A 5-Point Minimal Solver for Event Camera Relative Motion Estimation

Ling Gao[1*]   Hang Su[1*]   Daniel Gehrig[2]   Marco Cannici[2]   Davide Scaramuzza[2]   Laurent Kneip[1]

[1] Mobile Perception Lab, ShanghaiTech University, China
[2] Robotics and Perception Group, University of Zurich, Switzerland

Abstract

Event-based cameras are ideal for line-based motion estimation, since they predominantly respond to edges in the scene. However, accurately determining the camera displacement based on events continues to be an open problem. This is because line feature extraction and dynamics estimation are tightly coupled when using event cameras, and no precise model is currently available for describing the complex structures generated by lines in the space-time volume of events. We solve this problem by deriving the correct non-linear parametrization of such manifolds, which we term eventails, and demonstrate its application to event-based linear motion estimation, with known rotation from an Inertial Measurement Unit. Using this parametrization, we introduce a novel minimal 5-point solver that jointly estimates line parameters and linear camera velocity projections, which can be fused into a single, averaged linear velocity when considering multiple lines. We demonstrate on both synthetic and real data that our solver generates more stable relative motion estimates than other methods while capturing more inliers than clustering based on spatio-temporal planes. In particular, our method consistently achieves a 100% success rate in estimating linear velocity where existing closed-form solvers only achieve between 23% and 70%. The proposed eventails contribute to a better understanding of spatio-temporal event-generated geometries and we thus believe it will become a core building block of future event-based motion estimation algorithms.

Project page: **https://mgaoling.github.io/eventail/**

1. Introduction

Event-based cameras are bio-inspired vision sensors that naturally react to edges moving in the scene with microsecond temporal resolution and minimal motion blur. These intrinsic properties make events ideal for accurate relative motion estimation, especially under challenging motion and lighting conditions where standard cameras often fall short. Nevertheless, estimating motion from event measure-
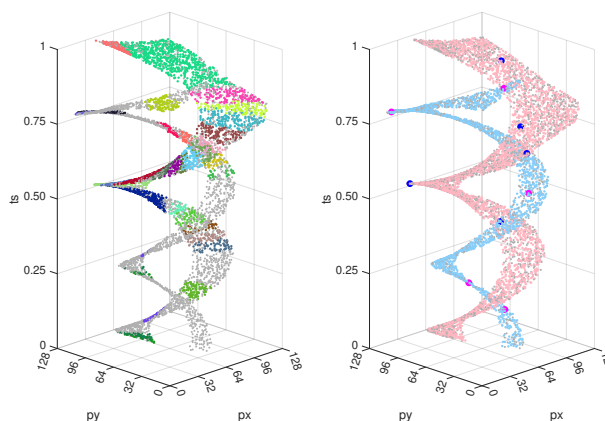
---
*indicates equal contribution

Figure 1. An event camera observing two non-parallel lines and moving with constant linear and angular velocity. The events triggered by each line lie on a manifold, which we call an eventail. We derive a minimal 5-point solver to estimate the parameters of the manifold, which includes both camera motion and scene geometry. Clustering these events based on spatio-temporal planes as done in previous work [24, 33] generates many spurious clusters (colorful points) with many outliers (grey points). Instead, eventails result in two large clusters with fewer outliers, and a velocity direction error of only 0.01 rad.

ments is an open challenge, as motion cues need to be inferred from the complex spatio-temporal structures formed by events, which typical vision-based algorithms struggle to grasp. Although event-based cameras have recently demonstrated unprecedented performance [41], the recent development of autonomous systems has created an increased demand for more accurate and reliable solutions which could better exploit the opportunity of improved motion modeling offered by these sensors.

However, while with a traditional camera, solving for relative motion simply means aligning two views with sufficient overlap, this problem is not as straightforward to define for an event-based camera since views are not even available in the first place. Furthermore, even if the fields of view of the camera share substantial overlap at two different points in time, the structure of the perceived events at those two moments remains very much a function of local

camera dynamics. In the worst case, if the camera ceases to move at all, no more events are triggered, and relative pose estimation becomes an ill-posed problem. It is intuitively clear that—for a dynamic vision sensor—the most fundamental problem of relative motion estimation is therefore given by the determination of local camera dynamics from a relatively short interval of events. The present paper introduces a geometric, deterministic solution to this problem.

The sparse and noisy nature of events has pushed the geometric vision community towards semi-dense approaches that make use of or optimize edge maps [37, 13, 52]. Based on the assumption that the gradient map contains straight lines, a promising area of research, therefore, looks at line features as a possible alternative to assist the geometric solution of relative event camera motion. Works in this area [24, 33], however, have inherent limitations that stem from a wrong assumption made during the initial feature extraction step. Indeed, they perform feature extraction independently of the relative camera displacement information, and they rely on a simple clustering strategy that models the space-time volume of events generated by a line under motion as a plane. However, as will be explained in detail in this work, lines do not form flat planes in the space-time volume of events, even if the camera undergoes constant linear and angular velocity, as evident from Figure 1.

It is thus clear that the problem of line feature extraction in the space-time volume of events can no longer be considered apart from the problem of dynamics estimation. In the present paper, we depart from this approximation and introduce a novel feature extractor that relies on a rigorously derived geometrical model of line-generated manifolds. Clustering the events of one manifold entails the identification of the manifold parameters, thereby leading to an implicit solution of the linear camera velocity from given angular rates measured by an Inertial Measurement Unit (IMU). Specifically, we make the following contributions:

- We introduce a minimal geometric parametrization of the manifolds that contain all events generated by the observation of a single line under the assumption of locally constant, linear velocity. The parametrization involves the velocity components that are non-parallel to the line, as well as a minimal 3D parametrization of the line itself.
- Based on this incidence relationship, we propose a minimal 5-point solver for the manifold parameters, and demonstrate its application in robust clustering for line feature detection and partial camera dynamics determination.
- We conclude with an averaging scheme that fuses the partial camera dynamics observations from each line-generated cluster into a complete estimate of the linear camera velocity, and thereby presents a rigorous theory for deterministic event camera motion initialization.

The present paper focuses on a theoretical understanding of line-generated manifold features, and thereby contributes to a better understanding of the geometry of the temporally dense-sampling event cameras. The theory is thoroughly evaluated on simulated data, and the advantage of the method is also demonstrated in a few concluding real-world examples. In particular, we show that using our method can consistently achieve a 100% success rate in estimating linear velocity where existing closed-form solvers achieving a success rate 23% and 70%.

2. Related work

Vision-based camera motion estimation is a long-studied problem, and there have been countless solutions for single-camera, multi-camera, and visual-inertial scenarios. The interested reader is kindly referred to the survey of Cadena *et al.* [6] for a relatively recent overview. A work worth separate mention, though, is by Weiss *et al.* [45], who directly estimate camera velocity.

