVS and IMU sensors are considered as optimization states. To find 6-DoF initial pose using optical flow, we need to know the depth of events and to estimate depth we need the 6-DoF pose. We iteratively estimate an initial depth then use it to correct for accurate pose and twist estimation.

4.3.1 Initial Depth Estimation

The line detection algorithm provides the line parameters (center point, line vector and principal optical flow) and the assigned events to each line. A 2D projected line on the image plane may have varying depth in 3D. However, the depth of events around the line's center point presents small depth variations (see Figure 3). We use the estimated optical flow and the IMU measurements to estimate the depth according to (5). Since linear velocity is obtained from single integration of IMU measurements and the angular velocity is directly provided, we use a sliding average window to alleviate the effect of accelerometer white noise without removing the gravity vector offset. Gyroscope angular velocities bias offset and white noise are filtered out using a band pass filter. For each set of events around a line, we use Equation (5) where the only unknown is the inverse depth so each optical flow gives two depth values and equation becomes:

$$\begin{bmatrix} \frac{1}{Z_{cx}} \\ \frac{1}{Z_{cy}} \end{bmatrix} \frac{1}{Z_{cx}} = (\dot{x} - B(x_c, y_c)\Omega) / (A(x_c, y_c)\mathbf{V}_c) \quad (25)$$

where the division here is element-wise division. The depth ratio $\left(\frac{1}{Z_{cx}} / \frac{1}{Z_{cy}}\right)$ should be identity because they belong to the same event. If the depth ratio is not in a bounded interval $[th_1, th_2]$, this implies that the estimated optical flow is highly corrupt and will be rejected. The initial depth assigned to all events of the line is the mean of the estimated depth around the center after rejecting outliers. This initialization method is only effective if the depth does not vary much along each line, i.e. downward facing cameras of drones or cameras moving indoor in front of walls.

4.3.2 Initial Pose and Twist Estimation

Using estimated depth of all events around center point of detected lines and after rejecting outlier optical flow, we re-inject the depth values into Equation (5) after modifying it so that it becomes (for a single event):

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{-f}{Z_c} & 0 & \frac{(x_c-c_u)}{Z_c} & \cdots \\ \cdots & \frac{(x_c-c_u)(y_c-c_v)}{f} & -\left(f + \frac{(x_c-c_u)^2}{f}\right) & (y_c - c_v) \\ 0 & \frac{-f}{Z_c} & \frac{(y_c-c_v)}{Z_c} & \cdots \\ \cdots & \left(f + \frac{(y_c-c_v)^2}{f}\right) & -\frac{(x_c-c_u)(y_c-c_v)}{f} & -(x_c - c_y) \end{bmatrix} \zeta^\vee$$

$$\dot{x}_c = C(x_c, y_c, Z_c)\zeta^\vee \quad (26)$$

In Equation (26), the twist vector $\zeta^\vee$ is the only unknown. We can stack the optical flow information for all events as:

$$\begin{bmatrix} C_1(x_c, y_c, Z_c) \\ \vdots \\ C_n(x_c, y_c, Z_c) \end{bmatrix} \zeta^\vee = \begin{bmatrix} \dot{x}_{c1} \\ \vdots \\ \dot{x}_{cn} \end{bmatrix}. \quad (27)$$

Equation (27) can be solved for $\zeta^\vee$ using least square method for $Ax = b$ where the solution would be $(A^T A)^{-1} A^T b$. Estimating the depth and twist is repeated iteratively until convergence to make sure initialized depth and twist are correctly estimated. The initial pose is estimated by integrating the twist vector.

5 EXPERIMENTAL SETUP

Our proposed visual-inertial odometry scheme performs in structured environments containing lines with low depth variations. For this purpose, we choose sequences fulfilling these criteria in order to provide a fair assessment. We used one of `IBISCape` sequences provided in (Soliman et al., 2022) of a car moving in an environment augmented with white walls and black rectangles at different depths. Additionally, we used the sequence of `shapes_6dof` provided in (Mueggler et al., 2017) of a handheld camera moving randomly in front of different geometric shapes depicted on a wall. These sequence were, first, passed through the optical flow estimator then the lines detector to have all the required information for optimization (see Figure 4).

We use Ceres solver (Agarwal et al., 2022) as an optimizer for its automatic differentiation capability. Our algorithm run on a $3GHz$ Core $i7$ 16 core Linux machine. We have set our time step to 0.025 s where 5 IMU measurements are preintegrated for `IBISCape`'s sequence and 25 measurements are preintegrated for

Table 1: Specifications of the used sequences.

