

# An N-Point Linear Solver for Line and Motion Estimation with Event Cameras

Ling Gao<sup>1\*</sup> Daniel Gehrig<sup>2\*</sup> Hang Su<sup>1</sup> Davide Scaramuzza<sup>2</sup> Laurent Kneip<sup>1</sup>

<sup>1</sup> Mobile Perception Lab, ShanghaiTech University, China

<sup>2</sup> Robotics and Perception Group, University of Zurich, Switzerland

## Abstract

Event cameras respond primarily to edges—formed by strong gradients—and are thus particularly well-suited for line-based motion estimation. Recent work has shown that events generated by a single line each satisfy a polynomial constraint which describes a manifold in the space-time volume. Multiple such constraints can be solved simultaneously to recover the partial linear velocity and line parameters. In this work, we show that, with a suitable line parametrization, this system of constraints is actually linear in the unknowns, which allows us to design a novel linear solver. Unlike existing solvers, our linear solver (i) is fast and numerically stable since it does not rely on expensive root finding, (ii) can solve both minimal and overdetermined systems with more than 5 events (i.e.  $N \geq 5$ ), and (iii) admits the characterization of all degenerate cases and multiple solutions. The found line parameters are singularity-free and have a fixed scale, which eliminates the need for auxiliary constraints typically encountered in previous work. To recover the full linear camera velocity we fuse observations from multiple lines with a novel velocity averaging scheme that relies on a geometrically-motivated residual, and thus solves the problem more efficiently than previous schemes which minimize an algebraic residual. Extensive experiments in synthetic and real-world settings demonstrate that our method surpasses the previous work in numerical stability, and operates over 600 times faster.

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

## 1. Introduction

Man-made scenes contain a multitude of straight lines, and exploiting these lines for motion estimation is an important feature of modern mobile vision systems like AR/VR devices and robotic systems [17, 23, 25, 42]. However, computer vision algorithms aimed at leveraging these line features still suffer from fundamental limitations when using standard frame-based sensing: During high-speed motion and challenging illumination conditions, these sensors

suffer from motion blur and saturation effects, which have deleterious effects on line feature extraction. Event cameras [10] are biologically inspired sensors that address the limitation of frame-based sensors by instead only measuring the *changes in intensity* at a per-pixel level, and they do this with high dynamic range, low motion blur, high temporal resolution, and high spatial data-sparsity.

Due to their working principle, event cameras respond primarily to edges—formed by strong gradients—and are thus particularly well-suited for line-based motion estimation. A recent breakthrough [12] in event-based motion estimation introduced an incidence relation that enforces the intersection of bearing vectors emitted by events and a corresponding line that generates those events. Using this relation, a 5-point minimal solver was designed that recovers the parameters of a minimal two-point-two-plane parametrization of the line and two velocity ratios in the plane perpendicular to the line, using the Gröbner basis method and polynomial elimination theory. However, this solver suffers from several limitations: First, it relies on a non-minimal line representation using four degrees of freedom (DoF) that (i) fails to realize that, in the absence of scale, only three DoF are needed, and (ii) encounters singularities when describing lines parallel to the two planes. Secondly, to solve for the motion and line parameters, previous work employs a polynomial solver, which (i) is by definition minimal and thus incapable of incorporating more than five events, and (ii) relies on root finding algorithms that are expensive to run and suffer from instabilities that are not easily detected.

This work addresses these limitations with two important innovations: First, it introduces a new line representation based on the angle-axis representation of a rotation matrix, which is singularity-free and only depends on three DoF and thus implicitly enforces scale ambiguity. Second, in formulating the incidence relation with this new parametrization, we derive a simple algorithm for determining motion and line parameters that only relies on solving a linear system and simple vector operations and is thus orders of magnitude faster than the polynomial solver in [12]. This linear system is easily extended to  $N > 5$  events, with a minimal increase in complexity, which enables solution refine-

\*indicates equal contribution

ment with inliers when employing random sample consensus (RANSAC) schemes.

The proposed solver sheds light on all possible cases of degenerate solutions, how they arise, and all additional solutions that arise from symmetries in the incidence relation. Finally, expressing the incidence relation in terms of our new line parametrization allows us to fully characterize and visualize the types of manifolds circumscribed by events generated by a single line. Our contributions are:

- A minimal, three DoF representation of lines based on the angle-axis representation of a rotation matrix. This representation encodes a reference frame centered at the 3D line, and enforces a unit distance to the closest point.
- A linear algorithm for determining line and motion parameters from a set of events triggered by a line. This algorithm is fast, and extensible to multiple events, and sheds light on degenerate and multiplicitous solutions.
- A simpler and faster scheme for fusing partial linear velocity measurements from multiple lines, based on geometrically-motivated constraints, instead of solving algebraic equations.

We validate our method extensively in simulated and real-world settings. In particular, our solver is on average 600 times faster, taking only 3.25  $\mu$ s for five events, compared to 2046  $\mu$ s for [12], while achieving a similar performance.

## 2. Related Work

Ego-motion estimation is crucial for intelligent mobile devices and thus has been the subject of extensive research over the past few decades. The sub-class of vision-based solutions comprises single-camera, stereo-camera, and multi-camera solutions that are potentially supported by an inertial measurement unit. A review would go beyond the scope of the present paper, and the reader is kindly referred to recent reviews such as the one by Cadena *et al.* [4]. This paper focuses on motion estimation with event cameras.

The latter is a challenging problem that is initially often addressed for constrained scenarios such as 2D motion [39], known depth or 3D structure [3, 5–7, 27, 40, 46], pure rotation [8], and homographic warping [9, 20, 24, 31, 34]. The community has furthermore explored the combination with other sensors such as standard cameras [16, 20, 38], inertial units [38, 45], or a second event camera [44]. In turn, the present paper considers motion in 3D with a single event camera and in arbitrary environments. Different optimization-based [21, 28, 32], filter-based [19, 45] and learning-based [13, 26] solutions have already been presented. Of particular interest to the present work are methods that rely on line features [21, 43].

In the spirit of original works on monocular visual odometry [29], the present work addresses the relatively unexplored topic of geometric incidence relationships for local relative motion calculation with an event camera. Ge-

ometric solutions remain important to date owing to their ability to find solutions with optimality guarantees, and potentially certificates, under known assumptions, unlike optimization-, filter- or learning-based solvers, which often lack these guarantees. Given that events are primarily triggered by high-gradient appearance boundaries, the dominant feature for event-based motion estimation is given by lines. Weng *et al.* [41] and Hartley *et al.* [15] have proposed closed-form solutions for frame-based cameras. Trifocal tensor geometry inspired the first closed-form solution for event cameras [30]. An important characteristic of this solver is that it relies on a local constant velocity motion model and makes use of the first-order dynamics of the camera. This is important as it permits the inclusion of the time-stamped, asynchronous measurements produced by an event camera. Nonetheless, the method by Peng *et al.* [30] is not general as it still depends on approximate event-based line-feature extractors [2, 36, 37] rather than events, only.

