Event-based Stereo Visual Odometry

Yi Zhou, Guillermo Gallego, Shaojie Shen

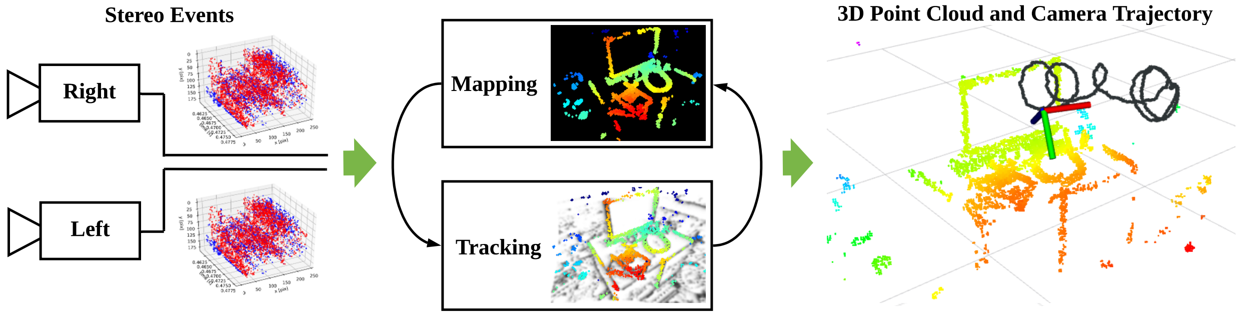


Fig. 1: The proposed system takes as input the asynchronous data acquired by a pair of event cameras in stereo configuration (Left) and recovers the motion of the cameras as well as a semi-dense map of the scene (Right). It exploits spatio-temporal consistency of the events across the image planes of the cameras to solve both localization (i.e., 6-DoF tracking) and mapping (i.e., depth estimation) subproblems of visual odometry (Middle). The system runs in real time on a standard CPU.

***Abstract*—Event-based cameras are bio-inspired vision sensors whose pixels work independently from each other and respond asynchronously to brightness changes, with microsecond resolution. Their advantages make it possible to tackle challenging scenarios in robotics, such as high-speed and high dynamic range scenes. We present a solution to the problem of visual odometry from the data acquired by a stereo event-based camera rig. Our system follows a parallel tracking-and-mapping approach, where novel solutions to each subproblem (3D reconstruction and camera pose estimation) are developed with two objectives in mind: being principled and efficient, for real-time operation with commodity hardware. To this end, we seek to maximize the spatio-temporal consistency of stereo event-based data while using a simple and efficient representation. Specifically, the mapping module builds a semi-dense 3D map of the scene by fusing depth estimates from multiple viewpoints (obtained by spatio-temporal consistency) in a probabilistic fashion. The tracking module recovers the pose of the stereo rig by solving a registration problem that naturally arises due to the chosen map and event data representation. Experiments on publicly available datasets and on our own recordings demonstrate the versatility of the proposed method in natural scenes with general 6-DoF motion. The system successfully leverages the advantages of event-based cameras to perform visual odometry in challenging illumination conditions, such as low-light and high dynamic range, while running in real-time on a standard CPU. We release the software and dataset under an open source licence to foster research in the emerging topic of event-based SLAM.**

Multimedia Material

Supplemental video: https://youtu.be/3CPPs1gz04k
Code: https://github.com/HKUST-Aerial-Robotics/ESVO.git

Yi Zhou and Shaojie Shen are with the Robotic Institute, the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology, Hong Kong, China. E-mail: {eeyzhou, eeshaojie}@ust.hk. Guillermo Gallego is with the Technische Universität Berlin and the Einstein Center Digital Future, Berlin, Germany.

This work was supported by the HKUST Institutional Fund. (Corresponding author: Yi Zhou.)

I. Introduction

Event cameras are novel bio-inspired sensors that report the pixel-wise intensity changes asynchronously at the time they occur, called "events" [1], [2]. Hence, they do not output grayscale images nor they operate at a fixed rate like traditional cameras. This asynchronous and differential principle of operation suppresses temporal redundancy and therefore reduces power consumption and bandwidth. Endowed with microsecond resolution, event cameras are able to capture high-speed motions, which would cause severe motion blur on standard cameras. In addition, event cameras have a very high dynamic range (HDR) (e.g., 140 dB compared to 60 dB of standard cameras), which allows them to be used on broad illumination conditions. Hence, event cameras open the door to tackle challenging scenarios in robotics such as high-speed and/or HDR feature tracking [3], [4], [5], camera tracking [6], [7], [8], [9], control [10], [11], [12] and Simultaneous Localization and Mapping (SLAM) [13], [14], [15], [16].

The main challenge in robot perception with these sensors is to design new algorithms that process the unfamiliar stream of intensity changes ("events") and are able to unlock the camera's potential [2]. Some works have addressed this challenge by combining event cameras with additional sensors, such as depth sensors [17] or standard cameras [18], [19], to simplify the perception task at hand. However, this introduced bottlenecks due to the combined system being limited by the lower speed and dynamic range of the additional sensor.

In this paper we tackle the problem of stereo visual odometry (VO) with event cameras in natural scenes and arbitrary 6-DoF motion. To this end, we design a system that processes a stereo stream of events in real time and outputs the ego-motion of the stereo rig and a map of the 3D scene (Fig. 1). The proposed system essentially follows a parallel tracking-and-mapping philosophy [20], where the main modules operate in an interleaved fashion estimating the ego-motion and the 3D structure, respectively (a more detailed overview of the system is given in Fig. 2). In summary, our *contributions* are:

- A novel mapping method based on the optimization of an objective function designed to measure spatio-temporal consistency across stereo event streams (Section IV-A).
- A fusion strategy based on the probabilistic characteristics of the estimated inverse depth to improve density and accuracy of the recovered 3D structure (Section IV-B).
- A novel camera tracking method based on $3\mathrm{D}-2\mathrm{D}$ registration that leverages the inherent distance field nature of a compact and efficient event representation (Section V).
- An extensive experimental evaluation, on publicly available datasets and our own, demonstrating that the system is computationally efficient, running in real time on a standard CPU (Section VI). The software, design of the stereo rig and datasets used have been open sourced.

This paper significantly extends and differs from our previous work [21], which only tackled the stereo mapping problem. Details of the differences are given at the beginning of Section IV. In short, we have completely reworked the mapping part due to the challenges faced for real-time operation.

Stereo visual odometry (VO) is a paramount task in robot navigation, and we aim at bringing the advantages of event-based vision to the application scenarios of this task. To the best of our knowledge, this is the first published stereo VO algorithm for event cameras (see Section II).

Outline: The rest of the paper is organized as follows. Section II reviews related work in 3D reconstruction and ego-motion estimation with event cameras. Section III provides an overview of the proposed event-based stereo VO system, whose mapping and tracking modules are described in Sections IV and V, respectively. Section VI evaluates the proposed system extensively on publicly available data, demonstrating its effectiveness. Finally, Section VII concludes the paper.