Event-based motion estimation. The present work looks at motion estimation with an event camera, for which the last decade has already seen a number of solutions. Weikersdorfer *et al.* [44] originally propose a 2D-SLAM system with a dynamic vision sensor by employing a particle filter. The same group also proposes an event-based 3D SLAM framework by fusing events with a classical frame-based RGB-D camera [43]. Other event-based visual odometry systems make use of known depth or 3D structure [7, 30, 12, 5, 8], or are simply limited to the pure rotation scenario [13]. Contrast maximization [13, 11] is proposed as a unifying framework applicable to several event-based vision tasks. It draws substantial attention [22, 39, 26, 32, 31], but it is still limited to homographic warping scenarios. Full 6-DoF estimation is solved by Kim *et al.* [21] using a filtering approach, and Rebecq *et al.* [37] using an alternating tracking and mapping framework. Zhu *et al.* [51], Rebecq *et al.* [36], and Mueggler *et al.* [29] furthermore propose more reliable frameworks by fusing the measurements with an IMU.

More practical 6-DoF odometry and SLAM solutions keep being proposed by fusion with other sensors. Kueng *et al.* [22] combines the event camera with a standard camera to track features and build a probabilistic map. A similar sensor combination is used in Ultimate-SLAM [41], which improves robustness and accuracy by minimizing both vision and event-based residual errors. Zhou *et al.* [50] propose the first event-based stereo odometry system, while Zuo *et al.* [52] use a hybrid stereo setup composed of an event and a depth camera to realize DEVO, a semi-dense edge-tracking method. Finally, a recent work by Hidalgo-Carrió *et al.* [19] introduces EDS, a 6-DOF monocular direct visual odometry that combines events and frames.

Geometry-based motion estimation. Event-based motion estimation can be divided into optimization-based [37, 24, 29], filter-based [51, 44] and learning-based [27, 15] solutions. However, there is a lack of research on how fundamental geometry can be applied to event-based vision. For normal cameras, Weng *et al.* [46] and Hartley *et al.* [17] are among the first to introduce closed-form solutions for line-based motion estimation. Bartoli and Sturm [1, 2] introduce complete line-based structure from motion. More recently, based on modern line-feature extraction methods such as LSD [42], Zhang *et al.*, Pumarola *et al.*, and He *et al.* propose stereo [49], monocular [35], and visual-inertial fusion [18] based solutions to real-time, line feature-based SLAM, respectively.

Of particular interest to this work are event-based methods that rely on line features. Yuan *et al.* [47] and Le Gentil *et al.* [24] present optimization-based solutions, while Peng *et al.* [33] use tri-focal tensor geometry to present the first closed-form velocity initialization method. However, the methods make use of event-based line-feature extractors [4, 40] that fail to properly parametrize the line location in both space and time. Furthermore, their minimal solver needs at least 2 lines and therefore 10 events to make a single hypothesis.

Everding and Conradt [10] present a low-latency line tracker, while Mitrokhin *et al.* [28] present a learning-based method to track the surfaces generated by events. Ieng *et al.* [20] and Seok and Lim [38] finally propose model-based methods to fit and track the surfaces or curves generated by events. The perhaps most related work to ours is the work of Ieng *et al.* [20], who aim at understanding the spatio-temporal sub-space properties of the surfaces generated by events. However, to the best of our knowledge, our work is the first to establish a minimal parametrization of the surface as a function of the observable 3D spatial and dynamic parameters. Furthermore, we are the first to propose a deterministic minimal solver for this problem.

3. Theory

We assume to have a calibrated event camera under motion observing a scene that can be approximated by a set of 3D lines. We consider a temporal slice of events from which our objective is to initialize a first-order approximation of local camera dynamics. The motion of each observed line generates its own set of events, and—while the instantaneous reprojection of a line is still a line—each event cluster is generated by a line that moves and rotates through the space-time volume of events, thereby generating a DNA-like manifold distribution. In the following, we denote such a manifold an *eventail*[1]. A set of eventails from which we

wish to determine the camera velocity parameters is indicated in Figure 1.

The present section presents the theory of our method. We start with preliminaries and notations used throughout the paper. Next, we introduce a simple incidence relation that all events from one eventail need to satisfy. The constraint is transformed into minimal form, which not only reveals the intrinsic geometry of eventail manifolds and its dependency on spatio-temporal parameters, but also permits the derivation of a novel minimal solver that can be used for event clustering and a partial initialization of camera dynamics. To conclude, we present a complete velocity determination framework in which the partial observations from multiple eventails are merged into one common result.

3.1. Notations and preliminaries

We define the time interval of our slice of the space-time volume of events as $t \in [t_s - \Delta t, t_s + \Delta t]$. The set of events observed during this interval is given by $\mathcal{E} = \{\mathcal{E}_i\}_{i=1,...,N}$, where $\mathcal{E}_i$ represents the manifold cluster of events corresponding to the i-th 3D line $\mathbf{L}_i$. Each j-th event of the i-th cluster $e_{ij} = \{u_{ij}, v_{ij}, t_{ij}, p_{ij}\}$ is given by its u_{ij} and v_{ij} coordinates in the image plane, its timestamp t_{ij}, and its polarity p_{ij}[2]. The camera is assumed to be calibrated and we are given a function $[u\ v]^{\mathsf{T}} = \pi(\mathbf{P})$ that projects points $\mathbf{P} \in \mathbb{R}^3$ defined in the camera frame into the image plane. Conversely, we are given the inverse function $\mathbf{f} = \pi^{-1}(u, v)$ that transforms image plane coordinates into a unit-norm 3D direction vector defined in the camera frame and pointing towards the corresponding point in 3D. We furthermore define the first-order dynamics parameters $\mathbf{v}$ and $\boldsymbol{\omega}$, which represent the instantaneous translational and rotational velocity of the camera, respectively[3].

A line $\mathbf{L}$ in $\mathbb{R}^3$ can be represented by its direction vector $\mathbf{d}$ and a point $\mathbf{P}$ that lies on the line. The *Plücker coordinates* of this line are then defined as $[\mathbf{d}^{\mathsf{T}}\ \mathbf{m}^{\mathsf{T}}]^{\mathsf{T}} \doteq [\mathbf{d}^{\mathsf{T}}\ (\mathbf{P} \times \mathbf{d})^{\mathsf{T}}]^{\mathsf{T}}$, and the vector $\mathbf{m}$ is referred to as the *moment vector*. If two nonparallel lines $\mathbf{L}_1 = [\mathbf{d}_1^{\mathsf{T}}\ \mathbf{m}_1^{\mathsf{T}}]^{\mathsf{T}}$ and $\mathbf{L}_2 = [\mathbf{d}_2^{\mathsf{T}}\ \mathbf{m}_2^{\mathsf{T}}]^{\mathsf{T}}$ intersect, the following product is required to vanish, i.e.

$$\langle \mathbf{d}_1, \mathbf{m}_2 \rangle + \langle \mathbf{d}_2, \mathbf{m}_1 \rangle = 0, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ stands for the inner product. Note that, as long as the moment vector is strictly defined as the cross-product of a point on the line and the exact same direction vector that is used as the first three entries of the Plücker vector [34],

---

[1]The word is derived from the French word *éventail*, as the translating and rotating instantaneous line reprojection indeed resembles the geometry of the support sticks of a traditional fan.

[2]Note that, in this work, polarity is ignored.

[3]Note that the constant velocity motion assumption is violated if the camera experiences more jerky motion. One may argue that the approximation still holds within smaller time intervals, which will however limit the number of events and the baseline experienced over the course of the interval, and thus reduce the observability of depth and motion. As with any constant velocity approximation, our approach depends on sufficiently smooth motion.

the constraint is generally valid and there is no requirement on the norm of the direction vector.

