sensors

MDPI

Article

Visual Odometry with an Event Camera Using Continuous Ray Warping and Volumetric Contrast Maximization

Yifu Wang [1,†], Jiaqi Yang [1,†], Xin Peng [1], Peng Wu [1], Ling Gao [1], Kun Huang [1], Jiaben Chen [1] and Laurent Kneip [1,2,*]

[1] School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China; wangyf8@shanghaitech.edu.cn (Y.W.); yangjq@shanghaitech.edu.cn (J.Y.); pengxin1@shanghaitech.edu.cn (X.P.); wupeng2@shanghaitech.edu.cn (P.W.); gaoling@shanghaitech.edu.cn (L.G.); huangkun1@shanghaitech.edu.cn (K.H.); chenjb1@shanghaitech.edu.cn (J.C.)

[2] Shanghai Engineering Research Center of Intelligent Vision and Imaging, ShanghaiTech University, Shanghai 201210, China

* Correspondence: lkneip@shanghaitech.edu.cn

† These authors contributed equally to this work.

Abstract: We present a new solution to tracking and mapping with an event camera. The motion of the camera contains both rotation and translation displacements in the plane, and the displacements happen in an arbitrarily structured environment. As a result, the image matching may no longer be represented by a low-dimensional homographic warping, thus complicating an application of the commonly used Image of Warped Events (IWE). We introduce a new solution to this problem by performing contrast maximization in 3D. The 3D location of the rays cast for each event is smoothly varied as a function of a continuous-time motion parametrization, and the optimal parameters are found by maximizing the contrast in a volumetric ray density field. Our method thus performs joint optimization over motion and structure. The practical validity of our approach is supported by an application to AGV motion estimation and 3D reconstruction with a single vehicle-mounted event camera. The method approaches the performance obtained with regular cameras and eventually outperforms in challenging visual conditions.

Keywords: computer vision; event-based vision; SLAM; visual localization and mapping

check for updates

Citation: Wang, Y.; Yang, J.; Peng, X.; Wu, P.; Gao, L.; Huang, K.; Chen, J.; Kneip, L. Visual Odometry with an Event Camera Using Continuous Ray Warping and Volumetric Contrast Maximization. *Sensors* **2022**, *22*, 5687. https://doi.org/10.3390/s22155687

Academic Editors: Anastasios Doulamis and Stefania Perri

Received: 29 May 2022
Accepted: 26 July 2022
Published: 29 July 2022

1. Introduction

Vision-based localization and mapping is an important technology with many applications in robotics, intelligent transportation, and intelligence augmentation. Although several decades of active research have led to a certain level of maturity, we keep facing challenges in scenarios with high dynamics, low texture distinctiveness, or challenging illumination conditions [1,2]. Event cameras—also called dynamic vision sensors—present an interesting alternative in this regard, as they pair High Dynamic Range (HDR) with high temporal resolution. The advantages and challenges of event-based vision are well explained by the original work of Brandli et al. [3] as well as the recent survey by Gallego et al. [4].

Previous works have employed time-continuous parametrizations of image warping functions. Based on the assumption that events are pre-dominantly triggered by high-gradients edges in the image, the optimal image warping parameters will cause the events to warp onto a sharp edge map in a reference view called the Image of Warped Events (IWE). The optimal warping parameters are hence found by maximizing contrast in the IWE. Various reward functions to evaluate contrast have been presented and analyzed in the recent works of Gallego et al. [5,6] and Stoffregen and Kleeman [7], and successfully used for solving a variety of problems with event cameras such as optical flow [8–13], segmentation [14–16], 3D reconstruction [17–19], and motion estimation [20–27]. The main problem with the construction of the IWE is that it relies on a low-dimensional image-to-image warping function, which—in the case of both translational and rotational displacements—is only possible if the model is homographic or if knowledge about the depth of the scene is prior available.

Past solutions to event-based localization and mapping, therefore, looked at alternative solution attempts. Note that there are lots of works on the localization and mapping problems individually, a listing of which would go beyond the scope of this introduction. Here we only focus on combined solutions to both problems that use only a single event camera and that can handle combined rotational and translational displacements in unknown, arbitrarily structured environments. There are surprisingly few works that solve this problem, which is proof of its difficulty. The first solution to this problem is given by Kim et al. [28], who propose a complex framework of three individual filters. Results are limited to small-scale environments and small, dedicated motions. A geometric attempt is given by Rebecq et al. [29], who present a combination of a tracker and their ray-density-based structure extraction method EMVS [17]. However, the framework alternates between the tracking and mapping solutions, which leaves open questions as to how to bootstrap the system safely. Zhu et al. [30] finally present a promising learning-based approach. However, it depends on vast amounts of training data and provides no guarantees of optimality or generality.

Our work makes the following contributions:

- We perform contrast maximization in 3D. Using a time-continuous trajectory model, the 3D location of the landmarks corresponding to events is modelled by time-continuous ray warping in space, and the optimal motion parameters are found by maximizing contrast within a volumetric ray density field, denoted by Volume of Warped Events (VWE).
- Our method is the first to perform joint optimization over motion and structure for event cameras exerting both translational and rotational planar displacements in an arbitrarily structured environment.
- We successfully apply our framework to Autonomous Ground Vehicle (AGV) motion estimation with a forward-facing event camera. We prove that by using only an event camera, we can provide good quality, continuous visual localization, and mapping results able to compete with regular camera alternatives, especially as visual conditions degrade.

2. Contrast Maximization

We are given a set of N events $\mathcal{E} = \{e_k\}_{k=1}^{N}$ happening over a certain time interval, where each event $e_k = \{\mathbf{x}_k, t_k, b_k\}$ is defined by its image location $\mathbf{x}_k = [x_k \quad y_k]^T$, timestamp t_k, and polarity b_k. Note that the set is ordered, meaning that if $\mathcal{E} = \{\ldots, e_i, \ldots, e_j, \ldots\}$, then $t_i \leq t_j$. We furthermore assume that image warping during the entire time interval can be parametrized as a continuous-time function of a certain parameter vector $\boldsymbol{\theta}$, and define the warping function $\mathbf{x'}_k = \mathbf{W}(\mathbf{x}_k, t_k|\boldsymbol{\theta})$ that warps an event with location $\mathbf{x}_k$ and timestamp t_k into a reference view at t_r.

Gallego et al. recently proposed a unifying framework for solving motion estimation problems with event cameras [5]. If the motion is estimated correctly, events that are triggered by the same point will be accumulated by the same pixel in the reference view, and the resulting Image of Warped Events (IWE) will, therefore, become a sharp edge map. The question is how the accumulation is done and how the sharpness of the IWE is characterized.