II. Related Work

Event-based stereo VO is related to several problems in structure and motion estimation with event cameras. These have been intensively researched in recent years, notably since event cameras such as the Dynamic Vision Sensor (DVS) [1] became commercially available (2008). Here we review some of those works. A more extensive survey is provided in [2].

A. Event-based Depth Estimation (3D Reconstruction)

a) Instantaneous Stereo: The literature on event-based stereo depth estimation is dominated by methods that tackle the problem of 3D reconstruction using data from a pair of synchronized and rigidly attached event cameras during a very short time (ideally, on a per-event basis). The goal is to exploit the advantages of event cameras to reconstruct dynamic scenes at very high speed and with low power. These works [22], [23], [24] typically follow the classical two-step paradigm of finding epipolar matches and then triangulating the 3D point [25]. Event matching is often solved by enforcing several constraints, including temporal coherence (e.g., simultaneity) of events across both cameras. For example, [26] combines epipolar constraints, temporal inconsistency, motion inconsistency and photometric error (available only from grayscale events given by ATIS cameras [27]) into an objective function to compute the best matches. Other works, such as [28], [29], [30], extend cooperative stereo [31] to the case of event cameras [32]. These methods work well with static cameras in uncluttered scenes, so that event matches are easy to find among few moving objects.

b) Monocular: Depth estimation with a single event camera has been shown in [13], [33], [34]. Since instantaneous depth estimation is brittle in monocular setups, these methods tackle the problem of depth estimation for VO or SLAM: hence, they require knowledge of the camera motion to integrate information from the events over a longer time interval and be able to produce a semi-dense 3D reconstruction of the scene. Event simultaneity does not apply, hence temporal coherence is much more difficult to exploit to match events across time and therefore other techniques are devised.

B. Event-based Camera Pose Estimation

Research on event-based camera localization has progressed by addressing scenarios of increasing complexity. From the perspective of the type of motion, constrained motions, such as pure rotation [35], [36], [8], [37] or planar motion [38], [39] have been studied before investigating the most general case of arbitrary 6-DoF motion. Regarding the type of scenes, solutions for artificial patterns, such as high-contrast textures and/or structures (line-based or planar maps) [38], [40], [6], have been proposed before solving more difficult cases: natural scenes with arbitrary 3D structure and photometric variations [36], [7], [9].

From the methodology point of view, probabilistic filters [38], [36], [7] provide event-by-event tracking updates thus achieving minimal latency (µs), whereas frame-based techniques (often non-linear optimization) trade off latency for more stable and accurate results [8], [9].

C. Event-based VO and SLAM

a) Monocular: Two methods stand out as solving the problem of monocular event-based VO for 6-DoF motions in natural 3D scenes. The approach in [13] simultaneously runs three interleaved Bayesian filters, which estimate image intensity, depth and camera pose. The recovery of intensity information and depth regularization make the method computationally intensive, thus requiring dedicated hardware (GPU) for real-time operation. In contrast, [14] proposes a geometric approach based on the semi-dense mapping technique in [33] (focusing events [41]) and an image alignment tracker that works on event images. It does not need to recover absolute intensity and runs in real time on a CPU. So far, none of these methods have been open sourced to the community.

b) Stereo: The authors are aware of the existence of a stereo VO demonstrator built by event camera manufacturer [42]; however its details have not been disclosed. Thus, to the best of our knowledge, this is the first published stereo VO algorithm for event cameras. In the experiments (Section VI)

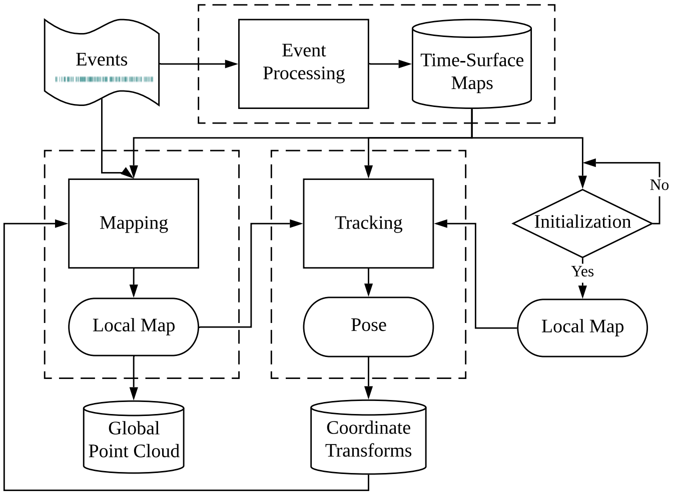


Fig. 2: *Proposed system flowchart*. Core modules of the system, including event pre-processing (Section III-A), mapping (Section IV) and tracking (Section V) are marked with dashed rectangles. The only input to the system comprises raw stereo events from calibrated cameras, and the output consists of camera rig poses and a point cloud of 3D scene edges.

we compare the proposed algorithm against an iterative-closest point (ICP) method, which is the underlying technology of the above demonstrator.

Our method builds upon our previous mapping work [21], reworked, and a novel camera tracker that re-utilizes the data structures used for mapping. For mapping, we do not follow the classical paradigm of event matching plus triangulation, but rather a forward-projection approach that enables depth estimation without establishing event correspondences explicitly. Instead, we reformulate temporal coherence using the compact representation of space-time provided by time surfaces [43]. For tracking, we use non-linear optimization on time surfaces, thus resembling the frame-based paradigm, which trades off latency for efficiency and accuracy. Like [14] our system does not need to recover absolute intensity and is efficient, able to operate in real time without dedicated hardware (GPU); standard commodity hardware such as a laptop's CPU suffices.

III. System Overview

The proposed stereo VO system takes as input only raw events from calibrated cameras and manages to simultaneously estimate the pose of the stereo event camera rig while reconstructing the environment using semi-dense depth maps. An overview of the system is given in Fig. 2, in which the core modules are highlighted with dashed lines. Similarly to classical SLAM pipelines [20], the core of our system consists of two interleaved modules: mapping and tracking. Additionally, there is a third key component: event pre-processing.

Let us briefly introduce the functionality of each module and explain how they work cooperatively. First of all, the event processing module generates an event representation, called time-surface maps (or simply "time surfaces", see Section III-A), used by the other modules. Theoretically, these time maps are updated asynchronously, with every incoming event (µs resolution). However, considering that a single event does not bring much information to update the state of a VO system, the stereo time surfaces are updated at a more practical rate: e.g., at the occurrence of a certain number of events or at a fixed rate (*e.g.*, 100 Hz in our implementation). A short history of time surfaces is stored in a database (top right of Fig. 2) for access by other modules. Secondly, after an initialization phase (see below), the tracking module continuously estimates the pose of the left event camera with respect to the local map. The resulting pose estimates are stored in a database of coordinate transforms (*e.g.*, TF in ROS [44]), which is able to return the pose at any given time by interpolation in SE(3). Finally, the mapping module takes the events, time surfaces and pose estimates and refreshes a local map (represented as a probabilistic semi-dense depth map), which is used by the tracking module. The local maps are stored on a database of global point cloud for visualization.

