








Motion and Structure from Event-based Normal Flow

Zhongyang Ren^{1*} , Bangyan Liao^{2*} , Delei Kong¹ , Jinghang Li¹ ,
Peidong Liu² , Laurent Kneip³ , Guillermo Gallego⁴ , and Yi Zhou¹  

¹ School of Robotics, Hunan University

² School of Engineering, Westlake University

³ School of Information Science and Technology, ShanghaiTech University


⁴ TU Berlin, ECDF, SCIOI Excellence Cluster and Robotics Institute Germany

Abstract. Recovering the camera motion and scene geometry from visual data is a fundamental problem in computer vision. Its success in conventional (frame-based) vision is attributed to the maturity of feature extraction, data association and multi-view geometry. The emergence of asynchronous (event-based) cameras calls for new approaches that use raw event data as input to solve this fundamental problem. State-of-the-art solutions typically infer data association implicitly by iteratively reversing the event data generation process. However, the nonlinear nature of these methods limits their applicability in real-time tasks, and the constant-motion assumption leads to unstable results under agile motion. To this end, we reformulate the problem in a way that aligns better with the differential working principle of event cameras. We show that event-based normal flow can be used, via the proposed geometric error term, as an alternative to the full (optical) flow in solving a family of geometric problems that involve instantaneous first-order kinematics and scene geometry. Furthermore, we develop a fast linear solver and a continuous-time nonlinear solver on top of the proposed geometric error term. Experiments on both synthetic and real data show the superiority of our linear solver in terms of accuracy and efficiency, and its practicality as an initializer for previous nonlinear solvers. Besides, our continuous-time non-linear solver exhibits exceptional capabilities in accommodating sudden variations in motion since it does not rely on the constant-motion assumption. Our project page can be found at <https://nail-hnu.github.io/EvLinearSolver/>.

Keywords: Event cameras · Geometric model fitting · Normal flow

1 Introduction

Neuromorphic event-based cameras are bio-inspired sensors that report only brightness changes at each pixel location asynchronously. Their high temporal

* equal contribution;  corresponding author (eeyzhou@hnu.edu.cn).

resolution enables them to sense at exactly the same rate as the scene dynamics, and thus, event cameras are suitable for dealing with robotic perception [33, 38, 39, 41, 43], navigation [21, 34, 35, 42], and control [6, 9] tasks that involve agile motion. A general and fundamental problem among these tasks is to recover the camera’s ego-motion and scene geometry from event data. One of the bottlenecks for solving this kind of problems at very high speeds is the lack of methods for efficient and robust data association.

Unlike its standard-vision counterparts, an event camera cannot observe visual information (*i.e.* intensity changes) when motion is parallel to edges. Such a partial observability of event data makes optical flow estimation (as well as feature detection and matching) an ill-posed problem. To this end, many existing methods [14, 17, 19, 22, 30] for solving the problem of motion-model fitting on event data propose specific metrics that assess event-data association established in either an implicit manner or an explicitly model-driven style. The best fit can be retrieved by iteratively optimizing the data association, often expressed in terms of event alignment. However, we argue two flaws of these methods: (*i*) The success of these non-linear solutions requires a proper initialization. While a brute-force search may be applicable, it is computationally expensive for real-time application, specially if the dimension of the parameter is high. Efficient and deterministic initialization methods are lacking and urgently needed. (*ii*) The common assumption of constant velocity only holds within a short time interval. Nevertheless, a time window that is too narrow may contain insufficient number of events for a valid model fitting. The heuristic of using a constant number of events may violate the constant velocity assumption, especially in the case of sudden changes in speed.

In this paper, we rethink the problem formulation in a way that is more consistent with the differential working principle of event cameras. Unlike existing pipelines that use events occurring within a time interval to resolve the relative motion, we attempt to recover the structure and instantaneous first-order kinematic parameters involved in the motion field equation, using as input an instantaneous observation, namely event-based normal flow. To this end, we explore effective constraints between normal-flow observations and model parameters for solving a family of motion-and-structure estimation problems, such as the estimation of depth and full optical flow, instantaneous angular velocity and linear velocity, and differential homography (see Fig. 1).

The contributions of this paper can be summarized as follows:

- A normal flow constraint with a geometric connection to event data. It overcomes the partial observability issue (*i.e.*, errors caused by using normal flow as replacement for full optical flow) in general problems of geometric model fitting when normal flow is used as input.
- Two solvers for a family of motion-and-structure estimation problems using as input sparse event-based normal flow. Our linear solver (can be used with RANSAC) leads to closed-form and deterministic solutions that can be used as an initialization to existing nonlinear methods. Our nonlinear solver,

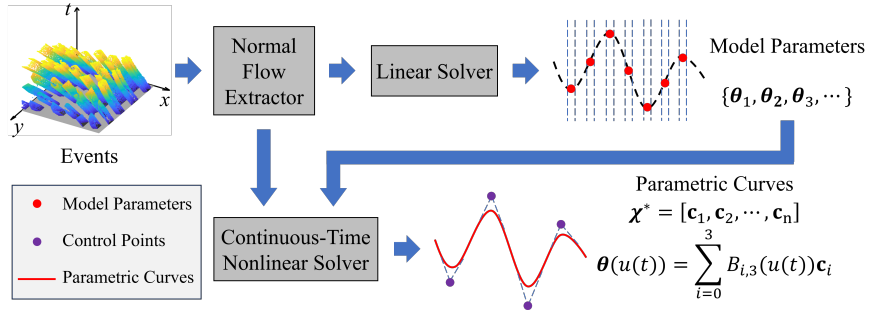


Fig. 1: Overview of our event-based motion estimation approach. Normal flow vectors are computed from the raw event data and are used as input to the two solvers. The result of the linear solver consists of fitted motion models $\{\theta_1, \theta_2, \dots\}$ at discrete time instants, which can be used to initialize the continuous-time nonlinear solver.

under a continuous-time formulation, can handle sudden speed variations, and thus, is free of the constant-motion assumption.

- A thorough evaluation on the two proposed solvers, comparing against state-of-the-art methods, and also including an investigation on the performance of existing non-linear methods initialized using our linear solver.

Outline: The rest of the paper is organized as follows. We first discuss the related work in Sec. 2. Then a compact review of preliminary knowledge is given in Sec. 3. Our approach is presented in Sec. 4, including the normal-flow constraint, the linear models of a family of motion-and-structure estimation problems, and the proposed linear and nonlinear solvers. Experimental evaluation is conducted in Sec. 5 and conclusion are drawn in Sec. 6.