Gallego et al. propose to optimize the alignment of the events by maximizing the contrast in the IWE. Formally, the IWE at point $\mathbf{x}$ is defined by $I(\mathbf{x}|\grave{}) = \sum_{k=1}^{N} e^{-\frac{\|\mathbf{x}-\mathbf{x'}_k\|^2}{2\sigma}}$, and it is evaluated discretely for each pixel center location. While the application of a Gaussian kernel makes sure that events that are closer to a certain pixel contribute more than events that are further away, it also makes sure that the IWE and its contrast remain smooth functions of the motion parameters and thus optimizable through gradient-based methods. According to [6,7], the contrast or sharpness of the IWE may finally be evaluated using one of several possible focus loss functions. Here we use the perhaps most common one, given by the IWE variance $f_{Var}(I) = \frac{1}{N_p} \sum_{i,j} (I(\mathbf{x}_{ij}|\boldsymbol{\theta}) - \mu_I)^2$. μ_I is the mean value of

I, N_p the number of pixels in I, and i and j are indices that loop through all the rows and columns of the IWE. As shown in the heat maps of [6], the highest variance of the IWE gives the highest contrast location, and thus the optimal motion parameters cause the best alignment of the warped events.

The framework allows us to tackle several important motion estimation problems for event-based vision, such as optical flow estimation, motion segmentation, or pure rotational motion estimation. However, note that for an arbitrary point to be warped into the reference view, the warping must either be homographic or the parameter vector θ must contain the depth for each event at the time it was captured. Both are rather restrictive towards general camera motion estimation in arbitrary environments. Current state-of-art contrast maximization methods can, therefore, only handle a particular set of problems such as motion in front of a plane or pure rotation.

3. Volumetric Contrast Maximization Using Ray Warping

Let us now proceed to our main contribution, which consists of extending the idea of contrast maximization into 3D, a technique that will enable us to handle situations in which we perceive non-planar environments under arbitrary motion and with no priors on the depth of events. Our main idea is illustrated in Figure 1. We introduce a continuous-time camera trajectory model as done in Furgale et al. [31], which parametrizes both the position and the orientation of the sensor as a smooth, continuous function of time. For a given event, we may then use its timestamp to extrapolate the position and orientation of the event camera at the time the event was captured. Combined with the normalized spatial direction of the event inside the camera frame, each event can be translated into a spatial ray for which the starting point and orientation depend on the continuous trajectory parameters. Rather than evaluating the density of points for pixels in the image, we then propose to evaluate the density of rays at discrete locations in a volume in front of a reference view. We denote this volumetric density field the Volume of Warped Events (VWE). The intuition is analogous to the IWE: the assumption is that there is a limited number of spatial (appearance or geometric) edges that will cause sufficiently large gradients in the image. Under the optimal motion parameters, the rays of the events will therefore intersect along those spatial edges and cause maximum ray density in those regions. In other words, the optimal motion parameters may be found by maximizing the contrast in the VWE. The important question is again given by how to express the ray density in the VWE.

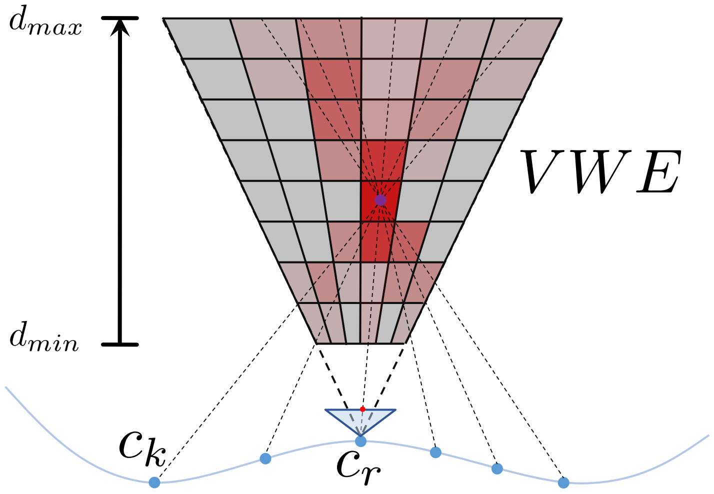


Figure 1. Volume of Warped Events: Events are transformed into rays that are warped in space based on a continuous time trajectory model. We evaluate the ray density in a volume in front of a reference view and maximize its contrast as a function of the continuous motion parameters.

The structure of the VWE field is inspired by the space-sweeping approach of [17] et al., who propose to estimate 3D structure regardless of explicit data associations and photometric information by finding local maxima in a spatial ray density field. However, their method assumes known camera poses, using an alternative camera tracking scheme in their previous work [29]. To the best of our knowledge, we are the first to propose the maximization of the contrast in the volumetric ray density field and thus implicitly perform joint optimization over the continuous camera trajectory parameters and the 3D structure.

3.1. Continuous Ray Warping

Suppose our event camera is pre-calibrated and camera-to-image as well as image-to-camera transformation functions $\pi(\cdot)$ and $\pi^{-1}(\cdot)$ are given. The latter transforms image locations into spatial directions in the camera frame by $\mathbf{f}_k = \pi^{-1}(\mathbf{x}_k)$. In terms of the extrinsic, the trajectory of the camera is kept general for now and simply represented by a minimal, time-continuous, smoothly varying 6-vector $\mathbf{s}(t|\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{t}(t|\boldsymbol{\theta}) \\ \mathbf{q}(t|\boldsymbol{\theta}) \end{bmatrix}$, where $\boldsymbol{\theta}$ still represents a set of continuous motion parameters, $\mathbf{t}$ the position of the camera expressed in a world frame, and $\mathbf{q}$ its orientation as a Rodriguez vector. Note that the dimensionality of $\boldsymbol{\theta}$ is left unspecified for now. However, as will be shown in Section 4, it may indeed have only one or two parameters for certain special types of planar displacements. Besides its inherently smooth property, the continuous-time trajectory model has the obvious ability to be able to register information coming from temporally dense sampling sensors, such as event cameras. The transformation from camera to world at time t is given by $\mathbf{T}(t|\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{R}(\mathbf{q}(t|\boldsymbol{\theta})) & \mathbf{t}(t|\boldsymbol{\theta}) \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$. With reference to Figure 1, c_k represents the camera frame at time t_k where a certain event e_k has been captured. The absolute pose of the frame at the time of capturing e_k is given by $\mathbf{T}_{wk} = \mathbf{T}(t_k|\boldsymbol{\theta})$. Now let c_r be the reference frame in which we define the projective sampling volume for the VWE. The absolute pose of c_r is given by $\mathbf{T}_{wr} = \mathbf{T}(t_r|\boldsymbol{\theta})$. The relative transformation is finally given as

$$\mathbf{T}_{rk} = \begin{bmatrix} \mathbf{R}_r(t_k|\boldsymbol{\theta}) & \mathbf{t}_r(t_k|\boldsymbol{\theta}) \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \mathbf{T}_{wr}^{-1}\mathbf{T}_{wk}. \tag{1}$$

Finally, let λ represent the unknown depth along the ray. Any point on the ray seen from the reference view can be parametrized by $\mathbf{p}_k(\lambda) = \lambda \mathbf{R}_r(t_k|\boldsymbol{\theta})\mathbf{f}_k + \mathbf{t}_r(t_k|\boldsymbol{\theta})$.