3.2. Incidence Relationship

The entire problem is formulated relative to the camera frame at time t_s. Let $\mathbf{L}_i = [\mathbf{d}_i^\mathsf{T}\ \mathbf{m}_i^\mathsf{T}]^\mathsf{T}$ be the Plücker coordinates of a line defined in the said reference frame. $e_{ij} \in \mathcal{E}_i$ is the j-th event triggered by the moving reprojection of the i-th line $\mathbf{L}_i$. Using our first-order dynamics approximation, we can easily define the position of the camera center at time t_{ij} seen from the reference frame at time t_s as $\mathbf{C}\,[t_{ij}] = \mathbf{v} \cdot (t_{ij} - t_s)$, and the rotation that takes points from the camera frame at time t_{ij} back to the reference frame at time t_s as $\mathbf{R}\,[t_{ij}] = \exp(\lfloor \boldsymbol{\omega} \rfloor_\times (t_{ij} - t_s))$. Here, $\lfloor \boldsymbol{\omega} \rfloor_\times \in \mathbb{R}^{3 \times 3}$ is the skew-symmetric matrix formed from the angular rate $\boldsymbol{\omega} \in \mathbb{R}^3$. Note that, similar to [33], we assume the angular velocity of the camera to be known, as it can be reliably measured by modern IMU's. We will present a sensitivity analysis to these measurements in the experimental section. Last, the direction vector pointing from the camera frame at time t_{ij} to the 3D point on the line $\mathbf{L}_i$ that triggered the event is given by $\mathbf{f}_{ij} = \pi^{-1}\,(u_{ij}, v_{ij})$.

The measurement of the event can now easily be expressed as a ray defined in the reference frame at time t_s where $\mathbf{R}\,[t_{ij}]\,\mathbf{f}_{ij}$ is the direction of the ray, and $\mathbf{C}\,[t_{ij}]$ represents the origin of the ray (i.e. a point on the line). The ray is therefore described by the Plücker coordinates

$$[(\mathbf{R}\,[t_{ij}]\,\mathbf{f}_{ij})^\mathsf{T}\ \ (\mathbf{C}\,[t_{ij}] \times (\mathbf{R}\,[t_{ij}]\,\mathbf{f}_{ij}))^\mathsf{T}]^\mathsf{T}\,. \quad (2)$$

Finally, the incidence relation is given by (1), namely

$$\langle \mathbf{d}_i\,, \mathbf{C}\,[t_{ij}] \times (\mathbf{R}\,[t_{ij}]\,\mathbf{f}_{ij}) \rangle + \langle \mathbf{R}\,[t_{ij}]\,\mathbf{f}_{ij}\,, \mathbf{m}_i \rangle = 0\,. \quad (3)$$

Next, we assume that rotational velocity is already known. In practice, this assumption is easily satisfied by the addition of an IMU. In the continuation, we therefore assume that events are directly represented by their unrotated, normalized coordinates $\mathbf{f}'_{ij} = \mathbf{R}\,[t_{ij}]\,\mathbf{f}_{ij}$. Our incidence relation becomes

$$\langle \mathbf{d}_i\,, \left(\mathbf{v} \cdot t'_{ij}\right) \times \mathbf{f}'_{ij} \rangle + \langle \mathbf{f}'_{ij}\,, \mathbf{m}_i \rangle = 0\,, \quad (4)$$

where $t'_{ij} = t_{ij} - t_s$. Figure 2 indicates the geometry of the problem. The above incidence relation is already in polynomial form, and relates our measurements $\mathbf{f}'_{ij}$ and t'_{ij} to the remaining unknowns $\mathbf{v}$ and $\mathbf{L}_i$. However, it is not a minimal parametrization of the eventail manifold because Plücker coordinates are not minimal representations of lines, and the velocity component that is parallel to the line indeed has no influence on the incidence condition (i.e. it is unobservable, as sliding the camera along the direction of the line is unable to cause residual errors, a condition which is also known as the *aperture problem*).
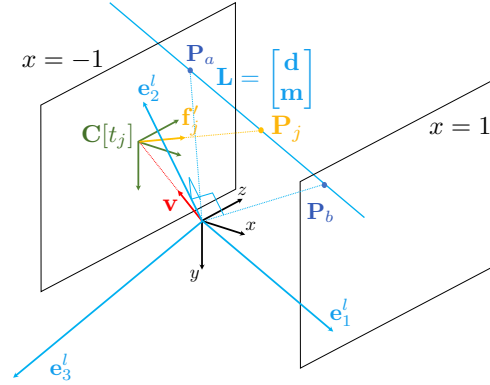

Figure 2. Incidence relationship between the line $\mathbf{L}$ with the two-point-two-plane parametrization, and the event with the bearing vector $\mathbf{f}'_j$. Camera velocity is given in the line-dependent reference frame $\mathbf{R}_\ell = [\mathbf{e}_1^\ell\ \mathbf{e}_2^\ell\ \mathbf{e}_3^\ell]$.

3.3. Transition into a minimal form

The following three subsections are with respect to a single cluster, only, which is why index i is dropped from the formulation. The transition into minimal form consists of two steps. We start by replacing the representation of the unknown 3D line by a two-point-two-plane parametrization [16], which is minimal. More specifically, we define the 3D line $\mathbf{L}$ by the intersection points with the planes $x = 1$ and $x = -1$ defined in the reference frame at time t_s. The two points in the reference frame are given by

$$\mathbf{P}_a = [-1, y_a, z_a]^\mathsf{T}\,, \ \mathbf{P}_b = [1, y_b, z_b]^\mathsf{T}\,. \quad (5)$$

As required, the new, minimal parametrization has only four degrees of freedom. Note that the orientation of the support planes for the two points can be arbitrarily changed, and their normal vectors can be chosen such that they are parallel to the approximate direction of the line in the image plane at time t_s (the latter can be set by for example considering the line connecting event samples). Without loss of generality, here we assume that the reference frame is already defined such that the support planes are given as defined above.

Next, we take care of the above-discussed problem that only part of the velocity vector is observable from a single-line observation under constant first-order dynamics. A minimal parametrization of the eventail manifold can only depend on part of the velocity vector, which notably are the components that are orthogonal to the line direction vector. To explicitly parametrize the vector as such, we introduce an intermediate, line-dependent reference frame in which the observable part of the camera velocity can be simply defined. The intermediate reference frame is given by

$$\mathbf{e}_1^\ell = \mathbf{P}_b - \mathbf{P}_a\,, \ \mathbf{e}_2^\ell = \mathbf{P}_b \times \mathbf{P}_a\,, \ \mathbf{e}_3^\ell = \mathbf{e}_1^\ell \times \mathbf{e}_2^\ell\,. \quad (6)$$

$\mathbf{e}_1^\ell$ is parallel to the line direction, $\mathbf{e}_2^\ell$ is orthogonal to plane traversing the camera center and the line $\mathbf{L}$, and $\mathbf{e}_3^\ell$ is contained in the latter plane but pointing orthogonally away from the line. Only velocity components along $\mathbf{e}_2^\ell$ and $\mathbf{e}_3^\ell$ are observable, hence the velocity can be substituted by

$$\mathbf{v} = \begin{bmatrix} \mathbf{e}_1^\ell & \mathbf{e}_2^\ell & \mathbf{e}_3^\ell \end{bmatrix} \cdot \begin{bmatrix} 0 & v_y^\ell & v_z^\ell \end{bmatrix}^\mathsf{T} = \mathbf{R}_\ell \mathbf{v}_\ell. \quad (7)$$

As required, this parametrization has only two additional degrees of freedom (i.e. v_y and v_z). Note that $\mathbf{R}_\ell$ is not an orthonormal rotation matrix, but it only represents an orthogonal basis for the minimal definition of the velocity. Using the new parameters, incidence relation (4) becomes

$$t_j'(\mathbf{P}_b - \mathbf{P}_a)^\mathsf{T}((\mathbf{R}_\ell \mathbf{v}_\ell) \times \mathbf{f}_j') - \mathbf{f}_j'^\mathsf{T}(\mathbf{P}_b \times \mathbf{P}_a) = 0. \quad (8)$$

The two-point-two-plane parametrization and the intermediate, line-dependent reference frame are shown in Figure 2.