2 Related Work

2.1 Event-based Geometric Model Fitting

The major body of literature on this specific topic follows an iterative scheme, which typically has its own special metric evaluating quantitatively the association between events, and solve the model fitting as a nonlinear optimization problem. Among the first works is the *Contrast Maximization* method [13, 14], which provides a unified framework for solving a family of motion-model fitting problems. It searches the optimal model parameters that ultimately lead to a sharp image of warped events (IWE). The goodness of fit is evaluated by the alignment of all involved events along point trajectories on the image plane, and such alignment is assessed by the strength of the contours of the IWE. The computational complexity of CMax is high because of iteratively updating of IWE, whose cost scales with the spatial resolution and the amount of event data.

To lessen the dependency on IWE, Nunes *et al.* proposes an entropy minimization framework (EMin) [30] that replaces the dispersion metric of [14] with

a similarity measurement between events, enabling to evaluate the goodness of fit in a higher dimensional space. EMin is, still, too expensive for real-time application. Liu *et al.* [22] propose a two-step iterative method for the problem of 3-D rotation estimation using events as input. Given a rotation estimate, the first step determines event correspondences by looking for the nearest neighbour in the spatio-temporal domain. The second step employs a spatio-temporal registration which aligns corresponding events by minimizing the sum of squared geodesic residual. The least-square nature of the energy function in the second step leads to a computation friendly performance. Gu *et al.* [17] present a probabilistic approach for spatio-temporal alignment of events. The aligned events at a particular pixel are modeled as a spatio-temporal Poisson point process and the goodness of fit can be quantified using the likelihood of the data. The optimal model is retrieved when the likelihood is maximized. More recently, an event-to-map alignment scheme is proposed in [19], which adopts the time surface (TS) map as a reference for event alignment. The proposed TS loss function drives later events to get increasingly closer to pixel positions of corresponding earlier events, thus, leading to a progressively sharp TS map as iterations proceed.

In summary, all the above-mentioned methods are nonlinear, and thus, a proper initialization is necessary. Although a brute-force search is widely adopted, sometimes it can be too expensive to locate the convergence basin in parameter space. Moreover, [14,30] may come across a convergence to false-positive minima, known as event collapse [36,37]. Hence, a quick and deterministic solver (e.g., a closed-form solution) is desirable. Besides, most of these methods assume a constant-motion model, which limits their applicability to scenarios of sudden speed variations. In addition, we have noticed that a recent work [16] can jointly estimate the line parameters and the camera’s linear velocities. However, this method is not a general one due to the dependence on line structures and the usage of an IMU. Thus, it is not within the scope of our discussion.

2.2 Motion and Structure from Normal Flow

Due to the aperture problem, a local estimate of optical flow recovers only the partial component of the full flow vector along the direction of image gradient. Preliminary knowledge about normal flow can be found in the supplementary material. Recovering motion and structure from normal flow on intensity images is a well-studied topic. Its main body of literature typically proposes a qualitatively deterministic solution by which a set (even infinite) of applicable ego motions are determined via an analysis of normal flow classification [3, 11, 20]. Only recently, researches devised ways to formulate the Cheirality constraint as a differentiable term and use it in a continuous optimization approach [1] or as a loss function for training an artificial neural network [31].

In event-based perception and navigation, similar methodologies as the frame-based ones are also witnessed, where normal flow computed from event data is used as input to determine the Focus of Expansion (FoE) for drones’ course estimation [10] and obstacle avoidance [5].

Most of these heuristic methods, while effective, are cumbersome for practical application. In contrast, our method delivers a deterministic way to retrieve motion and structure from normal flow. Additionally, it is noticed that normal flow can replace full flow in some specific problems of geometric model fitting, such as affine models [26, 27]. In this way, the geometric model fitting problem can be easily solved in closed-form. Nevertheless, these methods do not consider the observability bias in normal flows, and thus, such naive substitution may lead to inaccurate model fitting. To this end, we present a robust linear solver based on a carefully-designed geometric measurement model that can overcome the partial observability of event-based normal flow. It is worth mentioning that a recent work [24] also determines instantaneous linear velocity using as input the normal flow. Our difference is seen in the scope of applicability. The usage of normal flow is restricted in the motion flow equation of [24], while our paper extends it to a wider scope (*i.e.*, a family of geometric model fitting problems) by combining normal flow with *e.g.*, differential epipolar geometry, differential homography, *etc.*

3 Preliminaries

To make this paper self-contained, some preliminary knowledge is revisited. Due to space limitations, here we only list essential concepts. A more detailed review of preliminary knowledge can be found in the supplementary material.

Motion Field. Consider a perspective camera moving in a static environment, with instantaneous angular and linear velocities in the camera frame $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^\top$ and $\boldsymbol{\nu} = (\nu_x, \nu_y, \nu_z)^\top$, respectively. The motion field at pixel $\mathbf{x} = (x, y)^\top$ produced by projecting the 3D motion at point with depth $Z(\mathbf{x})$ is

$$\mathbf{u} = \frac{1}{Z(\mathbf{x})} \mathbf{A}(\mathbf{x})\boldsymbol{\nu} + \mathbf{B}(\mathbf{x})\boldsymbol{\omega}, \quad (1)$$

where $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are 2×3 matrices that solely depend on \mathbf{x} . Equation (1) is the motion field equation [40, Ch.8] [23]. Optical flow in rigid scenes aims at estimating the motion field from visual data, which is only possible in regions that do not suffer from the aperture problem.