The most related works to ours are the works by Ieng *et al.* [18], Seok and Lim [33], and Gao *et al.* [12], who propose model-based fitting of the manifold locations of the events generated by the observation of a line under motion. In particular, Gao *et al.* [12] introduces an exact incidence relation that all such events must obey under constant linear velocity. It depends on observable camera motion and 3D line parameters, and thus serves as a basis for joint manifold fitting and motion estimation. Based on this foundation, the present work develops the first linear  $N$ -point solution to this problem, which not only unlocks unprecedented efficiency, but also a simplified understanding of degenerate geometric conditions.

## 3. Methodology

Assume a calibrated event camera undergoing an arbitrary six DoF motion, while observing a set of  $M$  lines  $\{\mathbf{L}_i\}_{i=1}^M$ . Each line generates a set of  $N_i$  events  $\mathcal{E}_i = \{e_{ij}\}_{j=1}^{N_i}$  where each event  $e_{ij} = (\mathbf{x}_{ij}, t_{ij}, p_{ij})$  is characterized by its pixel coordinate  $\mathbf{x}_{ij}$  in the image plane, timestamp  $t_{ij}$  (with  $\mu$ s resolution), and polarity  $p_{ij}$ .

For a small time window  $[t_s - \Delta t, t_s + \Delta t]$ , centered at reference time  $t_s$ , such that the camera motion can be approximated by linear dynamics, the events generated by a single line circumscribe a manifold termed *eventail* [12]. In Sec. 3.1, we describe the geometric incidence relation [12] which needs to be satisfied by events on this manifold and depends on the observable components of the velocity  $\mathbf{v}$  and the line  $\mathbf{L}_i$  for which we introduce a minimal parametrization. Then, in Sec. 3.2 we will introduce a solver that recovers the line parameters, and observable linear velocity parameters from a set of  $N_i \geq 5$  events that lie on the manifold. Finally, in Sec. 3.6, we will explore how to recover the full linear velocity from a set of  $M \geq 2$  partial velocity observations.

### 3.1. Incidence Relationship

We reiterate here briefly the incidence relation introduced in [12] and use Fig. 1 for illustration purposes. For simplicity, we will first consider the case of one line, and thus drop the index  $i$  from the variables. We furthermore express all quantities in the camera frame centered at time  $t_s$ . The incidence relation enforces that events are triggered by points on the line, such that the line  $\mathbf{L}_j = [\mathbf{d}_j^T \mathbf{m}_j^T]^T$  (in Plücker coordinates) emanating from an individual event  $e_j$  triggered at time  $t_j$  (orange line in Fig. 1) intersects the line  $\mathbf{L} = [\mathbf{d}^T \mathbf{m}^T]^T$  (blue line in Fig. 1). The condition for intersection of two non-parallel lines is

$$\mathbf{d}^T \mathbf{m}_j + \mathbf{m}^T \mathbf{d}_j = 0. \quad (1)$$

Plücker line coordinates comprise the line direction  $\mathbf{d} \in \mathbb{R}^3$  and moment  $\mathbf{m} = \mathbf{P} \times \mathbf{d}$ , with  $\mathbf{P} \in \mathbb{R}^3$  being an arbitrary point on the line. As [12] for the event ray  $j$  we use the camera position  $\mathbf{C}[t_j] = \mathbf{P}_j$  at time  $t_j$ , and the event bearing vector  $\mathbf{f}'_j = \mathbf{d}_j$  rotated into the reference frame at time  $t_s$ . Under first order dynamics, these are  $\mathbf{C}[t_j] = t'_j \mathbf{v}$  and  $\mathbf{f}'_j = \mathbf{R}[t_j] \mathbf{f}_j$  with  $t'_j = t_j - t_s$ . As [12], we assume  $\mathbf{R}[t_j]$  given and computed by integrating angular rate measurements  $\boldsymbol{\omega}$  from an available inertial measurement unit (IMU) with  $\mathbf{R}[t_j] = \exp([\boldsymbol{\omega}]_{\times} t'_j)$ . Here  $\exp(\cdot)$  denotes the matrix exponential and  $[\boldsymbol{\omega}]_{\times}$  denotes the skew-symmetric matrix associated with the angular rate  $\boldsymbol{\omega} \in \mathbb{R}^3$ . The Plücker coordinates are thus  $\mathbf{L}_j = [\mathbf{f}'_j{}^T \quad (\mathbf{C}[t_j] \times \mathbf{f}'_j)^T]^T$  and the incidence relation becomes

$$\mathbf{d}^T (\mathbf{C}[t_j] \times \mathbf{f}'_j) + \mathbf{m}^T \mathbf{f}'_j = 0. \quad (2)$$

The above equation relates measurements  $\mathbf{f}_j$  and  $t_j$  to our unknowns  $\mathbf{d}$ ,  $\mathbf{m}$  and  $\mathbf{v}$ , but is still not in a minimal form. This is because, firstly Plücker coordinates are not minimal, and indeed true minimal line representations only have four DoF. Second, in a monocular setup, there is scale ambiguity, and this dictates that the absolute scale of  $\mathbf{v}$  and  $\mathbf{L}$  is unobservable. Finally, the velocity component along the line direction is unobservable due to the aperture problem. To encapsulate these constraints, in the next part, we will introduce a line representation based on the angle-axis representation of rotation matrices. This rotation matrix simultaneously spans a coordinate frame in which we will express our camera velocity, and the aperture problem can be enforced succinctly.

### 3.2. Transition into a Minimal Form

The transition into minimal form consists of two steps. We first derive the angle-axis-based line representation by successively eliminating internal constraints in the Plücker line coordinate representation, and then proceed with finding the minimal camera velocity representation. We start by observing that scaling both  $\mathbf{d}$  and  $\mathbf{m}$  yields the same line, and

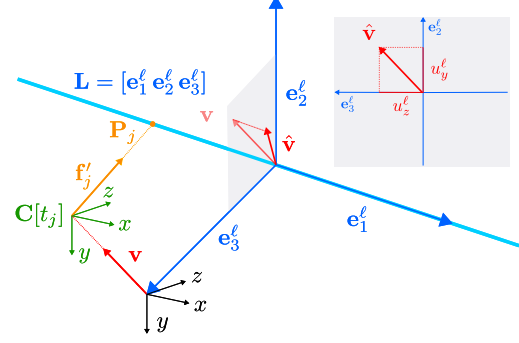


Figure 1. Incidence relationship between the line  $\mathbf{L}$ , and the bearing vector  $\mathbf{f}'_j$  of event. We parameterize the line with the rotation matrix  $\mathbf{R}_\ell = [\mathbf{e}_1^\ell \mathbf{e}_2^\ell \mathbf{e}_3^\ell]$ . Since scale is unobservable, we select the point  $-\mathbf{e}_3^\ell$  at unit depth to lie on the line, and  $\mathbf{e}_1^\ell$  to indicate its direction. Due to the aperture problem, we can only observe the projected camera velocity  $\hat{\mathbf{v}}$  with components  $u_y^\ell$  and  $u_z^\ell$  in the  $\mathbf{e}_2^\ell$  and  $\mathbf{e}_3^\ell$  direction respectively.