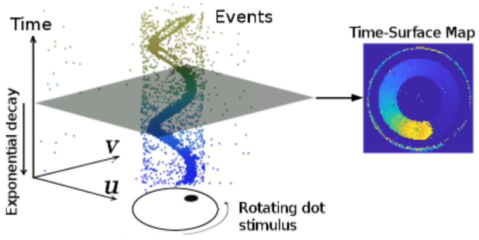


Fig. 3: *Event Representation*. Left: Output of an event camera when viewing a rotating dot. Right: Time-surface map (1) at time t, $\mathcal{T}(\mathbf{x}, t)$, which essentially measures how far in time (with respect to t) the last event spiked at each pixel $\mathbf{x} = (u, v)^\top$. The brighter the color, the more recently the event was triggered. Figure adapted from [46].

Initialization: To bootstrap the system, we apply a stereo method (a modified SGM method [45], as discussed in Section VI-C) that provides a coarse initial map. This enables the tracking module to start working while the mapping module is also started and produces a better semi-dense inverse depth map (more accurate and dense).

A. Event Representation

As illustrated in Fig. 3-left, the output of an event camera is a stream of asynchronous events. Each event $e_k = (u_k, v_k, t_k, p_k)$ consists of the space-time coordinates where an intensity change of predefined size happened and the sign (polarity $p_k \in \{+1, -1\}$) of the change.

The proposed system (Fig. 2) uses both individual events and an alternative representation called Time Surface (Fig. 3-right). A time surface (TS) is a 2D map where each pixel stores a single time value, e.g., the timestamp of the last event at that pixel [47]. Using an exponential decay kernel [43] TSs emphasize recent events over past events. Specifically, if t_{last} is the timestamp of the last event at each pixel coordinate $\mathbf{x} = (u, v)^\top$ the TS at time $t \geq t_{\text{last}}(\mathbf{x})$ is defined by

$$\mathcal{T}(\mathbf{x}, t) \doteq \exp\left(-\frac{t - t_{\text{last}}(\mathbf{x})}{\eta}\right), \quad (1)$$

where η, the decay rate parameter, is a small constant number (e.g., $30\,\mathrm{ms}$ in our experiments). As shown in Fig. 3-right, TSs represent the recent history of moving edges in a compact way (using a 2D grid). A discussion of several event representations (voxel grids, event frames, etc.) can be found in [2], [48].

We use TSs because they are memory- and computationally efficient, informative (edges are the most descriptive regions of a scene for SLAM), interpretable and because they have proven to be successful for motion (optical flow) [47], [49], [50] and depth estimation [21]. Specifically, for mapping (Section IV) we propose to do pixel-wise comparisons on a stereo pair of TSs [21] as a replacement for the photo-consistency criterion of standard cameras [51]. Since TSs encode temporal information, comparison of TS patches amounts to measuring *spatio-temporal consistency* over small data volumes on the image planes. For tracking (Section V), we exploit the fact that a TS acts like an anisotropic distance field [52] defined by the most recent edge locations to register events with respect to the 3D map. For convenient visualization and processing, (1) is rescaled from $[0,1]$ to the range $[0,255]$.

IV. Mapping: Stereo Depth Estimation by Spatio-temporal Consistency and Fusion

The mapping module consists of two steps: (i) computing depth estimates of events (Section IV-A and Algorithm 1) and (ii) fusing such depth estimates into an accurate and populated depth map (Section IV-B). An overview of the mapping module is provided on Fig. 7(a).

The underlying principles often leveraged for event-based stereo depth estimation are *event co-occurrence* and the epipolar constraint, which simply state that a 3D edge triggers two simultaneous events on corresponding epipolar lines of both cameras. However, as shown in [53], [28], stereo temporal coincidence does not strictly hold at the pixel level because of delays, jitter and pixel mismatch (e.g., differences in event firing rates). Hence, we define a stereo temporal consistency criterion across space-time neighborhoods of the events rather than by comparing the event timestamps at two individual pixels. Moreover we represent such neighborhoods using time surfaces (due to their properties and natural interpretation as temporal information, Section III-A) and cast the stereo matching problem as the minimization of such a criterion.

The above two-step process and principle was used in our previous work [21]. However, we contribute some fundamental differences guided by a real-time design goal: (i) The objective function is built only on the temporal inconsistency across one stereo event time-surface map (Section IV-A) rather than over longer time spans (thus, the proposed approach becomes closer to the strategy in [51] than that in [54]). This needs to be coupled with (ii) a novel depth-fusion algorithm (Section IV-B), which is provided after investigation of the probabilistic characteristics of the temporal residuals and inverse depth estimates, to enable accurate depth estimation over longer time spans than a single stereo time-surface map. (iii) The initial guess to minimize the objective is determined using a block matching method, which is more efficient than brute-force search [21]. (iv) Finally, on a more technical note, non-negative per-patch residuals [21] are replaced with signed per-pixel residuals, which guarantee non-zero Jacobians for valid uncertainty propagation and fusion.

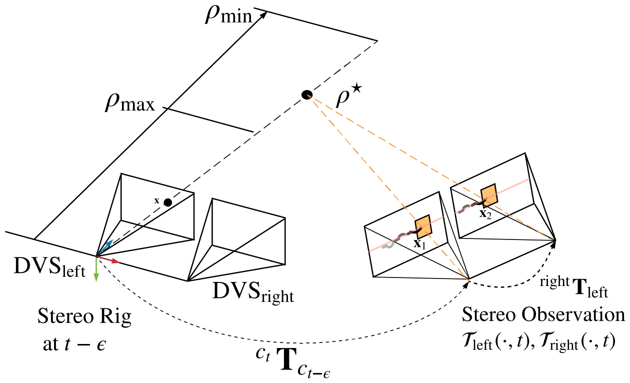


Fig. 4: *Mapping. Geometry of (inverse) depth estimation.* 3D points compatible with an event $e=(\mathbf{x},t-\epsilon,p)$ on the left camera are parametrized by inverse depth ρ on the viewing ray through pixel $\mathbf{x}$ at time $t-\epsilon$. The true location of the 3D point that triggered the event corresponds to the value $\rho^\star$ that maximizes the temporal consistency across the stereo observation $\big(\mathcal{T}_{\text{left}}(\cdot,t),\mathcal{T}_{\text{right}}(\cdot,t)\big)$. A search interval $[\rho_{\min},\rho_{\max}]$ is defined to bound the optimization along the viewing ray.

A. Inverse Depth Estimation for an Event

We follow an energy optimization framework to estimate the inverse depth of events occurred before the stereo observation at time t. Fig. 4 illustrates the geometry of the proposed approach. Without loss of generality, we parametrize inverse depth using the left camera. A *stereo observation* at time t refers to a pair of time surfaces $\big(\mathcal{T}_{\text{left}}(\cdot,t),\mathcal{T}_{\text{right}}(\cdot,t)\big)$ created using (1) (see also Figs. 5(c) and 5(d)).