3.2. VWE and Spatial Contrast Maximization

We are now going back to our question of how to express the ray density in the VWE. The VWE is defined in a volumetric, projective sampling grid, as illustrated in Figure 2. Let $\mathbf{v}$ be the center of a voxel. The density of the rays in a voxel is now expressed as a function of the orthogonal distance between the voxel center $\mathbf{v}$ (expressed in the reference view) and each individual ray. This spatial point-to-line distance is also called the *object space distance*, and it is given by

$$\epsilon_k^r(\mathbf{v}|\boldsymbol{\theta}) = \|(\mathbf{I} - \mathbf{V}_k)(\mathbf{R}_r^\mathsf{T}(t_k|\boldsymbol{\theta})(\mathbf{v} - \mathbf{t}_r(t_k|\boldsymbol{\theta})))\|, \tag{2}$$

where we have used the rotation $\mathbf{R}_r^\mathsf{T}(t_k|\boldsymbol{\theta})$ and translation $-\mathbf{R}_r^\mathsf{T}(t_k|\boldsymbol{\theta})\mathbf{t}_r(t_k|\boldsymbol{\theta})$ to transform the voxel centre $\mathbf{v}$ into the camera viewpoint at time t_k, and $(\mathbf{I} - \mathbf{V}_k) = (\mathbf{I} - \frac{\mathbf{f}_k\mathbf{f}_k^T}{\mathbf{f}_k^T\mathbf{f}_k})$ is the householder matrix to project this point onto the normal plane of the observation direction $\mathbf{f}_k$. An example of object space distances for one voxel is indicated in Figure 2.

Supposing that we have N events, the final VWE is again given in smooth form by applying a Gaussian kernel and summing up the object space distances of every event with respect to the voxel center $\mathbf{v}$

$$V^r(\mathbf{v}|\boldsymbol{\theta}) = \sum_{k=1}^{N} e^{-\frac{\epsilon_k^r(\mathbf{v}|\boldsymbol{\theta})^2}{2\sigma}}. \tag{3}$$

The standard deviation σ of the Gaussian kernels is not constant but chosen as a function of the depth of the corresponding voxel from the center of the reference view.

The final optimization objective is given by maximizing the variance of the VWE, which expresses the sharpness of the edges reflected in the volumetric density field

$$f_{Var}(V^r) = \frac{1}{N_v} \sum_{m,n,l} (V^r(\mathbf{v}_{mnl}|\boldsymbol{\theta}) - \mu_{V^r})^2. \quad (4)$$

μ_{V^r} is the mean value of V^r, N_v the number of voxels in the entire volume, and m, n, and l now iterate through the voxels in the volume. Figure 3 visualizes an example VWE for wrong and correct motion parameters. For correct motion parameters (cf. Figure 3a,b), which is the exact rotation and translation displacement obtained from groundtruth, the density field presents higher values and more contrast than for wrong motion parameters (cf. Figure 3c,d), which we add a large perturbation on both rotation and translation displacement.

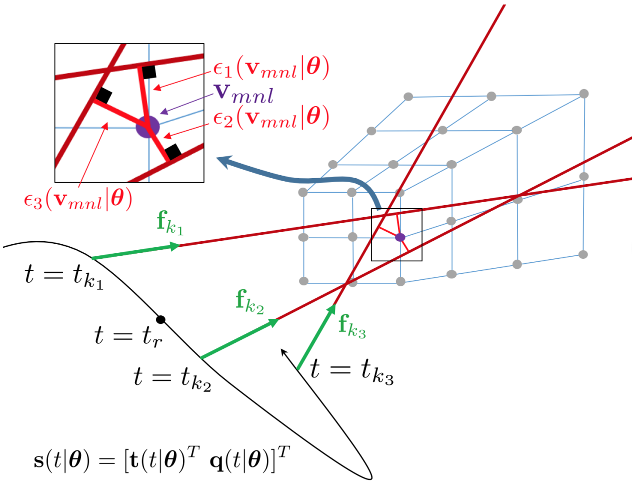

$\epsilon_1(\mathbf{v}_{mnl}|\boldsymbol{\theta})$
$\mathbf{v}_{mnl}$
$\epsilon_2(\mathbf{v}_{mnl}|\boldsymbol{\theta})$
$\epsilon_3(\mathbf{v}_{mnl}|\boldsymbol{\theta})$
$\mathbf{f}_{k_1}$
$t = t_{k_1}$
$\mathbf{f}_{k_2}$
$\mathbf{f}_{k_3}$
$t = t_r$
$t = t_{k_2}$
$t = t_{k_3}$
$\mathbf{s}(t|\boldsymbol{\theta}) = [\mathbf{t}(t|\boldsymbol{\theta})^T \ \mathbf{q}(t|\boldsymbol{\theta})]^T$

Figure 2. Warped rays with object space distances for an example voxel $\mathbf{v}_{mnl}$.

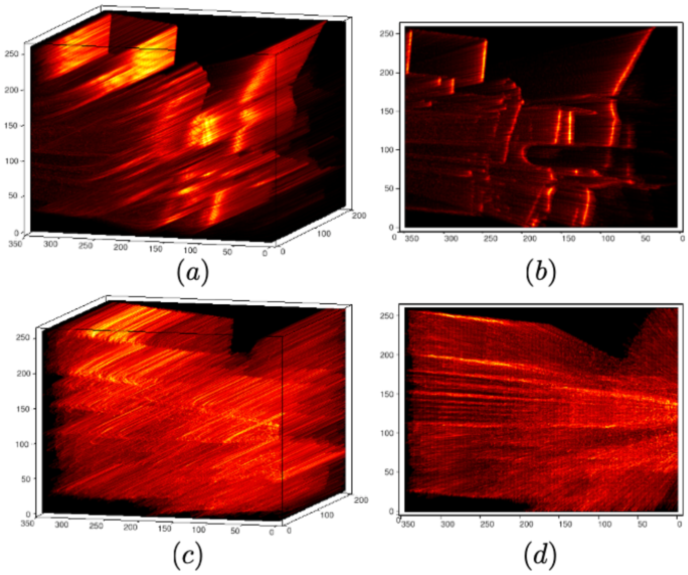


Figure 3. Volumetric ray density fields for correct (**a**,**b**) and wrong (**c**,**d**) motion parameters.