thus we may choose to fix the scale of  $\mathbf{d}$  to be unity. Next, we observe that  $\mathbf{m} = \mathbf{P} \times \mathbf{d}$  is by definition perpendicular to  $\mathbf{d}$ . Moreover, we see that any  $\mathbf{P}$  on the same line results in the same moment, and thus we may choose to select it closest to the origin, such that it is perpendicular to  $\mathbf{d}$ . Since the scale is unobservable, we may furthermore fix the distance from  $\mathbf{P}$  to the origin to be unity. Summarizing these observations, we conclude that we may select  $\mathbf{P}$  and  $\mathbf{d}$  to be perpendicular unit vectors, and in particular we will select  $\mathbf{P} = -\mathbf{e}_3^\ell$  and  $\mathbf{d} = \mathbf{e}_1^\ell$ , such that the resulting moment is  $\mathbf{m} = -\mathbf{e}_2^\ell$ . We visualize the three unit vectors in Fig. 1, and observe that they span a *line-dependent* coordinate frame via the rotation matrix  $\mathbf{R}_\ell = [\mathbf{e}_1^\ell \mathbf{e}_2^\ell \mathbf{e}_3^\ell]$ . Rotation matrices belong to  $SO(3)$ , and thus it becomes apparent that this line representation can be further compressed via the matrix logarithm  $\boldsymbol{\theta}_\ell = (\log(\mathbf{R}_\ell))^\vee = [\theta_x^\ell \theta_y^\ell \theta_z^\ell]$ , to yield only three DoF, which is a minimal line representation in the absence of scale. Here  $(\cdot)^\vee$  maps the skew-symmetric matrix in the argument to the associated vector.

Note that compared to the two-point-two-plane parametrization [14] in [12], this representation is (i) minimal, relying on three instead of four DoF, and (ii) more flexible, since it does not need a reparametrization step for lines that are almost parallel to the  $yz$ -plane.

We now address the camera velocity parametrization, which we express in the line-dependent coordinate frame.

$$\mathbf{v} = \mathbf{R}_\ell \mathbf{u}_\ell, \quad (3)$$

introduces the camera velocity  $\mathbf{u}_\ell = [u_x^\ell \ u_y^\ell \ u_z^\ell]$  expressed in the line coordinate frame. Using these new parametrizations, the incidence relation Eq. 2 becomes

$$t'_j \mathbf{e}_1^{\ell T} ((\mathbf{R}_\ell \mathbf{u}_\ell) \times \mathbf{f}'_j) - \mathbf{f}'_j{}^T \mathbf{e}_2^\ell = 0. \quad (4)$$

Cycling through the triple product in the first summand, *i.e.*  $\mathbf{a}^\top(\mathbf{b} \times \mathbf{c}) = \mathbf{c}^\top(\mathbf{a} \times \mathbf{b})$ , we arrive at

$$t'_j \mathbf{f}'_j{}^\top (\mathbf{e}_1^\ell \times (\mathbf{R}_\ell \mathbf{u}_\ell)) - \mathbf{f}'_j{}^\top \mathbf{e}_2^\ell = 0, \quad (5)$$

which can be expanded and further simplified to

$$t'_j \mathbf{f}'_j{}^\top (u_z^\ell \mathbf{e}_2^\ell - u_y^\ell \mathbf{e}_3^\ell) + \mathbf{f}'_j{}^\top \mathbf{e}_2^\ell = 0. \quad (6)$$

Note that due to the cross product with  $\mathbf{e}_1^\ell$ , the camera velocity  $u_x^\ell$  becomes unobservable within this incidence relation, *i.e.* changing it does not affect the residual. This confirms our intuition that the aperture problem should make velocities along the line direction unobservable. As a result, we focus on only solving for  $u_y^\ell$  and  $u_z^\ell$ . Furthermore, observe that these equations are in terms of  $\mathbf{e}_2^\ell$  and  $\mathbf{e}_3^\ell$  instead of the minimal parameters  $\theta_\ell$ . In what follows we design the solver around recovering  $\mathbf{e}_2^\ell$  and  $\mathbf{e}_3^\ell$  since it yields a simpler algorithm, yet it should be remembered that the minimal representation can always be recovered using the matrix logarithm of  $\mathbf{R}_\ell$ . We now discuss how to recover the unknowns from a set of incidence relations from multiple events, which is summarized in Alg. 1.

### 3.3. Five-point Minimal Solver

The incidence relationship in Eq. 6 has five unknowns, three from  $\theta_\ell = [\theta_x^\ell \theta_y^\ell \theta_z^\ell]$  and two from  $\mathbf{u}_\ell = [0 \ u_y^\ell \ u_z^\ell]$ , and thus can be solved by stacking a minimum of five such constraints. Since each such constraint originates from a single event, this means that five events are the minimum number to solve this system. This stack of equations is

$$\begin{aligned} t'_1 \mathbf{f}'_1{}^\top (u_z^\ell \mathbf{e}_2^\ell - u_y^\ell \mathbf{e}_3^\ell) + \mathbf{f}'_1{}^\top \mathbf{e}_2^\ell &= 0 \\ t'_2 \mathbf{f}'_2{}^\top (u_z^\ell \mathbf{e}_2^\ell - u_y^\ell \mathbf{e}_3^\ell) + \mathbf{f}'_2{}^\top \mathbf{e}_2^\ell &= 0 \\ &\vdots \\ t'_5 \mathbf{f}'_5{}^\top (u_z^\ell \mathbf{e}_2^\ell - u_y^\ell \mathbf{e}_3^\ell) + \mathbf{f}'_5{}^\top \mathbf{e}_2^\ell &= 0. \end{aligned}$$

This system of equations is linear in the unknowns and can be rewritten as a single matrix equation

$$\underbrace{\begin{bmatrix} t'_1 \mathbf{f}'_1{}^\top & \mathbf{f}'_1{}^\top \\ \vdots & \vdots \\ t'_5 \mathbf{f}'_5{}^\top & \mathbf{f}'_5{}^\top \end{bmatrix}}_{\doteq \mathbf{A} \in \mathbb{R}^{5 \times 6}} \underbrace{\begin{bmatrix} u_z^\ell \mathbf{e}_2^\ell - u_y^\ell \mathbf{e}_3^\ell \\ \mathbf{e}_2^\ell \end{bmatrix}}_{\doteq \mathbf{x} \in \mathbb{R}^{6 \times 1}} = \mathbf{0}. \quad (7)$$

Note that this formulation successfully groups the terms from the events in  $\mathbf{A}$  and unknowns in  $\mathbf{x}$ . We can thus solve for  $\mathbf{x}$ , and then reconstruct the unknowns from the found solution. Solving Eq. 7 can be done with a singular value decomposition of  $\mathbf{A}$  and then selecting the last column of  $\mathbf{V}$  corresponding to the smallest singular value of  $\mathbf{A}$ . Let us denote this solution with  $\hat{\mathbf{x}}$ . Note that  $\hat{\mathbf{x}}$  is normalized,

and, due to the homogeneous nature of Eq. 7, only known up to parity, *i.e.* both  $\pm \hat{\mathbf{x}}$  are solutions. Note also that this procedure is not limited to using only five events, but can be applied to  $N \geq 5$ , however in this case the solution will no longer be exact, and instead approximate but with a globally minimal squared residual equal to the smallest singular value of  $\mathbf{A}$ . The ability to process more than five events sets this method apart from the solver in [12] which uses a fixed elimination template tailored to only five events. Next, we discuss how to recover the unknowns from a solution  $\hat{\mathbf{x}}$ .