Brightness constancy is the assumption stating that the true motion trajectories $\mathbf{x}(t)$ followed by objects on the image plane are those at which brightness is constant: $I(\mathbf{x}(t), t) = \text{const}$. Differentiating in time (total derivative) gives the equation at (\mathbf{x}, t) : $\nabla I \cdot \dot{\mathbf{x}} + \partial_t I = 0$, where $\dot{\mathbf{x}} = d\mathbf{x}/dt$ is the velocity of image point $\mathbf{x}(t)$. Given I , this equation provides one constraint per point (\mathbf{x}, t) to estimate $\dot{\mathbf{x}}$, now called optical flow. Without additional equations, only the normal component of the optical flow can be determined [40],

$$\mathbf{n} = -\frac{\partial_t I}{\|\nabla I\|^2} \nabla I. \quad (2)$$

Normal Flow. Decomposing the optical flow $\mathbf{u} = \mathbf{u}_\perp + \mathbf{u}_\parallel$ into its normal and parallel components to the local image gradient (∇I) , the normal flow is $\mathbf{n} = \mathbf{u}_\perp$, and the dot-product with \mathbf{n} gives

$$\mathbf{n}^\top \mathbf{u} = \|\mathbf{n}\|^2, \quad \text{or equivalently} \quad (\mathbf{u} - \mathbf{n}) \cdot \mathbf{n} = 0. \quad (3)$$

Differential Homography. (*Continuous Homography Matrix* [25, Ch.5]). If all scene points lie on a plane, their coordinates are related to their image projections via a homography transformation (which encodes the information of the scene plane $\{\mathbf{N}, d\}$ and the camera pose). In this case, the motion field (1) can be rewritten to adopt the form in homogeneous coordinates

$$\hat{\mathbf{u}}(\mathbf{x}) = (\mathbf{1} - \hat{\mathbf{x}}\mathbf{e}_3^\top) \mathbf{H}_d \hat{\mathbf{x}}, \quad (4)$$

where $\mathbf{H}_d \doteq -([\boldsymbol{\omega}]_\times + d^{-1}\boldsymbol{\nu}\mathbf{N}^\top)$ denotes the differential homography matrix ($[\cdot]_\times$ refers to the cross-product, skew-symmetric matrix), $\mathbf{1}$ is the identity matrix, $\hat{\mathbf{x}} = (x, y, 1)^\top$, $\hat{\mathbf{u}} = (u_x, u_y, 0)^\top$ and $\mathbf{e}_3 = (0, 0, 1)^\top$.

Differential Epipolar Geometry. (*Continuous Epipolar Geometry* [25, Ch.5]). The epipolar geometry examines the distance between a potential match to the determined epipolar line, and can be expressed as $\hat{\mathbf{x}}'^\top \mathbf{E} \hat{\mathbf{x}} = 0$. The differential epipolar geometry can be derived analogously, providing a relationship between \mathbf{x}, \mathbf{u} and the camera velocities $\boldsymbol{\nu}, \boldsymbol{\omega}$. In homogeneous coordinates:

$$\hat{\mathbf{u}}(\mathbf{x})^\top [\boldsymbol{\nu}]_\times \hat{\mathbf{x}} - \hat{\mathbf{x}}^\top \mathbf{s} \hat{\mathbf{x}} = 0, \quad (5)$$

where $\mathbf{s} = \frac{1}{2}([\boldsymbol{\nu}]_\times [\boldsymbol{\omega}]_\times + [\boldsymbol{\omega}]_\times [\boldsymbol{\nu}]_\times)$ is a 3×3 symmetric matrix.

4 Methodology

Figure 1 summarizes our approach, grounded upon event-based normal flow. In this section, we first disclose the normal-flow constraint that can overcome the partial observability of normal flow when replacing full optical flow with normal flow in general tasks of geometric model fitting (Sec. 4.1). By combining the constraint and the theory in Sec. 3, we derive linear equations for a family of motion-and-structure estimation problems (Sec. 4.2). To solve these problems, we present a linear solver and a continuous-time nonlinear solver (Sec. 4.3).

4.1 Normal Flow Constraint

Normal flow can be computed easily from the event data, for example by fitting planes locally to the space-time “cloud” of events on image space [2, 32]. Specifically, assuming that the event data $e_k \doteq \{\mathbf{x}_k, t_k\}$ spans locally a surface in the spatio-temporal domain, the surface can be defined by a function $\Sigma_e : \mathbb{R}^2 \rightarrow \mathbb{R}; \mathbf{x}_k \mapsto t_k$ that maps the pixel coordinates to the latest event’s

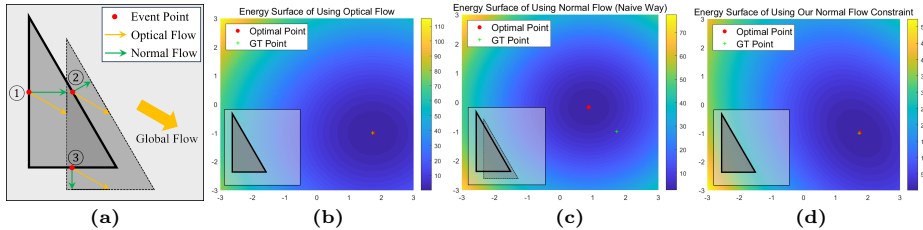


Fig. 2: Toy example of applying the proposed normal-flow constraint. (a): A 2D registration task using as input either optical flow or normal flow. The groundtruth displacement is defined by a global flow of $(1.732, -1)^\top$. (b)-(d): The loss landscape obtained using different geometric measurements in the registration task. The red dot denotes the resulting displacement, and the green cross the groundtruth displacement. The bottom-left regions in (b)-(d) display the corresponding registration results.

timestamp [2, 5]. The normal flow is parallel to the spatial derivative of Σ_e , namely $\nabla \Sigma_e(\mathbf{x})$. More specifically, it can be calculated by

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla \Sigma_e(\mathbf{x})}{\|\nabla \Sigma_e(\mathbf{x})\|^2}. \quad (6)$$

Readers may refer to the supplementary material for a more detailed derivation. Surely there are other methods to extract event-based normal flow and the solvers we develop in this section are independent of them, i.e., our solvers can also be fed with the extracted normal flow data from such methods.

Event-based normal flow cannot be used directly as a replacement for the full flow in (1) to solve for the motion parameters. However, we can leverage the relationship between both flows (3) to define an effective geometric error criterion that circumvents the partial observability of normal flow. If the flow \mathbf{u} depends on some unknown (“state”) variables $\boldsymbol{\theta}$, we propose estimating them based on how well the normal flow constraint (3) is satisfied, e.g., by minimizing some function of the error terms

$$\mathbf{n}(\mathbf{x})^\top \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}) - \|\mathbf{n}(\mathbf{x})\|^2 \doteq \epsilon_{\text{nf}} \in \mathbb{R}. \quad (7)$$

To illustrate the validity of the proposed normal-flow constraint over the naive way of simply using normal flow to replace full optical flow, we use a 2D registration task as toy example. As shown in Fig. 2, the task consists of registering the solid-line triangle to the dashed-line one using as input either optical flow or normal flow. We observe that using the proposed normal-flow constraint as a geometric measurement leads to a registration result that is almost as accurate as the one given by the full optical flow. However, due to the partial observability of normal flow, simply using it as a replacement for full optical flow leads to a biased registration result, as shown in Fig. 2c. As we see in all following experiments, similar conclusions can be made in general problems of geometric model fitting.