3.3. Global Optimization Over Longer Trajectories

We perform global optimization by simultaneously maximizing the contrast in multiple VWEs cast from neighboring reference views. Let $\{t_{r_1}, \dots, t_{r_M}\}$ be the time instants at which individual VWEs are placed. For simplicity, the time instants are regularly spaced such that $t_{r_{i+1}} - t_{r_i} = \tau_1$. We furthermore define time intervals $[t_{r_i} - \frac{\tau_2}{2} : t_{r_i} + \frac{\tau_2}{2}]$ for each corresponding field V^{r_i}, which define the subset of events that will be used for registration in that reference view. More specifically, event e_k is used in V^{r_i} if $t_k \in [t_{r_i} - \frac{\tau_2}{2} : t_{r_i} + \frac{\tau_2}{2}]$. The overall global optimization objective becomes

$$\max_{\boldsymbol{\theta}} \sum_{i=1}^{M} \frac{1}{N_v} \sum_{m,n,l} (V^{r_i}(\mathbf{v}_{mnl}|\boldsymbol{\theta}) - \mu_{V^{r_i}})^2,$$

$$\text{where } V^{r_i}(\mathbf{v}|\boldsymbol{\theta}) = \sum_{e_k \in \mathcal{E}_{t_{r_i}-\frac{1}{2}\tau_2}^{t_{r_i}+\frac{1}{2}\tau_2}} e^{-\frac{\epsilon_k^{r_i}(\mathbf{v}|\boldsymbol{\theta})^2}{2\sigma}}, \qquad (5)$$

and $\mathcal{E}_{t_{r_i}-\frac{1}{2}\tau_2}^{t_{r_i}+\frac{1}{2}\tau_2}$ is defined as the subset of all the events e_k for which $t_k \in [t_{r_i} - \frac{\tau_2}{2} : t_{r_i} + \frac{\tau_2}{2}]$. The global optimization strategy is depicted in Figure 4. Note that τ_1 may be chosen such that neighboring volumes are overlapping, and τ_2 may be chosen such that events are considered in more than just a single volumetric density field (i.e., $\tau_2 > \tau_1$). These choices guarantee that the implicit graph behind this optimization problem is well connected and effects such as scale propagation take place.

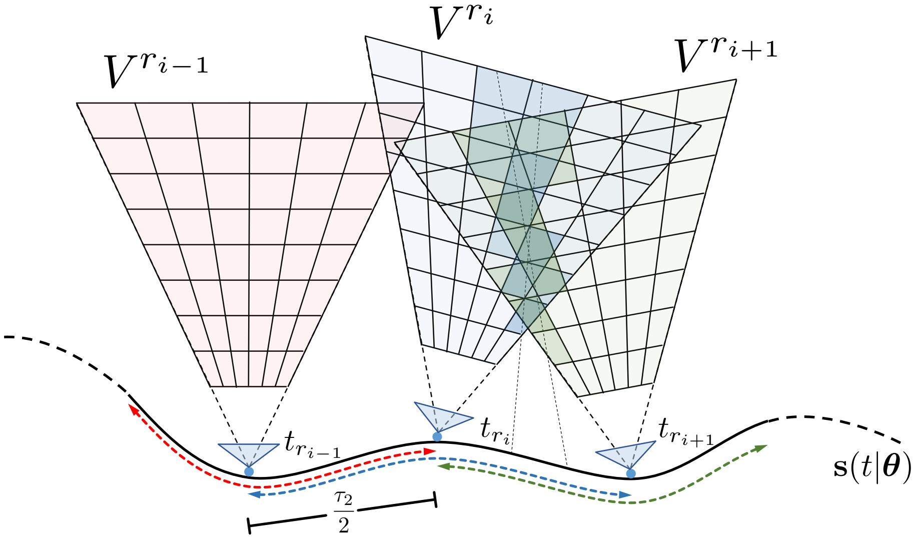


Figure 4. Global optimization over multiple reference volumes. The volumes may have spatial overlap. There is an individual time span $[t_{r_i} - \frac{1}{2}\tau_2 : t_{r_i} + \frac{1}{2}\tau_2]$ associated with each reference volume V^{r_i} from which events will be considered (marked by the red, blue and green arrows). The time spans may have temporal overlap. Two events may hence both appear in two distinct density fields, which reinforces scale propagation in the optimization.

4. Application to AGV with a Forward-Facing Event Camera

We evaluate our method on a planar Autonomous Ground Vehicle (AGV) on which we mount a single forward-facing event camera. Many solutions for regular, monocular cameras exist, such as simple relative pose solvers [32] or full visual SLAM frameworks [33]. The application of an event camera promises strong advantages in situations of high motion dynamics or—as shown in this work—challenging illumination conditions. Our motion estimation framework is divided into two sub-parts, a front-end module that initializes motion over shorter segments and a back-end module that refines the estimate over larger-scale sequences. Both will be introduced after a short overview of the framework.

4.1. Framework Overview

The complete Visual Odometry (VO) system is designed based on the above VWE method. There are two main modules in the pipeline. The front-end initialization module groups the events into sufficiently small subsets such that the motion on these subsets can be locally approximated using a simplified first-order constant velocity model. Furthermore, the front-end performs contrast maximization using a single VWE only. After a sufficient number of events and initial relative displacements have been accumulated, our method then proceeds to the back-end optimization part. The latter initializes a larger-scale, smooth, continuous-time trajectory model and executes the multi-volume optimization outlined in (5).

4.2. Front-End Single-Frame Optimization

For the local approximation of the motion, we use a parametrization that is inspired by [34,35]. Based on the assumptions of a driftless, non-holonomic platform and locally constant velocities, the continuous motion of the planar vehicle may be approximated to lie on an arc of a circle to which the heading of the vehicle remains tangential. This motion model is also known as the Ackermann motion model, and the center of the circle is commonly referred to as the Instantaneous Centre of Rotation (ICR). The model has only two degrees of freedom, which largely simplifies the geometry of the problem. It is given by the forward velocity v and the rotational velocity ω.

Using the convention and equations from [35], the relative transformation from a frame at time t_k to a nearby reference frame at time t_r is given by

$$\begin{aligned}\mathbf{R}_r^v(t_k|\boldsymbol{\theta}) &= \begin{bmatrix}\cos\omega(t_k-t_r) & -\sin\omega(t_k-t_r) & 0\\ \sin\omega(t_k-t_r) & \cos\omega(t_k-t_r) & 0\\ 0 & 0 & 1\end{bmatrix}\\ \mathbf{t}_r^v(t_k|\boldsymbol{\theta}) &= \frac{v}{\omega}\begin{bmatrix}1-\cos(\omega(t_k-t_r))\\ \sin(\omega(t_k-t_r))\\ 0\end{bmatrix}.\end{aligned} \tag{6}$$

Given that scale is unobservable, we fix the forward velocity v to the configured speed of the vehicle (correct scale propagation is taken into account in the later global optimization scheme). As a result, the local motion initialization scheme over a single volume has only 1-DoF, and the parameter vector becomes $\boldsymbol{\theta} = \omega$. Note furthermore that the original Ackermann model requires the camera to be mounted in the center of the non-steering axis, which—in practice—hardly ever is the case. We, therefore, add the extrinsic calibration parameters $\mathbf{R}_{vc}$ and $\mathbf{t}_{vc}$, which transform points back-and-forth between the camera and the vehicle reference frames. The reader is invited to read up [35] to see more foundations of the Ackermann motion model. Note that the variance of the VWE is a function of our unique degree of freedom ω, and the motion parameters can thus be efficiently solved by local gradient-based optimization methods once a rough initial guess is given.