### 3.4. Recovering the Unknowns from $\hat{\mathbf{x}}$

For simplicity, we will only treat the case with  $+\hat{\mathbf{x}}$ , but will state that the ambiguity in the parity of the solution to Eq. 7 gives rise to the solution pairs  $S_0, S_1$ , and  $S_2, S_3$ . Remembering the definition of  $\mathbf{x}$  we write that

$$\hat{\mathbf{x}} = \begin{bmatrix} \lambda \hat{\mathbf{x}}_{1:3} \\ \lambda \hat{\mathbf{x}}_{4:6} \end{bmatrix} = \begin{bmatrix} u_z^\ell \mathbf{e}_2^\ell - u_y^\ell \mathbf{e}_3^\ell \\ \mathbf{e}_2^\ell \end{bmatrix}. \quad (8)$$

Here  $\lambda$  is an unknown scaling factor. However, since the last three entries of  $\hat{\mathbf{x}}$  correspond to the unit vector  $\mathbf{e}_2^\ell$ , we can simply normalize  $\hat{\mathbf{x}}$  by the length of  $\hat{\mathbf{x}}_{4:6}$ . In what follows we assume that this normalization is done beforehand, and thus ignore this scaling factor by setting  $\lambda = 1$ . Straightforward manipulation yields

$$\mathbf{e}_2^\ell = \hat{\mathbf{x}}_{4:6} \quad (9a)$$

$$u_z^\ell = \hat{\mathbf{x}}_{1:3}^\top \hat{\mathbf{x}}_{4:6} \quad (9b)$$

$$u_y^\ell \mathbf{e}_1^\ell = \hat{\mathbf{x}}_{1:3} \times \hat{\mathbf{x}}_{4:6}. \quad (9c)$$

From the last equation we can recover  $u_y^\ell$  and  $\mathbf{e}_1^\ell$  by taking the norm, and normalized vector

$$u_y^\ell = \|\hat{\mathbf{x}}_{1:3} \times \hat{\mathbf{x}}_{4:6}\| \text{ and } \mathbf{e}_1^\ell = \frac{\hat{\mathbf{x}}_{1:3} \times \hat{\mathbf{x}}_{4:6}}{\|\hat{\mathbf{x}}_{1:3} \times \hat{\mathbf{x}}_{4:6}\|}. \quad (10)$$

Note that this decomposition is not unique, as we may simultaneously flip the signs of  $\mathbf{e}_1^\ell$  and  $u_y^\ell$ , resulting in the same product. Finally, we can recover  $\mathbf{e}_3^\ell = \mathbf{e}_1^\ell \times \mathbf{e}_2^\ell$ . While  $\hat{\mathbf{x}}$  is the globally optimal solution to the incidence relation constraints in Eq. 7, it is not immediately clear that the recovered  $\mathbf{R}_\ell, \mathbf{u}_\ell$  are also globally optimal with respect to this constraint. As proved in the supplementary material it turns out that  $\mathbf{R}_\ell, \mathbf{u}_\ell$  are globally optimal. Next, we will comment on verifying the correctness of the solution.

### 3.5. Degenerate Solutions and Solution Multiplicity

First, we state a theorem that the only degenerate cases arise when the matrix  $\mathbf{A}$  in Eq. 7 is rank-deficient. This means that, if  $\text{rank}(\mathbf{A}) \geq 5$ , the previously discussed decomposition always succeeds and yields four solutions.



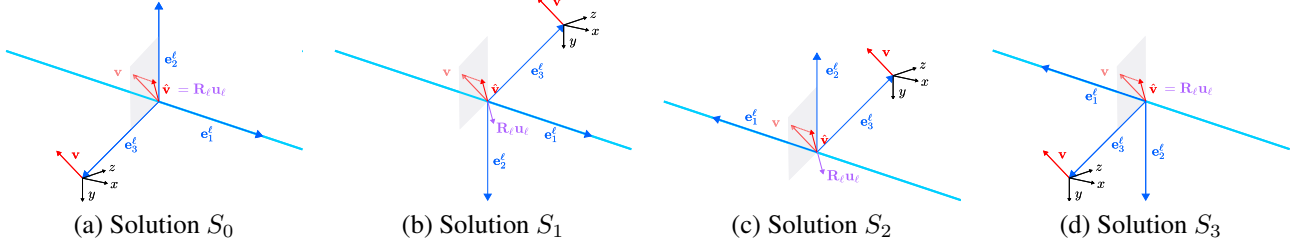


Figure 2. Multiplicity of solutions to the incidence relation in Eq. 6. While  $S_0$  and  $S_3$  have the line in front of the camera,  $S_1$  and  $S_2$  have the line behind the camera. The solution pairs  $S_0, S_3$  and  $S_1, S_2$  differ in the orientation of  $\mathbf{e}_1^\ell$ , which comes from the ambiguity of defining the line direction. In solutions with the line behind the camera, the measured projected camera velocity,  $\mathbf{R}_\ell \mathbf{u}_\ell$  returned by the solver is the negative of the true projected velocity  $\hat{\mathbf{v}}$ . However, these solutions can be discarded by enforcing the condition in Eq. 14.

**Theorem 1:** *If  $\text{rank}(\mathbf{A}) \geq 5$ , with  $\mathbf{A}$  defined in Eq. 7, the decomposition of  $\hat{\mathbf{x}}$  into  $\mathbf{u}_\ell$  and  $\mathbf{R}_\ell$  always succeeds and yields four distinct solutions. If  $\text{rank}(\mathbf{A}) < 5$  the solver returns infinitely many solutions.*

Note that this theorem also handles cases in which the line passes through the origin. Both the proof of the above theorem and the handling of this case are described in the supplementary material. Rank deficiency of  $\mathbf{A}$  occurs when events share the same timestamp  $t_j$  or motion-corrected bearing vector  $\mathbf{f}'_j$ . To identify this scenario, checking the matrix rank before solving for  $\hat{\mathbf{x}}$  can be done. After SVD, the second smallest singular value should be checked for being large, since small values indicate near-rank deficiency.