1) Problem Statement: The inverse depth $\rho^\star \doteq 1/Z^\star$ of an event $e_{t-\epsilon}\equiv(\mathbf{x},t-\epsilon,p)$ (with $\epsilon\in[0,\delta t]$) on the left image plane, which follows a camera trajectory $\mathbf{T}_{t-\delta t:t}$, is estimated by optimizing the objective function:

$$\rho^\star = \arg\min_{\rho} C(\mathbf{x},\rho,\mathcal{T}_{\text{left}}(\cdot,t),\mathcal{T}_{\text{right}}(\cdot,t),\mathbf{T}_{t-\delta t:t}) \tag{2}$$

$$C \doteq \sum_{\mathbf{x}_{1,i}\in W_1,\mathbf{x}_{2,i}\in W_2} r_i^2(\rho). \tag{3}$$

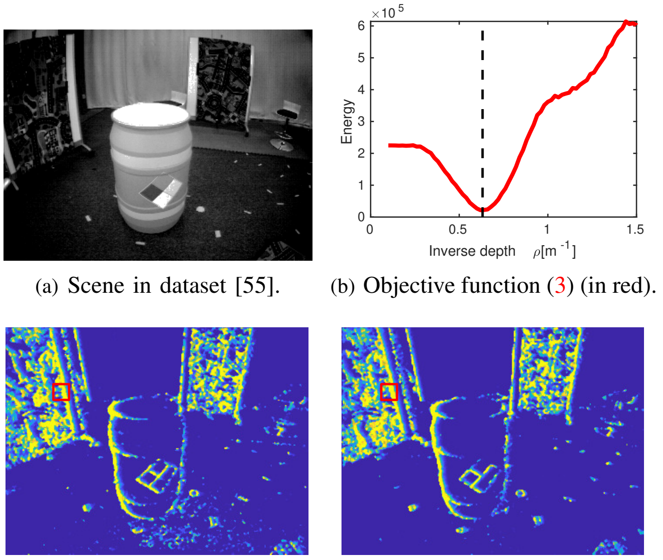
The residual

$$r_i(\rho) \doteq \mathcal{T}_{\text{left}}(\mathbf{x}_{1,i},t) - \mathcal{T}_{\text{right}}(\mathbf{x}_{2,i},t) \tag{4}$$

denotes the temporal difference between two corresponding pixels $\mathbf{x}_{1,i}$ and $\mathbf{x}_{2,i}$ inside neighborhoods (i.e., patches) W_1 and W_2, centered at $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. Assuming the calibration (intrinsic and extrinsic parameters) is known and the pose of the left event camera at any given time within $[t-\delta t,t]$ is available (e.g., via interpolation of $\mathbf{T}_{t-\delta t:t}$ in SE(3)), the points $\mathbf{x}_1$ and $\mathbf{x}_2$ are given by

$$\mathbf{x}_1 = \pi\big({}^{c_t}\mathbf{T}_{c_{t-\epsilon}}\cdot\pi^{-1}(\mathbf{x},\rho_k)\big), \tag{5a}$$

$$\mathbf{x}_2 = \pi\big({}^{\text{right}}\mathbf{T}_{\text{left}}\cdot{}^{c_t}\mathbf{T}_{c_{t-\epsilon}}\cdot\pi^{-1}(\mathbf{x},\rho_k)\big). \tag{5b}$$



(a) Scene in dataset [55].

(b) Objective function (3) (in red).

(c) Time surface (left DVS).

(d) Time surface (right DVS).

Fig. 5: *Mapping. Spatio-temporal consistency*. (a) An intensity frame shows the visual appearance of the scene. Our method does not use intensity frames; only events. (b) The objective function measures the inconsistency between the motion history content (time surfaces (c) and (d)) across left-right retinas, thus replacing the photometric error in frame-based stereo. Specifically, (b) depicts the variation of $C(\mathbf{x}, \rho, \mathcal{T}_{\text{left}}(\cdot, t), \mathcal{T}_{\text{right}}(\cdot, t), \mathbf{T}_{t-\delta t:t})$ with inverse depth ρ. The vertical dashed line (black) indicates the ground truth inverse depth. (c)-(d) show the time surfaces of the stereo event camera at the observation time, $\mathcal{T}_{\text{left}}(\cdot, t)$, $\mathcal{T}_{\text{right}}(\cdot, t)$, where the pixels for measuring the temporal residual in (b) are enclosed in red.

Note that each event is warped using the camera pose at the time of its timestamp. The function $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ projects a 3D point onto the camera's image plane, while its inverse function $\pi^{-1} : \mathbb{R}^2 \to \mathbb{R}^3$ back-projects a pixel into 3D space given the inverse depth ρ. ${}^{\text{right}}\mathbf{T}_{\text{left}}$ denotes the transformation from the left to the right event camera, which is constant. All event coordinates $\mathbf{x}$ are undistorted and stereo-rectified using the known calibration of the cameras.

Fig. 5 shows an example of the objective function from a real stereo event-camera sequence [55] that has ground truth depth. It confirms that the proposed objective function (3) does lead to the optimal depth for a generic event. It visualizes the profile of the objective function for the given event (Fig. 5(b)) and the stereo observation used (Figs. 5(c) and 5(d)).

Remark on Modeling Data Association: Note that our approach differs from classical two-step event-processing methods [22], [23], [24], [26] that solve the stereo matching problem first and then triangulate the 3D point. Such two-step approaches work in a "back-projection" fashion, mapping 2D event measurements into 3D space. In contrast, our approach combines matching and triangulation in a single step, operating in a forward-projection manner (3D→2D). As shown in Fig. 4, an inverse depth hypothesis ρ yields a 3D point, $\pi^{-1}(\mathbf{x}, \rho)$, whose projection on both stereo image planes at time t gives points $\mathbf{x}_1(\rho)$ and $\mathbf{x}_2(\rho)$ whose neighborhoods are compared in the objective function (3). Hence, an inverse depth hypothesis ρ establishes a candidate stereo event match, and the best match is provided by the ρ that minimizes the objective.

2) Non-Linear Solver for Depth Estimation: The proposed objective function (2)-(3) is optimized using non-linear least squares methods, such as the Gauss-Newton method, which iteratively find the root of the necessary optimality condition

$$\frac{\partial C}{\partial \rho} = 2\mathbf{J}^\top \mathbf{r} = 0, \tag{6}$$

where $\mathbf{r} \doteq (r_1, r_2, ..., r_{N^2})^\top$, N^2 is the size of the patch, and $\mathbf{J} = \partial \mathbf{r}/\partial \rho$. Substituting the linearization of $\mathbf{r}$ given by Taylor's formula, $\mathbf{r}(\rho + \varDelta\rho) \approx \mathbf{r}(\rho) + \mathbf{J}(\rho)\varDelta\rho$, we arrive at the normal equation $\mathbf{J}^\top \mathbf{J} \varDelta\rho = -\mathbf{J}^\top \mathbf{r}$, where $\mathbf{J}^\top \mathbf{J} = \|\mathbf{J}\|^2$ and we omitted the dependency of $\mathbf{J}$ and $\mathbf{r}$ with ρ for succinctness. The inverse depth solution ρ is iteratively updated by

$$\rho \leftarrow \rho + \varDelta\rho \quad \text{with} \quad \varDelta\rho = -(\mathbf{J}^\top \mathbf{r})/\|\mathbf{J}\|^2. \tag{7}$$

Analytical derivatives are used to speed up computations.