4.3. Back-End Multi-Frame Optimization

The front-end obviously estimates the motion over short time periods only and furthermore relies on the approximation of locally constant velocities and a circular arc trajectory. We add a global back-end optimization over the entire trajectory, which relies on a more general model representing smooth planar motion. We use a two-dimensional, p-th degree B-spline curve

$$\mathbf{c}_{2\times1}(t) = \sum_{i=0}^{n} N_{i,p}(t)\mathbf{p}_i, \qquad a \le t \le b, \tag{7}$$

where the $\{\mathbf{p}_i\}$ stand for the $n+1$ two-dimensional control points that control the shape of the smooth, planar trajectory, and the $\{N_{i,p}(t)\}$ are the known pth-degree B-spline basis functions. The reader is invited to read up [31,36] to see the foundations of B-splines and example applications. Here we only focus on establishing the link to our smooth camera pose functions used in the optimization objective (5).

The parameter vector $\boldsymbol{\theta}$ may be defined as the stacked control points of the spline expressed by $\boldsymbol{\theta} = [\mathbf{p}_0^T \ \ldots \ \mathbf{p}_n^T]^T$. The spline directly models the position in the plane, so we easily obtain $\mathbf{t}^v(t|\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{c}(t) \\ 0 \end{bmatrix}$. For planar motion, the orientation is given by a pure rotation about the vertical axis, and we furthermore exploit the fact that for driftless non-holonomic vehicles, the heading of the vehicle remains tangential to the trajectory. If the heading of the vehicle is still defined as the y axis, and the z axis points vertically upwards, the orientation of the vehicle is finally given as

$$\mathbf{R}^v(t|\boldsymbol{\theta}) = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \dot{\mathbf{c}}(t) & \dot{\mathbf{c}}(t) & \mathbf{0} \\ 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Note that only the temporal basis functions depend on time and that $\dot{\mathbf{c}}(t)$ therefore also is a spline-based function of the same control points. The control point vector is initialized from the approximate trajectory given by the front end using the spline curve approximation given by the automatic knots spacing algorithms (9.68) and (9.69) of [36].

5. Implementation and Validation

In this section, we briefly introduce implementation details of our method and then test our algorithm on multiple both synthetic and real datasets. We assess both the accuracy and quality of the estimated trajectories, as well as the quality of the implicitly modeled 3D structure.

5.1. Implementation Details

We utilize the event back-projection approach proposed in [17] to find the neighboring voxels of a spatial ray efficiently. The details of this algorithm can be found in Section 7.1 of [17]. We furthermore use a simple gradient-ascent scheme to solve our volumetric contrast maximization problems. Especially in (6), the fixation of the forward velocity v leaves the angular velocity ω as the only unknown parameter, thus making the front-end constraint a univariate problem. Finally, to recover the implicitly modeled 3D structure of the environment, we simply reuse the Event-based Multi-View Stereo (EMVS) method from [17].

5.2. Experiment Setup

To demonstrate the performance of our algorithm, we apply it to both synthetic and real datasets. In the synthetic case, we use large-scale outdoor sequences from the KITTI benchmark [37] and convert the image sequences into event data by using the method of Gehrig et al. [38]. The datasets are fully calibrated and contain images captured by a forward-looking camera mounted on a vehicle driving through a city. Experiments on real data are conducted by collecting several small-scale indoor sequences with a DAVIS346 event camera. The camera is mounted forward-facing on a turtlebot (cf. illustrated in Figure 5). It has a resolution of 346×260, and captures RGB images in parallel to the events.

We compare our approach against traditional camera alternatives. Our current implementation focuses on non-holonomic planar motion, which is why we use the 1-point RANSAC algorithm for Ackermann motion [34] as a solid baseline algorithm for the regular camera alternative. We also let our method compete against an established alternative from the open-source community: ORB-SLAM [33]. Note that we rescale all monocular, scale-invariant results to align as well as possible with groundtruth, which we obtain from the original KITTI datasets or an Opti-track system.

It should be noted that a direct comparison against alternative event-based VO/SLAM projects is difficult for several reasons. To date, there are no open-source implementations and we are the first to even evaluate a monocular, event-based pipeline on a popular, established benchmark sequence. Furthermore, as stated in Section III. D of [29] and Section 3.5 of [28], the few existing alternatives either depend strongly on the quality of an initial 3D map (cf. [29]) or suffer from slowly converging depth estimates (cf. [28]). As shown in their experiments, they, therefore, require hovering motion in front of the same scene to provide sufficient time for the mapping back-end to converge. In contrast, our method performs joint optimization of trajectory and structure in near real-time, and thus successfully handles the continuous forward-exploration scenario.

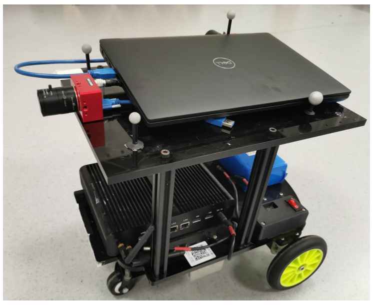


Figure 5. AGV equipped with a forward-facing event camera for vehicle motion estimation.