Next, we discuss the multiplicity of solutions. As previously stated, the designed solver returns four distinct solutions if the rank of  $\mathbf{A}$  is at least 5. Here we enumerate these solutions (visualized in Fig. 2), stated as a theorem:

**Theorem 2:** *Given a solution  $S_0 = \{\mathbf{e}_1^\ell, \mathbf{e}_2^\ell, \mathbf{e}_3^\ell, u_y^\ell, u_z^\ell\}$  to the incidence relation in Eq. 6, then*

$$\begin{aligned} S_1 &= \{\mathbf{e}_1^\ell, -\mathbf{e}_2^\ell, -\mathbf{e}_3^\ell, u_y^\ell, u_z^\ell\}, \\ S_2 &= \{-\mathbf{e}_1^\ell, \mathbf{e}_2^\ell, -\mathbf{e}_3^\ell, -u_y^\ell, u_z^\ell\}, \\ S_3 &= \{-\mathbf{e}_1^\ell, -\mathbf{e}_2^\ell, \mathbf{e}_3^\ell, -u_y^\ell, u_z^\ell\} \end{aligned}$$

are also solutions. For solutions  $S_1$  and  $S_2$  the closest point  $-\mathbf{e}_3^\ell$  on the line is behind the camera, while for solutions  $S_2$  and  $S_3$  the line direction  $\mathbf{e}_1^\ell$  is flipped, which represents an ambiguity in the definition of direction of  $\mathbf{L}$ .

We state the proof in the supplementary material. Note that two configurations correspond to flipping across the  $xy$ -plane. We eliminate these solutions by enforcing that the intersection point  $\mathbf{P}_j$  (see Fig. 1) between the line and event ray is in front of the camera. In the supplementary material, we use this geometric interpretation to characterize manifolds spanned by events in more detail.

---

**Algorithm 1** Linear Solver for Line and Partial Motion Parameters

---

**Input:** A set of events  $\mathcal{E}$  with rotated bearing vectors.

**Output:** Line parameters  $\theta_\ell$  and projected velocities  $\mathbf{u}_\ell$ .

- Form matrix  $\mathbf{A}$  from the set of events  $\mathcal{E}$  by Eq. 7 and make sure that  $\text{rank}(\mathbf{A}) \geq 5$ .
  - Apply SVD on  $\mathbf{A}$  and select the last column of  $\mathbf{V}$ , denoted with  $\hat{\mathbf{x}}$ . Both  $\pm \hat{\mathbf{x}}$  can be selected.
  - Normalize  $\hat{\mathbf{x}}$  by  $\hat{\mathbf{x}}_{4:6}$ , the last three elements.
  - Recover  $\mathbf{e}_2^\ell, u_z^\ell$  from Eq. 9.
  - Recover  $\mathbf{e}_1^\ell, u_y^\ell$  from Eq. 10. Both  $\{\mathbf{e}_1^\ell, u_y^\ell\}$  and  $\{-\mathbf{e}_1^\ell, -u_y^\ell\}$  can be selected.
  - Compile  $\mathbf{u}_\ell = [0 \ u_y^\ell \ u_z^\ell]$ .
  - Compute  $\mathbf{e}_3^\ell = \mathbf{e}_1^\ell \times \mathbf{e}_2^\ell$ .
  - Construct the rotation  $\mathbf{R}_\ell = [\mathbf{e}_1^\ell \ \mathbf{e}_2^\ell \ \mathbf{e}_3^\ell]$ .
  - Recover minimal line parameters  $\theta_\ell = (\log(\mathbf{R}_\ell))^\vee$ .
- 

### 3.6. Velocity Averaging from Multiple Manifolds

As previously discussed, the linear solver can only recover partial velocities perpendicular to the line generating the events in a single cluster. We now describe how to recover the full velocity from a set of partial observations from multiple lines  $\mathbf{L}_i$ . Each partial observation of the velocity  $\mathbf{v}$  is the projection  $\hat{\mathbf{v}}_i = \mathbf{H}_i \mathbf{v}$  into the  $\mathbf{e}_{2i}^\ell \mathbf{e}_{3i}^\ell$ -plane where  $\mathbf{H}_i = \mathbf{I} - \mathbf{e}_{1i}^\ell \mathbf{e}_{1i}^{\ell T}$ , which is visualized in Fig. 3(i). The observation is given by the two projections  $u_{y_i}^\ell$  and  $u_{z_i}^\ell$ , scaling the second and third basis vectors of  $\mathbf{R}_{\ell i}$ , respectively. Thus, the correct velocity estimate must satisfy

$$\hat{\mathbf{v}}_i = \mathbf{H}_i \mathbf{v} = \mathbf{e}_{2i}^\ell u_{y_i}^\ell + \mathbf{e}_{3i}^\ell u_{z_i}^\ell = \mathbf{R}_{\ell i} \mathbf{u}_{\ell i}. \quad (11)$$

Unlike the velocity averaging scheme in [12] we adopt the following geometrically motivated, but equivalent formalism to solve multiple such equations: Each such constraint (one for each line) can be converted into a homogeneous linear constraint following the steps in Fig. 3(i). We see that the  $90^\circ$  rotated velocity  $\mathbf{R}_{\ell i} \mathbf{R}_{\frac{\pi}{2}} \mathbf{u}_{\ell i}$  must be perpendicular to the projected camera velocity  $\hat{\mathbf{v}}_i$ . Forming the dot

product with this vector yields:

$$\hat{\mathbf{v}}_i^T \mathbf{R}_{\ell_i} \mathbf{R}_{\frac{\pi}{2}} \mathbf{u}_{\ell_i} = \mathbf{v}^T (u_{y_i}^{\ell} \mathbf{e}_{3i}^{\ell} - u_{z_i}^{\ell} \mathbf{e}_{2i}^{\ell}) = 0 \quad (12)$$

with  $\mathbf{R}_{\frac{\pi}{2}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$ .

Note that the above constraint remains, even when inserting the alternative solutions  $S_0, S_3$  or  $S_1, S_2$ , since these only lead to a change in the sign. Stacking  $M$  such equations, one for each line, can be summarized as:

$$\underbrace{\begin{bmatrix} u_{y_1}^{\ell} \mathbf{e}_{31}^{\ell T} - u_{z_1}^{\ell} \mathbf{e}_{21}^{\ell T} \\ \vdots \\ u_{y_M}^{\ell} \mathbf{e}_{3M}^{\ell T} - u_{z_M}^{\ell} \mathbf{e}_{2M}^{\ell T} \end{bmatrix}}_{\doteq \mathbf{D} \in \mathbb{R}^{M \times 3}} \mathbf{v} = \mathbf{0}. \quad (13)$$

We solve this equation again with SVD, by selecting the column of  $\mathbf{V}$  corresponding with the minimal singular value. From this system, we also conclude that we need a minimum of two lines to recover  $\mathbf{v}$ . Again, SVD only recovers  $\mathbf{v}$  up to an unknown parity, *i.e.* both  $\pm \mathbf{v}$  satisfy the equation. To disambiguate the solution, we enforce that the projected velocity must point in the same direction as  $\mathbf{R}_{\ell_i} \mathbf{u}_{\ell_i}$  for each line. This can be written concisely as

$$\hat{\mathbf{v}}_i^T \mathbf{R}_{\ell_i} \mathbf{u}_{\ell_i} = \mathbf{v}^T (u_{y_i}^{\ell} \mathbf{e}_{2i}^{\ell} + u_{z_i}^{\ell} \mathbf{e}_{3i}^{\ell}) > 0, \quad (14)$$

If at least one line does not satisfy this condition, the sign of  $\mathbf{v}$  should be flipped. Compared to [12], which uses a more expensive Shur-Complement step to eliminate unknown scale factors, our algorithm only requires a single step of SVD, but yields the same results. This is because our algorithm is actually equivalent to that of [12], as will be shown in the supplementary material.