3.4. Elementary properties of the minimal form

Before introducing the solver, here we list two corollaries along with their corresponding proofs.

Corollary 1: *The solution in terms of motion and structure parameters is scale invariant. In the case of our parametrization, the scale invariance is entirely reflected by the structure parameters. Scaling the line $\mathbf{L}$ will scale the velocity basis vectors $\mathbf{e}_2^\ell$ and $\mathbf{e}_3^\ell$ such that v_y^ℓ and v_z^ℓ remain unchanged. The latter are constant ratios.*

Proof: It is sufficient to prove that if scaling the line points $\mathbf{P}_a$ and $\mathbf{P}_b$ by a factor k, the basis vectors $\mathbf{e}_2^\ell$ and $\mathbf{e}_3^\ell$ will be equally scaled by k. Let us denote the intersection points with planes $x = -1$ and $x = 1$ of the scaled line by $\mathbf{P}_a'$ and $\mathbf{P}_b'$. $\mathbf{P}_a'$ and $\mathbf{P}_b'$ must lie on the line connecting $k\mathbf{P}_a$ and $k\mathbf{P}_b$ such that their first coordinate equals to -1 or 1, respectively. We therefore have

$$\begin{cases} \mathbf{P}_a' = k\mathbf{P}_a' + \lambda_a(k\mathbf{P}_b - k\mathbf{P}_a) = [-1 \; \cdot \; \cdot]^\mathsf{T} \\ \mathbf{P}_b' = k\mathbf{P}_a' + \lambda_b(k\mathbf{P}_b - k\mathbf{P}_a) = [1 \; \cdot \; \cdot]^\mathsf{T} \end{cases} \quad (9)$$

$$\Leftrightarrow \begin{cases} 2\lambda_a k - k = -1 \\ 2\lambda_b k - k = 1 \end{cases} \Leftrightarrow \begin{cases} \lambda_a = \frac{k-1}{2k} \\ \lambda_b = \frac{k+1}{2k} \end{cases}.$$

Back substituting in (9), we obtain

$$\begin{cases} \mathbf{P}_a' = \frac{k}{2}(\mathbf{P}_a + \mathbf{P}_b) - \frac{(\mathbf{P}_b - \mathbf{P}_a)}{2} \\ \mathbf{P}_b' = \frac{k}{2}(\mathbf{P}_a + \mathbf{P}_b) + \frac{(\mathbf{P}_b - \mathbf{P}_a)}{2} \end{cases}. \quad (10)$$

The new basis vector $\mathbf{e}_2'^\ell$ is finally given by

$$\begin{aligned} \mathbf{e}_2'^\ell &= \mathbf{P}_b' \times \mathbf{P}_a' = -k(\mathbf{P}_a + \mathbf{P}_b) \times \frac{\mathbf{P}_b - \mathbf{P}_a}{2} \\ &= -k\left(\frac{\mathbf{P}_a \times \mathbf{P}_b}{2} - \frac{\mathbf{P}_b \times \mathbf{P}_a}{2}\right) = k(\mathbf{P}_b \times \mathbf{P}_a) = k\mathbf{e}_2^\ell. \end{aligned}$$

Next, it is easy to see that $\mathbf{e}_1'^\ell = \mathbf{P}_b' - \mathbf{P}_a' = \mathbf{P}_b - \mathbf{P}_a = \mathbf{e}_1^\ell$. Subsequently, $\mathbf{e}_3'^\ell = \mathbf{e}_1'^\ell \times \mathbf{e}_2'^\ell = k\mathbf{e}_1^\ell \times \mathbf{e}_2^\ell = k\mathbf{e}_3^\ell$. ∎

Corollary 2: *There exists a solution duality, i.e. if $\{\mathbf{P}_a, \mathbf{P}_b, \mathbf{v}_\ell\}$ is a valid solution, then $\{\mathbf{P}_a' = -\mathbf{P}_b, \mathbf{P}_b' = -\mathbf{P}_a, \mathbf{v}_\ell' = \mathbf{v}_\ell\}$ is also a solution. It notably corresponds to a reversal of the velocity and a placement of the line behind the camera. It is furthermore interesting to note that only the line parameters are affected by the solution duality.*

Proof: $\mathbf{e}_2'^\ell = \mathbf{P}_b' \times \mathbf{P}_a' = -(\mathbf{P}_b \times \mathbf{P}_a) = -\mathbf{e}_2^\ell$, and $\mathbf{e}_1'^\ell = \mathbf{P}_b' - \mathbf{P}_a' = \mathbf{P}_b - \mathbf{P}_a = \mathbf{e}_1^\ell$, which is why $\mathbf{e}_3'^\ell = -\mathbf{e}_3^\ell$. Let $\mathbf{R}_\ell' = \begin{bmatrix} \mathbf{e}_1'^\ell & \mathbf{e}_2'^\ell & \mathbf{e}_3'^\ell \end{bmatrix}$. Given that the first coordinate of $\mathbf{v}_\ell$ is always zero, we therefore have $\mathbf{R}_\ell' \cdot \mathbf{v}_\ell' = -\mathbf{R}_\ell \cdot \mathbf{v}_\ell$. Finally, for our incidence relation, we have

$$\begin{aligned} & t_j'(\mathbf{P}_b' - \mathbf{P}_a')^\mathsf{T}((\mathbf{R}_\ell' \mathbf{v}_\ell') \times \mathbf{f}_j') - \mathbf{f}_j'^\mathsf{T}(\mathbf{P}_b' \times \mathbf{P}_a') \\ = & -t_j'(\mathbf{P}_b - \mathbf{P}_a)^\mathsf{T}((\mathbf{R}_\ell \mathbf{v}_\ell) \times \mathbf{f}_j') - \mathbf{f}_j'^\mathsf{T}(\mathbf{P}_a \times \mathbf{P}_b) \\ = & \; 0. \end{aligned}$$
∎

3.5. Five-point Minimal Solver

There exist six unknowns in our formulation. However, as mentioned in the above corollaries, there is an additional scale ambiguity in the line's support points. Hence, the inherent number of degrees of freedom is five, and a minimal solver for an eventail can be found by constructing five incidence constraints from five randomly picked events from the cluster. To remove the scale invariance, an additional constraint on the scale is added. Given that only the structure parameters are affected by the scale invariance, the scale constraint needs to include the related variables. We constrain the scale by adding the equation

$$(\mathbf{R}_\ell \mathbf{v}_\ell)^\mathsf{T} \cdot \mathbf{R}_\ell \mathbf{v}_\ell - 1 = 0. \quad (11)$$

Using Gröbner basis theory [9], it is easy to find out that this problem has only two solutions. However, the fact that part of the solution variables is uniquely defined leads to some peculiarities in the derivation of the elimination template [23]. Simply put, there are variable orderings for which the template leads to an equation where either v_y or v_z is a constant. More importantly though, after back substituting this variable into other equations, no simple solution for the remaining variables is apparent, but a second elimination template of substantial size needs to be solved.