3) Initialization of the Non-Linear Solver: Successful convergence of the inverse depth estimator (7) relies on a good initial guess ρ_0. For this, instead of carrying out an exhaustive search over an inverse depth grid [21], we apply a more efficient strategy exploiting the canonical stereo configuration: block matching along epipolar lines of the stereo observation $\big(\mathcal{T}_{\text{left}}(\cdot, t), \mathcal{T}_{\text{right}}(\cdot, t)\big)$ using an integer-pixel disparity grid. That is, we maximize the Zero-Normalized Cross-Correlation (ZNCC) using patch centers $\mathbf{x}'_1 = \mathbf{x}$ (pixel coordinates of event $e_{t-\epsilon}$) and $\mathbf{x}'_2 = \mathbf{x}'_1 + (d, 0)^\top$, where d is the disparity, as an approximation to the true patch centers (5). Note that temporal consistency is slightly violated here because the relative motion ${}^{c_t}\mathbf{T}_{c_{t-\epsilon}}$ in (5), corresponding to the time of the event, $t - \epsilon$, is not compensated for in $\mathbf{x}'_1, \mathbf{x}'_2$. Nevertheless, this approximation provides a reasonable and efficient initial guess ρ (using d) whose temporal consistency is refined in the subsequent nonlinear optimization.

4) Summary: Inverse depth estimation for a given event on the left camera is summarized in Algorithm 1. The inputs of the algorithm are: the event $e_{t-\epsilon}$ (space-time coordinates), a stereo observation (time surfaces at time t) $\mathcal{T}_{\text{left/right}}(\cdot, t)$, the incremental motion ${}^{c_t}\mathbf{T}_{c_{t-\epsilon}}$ of the stereo rig between the times of the event and the stereo observation, and the constant extrinsic parameters between both event cameras, ${}^{\text{right}}\mathbf{T}_{\text{left}}$. The inverse depth of each event considered is estimated independently; thus computations are parallelizable.

B. Semi-Dense Reconstruction

The 3D reconstruction method presented in Section IV-A (Algorithm 1) produces inverse depth estimates for individual events, and according to the parametrization (Fig. 4), each estimate has a different timestamp. This section develops a probabilistic approach for fusion of inverse depth estimates to produce a semi-dense depth map at the current time (Fig. 7), which is later used for tracking. Depth fusion is crucial since it allows us to refer all depth estimates to a common time, reduces uncertainty of the estimated 3D structure and improves

Algorithm 1 Inverse Depth Estimation

```
Input: event $e_{t-\epsilon}$, stereo event observation $\mathcal{T}_{\text{left}}(\cdot,t)$, $\mathcal{T}_{\text{right}}(\cdot,t)$ and relative transformation ${}^{c_t}\mathbf{T}_{c_{t-\epsilon}}$.
Initialize $\rho$: ZNCC-block matching on $\mathcal{T}_{\text{left}}(\cdot,t)$, $\mathcal{T}_{\text{right}}(\cdot,t)$.
while not converged do
    Compute residuals $\mathbf{r}(\rho)$ in (4).
    Compute Jacobian $\mathbf{J}(\rho)$ (analytical derivatives).
    Update: $\rho \leftarrow \rho + \Delta\rho$, using (7).
end while
return Converged inverse depth $\rho$ (i.e., $\rho^\star$ in Fig. 4).
```

density of the reconstruction. In the following, we first study the probabilistic characteristics of inverse depth estimates (Section IV-B1). Based on these characteristics, the fusion strategy is presented and incrementally applied as depth estimates on new stereo observations are obtained (Sections IV-B2 and IV-B3). Our fused reconstruction approaches a semi-dense level, producing depth values for most edge pixels.

1) Probabilistic Model of Estimated Inverse Depth: We model inverse depth at a pixel on the reference view not with a number ρ but with an actual probability distribution. Algorithm 1 provides an "average" value $\rho^\star$ (also in (2)). We now present how uncertainty (i.e., spread around the average) is propagated and carry out an empirical study to determine the distribution of inverse depth.

In the last iteration of Gauss-Newton's method (7), the inverse depth is updated by

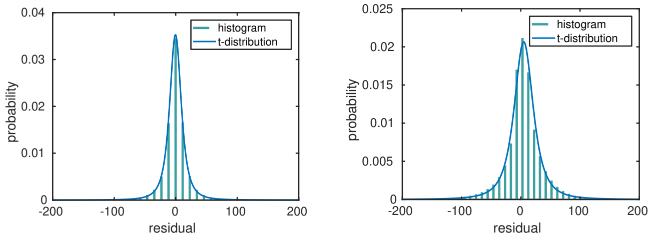
$$\rho^\star \leftarrow \rho + \Delta\rho(\mathbf{r}), \tag{8}$$

where $\Delta\rho$ is a function of the residuals (4) $\mathbf{r}$. Using events, ground truth depth and poses from two datasets we computed a large number of residuals (4) to empirically determine their probabilistic model. Fig. 6 shows the resulting histogram of the residuals r together with a fitted parametric model. In the experiment, we found that a *Student's* t distribution fits the histogram well. The resulting probabilistic model of r is denoted by $r \sim St(\mu_r, s_r^2, \nu_r)$, where μ_r, s_r, ν_r are the model parameters, namely the mean, scale and degree of freedom, respectively. The residual histograms in Fig. 6 seem to be well centered at zero (compared to their spread and to the abscissa range), and so we may set $\mu_r \approx 0$. Parameters of the fitted *Student's* t distributions are given in Table. I for the two sequences used from two different datasets.

Since Generalised Hyperbolic distributions (GH) are closed under affine transformations and the *Student's* t distribution is a particular case of GH, we conclude that the affine transformation $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b}$ (with non-singular matrix $\mathbf{A}$ and vector $\mathbf{b}$) of a random vector $\mathbf{x} \sim St(\boldsymbol{\mu}, \boldsymbol{S}, \nu)$ that follows a multivariate *Student's* t distribution (with mean vector $\boldsymbol{\mu}$, scale matrix $\boldsymbol{S}$ and degree of freedom ν), also follows a *Student's* t distribution [56], in the form $\mathbf{z} \sim St(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{S}\mathbf{A}^\top, \nu)$.

Applying this theorem to (7), with $r \sim St(\mu_r, s_r^2, \nu_r)$ and $\mathbf{A} \equiv -\sum_i J_i/\|\mathbf{J}\|^2$, $\mathbf{b} \equiv \mathbf{0}$, we have that the update $\Delta\rho$ approximately follows a *Student's* t distribution

$$\Delta\rho \sim St\left(-\frac{\sum J_i}{\|\mathbf{J}\|^2}\mu_r, \frac{s_r^2}{\|\mathbf{J}\|^2}, \nu_r\right). \tag{9}$$



(a) *simulation_3planes* [58]. (b) *upenn_flying1* [55].