Sequence	events [Mevent]	Total Time [s]	IMU [Hz]	V_{max} $[m/s]$	Ω_{max} $[^\circ/s]$
`IBISCape`	21.65	17.6	200 Hz	7.7	76
`shape_6dof`	17.96	59.7	1000 Hz	2.3	715

Table 2: Detailed quantitative analysis based-on the Average Root Mean Square Error metric of `IBISCape` and `shapes_6dof` sequence. We report results for 25, 50, and 75 percent of dropped events as milestones for brevity.

Method	`IBISCape` sequence				`shapes_6dof` sequence			
	μ [m]	σ [m]	μ [°]	σ [°]	μ [m]	σ [m]	μ [°]	σ [°]
EVO (Rebecq et al., 2017a)	0.1369	0.0082	1.7840	0.6214	0.09103	0.0051	5.0217	0.9851
Proposed (All events)	**0.1204**	**0.0079**	1.5602	0.7683	**0.0802**	**0.0043**	**2.5791**	**1.9732**
Proposed (25% dropped)	0.1231	0.0117	1.5874	0.8024	0.0841	0.0094	2.8041	1.8541
Proposed (50% dropped)	0.1217	0.0172	**1.4272**	**0.8401**	0.0971	0.0158	2.8460	1.9471
Proposed (75% dropped)	–	–	–	–	–	–	–	–

Table 3: Ablation study on the event-based VI system architecture. We report the mean position errors as a percentage of the sequence total distance [%].

Method	`shapes`		`poster`		`dynamic`	
	`6dof`	`translation`	`6dof`	`translation`	`6dof`	`translation`
IDOL (Le Gentil et al., 2020)	10.4	10.2	12.4	14.0	10.8	5.0
EVIO (Zihao Zhu et al., 2017)	2.69	2.42	3.56	0.94	4.07	1.90
Rebecq et al. (Rebecq et al., 2017b)	0.42	0.50	0.40	0.46	0.56	**0.39**
$(E+I)$ (Vidal et al., 2018)	0.48	**0.41**	**0.30**	0.15	0.38	0.59
Proposed (All events)	**0.41**	0.45	0.33	**0.11**	**0.17**	0.81

Table 4: Study on the effect of events dropping percentage on the total optimization time reported for the `shapes_6dof` sequence.

drop [%]	packet size [–]	packet time [s]	residual and jacobian time [s]	linear solver [s]	Total time [s]
–	50	0.25	0.420759	0.304085	0.724844
–	100	0.5	0.624733	0.496576	1.121309
–	150	0.75	0.956266	0.912470	1.868736
–	200	1	1.059416	0.998935	2.058351
25	50	0.25	0.262560	0.091245	0.353805
25	100	0.5	0.545977	0.215199	0.761176
25	150	0.75	0.729159	0.275548	1.004707
25	200	1	0.845035	0.317980	1.163015
50	50	0.25	0.235112	0.082975	0.318087
50	100	0.5	0.345446	0.104557	**0.450003**
50	150	0.75	0.465738	0.133461	**0.599199**
50	200	1	0.627081	0.188407	**0.815488**
75	50	0.25	0.691566	0.113761	0.805327
75	100	0.5	0.997071	0.208451	1.205522
75	150	0.75	1.285245	0.418131	1.703376
75	200	1	1.375911	0.537240	1.913151