5.3. Experiment on Synthetic Data

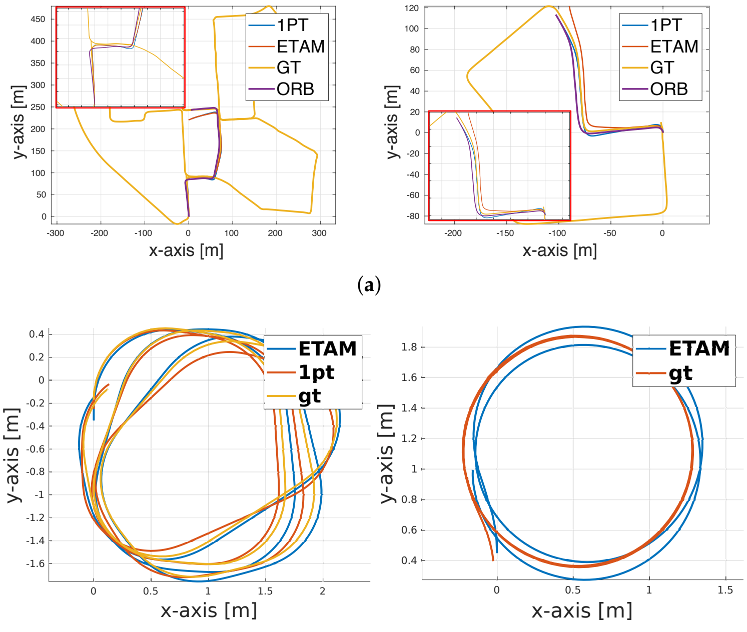
To prove the effectiveness of our method—which denote **ETAM**—, we apply it to synthetic sequences generated from the KITTI benchmark datasets [37]. These datasets represent a fairly normal use case without high motion dynamics or challenging illumination. We use the publicly available tool proposed [38] to convert the regular videos into event streams. We compare our method against two alternatives, which are the state-of-the-art **ORB-SLAM** algorithm [33] and the classical 1-point Ransac algorithm—denoted **1pt**—for planar motion [34]. The evaluation is performed on sequences *VO-00* and *VO-07*.

The qualitative performance is illustrated in Figure 6a. All algorithms successfully process the sequences without any gross errors, and our system is slightly less accurate than **ORB-SLAM** on these high-quality datasets. We furthermore believe that the decrease in performance is mostly explained by the approximate motion model, which ignores the slight pitch angle variations that could result from unevenness of the ground surface. Furthermore, we perform on par with **1pt**, which also relies on a non-holonomic planar motion model. To the best of our knowledge, this result is the first to demonstrate a monocular event camera solution that returns comparable results to regular camera alternatives.

5.4. Experiment on Real Data

To demonstrate the performance of our algorithm on real data, we collect further sequences with a DAVIS346 event camera. The datasets are captured indoors to simulate different illumination conditions and capture groundtruth via Opti-track. We first apply them to two shorter sequences in which the camera follows an either circular (*Circle*) or purely translational trajectory (*Str*). Next, we perform a test on a much longer sequence with a more complex motion (*Long2*). While the first three sequences are recorded under good illumination conditions, we conclude with another sequence with varying lighting conditions by toggling external illumination while the dataset is recorded (*HDR*).

ORB-SLAM proves to be fragile when applied to our indoor sequences. The images have low resolution and the proximity of the structure as well as fast vehicle rotations furthermore induce large frame-to-frame disparities, ultimately causing ORB-SLAM to break in such forward-exploration scenarios. We therefore assess the performance by a quantitative comparison of relative pose errors between **ETAM** and **1pt**. Results for all sequences are summarized in Table 1. It shows the root-mean-square or median of all deviations between estimated and groundtruth short-term relative rotation and translation displacements. Note that— to minimize the impact of unobservable scale—the error of the relative translation is evaluated by considering only the direction. Furthermore, errors are assessed per time, as it is clear that larger intervals may lead to more drift. We, therefore, employ the unit deg/s for both rotational and translational errors. The best performance is always highlighted in bold.



(**a**)

(**b**)

Figure 6. Results for both our method and 1pt-RANSAC on long outdoor trajectories (**top**) and indoor sequences (**bottom**). The indoor sequences are captured under normal (**left**) or challenging illumination conditions (**right**). (**a**) synthetically generated outdoor sequences. (**b**) real data indoor sequences.

It can be easily observed that **ETAM** outperforms **1pt** on most datasets, and it is able to continuously track entire sequences with high accuracy even as illumination conditions become more challenging. In contrast, regular camera-based visual odometry with 1-point RANSAC fails due to poor contrast or motion blur in dark or varying illumination settings (cf. Figure 7). Due to the forward-facing arrangement, the purely translational displacement, on the other hand, triggers much fewer events than trajectories with rotational displacements, hence the slightly inferior performance for this type of motion. Figure 6b visualizes top views of complete trajectories for both algorithms and groundtruth (denoted **gt**). The left figure is from the sequence *Long2*, and the right one is from the sequence *HDR*. Our event-based method can work robustly in all challenging conditions. We kindly refer the reader to our supplemental video file for further qualitative results of our method.

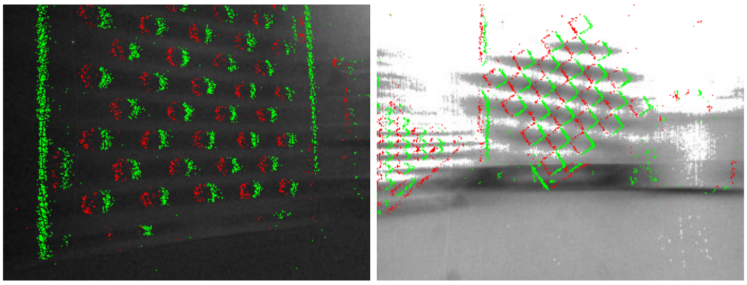


Figure 7. *Challenging illumination conditions*. Regular frames suffer from poor contrast (**left**) when lights are off or motion blur (**right**) when lights are on, which is caused by inappropriate exposure time under varying illumination conditions. Events, in turn, preserve the visual information of the structure.

Table 1. Accuracy on different sequences. Unit: $[deg/s]$.

method	Circular motion			
	Rmse($\mathbf{R}$)	Median($\mathbf{R}$)	Rmse($\mathbf{t}$)	Median($\mathbf{t}$)
1pt	2.4526	2.3330	0.5427	**0.0296**
ETAM	**1.3275**	**0.6443**	**0.2826**	0.0322
method	Purely translational motion			
	Rmse($\mathbf{R}$)	Median($\mathbf{R}$)	Rmse($\mathbf{t}$)	Median($\mathbf{t}$)
1pt	**0.6997**	**0.5300**	0.5179	0.0369
ETAM	0.9769	0.6334	**0.4637**	**0.0235**
method	Long trajectory			
	Rmse($\mathbf{R}$)	Median($\mathbf{R}$)	Rmse($\mathbf{t}$)	Median($\mathbf{t}$)
1pt	1.8516	1.5829	0.1675	0.1718
ETAM	**1.6901**	**1.3417**	**0.1631**	**0.1703**
method	Challenging illumination conditions			
	Rmse($\mathbf{R}$)	Median($\mathbf{R}$)	Rmse($\mathbf{t}$)	Median($\mathbf{t}$)
1pt	-	-	-	-
ETAM	**1.6042**	**0.9093**	**0.0686**	**0.0084**