Fig. 6: Probability distribution (PDF) of the temporal residuals r_i: empirical (green histogram) and *Student's* t fit (blue curve).

TABLE I: Parameters of the fitted *Student's* t distribution.

	Mean (μ)	Scale (s)	DoF (ν)	Std. (σ)
simulation_3planes [58]	-0.423	10.122	2.207	33.040
upenn_flying1 [55]	4.935	17.277	2.182	59.763

Next, applying the theorem to the affine function (8) and assuming $\mu_r \approx 0$ (Fig. 6) we obtain the approximate distribution

$$\rho \sim St\left(\rho^\star, \frac{s_r^2}{\|\mathbf{J}\|^2}, \nu_r\right), \tag{10}$$

with $\mathbf{J} \equiv \mathbf{J}(\rho^\star)$. The resulting variance is given by

$$\sigma_{\rho^\star}^2 = \frac{\nu_r}{\nu_r - 2}\frac{s_r^2}{\|\mathbf{J}\|^2}. \tag{11}$$

Robust Estimation: The obtained probabilistic model can be used for robust inverse depth estimation in the presence of noise and outliers, since the heavy tails of the *Student's* t distribution account for them. To do so, each squared residual in (3) is re-weighted by a factor $\omega(r_i)$, which is a function of the probabilistic model $p(r)$. The resulting optimization problem is solved using the Iteratively Re-weighted Least Squares (IRLS) method, replacing the Gauss-Newton solver in Algorithm 1. Details about the derivation of the weighting function are provided in [57], [52].

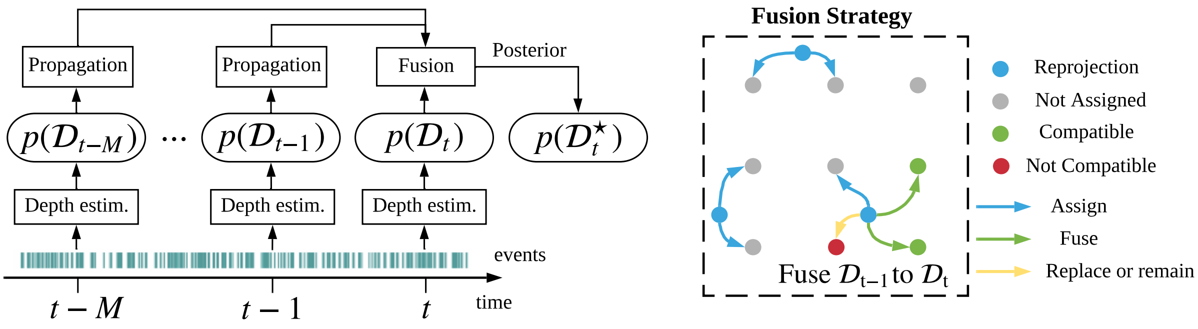
2) Inverse Depth Filters: The fusion of inverse depth estimates from several stereo pairs is performed in two steps. First, inverse depth estimates are propagated from the time of each event to the time of a stereo observation (i.e., the current time). This is simply done similarly to the uncertainty propagation operation in (9)-(10). Second, the propagated inverse depth estimate is fused (updated) with prior estimates at this pixel coordinate. The update step is performed using robust Bayesian filter for *Student's* t distribution. A *Student's* t filter is derived in [59]: given a prior $St(\mu_a, s_a, \nu_a)$ and a measurement $St(\mu_b, s_b, \nu_b)$, the posterior is approximated by a $St(\mu, s, \nu)$ distribution with parameters

$$\nu' = \min(\nu_a, \nu_b), \tag{12a}$$

$$\mu = \frac{s_a^2\mu_b + s_b^2\mu_a}{s_a^2 + s_b^2}, \tag{12b}$$

$$s^2 = \frac{\nu' + \frac{(\mu_a - \mu_b)^2}{s_a^2 + s_b^2}}{\nu' + 1} \cdot \frac{s_a^2 s_b^2}{s_a^2 + s_b^2}, \tag{12c}$$

$$\nu = \nu' + 1. \tag{12d}$$



(a) Flowchart of mapping module.

(b) Depth fusion rules at locations on a 3×3 pixel grid.

Fig. 7: *Mapping module*: (**a**) Stereo observations (time surfaces) are created at selected timestamps $t, \cdots, t-M$ (*e.g.*, 20 Hz) and fed to the mapping module along with the events and camera poses. Inverse depth estimates, represented by probability distributions $p(\mathcal{D}_{t-k})$, are propagated to a common time t and fused to produce an inverse depth map, $p(\mathcal{D}_t^\star)$. We fuse estimates from 20 stereo observations (i.e., $M = 19$) to create $p(\mathcal{D}_t^\star)$. (**b**) Taking the fusion from $t-1$ to t as an example, the fusion rules are indicated in the dashed rectangle, which represents a 3×3 region of the image plane (pixels are marked by a grid of gray dots). A 3D point corresponding to the mean depth of $p(\mathcal{D}_{t-1})$ projects on the image plane at time t at a blue dot. Such a blue dot and $p(\mathcal{D}_{t-1})$ influence (i.e., assign, fuse or replace) the distributions $p(\mathcal{D}_t^\star)$ estimated at the four closest pixels.

3) Probabilistic Inverse Depth Fusion: Assuming the propagated inverse depth follows a distribution $St(\mu_a, s_a^2, \nu_a)$, its corresponding location in the target image plane is typically a non-integer coordinate $\mathbf{x}^{\text{float}}$. Hence, the propagated inverse depth will have an effect on the distributions at the four nearest pixel locations $\{\mathbf{x}_j^{\text{int}}\}_{j=1}^4$ (see Fig. 7(b)). Using $\mathbf{x}_1^{\text{int}}$ as an example, the fusion is performed based on the following rules:

1) If no previous distribution exists at $\mathbf{x}_1^{\text{int}}$, initialize it with $St(\mu_a, s_a^2, \nu_a)$.
2) If there is already an inverse depth distribution at $\mathbf{x}_1^{\text{int}}$, e.g., $St(\mu_b, s_b^2, \nu_b)$, the compatibility between the two inverse depth hypotheses is checked to decide whether they may be fused. The compatibility of two hypotheses ρ_a, ρ_b is evaluated by checking
$$\mu_b - 2\sigma_b \leq \mu_a \leq \mu_b + 2\sigma_b, \tag{13}$$
where $\sigma_b = s_b\sqrt{\nu_b/(\nu_b - 2)}$. If the two hypotheses are compatible, they are fused into a single inverse depth distribution using (12), otherwise the distribution with the smallest variance remains.

The fusion strategy is illustrated in the dashed rectangle of Fig. 7(b) using as example the propagation and update from estimates of $\mathcal{D}_{t-1}$ to the inverse depth map $\mathcal{D}_t$.

4) Summary: Together with the inverse depth estimation introduced in Sec. IV-A, the overall mapping procedure is illustrated in Fig 7. The inverse depth estimation at a given timestamp t, using the stereo observation $\mathcal{T}_{\text{left/right}}(\cdot, t)$ and involved events as input, is tackled via nonlinear optimization (IRLS). Probabilistic estimates at different timestamps are propagated and fused to the inverse depth map distribution at the most recent timestamp t, $p(\mathcal{D}_t^\star)$. The proposed fusion leads to a semi-dense inverse depth map $\mathcal{D}_t^\star$ with reasonably good signal-noise ratio, which is required by the tracking method discussed in the following Section.