To solve this problem, we analyzed all 720 possible monomial orderings and related elimination templates. We found out that 240 out of the 720 orderings directly lead to the unique solution for either v_y and v_z and are unlikely to reveal the double solution for the line's support points. The remaining 480 orderings and related elimination templates directly lead to the two solutions. The elimination templates can vary quite substantially in size and reach from 100×428 to 244×820.

The smallest template that directly leads to the two solutions is indicated in Figure 3, and has a size of 154×578.

Figure 3. The chosen elimination template for our 5-point solver.

Note that, given that the elimination leads to only two solutions, no action matrix decomposition is needed. In turn, the actual solutions are straightforward to recover from the last few rows of the template. Among these, we select the one resulting in the following variable ordering: $v_y > v_z > y_a > z_a > z_b > y_b$. In particular, it leads to one quadratic equation in y_b, which can be back substituted into three further equations to lead to univariate constraints in y_a, z_a, and z_b, respectively. With the structure parameters known, any two events can be used to construct bi-variate linear constraints on v_y and v_z.

We embed the 5-point solver into RANSAC and use it to find the parameters of the eventail, which correspond to a partial dynamics recovery, as well as an event clustering. Sampling strategies and inlier criteria are discussed in Section 4.

3.6. Velocity averaging from multiple eventails

We are now moving back to multiple eventails, which is why the index i is reintroduced. The result we obtain from the clustering algorithm is a partial observation of the velocity $\mathbf{v}$ for each eventail. The observation is given by the two ratios v_{yi}^ℓ and v_{zi}^ℓ scaling the second and third basis vectors of $\mathbf{R}_{\ell i}$, respectively. Introducing the unobservable component κ_i along the first basis vector, we have

$$\mathbf{v} = \mathbf{e}_{1i}^\ell \cdot \kappa_i + \mathbf{e}_{2i}^\ell \cdot v_{yi}^\ell + \mathbf{e}_{3i}^\ell \cdot v_{zi}^\ell. \qquad (12)$$

By multiplying the equation from the left with either the transpose of $\mathbf{e}_{2i}^\ell$ or $\mathbf{e}_{3i}^\ell$, and exploiting the orthogonal property of the basis vectors, we easily obtain

$$\begin{cases} \mathbf{e}_{2i}^{\ell\mathsf{T}}\mathbf{v} = \mathbf{e}_{2i}^{\ell\mathsf{T}}\mathbf{e}_{1i}^\ell \cdot \kappa_i + \mathbf{e}_{2i}^{\ell\mathsf{T}}\mathbf{e}_{2i}^\ell \cdot v_{yi}^\ell + \mathbf{e}_{2i}^{\ell\mathsf{T}}\mathbf{e}_{3i}^\ell \cdot v_{zi}^\ell \\ \mathbf{e}_{3i}^{\ell\mathsf{T}}\mathbf{v} = \mathbf{e}_{3i}^{\ell\mathsf{T}}\mathbf{e}_{1i}^\ell \cdot \kappa_i + \mathbf{e}_{3i}^{\ell\mathsf{T}}\mathbf{e}_{2i}^\ell \cdot v_{yi}^\ell + \mathbf{e}_{3i}^{\ell\mathsf{T}}\mathbf{e}_{3i}^\ell \cdot v_{zi}^\ell \end{cases}$$

$$\Leftrightarrow \begin{cases} \mathbf{e}_{2i}^{\ell\mathsf{T}}\mathbf{v} = \|\mathbf{e}_{2i}^\ell\|_2^2 \cdot v_{yi}^\ell \\ \mathbf{e}_{3i}^{\ell\mathsf{T}}\mathbf{v} = \|\mathbf{e}_{3i}^\ell\|_2^2 \cdot v_{zi}^\ell \end{cases} \Leftrightarrow \begin{cases} \|\mathbf{e}_{2i}^\ell\|_2^{-2} \cdot \mathbf{e}_{2i}^{\ell\mathsf{T}}\mathbf{v} = v_{yi}^\ell \\ \|\mathbf{e}_{3i}^\ell\|_2^{-2} \cdot \mathbf{e}_{3i}^{\ell\mathsf{T}}\mathbf{v} = v_{zi}^\ell \end{cases}.$$

Stacking the result from all N observed lines and taking into account that each individual partial velocity observation is affected by an unknown scale factor, we obtain the linear averaging scheme

$$\begin{bmatrix} \|\mathbf{e}_{21}^\ell\|_2^{-2} \cdot \mathbf{e}_{21}^{\ell\mathsf{T}} & -v_{y1}^\ell & \cdots & 0 \\ \|\mathbf{e}_{31}^\ell\|_2^{-2} \cdot \mathbf{e}_{31}^{\ell\mathsf{T}} & -v_{z1}^\ell & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \|\mathbf{e}_{2N}^\ell\|_2^{-2} \cdot \mathbf{e}_{2N}^{\ell\mathsf{T}} & 0 & \cdots & -v_{yN}^\ell \\ \|\mathbf{e}_{3N}^\ell\|_2^{-2} \cdot \mathbf{e}_{3N}^{\ell\mathsf{T}} & 0 & \cdots & -v_{zN}^\ell \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{v} \\ \lambda_1 \\ \cdot \\ \cdot \\ \cdot \\ \lambda_N \end{bmatrix}}_{} = \mathbf{0}, \qquad (13)$$

$$[\mathbf{A}\ \mathbf{B}]$$

where the additionally requested scaling factors are given by $\lambda_1, \ldots, \lambda_N$. Multiplying the equation from the left with $[\mathbf{A}\ \mathbf{B}]^\mathsf{T}$, we obtain the form