4.2 Models of Motion and Structure

Based on the derivation in Sec. 4.1, we establish linear models to a family of motion and structure estimation problems using the event-based normal flow as input. In particular, we first discuss how to recover optical flow and depth by introducing the normal-flow constraint into the differential epipolar geometry. Then we show that the combination of the normal-flow constraint with the motion flow equation (1) results in solutions to the problem of instantaneous angular velocity estimation, and 6-DoF motion estimation given depth as prior, respectively. Finally, we show how to compute the differential homography.

Optical Flow. Given as input the accurate normal flow, and camera velocities, the full optical flow $\mathbf{u}(\mathbf{x})$ at pixel location \mathbf{x} can be calculated by solving the following linear system obtained from the combination of the normal-flow constraint (3) and the differential epipolar geometry (5):

$$\begin{pmatrix} \mathbf{n}(\mathbf{x})^\top \\ -\hat{\mathbf{x}}^\top [\boldsymbol{\nu}]_\times \end{pmatrix} \mathbf{u}(\mathbf{x}) = \begin{pmatrix} \|\mathbf{n}(\mathbf{x})\|^2 \\ \hat{\mathbf{x}}^\top \mathbf{s}\hat{\mathbf{x}} \end{pmatrix}. \quad (8)$$

Depth Estimation. Similarly, if the normal flow and camera velocities are known, the scene depth $Z(\mathbf{x})$ can be determined by combining the normal-flow constraint (3) with the motion field eq. (1), yielding the closed-form solution:

$$Z(\mathbf{x}) = \frac{\mathbf{n}(\mathbf{x})^\top \mathbf{A}(\mathbf{x})\boldsymbol{\nu}}{\|\mathbf{n}(\mathbf{x})\|^2 - \mathbf{n}(\mathbf{x})^\top \mathbf{B}(\mathbf{x})\boldsymbol{\omega}}. \quad (9)$$

Angular Velocity Estimation. Assuming that an event camera undergoes a pure rotational motion, the optical flow \mathbf{u} in (1) can be simplified as

$$\mathbf{u}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\boldsymbol{\omega}. \quad (10)$$

By substituting (10) into (3), we have the linear equation in $\boldsymbol{\omega}$:

$$\mathbf{n}(\mathbf{x})^\top \mathbf{B}(\mathbf{x})\boldsymbol{\omega} = \|\mathbf{n}(\mathbf{x})\|^2. \quad (11)$$

Since each normal flow provides only one constraint, the minimal problem requires three normal flow vectors as input.

6-DoF Motion Estimation. Assume the event camera undergoes a general 6-DoF motion with known depth of the observed scene, we can solve the first-order kinematics, namely instantaneous linear and angular velocities, by solving the following linear system obtained from the normal-flow constraint (3) and the motion-flow equation (1):

$$\mathbf{n}(\mathbf{x})^\top \mathbf{D}(\mathbf{x}) \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\omega} \end{bmatrix} = \|\mathbf{n}(\mathbf{x})\|^2, \quad (12)$$

where $\mathbf{D}(\mathbf{x}) = \left[\frac{1}{Z(\mathbf{x})} \mathbf{A}(\mathbf{x}) \mathbf{B}(\mathbf{x}) \right]$ is the 2×6 feature sensitivity matrix [4, 7]. The minimal problem in this context requires six normal flow measurements to determine the 6-DoF motion parameters.

Motion and Structure Estimation in Planar Scenes. For a moving camera that observes a planar scene, the motion and structure parameters are compatibly encoded in a homography matrix. In our case, we attempt to recover the differential homography \mathbf{h}_d (the vector form of \mathbf{H}_d in (4)) because the instantaneous linear and angular velocities are considered. By expanding (4) and rearranging equations, we have

$$\mathbf{u}(\mathbf{x}) = \mathbf{C}(\mathbf{x})\mathbf{h}_d, \quad (13)$$

where $\mathbf{C}(\mathbf{x})$ is a 2×9 matrix that solely depends on the pixel location \mathbf{x} . Subsequently, substituting (3) into (13), gives:

$$\mathbf{n}(\mathbf{x})^\top \mathbf{C}(\mathbf{x})\mathbf{h}_d = \|\mathbf{n}(\mathbf{x})\|^2. \quad (14)$$

As discussed in [44], retrieving the motion and structure parameters from the decomposition of \mathbf{h}_d is not trivial and beyond the scope of this paper. Therefore, we only focus on the numerical solution of \mathbf{h}_d .

4.3 Our Solvers

According to the motion-and-structure models derived in Sec. 4.2, we have established certain constraints between the model parameters and the normal flow observations. This allows us to simply develop linear solvers for each specific problem. Besides, we further propose a continuous-time nonlinear solver that can effectively handle the presence of sudden variations in motion and does not rely on the constant-motion assumption. To convey our solvers concisely, we generalize the motion-and-structure models into a unified form, as

$$\mathbf{n}(\mathbf{x})^\top \mathbf{O}(\mathbf{x})\boldsymbol{\theta} = \|\mathbf{n}(\mathbf{x})\|^2, \quad (15)$$

where $\mathbf{O}(\mathbf{x})$ is the system matrix⁵ and $\boldsymbol{\theta}$ are the motion/structure parameters.

Linear Solver. Given as input a set (*e.g.*, K) of sparse normal-flow estimates obtained at the same time⁶, our linear solver works out the following (overdetermined) linear system of equations:

$$\begin{pmatrix} \mathbf{n}(\mathbf{x}_1)^\top \mathbf{O}(\mathbf{x}_1) \\ \vdots \\ \mathbf{n}(\mathbf{x}_K)^\top \mathbf{O}(\mathbf{x}_K) \end{pmatrix} \boldsymbol{\theta} = \begin{pmatrix} \|\mathbf{n}(\mathbf{x}_1)\|^2 \\ \vdots \\ \|\mathbf{n}(\mathbf{x}_K)\|^2 \end{pmatrix}. \quad (16)$$

For robust estimation, RANSAC [12] is adopted to deal with noise and outliers.