Note that the above line averaging scheme may run into issues, if two lines are parallel. Making the common  $\mathbf{e}_1^{\ell}$  component of  $\mathbf{v}$  unobservable. However, this case actually induces a rank deficiency on  $\mathbf{D}$ , as proved in the supplementary material, and can thus easily be discarded.

## 4. Implementation

We integrate the aforementioned linear solver into a RANSAC framework for parameter determination of each manifold, followed by fitting over all inliers. Diverging from the approach of [12], our implementation adopts the GC-RANSAC framework [1] for robust geometric model estimation. GC-RANSAC enhances the original RANSAC by introducing a few versatile functionalities tailored for early termination, thereby expediting the overall process. Essentially, GC-RANSAC first iteratively selects a minimal, spatially coherent subset of events ( $N = 5$ ) from the incoming event stream, applies the proposed linear solver

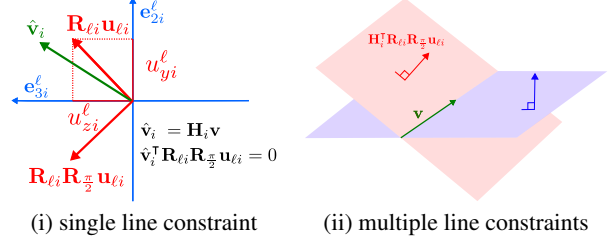


Figure 3. (i) The line constraint in Eq. 11 dictates that the 90-degree rotated measured velocity  $\mathbf{R}_{\ell_i} \mathbf{R}_{\frac{\pi}{2}} \mathbf{u}_{\ell_i}$  should be perpendicular to  $\hat{\mathbf{v}}_i$ , or  $\mathbf{v}^T \mathbf{H}_i^T \mathbf{R}_{\ell_i} \mathbf{R}_{\frac{\pi}{2}} \mathbf{u}_{\ell_i} = 0$ . (ii) Each such constraint spans a two-dimensional subspace, in which  $\mathbf{v}$  must reside. With a minimum of two such subspaces, the velocity can be found.

described in Alg. 1, and evaluates the quality of the resulting hypothesis. Each occurrence of a *so-far-the-best* hypothesis will trigger a local refinement within a subset of its inliers. This procedure is repeated  $M$  times to separately identify  $M$  manifolds. We will now delve into the critical aspects influencing this process.

**Spatially Coherent Sampler:** The manifold’s continuous structure allows for the examination of the data’s spatial coherency. We utilize NAPSAC [35] to sample from the incoming event stream. This approach starts by randomly selecting one event in the space-time volume followed by identifying four additional events within a hypersphere centered on the initial event, based on a predetermined radius  $r$ . In practice, these four points, located within the hyper-sphere, are likely to be inliers<sup>1</sup>.

**Angular Reprojection Residual:** We employ the angular reprojection residual [22] for inliers selection. The objective is to minimally correct the two lines, the bearing vector emanating from an event and the 3D line, so they could converge at a single point (*i.e.*  $\mathbf{P}_j$  in Fig. 1). Unlike the typical image reprojection residual, this measurement is invariant to both rotation and scale.

**Local Refinement:** RANSAC often results in numerous ineffective iterations. Therefore, when a promising hypothesis emerges, it is recommended to perform local refinement using the inlier sets, which can enhance the inlier ratio and decrease the total number of required iterations. In our work, we introduce two distinct methods for local refinement. The first method leverages the over-determined nature of our proposed linear solver, while the second employs non-linear optimization with Levenberg–Marquardt over an algebraic error (*i.e.* Eq. 6). As suggested in [1], we randomly select a subset from the inlier set ( $N = 10$ ) and repeat this procedure  $Q$  times.

<sup>1</sup>Note that this method does not conflict with the findings about ensuring spatial distribution among samples [12]. However, we adhere to a general rule of maintaining this spatial distribution within our defined hyper-sphere.

Table 1. General comparison between the two solvers.

Method	Runtime ( $\mu$ s)		Error Rate (%)	
	min.	avg.	$\geq 0.1^\circ$	$\geq 1.0^\circ$
Gröbner [12]	1893	2046	1.00	0.28
<b>Linear (ours)</b>	<b>3.00</b>	<b>3.25</b>	<b>0.00</b>	<b>0.00</b>

## 5. Experiments

We perform evaluations both on synthetic and real data. We first confirm the runtime improvement and numerical stability of our linear solver. Next, we discuss the impact of the number of used events or lines, over different noise setups. We conclude with experiments on a few public real-world sequences, demonstrating the advantage over existing methods. We quantify the accuracy of our results with the same criterion as [12], the direction error  $\phi$  between the estimated and the ground truth velocities, given that the scale is not observable. All experiments are conducted on a 32GB RAM desktop with an Intel Core i9-10900F Processor.

### 5.1. Simulation

We first evaluate the performance of the proposed linear solver under different setups over synthetic data. Readings from individual manifolds are generated as follows. We first sample randomly directed linear and angular velocities of 0.5 m/s and  $15^\circ$ /s magnitude, respectively. The time window length is set to 0.5 s, and the virtual event camera has a resolution of  $640 \times 480$  and a focal length of 320 pixels. Next, we sample a random line in 3D with a finite length and sample random events according to the spatiotemporal strategy in [12]. We study three types of noise with different magnitudes: pixel noise (0.5 px), timestamp jitter (0.5 ms), and gyroscope noise ( $5.0^\circ$ /s), which is assumed to be given by an IMU. The magnitude of the pixel noise and the noise on camera angular velocities are consistent within the same noise level but vary in direction. Timestamp noise follows a zero-mean Gaussian. For a more detailed sensitivity study for different noise sources and levels see the supplementary material. We generate one million random line, velocity, and event configurations, and report the mean and median angle error, as well as the minimum and mean runtime in milliseconds of our method and the one in [12].

**Runtime Analysis:** In each run we record the runtime of both solvers. As reflected in Tab. 1, our solver runs over 600 times faster than the Gröbner basis solver in [12].

**Numerical Stability:** We further analyze the numerical stability of both methods under the noise-free setup. We report the likelihood of the solver to converge to within a low ( $1.0^\circ$ , or  $0.1^\circ$ ) error. Our solver consistently reaches a zero error, unlike the polynomial solver, which, due to numerical instabilities of the elimination template, fails to

Table 2. Noise resilience of our linear solver and the solver in [12]. Left corresponds to the mean, right to the median error in degrees.

Method	num. events	Pixel Noise (0.5 px)	Time. Jitter (0.5 ms)	Gyro. Noise ( $5.0^\circ$ /s)
Gröbner [12]	5	7.80/1.67	3.61/0.83	7.48/3.09
<b>Linear (ours)</b>	5	5.53/1.24	2.87/0.73	6.53/2.47
<b>Linear (ours)</b>	10	<b>0.46/0.15</b>	<b>0.17/0.12</b>	<b>1.50/1.17</b>

converge within a  $0.1^\circ$  error range 1 % of times.