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^\mathsf{T} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \lambda_1 \\ \vdots \\ \lambda_N \end{bmatrix} = \mathbf{0}, \text{ where} \qquad (14)$$

$$\mathbf{U}_{3\times 3} = \sum_{i=1}^N \left(\frac{\mathbf{e}_{2i}^\ell \cdot \mathbf{e}_{2i}^{\ell\mathsf{T}}}{\|\mathbf{e}_{2i}^\ell\|_2^4} + \frac{\mathbf{e}_{3i}^\ell \cdot \mathbf{e}_{3i}^{\ell\mathsf{T}}}{\|\mathbf{e}_{3i}^\ell\|_2^4} \right)$$

$$\mathbf{V}_{N\times N} = \text{diag}\left((v_{y1}^\ell)^2 + (v_{z1}^\ell)^2, \ldots, (v_{yN}^\ell)^2 + (v_{zN}^\ell)^2 \right)$$

$$\mathbf{W}_{N\times 3}^\mathsf{T} = \begin{bmatrix} -\frac{v_{y1}^\ell}{\|\mathbf{e}_{21}^\ell\|_2^2}\mathbf{e}_{21}^{\ell\mathsf{T}} - \frac{v_{z1}^\ell}{\|\mathbf{e}_{31}^\ell\|_2^2}\mathbf{e}_{31}^{\ell\mathsf{T}} \\ \vdots \\ -\frac{v_{yN}^\ell}{\|\mathbf{e}_{2N}^\ell\|_2^2}\mathbf{e}_{2N}^{\ell\mathsf{T}} - \frac{v_{zN}^\ell}{\|\mathbf{e}_{3N}^\ell\|_2^2}\mathbf{e}_{3N}^{\ell\mathsf{T}} \end{bmatrix}.$$

Applying the Schur complement trick, we easily obtain

$$\left[\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^\mathsf{T}\right]\mathbf{v} = \mathbf{0}, \qquad (15)$$

which lets us find $\mathbf{v}$ via Eigen decomposition of a 3×3 matrix. Note that $\mathbf{V}^{-1}$ is computed in linear time by simply inverting each element along the diagonal of $\mathbf{V}$.

4. Experiments

We perform both simulation tests and real-world experiments. We first confirm the theoretical correctness of the proposed 5-point solver, its noise resilience, and discuss the implications of different event sampling strategies. Next, we test the influence of violations of the constant linear velocity motion assumption. We furthermore experimentally confirm the non-planar nature of the eventail manifold, and conclude with experiments on a public benchmark, demonstrating the advantage over existing bootstrapping methods. In order to evaluate the accuracy of the results, we adopt one of the criteria in [33], which is the direction error ϕ between the estimated and the ground truth velocities, since the scale is not observable.

4.1. Noise resilience on single cluster

We start by evaluating the performance of the 5-point solver under different noise setups over synthetic data, and discuss the impact of different event sampling strategies. Samples from individual manifolds are generated as follows. We first randomly generate two lines in the image
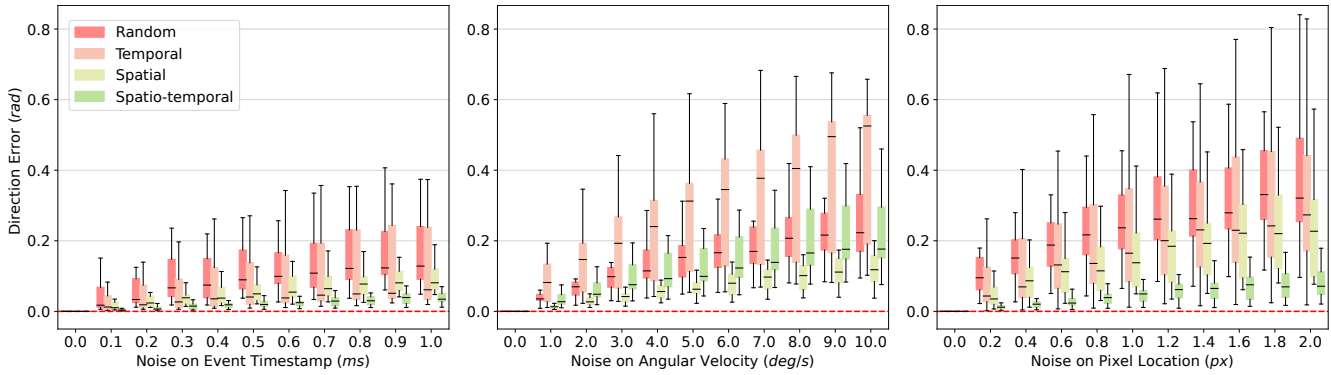
Figure 4. Results for the directional accuracy of the partially observed camera velocity as a function of noise in the event timestamps, the measured angular velocity, and the event locations. Each box denotes the range from the first quartile to the third quartile of the error distribution. The median is marked as the black line in the middle.

plane that represent the location of the line at the beginning and the end of the temporal window of events. The interval length is set to $0.5\,\mathrm{s}$. Next, we sample randomly directed linear and angular velocities of $1.0\,\mathrm{m/s}$ and $90\,°/\mathrm{s}$ magnitude, respectively. By factoring in the interval duration, we can deduce the relative camera location between the beginning and the end of the interval by linear motion model, and extract the corresponding 3D line via triangulation. The line is furthermore given a finite length in 3D. The virtual event camera has a resolution of 640×480 and a focal length of 320 pixels. The explained way of defining experiments ensures that the line passes through a sufficiently large area of the image canvas during the interval duration, thereby simulating well-posed problems where the camera exhibits sufficient displacement relative to the line in 3D. We randomly sample events, by first sampling a point on the 3D line, and then sampling the time stamp, at which that point is projected into the image plane. Each such projection is then denoted as an event. We analyze four strategies for sampling events in this way:

- *Random:* Event timestamps and 3D points on the line are both sampled randomly.

- *Temporal:* the time interval is evenly divided into five sub-intervals, and each of the five events is assigned a random timestamp within an individual sub-interval.

- *Spatial:* the line is evenly divided into five segments, and making sure that each of the five events is triggered by a random 3D point on an individual sub-segment.

- *Spatio-temporal:* a combination of the spatial and temporal sampling strategies.

To conclude the experiment setup, we introduce three types of noise sources with different magnitudes, namely pixel noise, timestamp jitter, and noise to the camera's angular velocity, which is assumed to be given by an auxiliary sensor. The magnitude of the pixel noise and the noise

on camera angular velocities is consistent within the same noise level but varies in direction. Zero-mean Gaussian noise is used for timestamp noise. Results are presented in Figure 4. Each box in the plot represents the mean errors from fifteen different geometry-motion configurations. Within each configuration, we conducted 100 evaluations with different event samples. Note that since the 5-point solver only takes readings from one cluster and produces partial observation, we compare the estimated velocity with the normalized ground truth velocity in the direction perpendicular to the line. Moreover, as is standard in SLAM evaluation pipelines, we do not report evaluations of the line parameters, as their error is subsumed in the motion errors.

Errors generally increase as noise levels increase. Without noise, our solver can always produce zero-error results, hence proving the theoretical correctness of the proposed method. There does not exist, however, a golden sampling strategy to produce minimum errors for different noise sources, though ensuring spatial distribution among the five sampled events is crucial to achieve accurate results. In the following RANSAC experiments, we therefore alternate between spatial and spatio-temporal sampling.