⁵ $\mathbf{O}(\mathbf{x})$ degenerates to 1 in the problem of optical flow estimation.

⁶ An approximation can be made based on the assumption of constant camera motion. In this way, the normal flow estimates can be collected within a short time interval.

Continuous-Time Nonlinear Solver. To handle the case of a sudden variation in speed that may violate the constant-motion assumption, we further propose a continuous-time nonlinear solver. In particular, we employ cubic B-splines to represent the continuous-time model parameters as

$$\boldsymbol{\theta}(u(t)) = \sum_{i=0}^3 B_{i,3}(u(t)) \mathbf{c}_i, \quad (17)$$

where $B_{i,k}$ denotes the basis function, i the index of control points, k ($=3$) the order of the spline and $\mathbf{c}_i \in \mathbb{R}^N$ the corresponding control point defined in the N -dimensional space of model parameters. Examples of cubic B-splines applied to event camera motion estimation are found in [18, 28]. The function $u(\cdot)$ is a normalization operator, which transfers time t to the spline’s parameter domain by means of basis translation [8]. In this way, the model parameters at any time instant t can be obtained via an interpolation using the known control points.

The formulation of the continuous-time model fitting problem goes as follows. The unknown variable consists of all the involved control points, $\boldsymbol{\mathcal{X}} \doteq [\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n]$, where n denotes the total number of control points. By introducing the continuous-time representation of model parameters to (15), we finally create the following objective function:

$$\boldsymbol{\mathcal{X}}^* = \arg \min_{\boldsymbol{\mathcal{X}}} \sum_{e_k \in \mathcal{E}} \left\| \mathbf{n}(\mathbf{x}_k)^\top \mathbf{O}(\mathbf{x}_k) \boldsymbol{\theta}(u(t_k)) - \|\mathbf{n}(\mathbf{x}_k)\|^2 \right\|^2, \quad (18)$$

where \mathcal{E} denotes the event set in which the norm flow observation is available for every involved event e_k . Note that this continuous-time formulation aligns better with the asynchronous nature of event-based normal flow, enabling a state estimation over a larger temporal window that may lead to a better signal-to-noise ratio. The linear solver can be used to initialize the continuous-time nonlinear solver. An illustrative explanation of our solvers is given in Fig. 1.

5 Experiments

To justify the effectiveness of the proposed algorithms, experiments are conducted on both simulation and real data. First, we introduce the datasets used and the applied metrics for evaluation (Sec. 5.1). Then we report the results of our linear solver and continuous-time nonlinear solver (Sec. 5.2 and Sec. 5.3). Finally, a discussion on the computational complexity and the limitation of our method is provided in Sec. 5.4.

5.1 Experimental Setup

Our evaluation utilizes four datasets, including the indoor event-camera dataset (ECD) captured with a DAVIS240C camera and motion-capture system for groundtruth, and a high-resolution dataset with two rotational motion sequences

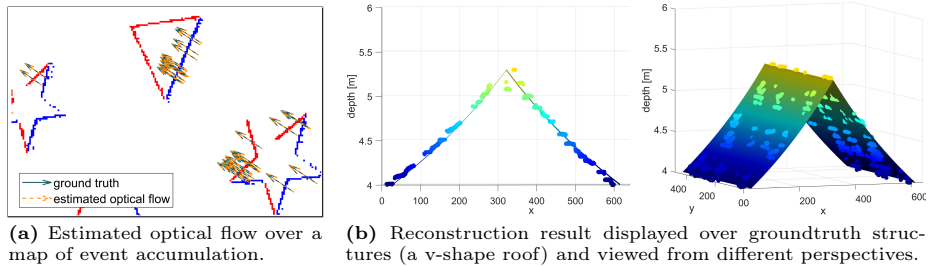


Fig. 3: Qualitative result of optical flow and depth on sequence *three_wall_translation* of our synthetic data. Note that our method returns sparse results.

using an iniVation DAVIS346 camera and Xsens IMU. The VECtor dataset, captured with a Prophesee Gen 3 camera and an accompanying RGB-D depth camera, is used for 6-DoF motion tracking. Additionally, a synthetic dataset generated in Blender is employed for motion-and-structure pattern analysis. Evaluation metrics for kinematics estimation include average error (AE) and root mean squared error (RMSE), while differential homography estimation is assessed using the Frobenius norm. Best results are in bold, and second-best are underlined. The supplementary material provides detailed descriptions of datasets and metrics, as well as implementation details of our methods.

5.2 Evaluation of the Linear Solver

In the following evaluation, we qualitatively and quantitatively assess the performance of our linear method (Ours) in each specific task, and compare against different state-of-the-art (SOTA) methods (CMax [14] and Pro-STR [19]) and their customized version initialized using our linear solver.

Optical Flow and Depth Estimation. Due to a lack of benchmark for extensive evaluation, we qualitatively evaluate our method on the tasks of optical flow and depth estimation. As illustrated in Fig. 3a, our linear solver demonstrates the capability to accurately reconstruct full flow using normal flow and known motion parameters. An example of the depth estimation result is given in Fig. 3b, showing that the recovered structure aligns well with the groundtruth.

Angular Velocity Estimation. As shown in Tab. 1, we see that our linear solver outperforms CMax and Pro-STR. Besides, we find the combination of our linear solver and CMax leads to the best result, indicating that the proposed linear solver is a complementary method to existing nonlinear methods.

Differential Homography Estimation. The result is consistent with that in the above task. Table 2 reveals that our linear method outperforms CMax and Pro-STR in terms of numerical accuracy. The combination of our linear solver and state-of-the-art nonlinear solver leads to the best result.

Table 1: Evaluation of our linear solver on the task of angular velocity estimation.