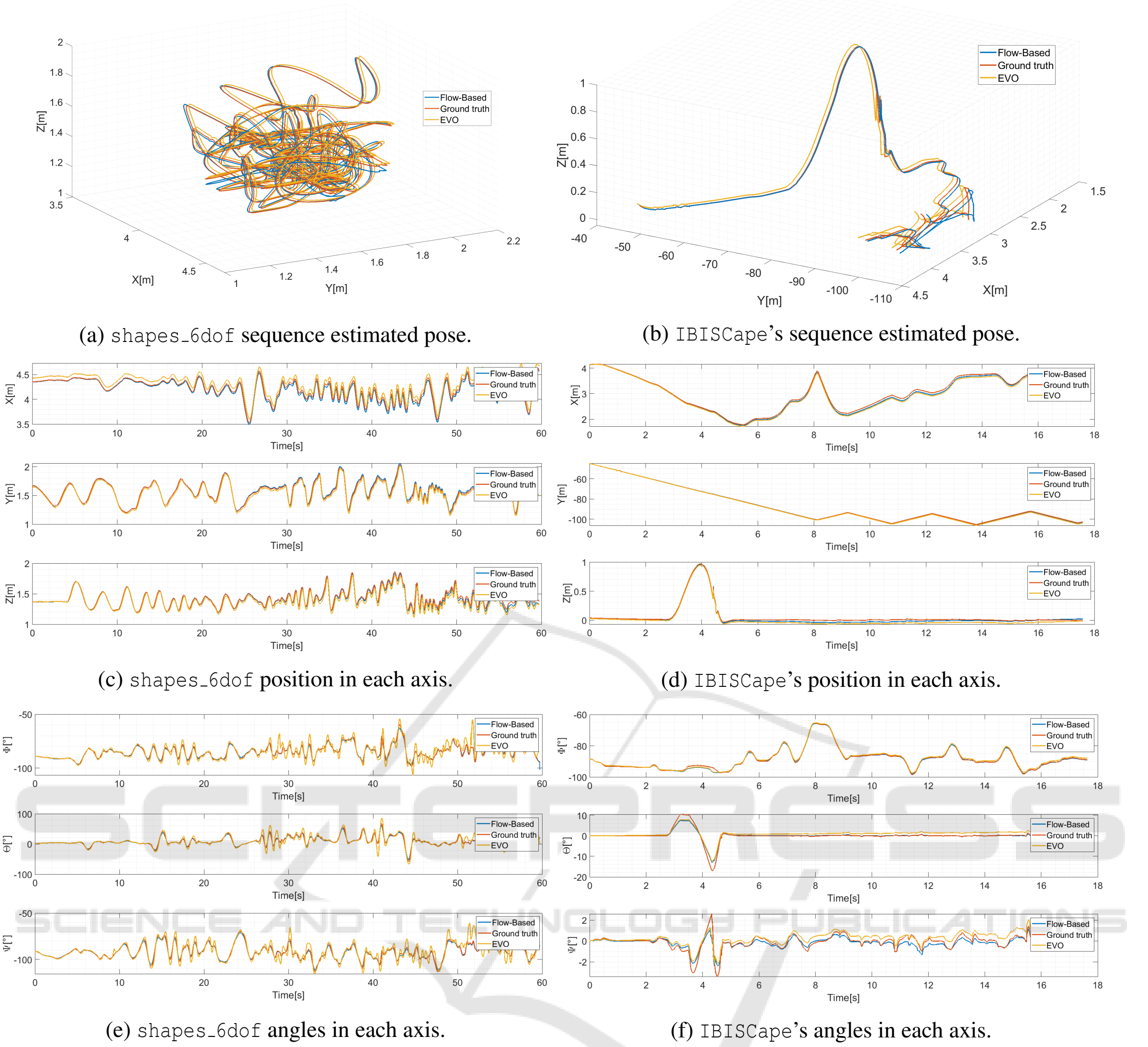
the `shapes_6dof` sequence. Being recorded with a handheld camera, `shapes_6dof` sequence undergoes high rotational speed and relatively low translational speed while `IBISCape`'s sequence have the opposite characteristics since it's recorded as a car's onboard camera.

6 RESULTS

`IBISCape`'s sequence had a higher RMSE for translation because of its high translational speed. In contrast, `shapes_6dof` sequence attained a lower RMSE for translation for the same reason. The rotational RMSE error is maintained relatively small because of the accuracy of the IMU measurements. Figure 7 shows the translational and rotational errors over time for the two line-based feasible applications. The first is the vehicle moving in a line staged textured road (`IBISCape` sequence), where the errors are reported in terms of 10's of [cm]. Whereas, the second application of a handheld DAVIS sensor facing shapes with clear lines, where the errors are reported in terms of 10's of [mm]. However, both sequences show a high standard deviation for the rotational errors results from the low accuracy in the gyroscope noise covariance estimation during IMU still calibration.

We ran many experiments to check for the accuracy of our system with and without dropping events to alleviate for real-time computation. The assumption that our scheme will still work in case of events being dropped is made since it only depends on optical flow (and not tracked features) and that the number of optimization residuals is always much lower than the amount of events at each time step. We found that our system can hold accurate results until we reach around 50% of dropped events for `shapes_6dof` sequence and about 60% of dropped events for `IBISCape`'s sequence (see Table 2). The amount of events that can be dropped depends on events frequency. `IBISCape`'s sequence maintained good results while more events were dropped because of its higher resolution and events' frequency.

The accuracy did not vary much before failure occurred with 75% events dropping, which validates the assumption that events can be dropped with a threshold depending on events frequency and camera resolution. Dropping the events can also be improved to maintain accuracy by choosing the dropped events being assigned to lines where each line should have a



(a) `shapes_6dof` sequence estimated pose.

(b) `IBISCape`'s sequence estimated pose.