4.2. Multiple clusters and validity of motion model

We have developed an Event-based Geometric wireframe Generator, entitled EGG, in order to conduct experiments over longer time periods. The wireframes are composed of finite lines which simulate strong gradient locations in the scene. Note that the direction of the gradient does not need to be specified, given that we ignore polarity in our formulation. The simulator supports different types of motion models, including both spline-based and non-polynomial motion. Events are generated whenever a projected line comes across a pixel center, thereby producing events with accurate timestamps. A realistic IMU model is also implemented with bias and random walk. In a nutshell, the EGG simulator we have developed is capable of pro-
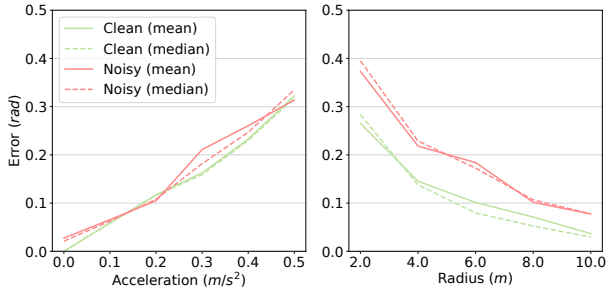
Figure 5. Average directional errors of fully estimated linear velocity over ten éventails. Results are evaluated for clean and noisy data, and different violations of the motion model assumptions.

ducing accurate ground truth readings of poses and twists, corrupted IMU readings, as well as line-generated events with precise timestamps.

We now investigate the robustness and accuracy of the overall velocity determination from multiple eventails. We generate ten line segments within the volume $[-2, 2] \times [-2, 2] \times [1.5, 3.0]$. Each event is labeled with its corresponding line to bypass clustering in the present experiment. In the next two sections, we evaluate the full RANSAC pipeline. Note, however, that we are still running RANSAC to fit the eventail parameters to each event cluster. We adopt the angular reprojection error [25] within the RANSAC algorithm. We conduct two types of motion, both of which violate the constant linear motion assumption. The first moves in a circular arc, with a tangential velocity of $1.0\,\mathrm{m/s}$ and with a radius ranging from $2\,\mathrm{m}$ to $10\,\mathrm{m}$ in increments of $2\,\mathrm{m}$. Each sequence lasts $0.3\,\mathrm{s}$, and so does the chosen time window. For each configuration, we generate twenty sets of events without noise and another twenty with noise. Perturbations are given by zero-mean Gaussian noise with a standard deviation of 1 pixel on location and $1\,\mathrm{ms}$ on timestamp. We also use corrupted IMU readings with a realistic bias. Finally, the ground truth camera motion has no rotation and faces in a constant direction. In the second simulation, we add acceleration with magnitude ranging from $0.1\,\mathrm{m/s^2}$ to $0.5\,\mathrm{m/s^2}$ to the linear velocity. Again, the time window is $0.3\,\mathrm{s}$.

Figure 5 shows that without noise and model violations, our method can always produce zero-error results. As the velocity becomes less constant (larger acceleration, smaller radius), the error increases, but the proposed solver still shows high noise robustness in these conditions.

4.3. Performance under high dynamics

To demonstrate the full potential of our parametrization, we generate sequences of $1\,\mathrm{s}$ with constant but very significant linear and angular velocities. Two non-parallel lines with comparable depth, i.e. from $[0, 0.75, 3]$ to $[0, 2, 3]$ and from $[0.38, -0.65, 3]$ to $[0.75, -1.3, 3]$ respectively, are

placed in front of the camera. The camera is moving towards the lines with a linear velocity of $[0.4, 0.4, 2.0]\mathrm{m/s}$, and with a self-rotation of $[0, 0, -2\pi]\mathrm{rad/s}$. The noise level is kept the same as in the previous simulation. As showcased in the example in Figure 1, our method successfully fits the two manifolds from five events with inlier ratios of $67.16\,\%$ and $73.22\,\%$, respectively, and a velocity direction error of only $0.01\,\mathrm{rad}$. On the other hand, traditional plane-based fitting fails in this case and splits up the set into 34 subsets. As explained next, this difference in clustering will impact the achievable final accuracy.

4.4. Real-world Experiment

We validate the method on four sequences from a public benchmark dataset [14] which feature clear line structures. The dataset provides VGA event recordings from a Gen3 Prophesee camera, $200\,\mathrm{Hz}$ ground truth camera poses from a MoCap system, and $200\,\mathrm{Hz}$ measurements from an XSens MTi-30 AHRS IMU. To increase efficiency we downsample the events in each sequence by a factor of ten. We then split the sequence into non-overlapping intervals of events with a $0.3\,\mathrm{s}$ duration each. To find individual line clusters we adopt two different approaches: In the first approach, we find clusters using cilantro [48] which finds spatio-temporal planes, as was used in [33], then when sampling 5 tuples of events from these clusters we count inliers over the entire event set, which has higher robustness than simply considering events within that cluster, as was done in [33]. In this experiment we set the maximum number of event clusters to five. The second one is unique to our method, and consists of running RANSAC on the entire window of events until a maximal inlier ratio is found, then extracting that line and partial velocity estimate together with the inlier events from the event set, and repeating this step for a total of five times. We use this strategy whenever cilantro fails to provide sufficient clusters. This means that our method is strictly more robust than [33] which cannot recover when cilantro fails. The estimated velocity is obtained by velocity averaging from multiple eventails and is further compared with the ground truth velocity in direction error ϕ. Comparative results are listed in Table 1 reporting the mean and median error of successful samples in each sequence. Since CELC+opt fails on a subset of the sequences, we also restrict the evaluation of our method to the subsequence where CELC+opt is successful, and denote these results with an asterisk *. Where there are too few clusters, CELC+opt fails to output valid results in some samples.

Results We find that while the success rate of [33] ranges between 23% and 70% our method consistently achieves 100%. Moreover, on the subset where [33] is successful both methods are comparable in terms of accuracy. This highlights that our methods signfcantly improves on the robustness of existing closed-form solvers for linear velocity.

Table 1. Real-world results on the VECtor benchmark [14]. We report the success rate, i.e. , the percentage of sequence's sections where the algorithm outputs reasonable results, as well as the velocity direction error ϕ^* only computed in these sub-sections. For our method, which always has 100% success rate, we also report the error ϕ over the full sequence.

Seq. Name	CELC+opt [33]			Ours				
	ϕ^*_{mean}	ϕ^*_{median}	Success	ϕ^*_{mean}	ϕ^*_{median}	ϕ_{mean}	ϕ_{median}	Success
board-slow	0.451	0.434	65.69 %	**0.429**	**0.385**	0.484	0.416	100 %
mountain-normal	**0.483**	**0.512**	56.70 %	0.542	0.528	0.584	0.586	100 %
desk-normal	0.464	**0.464**	69.86 %	**0.461**	0.474	0.461	0.466	100 %
sofa-normal	**0.419**	0.455	23.16 %	0.532	**0.438**	0.550	0.514	100 %
* subset where CELC+opt [33] does not fail								

5. Conclusion and Future Work

The present work provides a new understanding of the geometry of line-generated events perceived under constant linear and angular velocities. We have derived a theoretically correct parametrization of the manifold-distribution of such events, and showed how the parametrization partially involves camera dynamics parameters. Our presented minimal solver for the manifold parameters has been successfully embedded into RANSAC, based on which we studied resilience against noise and motion model violations. Extensive tests on real-world data have validated the superior ability of our method to fit éventail manifold parameters, thereby increasing accuracy and the overall success rate of the presented bootstrapping algorithm.