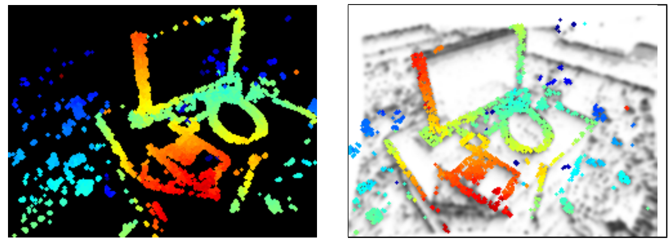
Remarks: All events are involved in creating time surfaces, which are used for tracking and mapping. However, depth is not estimated for every event because it is expensive and we aim at achieving real-time operation with limited computational resources (see Section VI-F).

The number of fused stereo observations, $M + 1 = 20$ in Fig. 7, was determined empirically as a sensible choice for having a good density of the semi-dense depth map in most sequences tested (Section VI). A more theoretical approach would be to have an adaptive number based on statistical criteria, such as the apparent density of points or the decrease of uncertainty in the fused depth, but this is left as future work.

V. Camera Tracking

Let us now present the tracking module in Fig. 2, which takes events and a local map as input and computes the pose of the stereo rig with respect to the map. In principle, each event has a different timestamp and hence also a different camera pose [16] as the stereo rig moves. Since it is typically not necessary to compute poses with microsecond resolution, we consider the pose of a stereo observation (*i.e.*, time surfaces).

Two approaches are now considered before presenting our solution. (i) Assuming a semi-dense inverse depth map is available in a reference frame and a subsequent stereo observation is temporally (and thus spatially) close to the reference frame, the relative pose (between the reference frame and the stereo observation) could be characterized as being the one that, transferring the depth map to both left and right frames of the stereo observation, yields minimal spatio-temporal inconsistency. However, this characterization is only a necessary condition for solving the tracking problem rather than a sufficient one. The reason is that a wrong relative pose might transfer the semi-dense depth map to the "blank" regions of both left and right time surfaces, which would produce



(a) Depth map in the reference viewpoint with known pose.

(b) Warped depth map overlaid on the time surface negative at the current time.

Fig. 8: *Tracking*. The point cloud recovered from the inverse depth map in (a) is warped to the time surface negative at the current time (b) using the estimated relative pose. The result (b) is a good alignment between the projection of the point cloud and the minima (dark areas) of the time surface negative.

an undesired minimum. (ii) An alternative approach to the spatio-temporal consistency criterion would be to consider only the left time surface of the stereo observation (since the right camera is rigidly attached) and use the edge-map alignment method from the monocular system [14]. However, this requires the creation of additional event images.

Instead, our solution consists of taking full advantage of the time surfaces already defined for mapping. To this end we present a novel tracking method based on global image-like registration using time surface "negatives". It is inspired by an edge-alignment method for RGB-D cameras using distance fields [52]. In the following, we intuitively and formally define the tracking problem (Sections V-A and V-B), and solve it using the forward compositional Lucas-Kanade method [60] (Section V-C). Finally, we show how to improve tracking robustness while maintaining a high throughput (Section V-D).

A. Exploiting Time Surfaces as Distance Fields

Time surfaces (TS) (Section III-A) encode the motion history of the edges in the scene. Large values of the TS (1) correspond to recently triggered events, i.e., the current location of the edge. Typically those large values have a ramp on one side (signaling the previous locations of the edge) and a "cliff" on the other one. This can be interpreted as an anisotropic distance field: following the ramp, one may smoothly reach the current location of the edge. Indeed, defining the "negative" (as in image processing) of a TS $\mathcal{T}(\mathbf{x},t)$ by

$$\bar{\mathcal{T}}(\mathbf{x},t) = 1 - \mathcal{T}(\mathbf{x},t), \tag{14}$$

allows us to interpret the small values as the current edge location and the ramps as a distance field to the edge. This negative transformation also allows us to formulate the registration problem as a minimization one rather than a maximization one. Like the TS, (14) is rescaled to the range $[0, 255]$.

The essence of the proposed tracking method is to align the dark regions of the TS negative and the support of the inverse depth map when warped to the TS frame by a candidate pose. Thus the method is posed as an image-like alignment method, with the images representing time information and the scene edges being represented by "zero time". A successful tracking example showing edge-map alignment is given in Fig. 8(b). Building on the findings of semi-dense direct tracking for frame-based cameras [51], we only use the left TS for tracking because incorporating the right TS does not significantly increase accuracy while it doubles the computational cost.

B. Tracking Problem Statement

More specifically, the problem is formulated as follows. Let $\mathcal{S}^{\mathcal{F}_\text{ref}} = \{\mathbf{x}_i\}$ be a set of pixel locations with valid inverse depth ρ_i in the reference frame $\mathcal{F}_\text{ref}$ (i.e., the support of the semi-dense depth map $\mathcal{D}^{\mathcal{F}_\text{ref}} \equiv \mathcal{D}^\star$). Assuming the TS negative at time k is available, denoted by $\bar{\mathcal{T}}_\text{left}(\cdot, k)$, the goal is to find the pose T such that the support of the warped semi-dense map $T(\mathcal{S}^{\mathcal{F}_\text{ref}})$ aligns well with the minima of $\bar{\mathcal{T}}_\text{left}(\cdot, k)$, as shown in Fig. 8. The overall objective of the registration is to find

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \sum_{\mathbf{x}\in\mathcal{S}^{\mathcal{F}_\text{ref}}} \left(\bar{\mathcal{T}}_\text{left}(W(\mathbf{x},\rho;\boldsymbol{\theta}),k)\right)^2, \tag{15}$$

where the warping function

$$W(\mathbf{x},\rho;\boldsymbol{\theta}) \doteq \pi_\text{left}(T(\pi_\text{ref}^{-1}(\mathbf{x},\rho), G(\boldsymbol{\theta}))), \tag{16}$$

transfers points from $\mathcal{F}_\text{ref}$ to the current frame. It consists of a chain of transformations: back-projection from $\mathcal{F}_\text{ref}$ into 3D space given the inverse depth, change of coordinates in space (using candidate motion parameters), and perspective projection onto the current frame. The function $G(\boldsymbol{\theta}) : \mathbb{R}^6 \to \mathrm{SE}(3)$ gives the transformation matrix corresponding to the motion parameters $\boldsymbol{\theta} \doteq (\mathbf{c}^\top, \mathbf{t}^\top)^\top$, where $\mathbf{c} = (c_1, c_2, c_3)^\top$ are the Cayley parameters [61] for orientation $\mathtt{R}$, and $\mathbf{t} = (t_x, t_y, t_z)^\top$ is the translation. The function $\pi_\text{ref}^{-1}(\cdot)$ back-projects a pixel $\mathbf{x}$ into space using the known inverse depth ρ, while $\pi_\text{left}(\cdot)$ projects the transformed space point onto the image plane of the left camera. $T(\cdot)$ performs a change of coordinates, transforming the 3D point with motion $G(\boldsymbol{\theta})$ from $\mathcal{F}_\text{ref}$ to the left frame $\mathcal{F}_k$ of the current stereo observation (time k). We assume rectified and undistorted stereo configuration, which simplifies the operations by using homogeneous coordinates.