**Analysis of the Number of Used Events:** In each simulation, we sample 1,000 signal events and introduce a type of representative noise to the measurements. From these, we use the first  $5 \leq K \leq 1,000$  events as input for our linear solver for a fair comparison and document the resulting error. We repeat this simulation a million times, varying  $K$ , and report the results in Fig. 4. We observe a clear trend that as the number of used events increases, the error decreases markedly, except when noise is introduced to the camera’s angular velocity. This exception occurs because the solver cannot average out the noise on the angular velocity, regardless of the number of events processed. The other two errors approach near zero when 1,000 events are used. A full analysis of the solver’s performance under each noise type can be found in the supplementary material.

**Analysis of the Number of Used Lines:** Finally, we validate our velocity averaging scheme. We extend our simulation to multiple manifolds. For each run, we sampled ten lines and, within each line, we selected ten signal events with known line associations as input to our solver. The first  $2 \leq K \leq 10$  solutions (line and partial motion parameters) from each manifold were taken into the linear averaging scheme, and the error was recorded. This simulation was executed 10,000 times. Fig. 4 shows that as the number of used lines increases, the error drops significantly. A comprehensive analysis of the solver’s performance against various noise types is available in the supplementary material. Additionally, we show quantitative results in Tab. 2. Both the Gröbner Solver and our linear solver use five lines with either five or ten events each. Our approach demonstrates a lower error with noisy measurements, and this margin grows further in overdetermined systems (*i.e.*  $N = 10$ ).

### 5.2. Real-world Experiment

Similarly to [12], we validate our method on the same data sequence from VECtor Benchmark [11]. Unlike the previous work, we first segmented the event data into non-overlapping intervals of 0.1 s each and reduced the overall size of the data to approximately 5,000 events per interval for efficiency. Motion-corrected bearing vectors are then calculated by fusing gyroscope readings from the attached IMU. Next, to construct the chosen sampler used in GC-

Table 3. Real-world experiment results. Left corresponds to the mean, right to the median error in degrees.

Seq. Name	CELC+opt [30]	Gröbner [12]	Linear only	Linear w/ non-min. solver	Linear w/ non-linear opt.
<i>mountain-normal</i>	27.7/29.3	33.5/33.6	25.2/21.4	17.0/17.2	<b>16.5/14.6</b>
<i>desk-normal</i>	26.6/26.6	26.4/26.7	22.7/23.4	<b>19.8/19.2</b>	22.1/20.7
<i>sofa-normal</i>	24.0/26.1	31.5/29.5	21.9/17.6	20.6/16.1	<b>19.9/15.0</b>

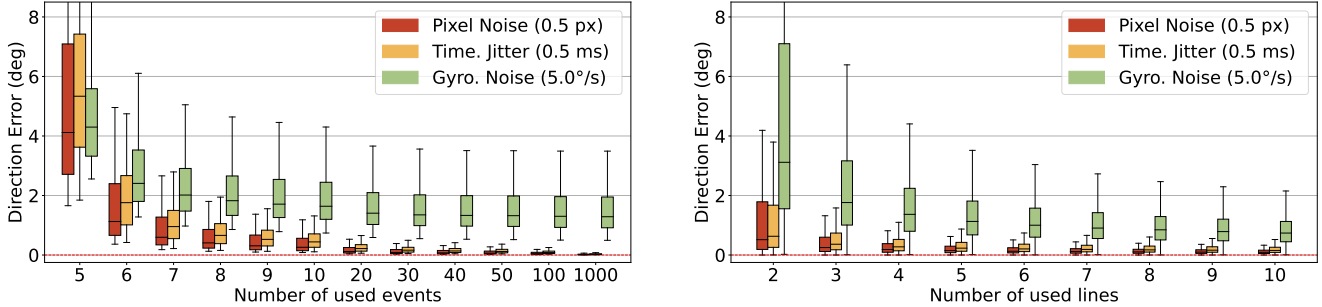


Figure 4. Condensed analysis of the number of used events (left) and used lines (right) over three types of representative noise.

RANSAC, we multiply the timestamp by a scale factor of 1,000 and established a radius of 50 to compose the spatially coherent graph in the space-time volume. We apply an angular reprojection threshold of  $0.2^\circ$  for inlier selection, consistent across both the primary iterations and the local refinement stages. The number of iterations for each manifold fitting is capped at 100 and evaluated manifolds is capped at 10. In Tab. 3, we summarized the performance, including both mean and median errors, across two baseline approaches and three variants of our proposed method. Importantly, as [12] reports, CELC+opt [30] is limited to certain sub-sequences, where spatial-temporal plane clustering is feasible, while, [12] does not suffer from this limitation. We test three configurations of our method: *linear only*, *linear with non-minimal refinement*, and *linear with non-linear optimization*. Each configuration uses GC-RANSAC with the spatially coherent sampler. The latter two configurations perform different operations when a new best hypothesis is found by the sampler: “Linear w/ non-min. solver” samples 10 events from the found inliers and feeds them to the linear solver, resulting in a refined solution. “Linear w/ non-linear opt.” runs on-manifold Levenberg-Marquardt optimization steps (with line parameters  $\mathbf{R}_\ell \in SO(3)$ ) over the 10 selected events and minimizes the algebraic error in Eq. 6. We use a subset of 10 inliers to enhance efficiency and reduce errors, as seen in simulated experiments.

**Results:** In each configuration, our method achieves a lower error than the two baseline approaches. Additionally, local refinement enhances accuracy significantly. Without refinement it already has a 10% lower mean error than [30], and 24% lower mean error than [12]. Introducing non-minimal solver refinement reduces the mean error by an

other 18% over the “Linear only” baseline and non-linear optimization reduces it by 16%.

## 6. Conclusion and Future Work

This work introduced a novel, efficient, and linear N-point solver for line-based relative motion estimation of an event camera. Compared to existing works that rely on polynomial system solvers, our method is more numerically stable, over 600 times faster, and allows the identification of degenerate cases explicitly. Moreover, we introduce a novel velocity averaging scheme that is simpler and faster than previous approaches. When combined with GC-RANSAC we show improved normalized velocity estimation compared to existing approaches, at a fraction of the runtime. Finally, the solutions found by our solver deliver new insights into event manifolds generated by lines and thus pave the way for line-based motion estimation with events. Moreover, despite focusing on event cameras in this work, our formulation is fully compatible with line detections from standard cameras. Thus the tools developed in this work can benefit both frame- and event-based computer vision. Our next steps consist of adding uncertainties to the partial velocity readings and applying the fusion strategy asynchronously over time as well as in conjunction with IMU measurements.

## Acknowledgments

This research has been supported by projects 22DZ1201900 and 22ZR1441300 funded by the Natural Science Foundation of Shanghai as well as project 62250610225 by the National Science Foundation of China (NSFC). This work was also supported by the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).