Prospective future directions include extending the proposed averaging scheme with uncertainty estimates from multiple manifolds, as well as optimizing over the cluster inliers for improved robustness and accuracy of the results. Moreover, while we showed that our sequential RANSAC approach was sufficient for fitting multiple lines into a given event stream, more sophisticated robust multi-model fitting techniques like Progressive-X [3] could be used to minimize the number of outliers. Finally, the current method omits the event polarity and does not perform any temporal smoothing or fusion of accelerometer or gyroscope readings from the IMU. Extending the approach in this direction would make it acceleration-aware, as [8], and lead to improved modeling of the camera motion. Despite these limitations, we believe that our findings lay the cornerstones for highly successful, incremental smoothing-based motion estimation.

References

[1] Adrien Bartoli and Peter Sturm. The 3d line motion matrix and alignment of line reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1. IEEE, 2001. 3

[2] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding (CVIU)*, 100(3):416–441, 2005. 3

[3] Dániel Baráth and Jiri Matas. Progressive-x: Efficient, anytime, multi-model fitting algorithm. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 06 2019. 9

[4] Christian Brändli, Jonas Strubel, Susanne Keller, Davide Scaramuzza, and Tobi Delbruck. ELiSeD—an event-based line segment detector. In *Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–7. IEEE, 2016. 3

[5] Samuel Bryner, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 325–331. IEEE, 2019. 2

[6] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics (T-RO)*, 32(6):1309–1332, 2016. 2

[7] Andrea Censi and Davide Scaramuzza. Low-latency event-based visual odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–710. IEEE, 2014. 2

[8] William Oswaldo Chamorro Hernández, Juan Andrade-Cetto, and Joan Solà Ortega. High-speed event camera tracking. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020. 2, 9

[9] David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013. 5

[10] Lukas Everding and Jörg Conradt. Low-latency line tracking using event-based dynamic vision sensors. *Frontiers in neurorobotics*, 12:4, 2018. 3

[11] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12280–12289, 2019. 2

[12] Guillermo Gallego, Jon EA Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-dof camera tracking for high-speed applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(2):2402–2412, 2016. 2

[13] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3867–3876, 2018. 2

[14] Ling Gao, Yuxuan Liang, Jiaqi Yang, Shaoxun Wu, Chenyu Wang, Jiaben Chen, and Laurent Kneip. VECtor: A versatile event-centric benchmark for multi-sensor slam. *IEEE Robotics and Automation Letters*, 7(3):8217–8224, 2022. 8, 9

[15] Mathias Gehrig, Sumit Bam Shrestha, Daniel Mouritzen, and Davide Scaramuzza. Event-based angular velocity regression with spiking networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4195–4202. IEEE, 2020. 3

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 4

[17] Richard I Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision (IJCV)*, 22(2):125–140, 1997. 3

[18] Yijia He, Ji Zhao, Yue Guo, Wenhao He, and Kui Yuan. PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features. *Sensors*, 18(4):1159, 2018. 3

[19] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza. Event-aided direct sparse odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5781–5790, 2022. 2

[20] Sio-Hoi Ieng, João Carneiro, and Ryad B Benosman. Event-based 3d motion flow estimation using 4d spatio temporal subspaces properties. *Frontiers in Neuroscience*, 10:596, 2017. 3

[21] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 349–364. Springer, 2016. 2

[22] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pages 16–23. IEEE, 2016. 2

[23] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 2, page 2008, 2008. 5

[24] 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 *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pages 5863–5870. IEEE, 2020. 1, 2, 3

[25] Seong Hun Lee and Javier Civera. Closed-form optimal two-view triangulation based on angular errors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2681–2689, 2019. 8

[26] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Globally optimal contrast maximisation for event-based motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6349–6358, 2020. 2

[27] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5419–5427, 2018. 3

[28] Anton Mitrokhin, Zhiyuan Hua, Cornelia Fermuller, and Yiannis Aloimonos. Learning visual motion segmentation using event surfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14414–14423, 2020. 3

[29] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics (T-RO)*, 34(6):1425–1440, 2018. 2, 3

[30] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pages 2761–2768. IEEE, 2014. 2

[31] Xin Peng, Ling Gao, Yifu Wang, and Laurent Kneip. Globally-optimal contrast maximisation for event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 44(7):3479–3495, 2021. 2

[32] Xin Peng, Yifu Wang, Ling Gao, and Laurent Kneip. Globally-optimal event camera motion estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–67. Springer, 2020. 2

[33] Xin Peng, Wanting Xu, Jiaqi Yang, and Laurent Kneip. Continuous event-line constraint for closed-form velocity initialization. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2021. 1, 2, 3, 4, 6, 8, 9

[34] Julius Plücker. On a new geometry of space. *Philosophical Transactions of the Royal Society of London*, 155:725–791, 1865. 3

[35] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer. PL-SLAM: Real-time monocular visual slam with points and lines. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4503–4508. IEEE, 2017. 3

[36] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using

keyframe-based nonlinear optimization. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 2

[37] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. 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, 2016. 2, 3

[38] Hochang Seok and Jongwoo Lim. Robust feature tracking in dvs event stream using bézier mapping. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1658–1667, 2020. 3

[39] Timo Stoffregen and Lindsay Kleeman. Event cameras, contrast maximization and reward functions: an analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12300–12308, 2019. 2

[40] David Reverter Valeiras, Xavier Clady, Sio-Hoi Ieng, and Ryad Benosman. Event-based line fitting and segment detection using a neuromorphic visual sensor. *IEEE Transactions on Neural Networks and Learning Systems*, 30(4):1218–1230, 2018. 3

[41] 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 Letters*, 3(2):994–1001, 2018. 1, 2

[42] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(4):722–732, 2008. 3

[43] David Weikersdorfer, David B Adrian, Daniel Cremers, and Jörg Conradt. Event-based 3d slam with a depth-augmented dynamic vision sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 359–364. IEEE, 2014. 2

[44] David Weikersdorfer, Raoul Hoffmann, and Jörg Conradt. Simultaneous localization and mapping for event-based vision systems. In *International Conference on Computer Vision Systems*, pages 133–142. Springer, 2013. 2, 3

[45] Stephan Weiss, Roland Brockers, and Larry Matthies. 4dof drift free navigation using inertial cues and optical flow. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pages 4180–4186. IEEE, 2013. 2

[46] Juyang Weng, Thomas S Huang, and Narendra Ahuja. Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(03):318–336, 1992. 3

[47] Wenzhen Yuan and Srikumar Ramalingam. Fast localization and tracking using event sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4564–4571. IEEE, 2016. 3

[48] Konstantinos Zampogiannis, Cornelia Fermuller, and Yiannis Aloimonos. cilantro: A lean, versatile, and efficient library for point cloud data processing. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 1364–1367, New York, NY, USA, 2018. ACM. 8

[49] Guoxuan Zhang, Jin Han Lee, Jongwoo Lim, and Il Hong Suh. Building a 3-d line-based map using stereo slam. *IEEE Transactions on Robotics (T-RO)*, 31(6):1364–1377, 2015. 3

[50] Yi Zhou, Guillermo Gallego, and Shaojie Shen. Event-based stereo visual odometry. *IEEE Transactions on Robotics (T-RO)*, 37(5):1433–1450, 2021. 2

[51] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5391–5399, 2017. 2, 3

[52] Yi-Fan Zuo, Jiaqi Yang, Jiaben Chen, Xia Wang, Yifu Wang, and Laurent Kneip. DEVO: Depth-event camera visual odometry in challenging conditions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2179–2185. IEEE, 2022. 2