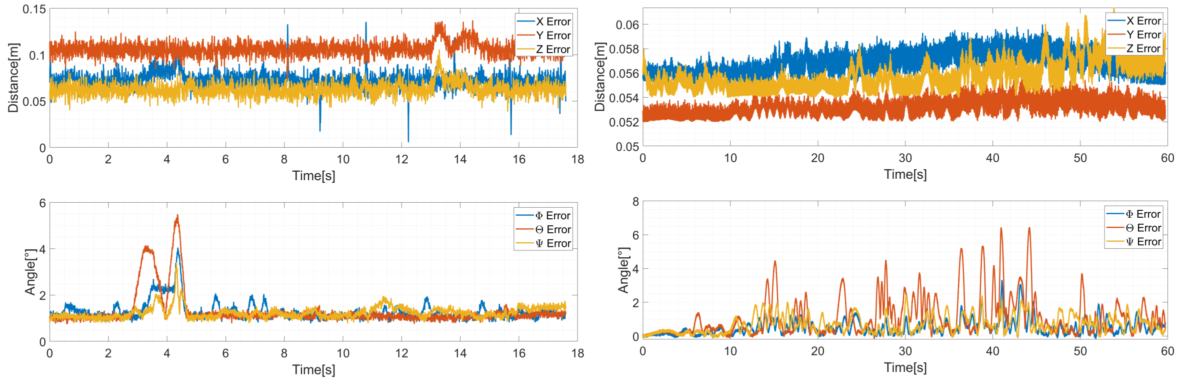
(c) `shapes_6dof` position in each axis.

(d) `IBISCape`'s position in each axis.

(e) `shapes_6dof` angles in each axis.

(f) `IBISCape`'s angles in each axis.

Figure 6: The estimated pose, position and angles of `shapes_6dof` and `IBISCape` sequences. Flow-Based method in blue, the ground truth in red and EVO in yellow.



(a) Position and angle errors of `IBISCape` sequence.

(b) Position and angle errors of `shapes_6dof` sequence.

Figure 7: Errors of our flow-based visual-inertial odometry method.

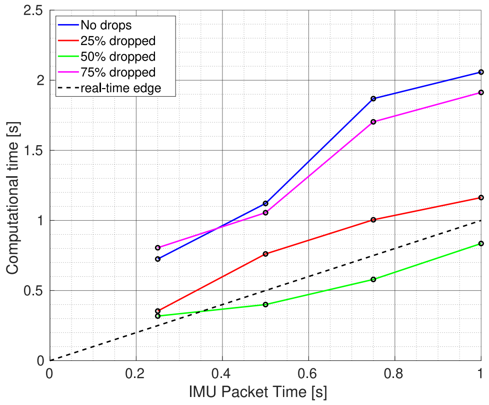


Figure 8: On-hardware real-time performance analysis.

minimum amount of events to avoid failure.

To measure the computational time of our scheme, measurements to be optimized are placed in a sliding window where previously optimized poses are considered constant and only the sliding window is optimized. Table 4 shows the computational time of different windows with different percentages of dropped events. The high computational time for `IBISCape`'s sequence is due to the high number of events generated by a 1024 × 1024 camera resolution. On the contrary, `shapes_6dof` sequence attained real-time performance for all the sliding windows with no dropped events.

The number of IMU measurements and the amount of events to be dropped defines the compromise to achieve real-time applicability (see Figure 8). We should keep the smallest possible sliding window with the maximum amount of events to be dropped which leads to a trade-off between computational time and accuracy (sliding windows allowing real-time performance are shown in bold within Table 4). We notice an increase in the computation time when 75% of the events are dropped as a result of an abrupt increase in the problem uncertainty due to the low number of optical flow edges as illustrated in Figure 2 (b), and hence, low information about the scene.

In Table 3, we represent an ablation study to intellect the contribution of the event-based VI system configuration on the pose estimation accuracy. The main conclusion from this quantitative analysis is that our method outperforms IDOL, an alternative state-of-the-art line-based method that does not incorporate optical flow, and can perform well in a line textured environments.

7 CONCLUSION

We introduce a flow-based visual-inertial odometry algorithm for neuromorphic vision sensors. The algorithm corrects optical flow information using IMU measurements in environments where lines can be detected. We run our algorithm without the need for triangulation or keyframe estimation, which provides the liberty to choose the size of our sliding window during optimization.

Instead of running for only scenarios where the depth of lines does not vary much, the optimal performance of our method can be witnessed when backed with a depth sensor. Integrating a depth sensor can also be used to estimate more accurate optical flow. Another improvement to our system would be adding a place recognition in order to have the ability to close the loop in a complete SLAM system.

REFERENCES

Agarwal, S., Mierle, K., and Team, T. C. S. (2022). Ceres Solver.

Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255.

Censi, A. and Scaramuzza, D. (2014). Low-latency event-based visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–710. IEEE.

Chirikjian, G. S. (2011). *Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications*, volume 2. Springer Science & Business Media.

Cyganek, B. and Siebert, J. P. (2011). *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons.

Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2016). On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21.

Furgale, P., Barfoot, T. D., and Sibley, G. (2012). Continuous-time batch estimation using temporal basis functions. In *2012 IEEE International Conference on Robotics and Automation*, pages 2088–2095. IEEE.

Gehrig, D., Gehrig, M., Hidalgo-Carrio, J., and Scaramuzza, D. (2020). Video to events: Recycling video datasets for event cameras. *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 3583–3592.

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer.

Khairallah, M. Z., Bonardi, F., Roussel, D., and Bouchafa, S. (2022a). Flow-based line detection and segmentation for neuromorphic vision sensors. In *2022 The*

29th IEEE International Conference on Image Processing (IEEE ICPR). IEEE.

Khairallah, M. Z., Bonardi, F., Roussel, D., and Bouchafa, S. (2022b). Pca event-based optical flow: A fast and accurate 2d motion estimation. In *2022 The 29th IEEE International Conference on Image Processing (IEEE ICIP)*. IEEE.

Kim, H., Handa, A., Benosman, R., Ieng, S.-H., and Davison, A. J. (2008). Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ*, 43:566–576.

Kim, H., Leutenegger, S., and Davison, A. J. (2016). Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer.

Kueng, B., Mueggler, E., Gallego, G., and Scaramuzza, D. (2016). Low-latency visual odometry using event-based feature tracks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 16–23. IEEE.

Le Gentil, C., Tschopp, F., Alzugaray, I., Vidal-Calleja, T., Siegwart, R., and Nieto, J. (2020). Idol: A framework for imu-dvs odometry using lines. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5863–5870. IEEE.

Liu, M.-y., Wang, Y., and Guo, L. (2017). 6-dof motion estimation using optical flow based on dual cameras. *Journal of Central South University*, 24(2):459–466.

Longuet-Higgins, H. C. and Prazdny, K. (1980). The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397.

Lupton, T. and Sukkarieh, S. (2011). Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76.

Mueggler, E., Gallego, G., Rebecq, H., and Scaramuzza, D. (2018). Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6):1425–1440.

Mueggler, E., Huber, B., and Scaramuzza, D. (2014). Event-based, 6-dof pose tracking for high-speed maneuvers. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2761–2768. IEEE.

Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., and Scaramuzza, D. (2017). The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149.

Muglikar, M., Gehrig, M., Gehrig, D., and Scaramuzza, D. (2021). How to calibrate your event camera. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1403–1409.

Rebecq, H., Gallego, G., Mueggler, E., and Scaramuzza, D. (2018). EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time. *Int. J. Comput. Vis.*, 126:1394–1414.

Rebecq, H., Horstschaefer, T., Gallego, G., and Scaramuzza, D. (2017a). EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600.

Rebecq, H., Horstschaefer, T., and Scaramuzza, D. (2017b). Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *BMVC*.

Soliman, A., Bonardi, F., Sidibé, D., and Bouchafa, S. (2022). IBISCape: A simulated benchmark for multi-modal SLAM systems evaluation in large-scale dynamic environments. *Journal of Intelligent & Robotic Systems*, 106(3):53.

Vidal, A. R., Rebecq, H., Horstschaefer, T., and Scaramuzza, D. (2018). Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001.

Weikersdorfer, D., Adrian, D. B., Cremers, D., and Conradt, J. (2014). Event-based 3d slam with a depth-augmented dynamic vision sensor. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 359–364. IEEE.

Weikersdorfer, D. and Conradt, J. (2012). Event-based particle filtering for robot self-localization. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 866–870. IEEE.

Weikersdorfer, D., Hoffmann, R., and Conradt, J. (2013). Simultaneous localization and mapping for event-based vision systems. In *International Conference on Computer Vision Systems*, pages 133–142. Springer.

Zhou, Y., Gallego, G., and Shen, S. (2021). Event-based stereo visual odometry. *IEEE Transactions on Robotics*, 37(5):1433–1450.

Zihao Zhu, A., Atanasov, N., and Daniilidis, K. (2017). Event-based visual inertial odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399.

Zucchelli, M. (2002). *Optical flow based structure from motion*. PhD thesis, Numerisk analys och datalogi.

Zucchelli, M., Santos-Victor, J., and Christensen, H. I. (2002). Multiple plane segmentation using optical flow. In *BMVC*, volume 2, pages 313–322.