5.5. Computational Efficiency

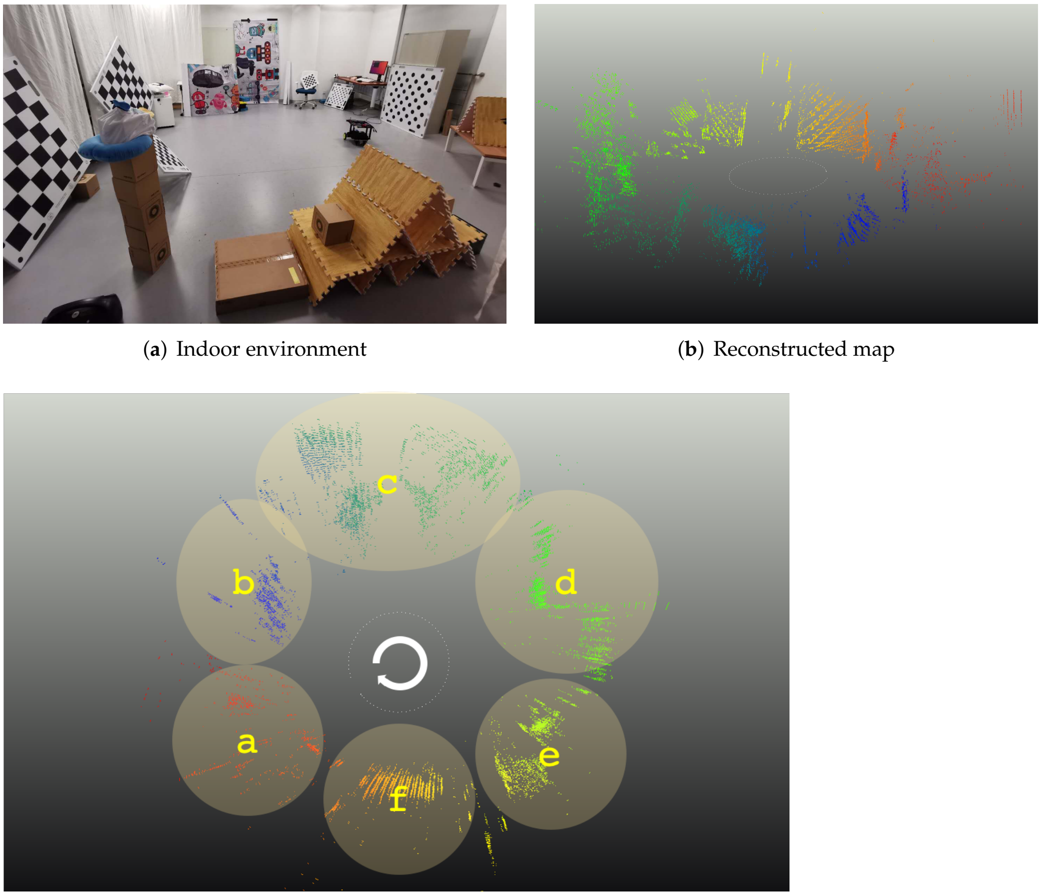
All experiments are conducted on an Intel Core i7 2.4 GHz CPU. The total cumulative processing time for each sequence is summarized in Table 2. It remains below the actual length of each dataset, thus indicating real-time capability.

Table 2. Processing time in seconds (s) for the proposed method.

	Circle	***Str***	***Long1***	***Long2***	***HDR***
Dataset length	14.0 s	10.4 s	43.3 s	40.4 s	17.9 s
Processing time	8.6 s	8.5 s	35.6 s	24.9 s	13.8 s

5.6. Reconstruction Result

Figures 8 and 9 finally visualize reconstruction results of the indoor scene (cf. video in Supplementary Material).Figure 8b,c show a side perspective and a birds-eye view of the final result. The colored semi-dense points represent the reconstructed structure, while the sparse white points in the center denote the discretized trajectory. As can be clearly observed, our method produces a visually reasonable reconstruction similar to what one would obtain using a sparse or semi-dense method on regular images. Further visualisations of re-projected point clouds overlayed onto real images are visualized in Figure 9a–f. The depth of points is indicated by the color, which reaches from red for closer points to blue for far-away points. Note that we clean up isolated noisy points by applying a radius filter. However, no additional depth fusion strategy is applied.

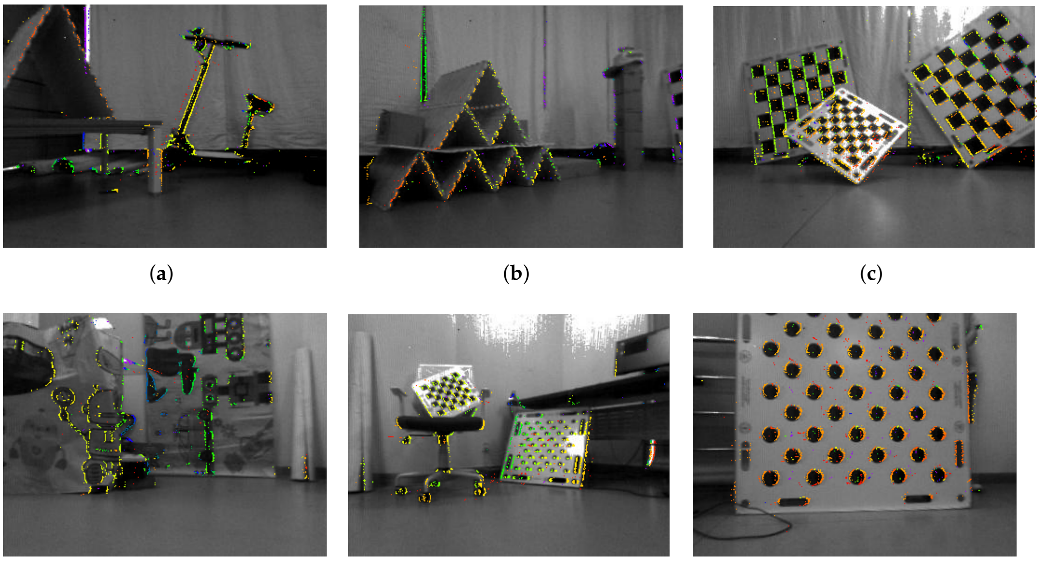


(**a**) Indoor environment

(**b**) Reconstructed map

(**c**) Topdown view

Figure 8. Reconstruction of an indoor scene. (**a**) shows a real image of the environment. (**b**,**c**) are different perspectives onto the reconstructed structure.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 9. (**a**–**f**) are back-projections of the marked structure parts shown in Figure 8 overlaid onto the corresponding images captured under those poses. Warmer colors indicate closer points, while colder colors indicate larger depth.

6. Conclusions

Our main novelty consists of a single, joint objective that optimizes smooth motion directly from events without the need for a prior derivation of 3D structure. This is achieved by constructing a volumetric ray density field, in which we then maximize contrast as a function of smooth motion parameters. As a result, the approach can bootstrap spatial motion in arbitrarily structured environments. The formulation is tested on the important application of ground vehicle motion estimation, and potential advantages under high dynamic motion or challenging illumination conditions are verified. While this is a highly promising result, our next step consists of extending the operation to more dynamic, full 3D motion, which we believe is possible if using the additional input of an IMU.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/s22155687/s1, Video S1: Supplementary video.