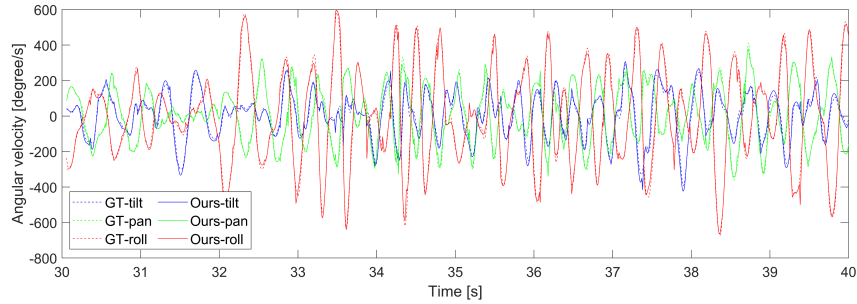
Method	<i>shapes_rotation</i> [29]		<i>ground_rotation</i> (our data)		<i>patterns_rotation</i> (our data)	
	e_ω (deg/s)	RMSE $_\omega$ (deg/s)	e_ω (deg/s)	RMSE $_\omega$ (deg/s)	e_ω (deg/s)	RMSE $_\omega$ (deg/s)
CMax [14]	15.60	26.38	9.48	15.58	5.95	9.07
Pro-STR [19]	90.61	145.81	31.07	44.74	10.47	11.88
Ours	<u>15.36</u>	<u>23.98</u>	<u>3.30</u>	<u>8.01</u>	4.70	6.08
Ours+CMax [14]	10.12	14.29	2.85	3.74	0.35	0.73
Ours+Pro-STR [19]	19.23	40.44	9.13	17.07	<u>4.31</u>	<u>5.23</u>
Method	<i>dynamic_rotation</i> [29]		<i>boxes_rotation</i> (our data)		<i>cubes_rotation</i> (our data)	
	e_ω (deg/s)	RMSE $_\omega$ (deg/s)	e_ω (deg/s)	RMSE $_\omega$ (deg/s)	e_ω (deg/s)	RMSE $_\omega$ (deg/s)
CMax [14]	16.14	26.48	8.18	13.56	6.77	10.62
Pro-STR [19]	54.64	85.63	22.72	32.05	11.07	12.79
Ours	<u>9.41</u>	<u>18.06</u>	2.60	<u>4.78</u>	2.31	3.02
Ours+CMax [14]	7.71	10.85	<u>2.64</u>	3.80	0.49	0.77
Ours+Pro-STR [19]	18.87	32.65	3.49	5.87	<u>2.04</u>	<u>2.76</u>

Table 2: Evaluation of our linear solver for differential homography estimation.

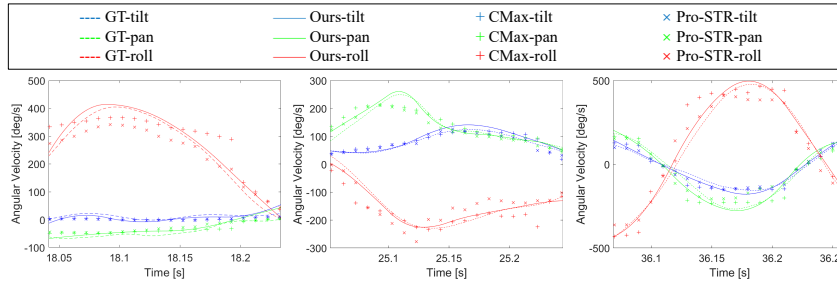
Method	<i>board_slow</i> [15]	<i>patterns_rotation</i>	<i>patterns_6dof</i>
	F-Norm	F-Norm	F-Norm
CMax [14]	1.16	1.33	1.27
Pro-STR [19]	2.07	1.22	1.37
Ours	0.24	0.20	0.43
Ours+CMax [14]	<u>0.23</u>	0.15	0.41
Ours+Pro-STR [19]	0.19	<u>0.17</u>	<u>0.42</u>

Table 3: Evaluation of our linear solver on the task of 6-DoF motion tracking.

Method	<i>corner_slow</i> [15]				<i>patterns_6dof</i> (our data)			
	e_ω (deg/s)	RMSE $_\omega$ (deg/s)	e_v (m/s)	RMSE $_v$ (m/s)	e_ω (deg/s)	RMSE $_\omega$ (deg/s)	e_v (m/s)	RMSE $_v$ (m/s)
CMax [14]	12.39	14.39	0.34	0.41	12.94	15.68	1.02	1.23
Pro-STR [19]	5.19	7.84	0.13	0.22	11.21	13.49	1.33	1.49
Ours	<u>1.31</u>	<u>1.79</u>	0.01	0.02	<u>0.56</u>	<u>0.89</u>	0.12	<u>0.16</u>
Ours+CMax [14]	2.19	2.82	0.11	0.17	0.55	0.87	0.18	0.30
Ours+Pro-STR [19]	1.06	1.34	<u>0.04</u>	<u>0.05</u>	1.07	1.56	0.12	0.15
Method	<i>mountain_normal</i> [15]				<i>cubes_6dof</i> (our data)			
	e_ω (deg/s)	RMSE $_\omega$ (deg/s)	e_v (m/s)	RMSE $_v$ (m/s)	e_ω (deg/s)	RMSE $_\omega$ (deg/s)	e_v (m/s)	RMSE $_v$ (m/s)
CMax [14]	11.01	14.17	0.44	0.52	10.74	14.33	0.78	1.05
Pro-STR [19]	5.23	10.53	0.14	0.25	9.75	11.40	1.29	1.44
Ours	<u>1.28</u>	<u>1.83</u>	<u>0.07</u>	0.09	0.79	1.02	0.11	0.14
Ours+CMax [14]	3.84	6.30	0.21	0.30	1.14	1.72	0.12	0.14
Ours+Pro-STR [19]	1.18	1.58	0.06	0.09	<u>1.05</u>	<u>1.33</u>	0.11	0.14



(a) Results of the continuous-time angular velocity estimator.



(b) Comparison against two state-of-the-art nonlinear solutions using data segments that involve sudden variations of angular velocity.

Fig. 4: Angular velocity estimation using the continuous-time nonlinear solver (Ours).

6-DoF Motion Tracking. Given known depth, we recover the linear and angular camera velocities on the task of 6-DoF motion estimation. As shown in Tab. 3, our linear solver reports more accurate results than both CMax and Pro-STR. In addition, we surprisingly find that the linear solver can sometimes outperform the combination of itself with the two nonlinear solvers. This is, according to our investigation, attributed to the violation of constant-motion assumption in some sequences, which is held for the two nonlinear solvers.