## References

- [1] Daniel Barath and Jiri Matas. Graph-cut ransac: Local optimization on spatially coherent structures. *IEEE TPAMI*, 44(9):4961–4974, 2021. [6](#)
- [2] Christian Brändli, Jonas Strubel, Susanne Keller, Davide Scaramuzza, and Tobi Delbruck. ELiSeD — an event-based line segment detector. In *EBCCSP*, pages 1–7, 2016. [2](#)
- [3] Samuel Bryner, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization. In *ICRA*, pages 325–331, 2019. [2](#)
- [4] 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 T-RO*, 32(6):1309–1332, 2016. [2](#)
- [5] Andrea Censi and Davide Scaramuzza. Low-latency event-based visual odometry. In *ICRA*, pages 703–710, 2014. [2](#)
- [6] William Oswaldo Chamorro Hernández, Juan Andrade-Cetto, and Joan Solà Ortega. High-speed event camera tracking. In *BMVC*, 2020.
- [7] Guillermo Gallego, Jon E.A. Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE TPAMI*, 40(2):2402–2412, 2016. [2](#)
- [8] 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 *CVPR*, pages 3867–3876, 2018. [2](#)
- [9] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. In *CVPR*, pages 12272–12281, 2019. [2](#)
- [10] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE TPAMI*, 44(1):154–180, 2020. [1](#)
- [11] 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 RA-L*, 7(3):8217–8224, 2022. [7](#)
- [12] Ling Gao, Hang Su, Daniel Gehrig, Marco Cannici, Davide Scaramuzza, and Laurent Kneip. A 5-point minimal solver for event camera relative motion estimation. In *ICCV*, pages 8015–8025, 2023. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [13] Mathias Gehrig, Sumit Bam Shrestha, Daniel Mouritzen, and Davide Scaramuzza. Event-based angular velocity regression with spiking networks. In *ICRA*, pages 4195–4202, 2020. [2](#)
- [14] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. [3](#)
- [15] Richard I. Hartley. Lines and points in three views and the trifocal tensor. *IJCV*, 22(2):125–140, 1997. [2](#)
- [16] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza. Event-aided direct sparse odometry. In *CVPR*, pages 5771–5780, 2022. [2](#)
- [17] Shi-Sheng Huang, Ze-Yu Ma, Tai-Jiang Mu, Hongbo Fu, and Shi-Min Hu. Lidar-monocular visual odometry using point and line features. In *ICRA*, pages 1091–1097, 2020. [1](#)
- [18] Sio-Hoi Teng, João Carneiro, and Ryad B. Benosman. Event-based 3d motion flow estimation using 4d spatio temporal subspaces properties. *Frontiers in Neuroscience*, 10:596, 2017. [2](#)
- [19] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *ECCV*, pages 349–364, 2016. [2](#)
- [20] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *IROS*, pages 16–23, 2016. [2](#)
- [21] Cedric Le Gentil, Florian Tschoop, Ignacio Alzugaray, Teresa Vidal-Calleja, Roland Siegwart, and Juan Nieto. IDOL: A framework for imu-dvs odometry using lines. In *IROS*, pages 5863–5870, 2020. [2](#)
- [22] Seong Hun Lee and Javier Civera. Closed-form optimal two-view triangulation based on angular errors. In *ICCV*, pages 2681–2689, 2019. [6](#)
- [23] Hyunjun Lim, Jinwoo Jeon, and Hyun Myung. UV-SLAM: Unconstrained line-based slam using vanishing points for structural mapping. *IEEE RA-L*, 7(2):1518–1525, 2022. [1](#)
- [24] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Globally optimal contrast maximisation for event-based motion estimation. In *CVPR*, pages 6348–6357, 2020. [2](#)
- [25] Junxin Lu, Zhijun Fang, Yongbin Gao, and Jieyu Chen. Line-based visual odometry using local gradient fitting. *Journal of Visual Communication and Image Representation*, 77:103071, 2021. [1](#)
- [26] 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 *CVPR*, pages 5419–5427, 2018. [2](#)
- [27] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *IROS*, pages 2761–2768, 2014. [2](#)
- [28] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE T-RO*, 34(6):1425–1440, 2018. [2](#)
- [29] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry. In *CVPR*, pages I–I, 2004. [2](#)
- [30] Xin Peng, Wanting Xu, Jiaqi Yang, and Laurent Kneip. Continuous event-line constraint for closed-form velocity initialization. In *BMVC*, 2021. [2](#), [8](#)
- [31] Xin Peng, Ling Gao, Yifu Wang, and Laurent Kneip. Globally-optimal contrast maximisation for event cameras. *IEEE TPAMI*, 44(7):3479–3495, 2022. [2](#)
- [32] 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 RA-L*, 2(2):593–600, 2016. [2](#)
- [33] Hochang Seok and Jongwoo Lim. Robust feature tracking in dvs event stream using bézier mapping. In *WACV*, pages 1658–1667, 2020. [2](#)
- [34] Timo Stoffregen and Lindsay Kleeman. Event cameras, contrast maximization and reward functions: An analysis. In *CVPR*, pages 12292–12300, 2019. [2](#)

- [35] Philip Hilaire Torr, Slawomir J. Nasuto, and John Mark Bishop. NAPSAC: High noise, high dimensional robust estimation - it's in the bag. In *BMVC*, page 3, 2002. 6
- [36] Alexander Vakhitov and Victor Lempitsky. Learnable line segment descriptor for visual slam. *IEEE Access*, 7:39923–39934, 2019. 2
- [37] David Reverter Valeiras, Xavier Clady, Sio-Hoi Ieng, and Ryad Benosman. Event-based line fitting and segment detection using a neuromorphic visual sensor. *IEEE TNNLS*, 30(4):1218–1230, 2019. 2
- [38] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaef, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE RA-L*, 3(2):994–1001, 2018. 2
- [39] David Weikersdorfer, Raoul Hoffmann, and Jörg Conradt. Simultaneous localization and mapping for event-based vision systems. In *ICVS*, pages 133–142, 2013. 2
- [40] David Weikersdorfer, David B. Adrian, Daniel Cremers, and Jörg Conradt. Event-based 3d slam with a depth-augmented dynamic vision sensor. In *ICRA*, pages 359–364, 2014. 2
- [41] Juyang Weng, Thomas S. Huang, and Narendra Ahuja. Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization. *IEEE TPAMI*, 14(3):318–336, 1992. 2
- [42] Khalid Yousif, Alireza Bab-Hadiashar, and Reza Hoseinezhad. An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems*, 1(4):289–311, 2015. 1
- [43] Wenzhen Yuan and Srikumar Ramalingam. Fast localization and tracking using event sensors. In *ICRA*, pages 4564–4571, 2016. 2
- [44] Yi Zhou, Guillermo Gallego, and Shaojie Shen. Event-based stereo visual odometry. *IEEE T-RO*, 37(5):1433–1450, 2021. 2
- [45] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *CVPR*, pages 5816–5824, 2017. 2
- [46] Yi-Fan Zuo, Jiaqi Yang, Jiaben Chen, Xia Wang, Yifu Wang, and Laurent Kneip. DEVO: Depth-event camera visual odometry in challenging conditions. In *ICRA*, pages 2179–2185, 2022. 2