Author Contributions: Conceptualization, X.P.; Data curation, L.G. and J.C.; Methodology, Y.W.; Software, J.Y. and K.H.; Supervision, L.K.; Validation, P.W. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the funding sponsored by the Natural Science Foundation of Shanghai (grant number: 22ZR1441300).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J.M. Visual simultaneous localization and mapping: A survey. *Artif. Rev.* **2015**, *43*, 55–81.
2. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332.
3. Brandli, C.; Berner, R.; Yang, M.; Liu, S.-C.; Delbruck, T. A 240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE J. -Solid-State Circuits* **2014**, *49*, 2333–2341.
4. Gallego, G.; Delbruck, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.; Conradt, J.; Daniilidis, K. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach.* **2020**, *44*, 154–180.
5. Gallego, G.; Rebecq, H.; Scaramuzza, D. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3867–3876.
6. Gallego, G.; Gehrig, M.; Scaramuzza, D. Focus is all you need: Loss functions for event-based vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 12280–12289.
7. Stoffregen, T.; Kleeman, L. Event cameras, contrast maximization and reward functions: An analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 12300–12308.
8. Benosman, R.; Clercq, C.; Lagorce, X.; Ieng, S.; Bartolozzi, C. Event-based visual flow. *IEEE Trans. Neural Networks Learn. Syst.* **2013**, *25*, 407–417.
9. Zhu, A.Z.; Atanasov, N.; Daniilidis, K. Event-based feature tracking with probabilistic data association. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4465–4470.
10. Stoffregen, T.; Kleeman, L. Simultaneous optical flow and segmentation (sofas) using dynamic vision sensor. In Proceedings of the 2017 Australasian Conference on Robotics and Automation (ACRA), Sydney, Australia, 11–13 December 2017; pp. 52–61.
11. Ye, C.; Mitrokhin, A.; Parameshwara, C.; Fermüller, C.; Yorke, J.A.; Aloimonos, Y. Unsupervised learning of dense optical flow and depth from sparse event data. *arXiv* **2018**, arXiv:1809.08625.
12. Zhu, A.Z.; Yuan, L.; Chaney, K.; Daniilidis, K. Unsupervised event-based learning of optical flow, depth, and egomotion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 989–997.
13. Zhu, A.Z.; Yuan, L.; Chaney, K.; Daniilidis, K. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv* **2018**, arXiv:1802.06898
14. Stoffregen, T.; Gallego, G.; Drummond, T.; Kleeman, L.; Scaramuzza, D. Event-based motion segmentation by motion compensation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 7244–7253.

15. Mitrokhin, A.; Fermüller, C.; Parameshwara, C.; Aloimonos, Y. Event-based moving object detection and tracking. In Proceedings of the RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–9.
16. Zhou, Y.; Gallego, G.; Lu, X.; Liu, S.; Shen, S. Event-based motion segmentation with spatio-temporal graph cuts. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–13. doi:10.1109/TNNLS.2021.3124580.
17. Rebecq, H.; Gallego, G.; Mueggler, E.; Scaramuzza, D. Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time. *Int. J. Comput. Vis.* **2018**, *126*, 1394–1414.
18. Zhu, A.Z.; Chen, Y.; Daniilidis, K. Realtime time synchronized event-based stereo. In *European Conference on Computer Vision*, Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 438–452.
19. Zhou, Y.; Gallego, G.; Rebecq, H.; Kneip, L.; Li, H.; Scaramuzza, D. Semi-dense 3d reconstruction with a stereo event camera. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 235–251.
20. Gallego, G.; Lund, J.E.; Mueggler, E.; Rebecq, H.; Delbruck, T.; Scaramuzza, D. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2402–2412.
21. Gallego, G.; Scaramuzza, D. Accurate angular velocity estimation with an event camera. *IEEE Robot. Autom. Lett.* **2017**, *2*, 632–639.
22. Bryner, S.; Gallego, G.; Rebecq, H.; Scaramuzza, D. Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), IEEE, Xi'an, China, 30 May–5 June 2021, 2019; pp. 325–331.
23. Peng, X.; Wang, Y.; Kneip, L. Globally optimal event camera motion estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
24. Liu, D.; Parra, A.; Chin, T.-J. Globally optimal contrast maximisation for event-based motion estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6349–6358.
25. Liu, D.; Parra, A.; Chin, T.-J. Spatiotemporal registration for event-based visual odometry. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, virtual, 19–25 June 2021; pp. 4937–4946.
26. Jiao, J.; Huang, H.; Li, L.; He, Z.; Zhu, Y.; Liu, M. Comparing Representations in Tracking for Event Camera-based SLAM. In Proceeding of the IEEE/CVF Conference On Computer Vision And Pattern Recognition Workshops (CVPRW), virtual, 19–25 June 2021; pp. 1369–1376.
27. Zhou, Y.; Gallego, G.; Shen, S. Event-based stereo visual odometry. *IEEE Trans. Robot.* **2021**, *37*, 1433–1450.
28. Kim, H.; Leutenegger, S.; Davison, A.J. Real-time 3d reconstruction and 6-dof tracking with an event camera. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 349–364.
29. 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.* **2016**, *2*, 593–600.
30. Zhu, D.; Xu, Z.; Dong, J.; Ye, C.; Hu, Y.; Su, H.; Liu, Z.; Chen, G. Neuromorphic visual odometry system for intelligent vehicle application with bio-inspired vision sensor. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 2225–2232.
31. Furgale, P.; Tong, C.H.; Barfoot, T.D.; Sibley, G. Continuous-time batch trajectory estimation using temporal basis functions. *Int. J. Robot. Res.* **2015**, *34*, 1688–1710, .
32. Nistér, D.; Naroditsky, O.; Bergen, J. Visual odometry. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004; pp. 652–659.
33. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262.
34. Scaramuzza, D.; Fraundorfer, F.; Siegwart, R. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 4293–4299.
35. Huang, K.; Wang, Y.; Kneip, L. Motion estimation of non-holonomic ground vehicles from a single feature correspondence measured over n views. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Long Beach, CA, USA, 16–20 June 2019; pp. 12706–12715.
36. Piegl, L.; Tiller, W., *The NURBS Book*; Springer Science & Business Media: Berlin, Germany, 2012.
37. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res. (IJRR)* **2013**, *32*, 1231–1237.
38. Gehrig, D.; Gehrig, M.; Hidalgo-Carrió, J.; Scaramuzza, D. Video to events: Recycling video datasets for event cameras. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, Seattle, WA, USA, 14–19 June 2020.