5.3 Evaluation of the Continuous-Time Solver

To justify the claimed advantage of the proposed continuous-time nonlinear solver, we use the exemplary task of angular velocity estimation. The sequence used (*shapes_rotation* [29]) captures a purely rotating event camera, and some parts of the data involves aggressive motion. As illustrated in Fig. 4a, our nonlinear solver gives accurate estimation results with a high stability. Moreover, we select three segments that witness a sudden variation of angular speed within a short time interval and compare our continuous-time nonlinear solver against CMax and Pro-STR. Note that all of them are initialized by our linear method for a fair comparison. As Fig. 4b shows, our results are more consistent with the groundtruth angular velocity than the other two methods, which are less accurate due to the dependence on the constant-motion assumption.

Table 4: Runtime of 6-DoF tracking on sequence *mountain_normal* (640×480 px).

Component	Runtime
Normal flow extraction	20 ms (for extracting 2000 normal flow vectors)
Linear solver	25.97 ms
CT solver	28.29 ms

5.4 Discussion

Computational Complexity. The computational complexity of the linear solver depends on the specific task of model fitting. For optical flow and depth estimation, it is linearly proportional to the size of the input event flow, and thus, the complexity is $\mathcal{O}(N)$. For the other tasks, the time cost predominantly hinges on the total number of RANSAC iterations and the computations performed per iteration. In each iteration, we typically solve a linear overdetermined system via QR decomposition at a complexity of $\mathcal{O}(cN^2)$, where c is the cardinality of a minimal problem (*e.g.*, $c = 3$ for the problem of angular velocity estimation).

Runtime. All experiments are conducted on a standard computer with Intel i7-13700F CPU, 16G memory and implemented with MATLAB (version R2022a). Using the 6-DoF tracking task as an example, we measure the runtime of the proposed linear solver and continuous-time (CT) solver, as reported in Tab. 4.

Limitations. To make it clear, we emphasize that our method, like other compared methods, does not resolve the motion and structure parameters simultaneously. It should also be noted that the success of our method relies heavily on the quality of the input normal flow. Particularly, our normal flow extractor (including some other open source solutions for computing normal flow from event data) may fail in the presence of densely repetitive textures, leading to a failure of the modeling fitting scheme. To assess the sensitivity of our method to the quality of event-based normal flow, we propose an analysis on numerical stability using simulation data in the supplementary material.

6 Conclusions

This work presents a solution to the general problem of geometric model fitting with event data. Considering the differential working principle of event cameras, our method establishes specific constraints between instantaneous motion-and-structure parameters and event-based normal flow. Experiments show that the proposed linear solver leads to closed-form and deterministic solutions, which can be used to initialize existing state-of-the-art nonlinear methods. The proposed continuous-time nonlinear solver, furthermore, demonstrates the capability to handle sudden speed variations, obviating the dependence on the constant-motion assumption. Finally, we hope this work can inspire future research on event-based geometric model fitting, a paramount topic to many downstream techniques, such as event-based 3-D vision, SLAM, and scene segmentation.

Acknowledgments

We thank Dr. Yi Yu for proof reading. This work was supported by the National Key Research and Development Project of China under Grant 2023YFB4706600. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC 2002/1 “Science of Intelligence” – project number 390523135.

References

1. Barranco, F., Fermüller, C., Aloimonos, Y., Ros, E.: Joint direct estimation of 3D geometry and 3D motion using spatio temporal gradients. *Pattern Recog.* **113**, 107759 (2021). <https://doi.org/10.1016/j.patcog.2020.107759>
2. Benosman, R., Clercq, C., Lagorce, X., Ieng, S.H., Bartolozzi, C.: Event-based visual flow. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(2), 407–417 (2014). <https://doi.org/10.1109/TNNLS.2013.2273537>
3. Brodskỳ, T., Fermüller, C., Aloimonos, Y.: Structure from motion: Beyond the epipolar constraint. *Int. J. Comput. Vis.* **37**, 231–258 (2000)
4. Bryner, S., Gallego, G., Rebecq, H., Scaramuzza, D.: Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization. In: *IEEE Int. Conf. Robot. Autom. (ICRA)* (2019). <https://doi.org/10.1109/ICRA.2019.8794255>
5. Clady, X., Clercq, C., Ieng, S.H., Houseini, F., Randazzo, M., Natale, L., Bartolozzi, C., Benosman, R.: Asynchronous visual event-based time-to-contact. *Front. Neurosci.* **8**(9) (2014). <https://doi.org/10.3389/fnins.2014.00009>
6. Conradt, J., Cook, M., Berner, R., Lichtsteiner, P., Douglas, R.J., Delbruck, T.: A pencil balancing robot using a pair of AER dynamic vision sensors. In: *IEEE Int. Symp. Circuits Syst. (ISCAS)*. pp. 781–784 (2009). <https://doi.org/10.1109/ISCAS.2009.5117867>
7. Corke, P.: *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics, Springer (2017). <https://doi.org/10.1007/978-3-319-54413-7>
8. De Boor, C.: On calculating with b-splines. *Journal of Approximation theory* **6**(1), 50–62 (1972)
9. Delbruck, T., Lang, M.: Robotic goalie with 3ms reaction time at 4% CPU load using event-based dynamic vision sensor. *Front. Neurosci.* **7**, 223 (2013). <https://doi.org/10.3389/fnins.2013.00223>
10. Dinaux, R., Wessendorp, N., Dupeyroux, J., De Croon, G.C.: Faith: Fast iterative half-plane focus of expansion estimation using optic flow. *IEEE Robotics and Automation Letters* **6**(4), 7627–7634 (2021)
11. Fermüller, C.: Passive navigation as a pattern recognition problem. *Int. J. Comput. Vis.* **14**(2), 147–158 (1995)
12. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981). <https://doi.org/10.1145/358669.358692>
13. Gallego, G., Gehrig, M., Scaramuzza, D.: Focus is all you need: Loss functions for event-based vision. In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. pp. 12272–12281 (2019). <https://doi.org/10.1109/CVPR.2019.01256>

14. Gallego, G., Rebecq, H., Scaramuzza, D.: A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. pp. 3867–3876 (2018). <https://doi.org/10.1109/CVPR.2018.00407>
15. Gao, L., Liang, Y., Yang, J., Wu, S., Wang, C., Chen, J., Kneip, L.: Vector: A versatile event-centric benchmark for multi-sensor slam. *IEEE Robotics and Automation Letters* **7**(3), 8217–8224 (2022). <https://doi.org/10.1109/LRA.2022.3186770>
16. Gao, L., Su, H., Gehrig, D., Cannici, M., Scaramuzza, D., Kneip, L.: A 5-point minimal solver for event camera relative motion estimation. In: *Int. Conf. Comput. Vis. (ICCV)*. pp. 8015–8025 (2023). <https://doi.org/10.1109/ICCV51070.2023.00739>
17. Gu, C., Learned-Miller, E., Sheldon, D., Gallego, G., Bideau, P.: The spatio-temporal Poisson point process: A simple model for the alignment of event camera data. In: *Int. Conf. Comput. Vis. (ICCV)*. pp. 13495–13504 (2021). <https://doi.org/10.1109/ICCV48922.2021.01324>
18. Guo, S., Gallego, G.: CMax-SLAM: Event-based rotational-motion bundle adjustment and SLAM system using contrast maximization. *IEEE Trans. Robot.* pp. 1–20 (2024). <https://doi.org/10.1109/TRO.2024.3378443>
19. Huang, X., Zhang, Y., Xiong, Z.: Progressive spatio-temporal alignment for efficient event-based motion estimation. In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. pp. 1537–1546 (2023)
20. Ji, H., Fermuller, C.: A 3D shape constraint on video. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(6), 1018–1023 (2006)
21. Kim, H., Leutenegger, S., Davison, A.J.: Real-time 3D reconstruction and 6-DoF tracking with an event camera. In: *Eur. Conf. Comput. Vis. (ECCV)*. pp. 349–364 (2016). https://doi.org/10.1007/978-3-319-46466-4_21
22. Liu, D., Parra, A., Chin, T.J.: Spatiotemporal registration for event-based visual odometry. In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. pp. 4935–4944 (2021). <https://doi.org/10.1109/CVPR46437.2021.00490>
23. Longuet-Higgins, H.C., Prazdny, K.: The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences* **208**(1173), 385–397 (1980). <https://doi.org/10.1098/rspb.1980.0057>
24. Lu, X., Zhou, Y., Niu, J., Zhong, S., Shen, S.: Event-based visual inertial velometer (2024)
25. Ma, Y., Soatto, S., Košecák, J., Sastry, S.: *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer (2004)
26. Meyer, F., Bouthemy, P.: Estimation of time-to-collision maps from first order motion models and normal flows. In: *1992 11th IAPR International Conference on Pattern Recognition*. vol. 1, pp. 78–82. IEEE Computer Society (1992)
27. Meyer, F.G.: Time-to-collision from first-order models of the motion field. *IEEE transactions on robotics and automation* **10**(6), 792–798 (1994)
28. Mueggler, E., Gallego, G., Rebecq, H., Scaramuzza, D.: Continuous-time visual-inertial odometry for event cameras. *IEEE Trans. Robot.* **34**(6), 1425–1440 (Dec 2018). <https://doi.org/10.1109/tro.2018.2858287>
29. Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., Scaramuzza, D.: The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Research* **36**(2), 142–149 (2017). <https://doi.org/10.1177/0278364917691115>
30. Nunes, U.M., Demiris, Y.: Entropy minimisation framework for event-based vision model estimation. In: *Eur. Conf. Comput. Vis. (ECCV)*. pp. 161–176 (2020). https://doi.org/10.1007/978-3-030-58558-7_10

31. Parameshwara, C.M., Hari, G., Fermüller, C., Sanket, N.J., Aloimonos, Y.: Diff-posenet: direct differentiable camera pose estimation. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 6845–6854 (2022)
32. Pijnacker Hordijk, B.J., Scheper, K.Y., De Croon, G.C.: Vertical landing for micro air vehicles using event-based optical flow. *J. Field Robot.* **35**(1), 69–90 (2018)
33. Rebecq, H., Gallego, G., Scaramuzza, D.: EMVS: Event-based multi-view stereo. In: British Mach. Vis. Conf. (BMVC) (2016). <https://doi.org/10.5244/C.30.63>
34. Rebecq, H., Horstschafer, T., Scaramuzza, D.: Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In: British Mach. Vis. Conf. (BMVC) (2017)
35. Rebecq, H., Horstschäfer, T., Gallego, G., Scaramuzza, D.: EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time. *IEEE Robot. Autom. Lett.* **2**(2), 593–600 (2017). <https://doi.org/10.1109/LRA.2016.2645143>
36. Shiba, S., Aoki, Y., Gallego, G.: Event collapse in contrast maximization frameworks. *Sensors* **22**(14), 1–20 (2022). <https://doi.org/10.3390/s22145190>
37. Shiba, S., Aoki, Y., Gallego, G.: A fast geometric regularizer to mitigate event collapse in the contrast maximization framework. *Adv. Intell. Syst.* p. 2200251 (2022). <https://doi.org/10.1002/aisy.202200251>
38. Shiba, S., Aoki, Y., Gallego, G.: Secrets of event-based optical flow. In: Eur. Conf. Comput. Vis. (ECCV). pp. 628–645 (2022). https://doi.org/10.1007/978-3-031-19797-0_36
39. Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., Scaramuzza, D.: Event-based motion segmentation by motion compensation. In: Int. Conf. Comput. Vis. (ICCV). pp. 7243–7252 (2019). <https://doi.org/10.1109/ICCV.2019.00734>
40. Trucco, E., Verri, A.: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA (1998)
41. Zhou, Y., Gallego, G., Rebecq, H., Kneip, L., Li, H., Scaramuzza, D.: Semi-dense 3D reconstruction with a stereo event camera. In: Eur. Conf. Comput. Vis. (ECCV). pp. 242–258 (2018). https://doi.org/10.1007/978-3-030-01246-5_15
42. Zhou, Y., Gallego, G., Shen, S.: Event-based stereo visual odometry. *IEEE Trans. Robot.* **37**(5), 1433–1450 (2021). <https://doi.org/10.1109/TRO.2021.3062252>
43. Zhu, A.Z., Chen, Y., Daniilidis, K.: Realtime time synchronized event-based stereo. In: Eur. Conf. Comput. Vis. (ECCV). pp. 438–452 (2018). https://doi.org/10.1007/978-3-030-01231-1_27
44. Zhuang, B., Tran, Q.H.: Image stitching and rectification for hand-held cameras. In: Eur. Conf. Comput. Vis. (ECCV) (2020). https://doi.org/10.1007/978-3-030-58571-6_15