C. Compositional Algorithm

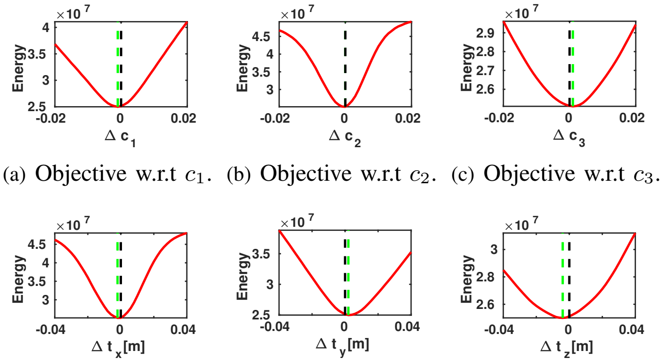
We reformulate the problem (15) using the forward compositional Lucas-Kanade method [60], which iteratively refines the incremental pose parameters. It minimizes

$$F(\Delta\boldsymbol{\theta}) \doteq \sum_{\mathbf{x}\in\mathcal{S}^{\mathcal{F}_\text{ref}}} \left(\bar{\mathcal{T}}_\text{left}(W(W(\mathbf{x},\rho;\Delta\boldsymbol{\theta});\boldsymbol{\theta}),k)\right)^2, \tag{17}$$

with respect to $\Delta\boldsymbol{\theta}$ in each iteration and then updates the estimate of the warp as:

$$W(\mathbf{x},\rho;\boldsymbol{\theta}) \leftarrow W(\mathbf{x},\rho;\boldsymbol{\theta}) \circ W(\mathbf{x},\rho;\Delta\boldsymbol{\theta}). \tag{18}$$

The compositional approach is more efficient than the additive method (15) because some parts of the Jacobian remain constant throughout the iteration and can be precomputed. This is due to the fact that linearization is always performed at the position of zero increment. As an example, Fig. 9 shows slices of the objective function with respect to each degree of freedom of $\boldsymbol{\theta}$, evaluated around ground-truth relative pose



(a) Objective w.r.t c_1. (b) Objective w.r.t c_2. (c) Objective w.r.t c_3.

(d) Objective w.r.t t_x. (e) Objective w.r.t t_y. (f) Objective w.r.t t_z.

Fig. 9: *Tracking*. Slices of the objective function (15). Plots (a)-(c) and (d)-(f) show the variation of the objective function with respect to each DoF in orientation and translation, respectively. The vertical black dashed line indicates the ground truth pose, while the green one depicts the function's minimizer.

$\boldsymbol{\Delta\theta} = \mathbf{0}$. It is clear that the objective function formulated using the compositional method is smooth, differentiable and has unique local optimum near the ground truth. To enlarge the width of the convergence basin, a Gaussian blur (kernel size of 5 pixels) is applied to the TS negative.

D. Robust and Efficient Motion Estimation

As far as we have observed, the non-linear least-squares solver is already accurate enough. However, to improve robustness in the presence of noise and outliers in the inverse depth map, a robust norm is considered. For efficiency, the Huber norm is applied and the iteratively reweighted least squares (IRLS) method is used to solve the resulting problem.

To speed up the optimization, we solve the problem using the Levenberg-Marquardt (LM) method with stochastic sampling strategy (as in [14]). At each iteration, only a batch of N_p 3D points are randomly picked in the reference frame and used for evaluating the objective function (typically $N_p = 300$). The LM method can deal with the non-negativeness of the residual $\bar{\mathcal{T}}_{\text{left}}(\cdot, k)$ and it is run only one iteration per batch. We find that five iterations are often enough for a successful convergence because the initial pose is typically close to the optimum.

VI. Experiments

Let us now evaluate the proposed event-based stereo VO system. First we present the datasets and stereo camera rig used as source of event data (Section VI-A). Then, we evaluate the performance of the method with two sets of experiments.

In the first set, we show the effectiveness of the mapping module alone by using ground truth poses provided by an external motion capture system. We show that the proposed *Student's t* probabilistic approach leads to more accurate inverse depth estimates than standard least squares (Section VI-B), and then we compare the proposed mapping method against three stereo 3D reconstruction baselines (Section VI-C).

In the second set of experiments, we evaluate the performance of the full system by feeding only events and comparing the estimated camera trajectories against the ground truth ones (Section VI-D). We further demonstrate the capabilities of our approach to unlock the advantages of event-based cameras in order to perform VO in difficult illumination conditions, such as low light and HDR (Section VI-E). Finally, we analyze the computational performance of the VO system (Section VI-F), discuss its limitations (Section VI-G), and motivate research on difficult motions for space-time consistency (Section VI-H).

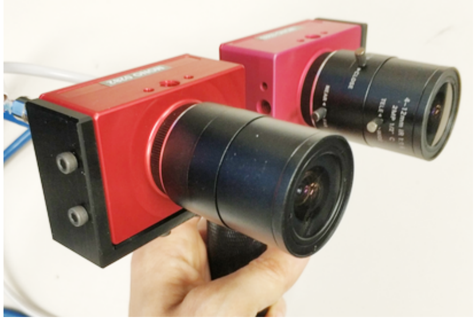


Fig. 10: Custom stereo event-camera rig consisting of two DAVIS346 cameras with a horizontal baseline of $7.5\,\text{cm}$.

TABLE II: Parameters of various stereo event-camera rigs used in the experiments.

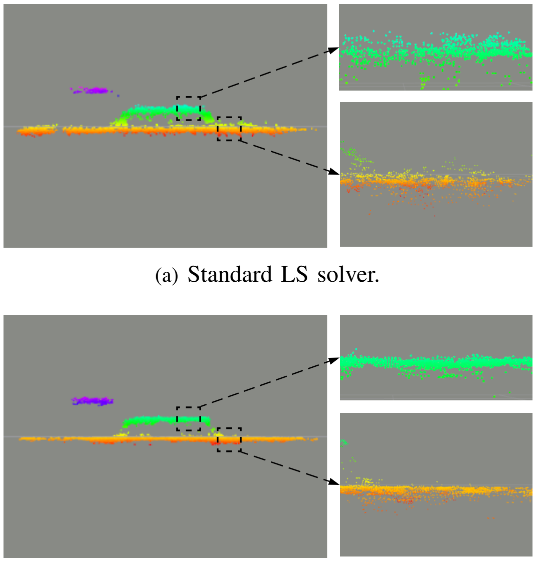
Dataset	Cameras	Resolution (pix)	Baseline (cm)	FOV (°)
[21]	DAVIS240C	240 × 180	14.7	62.9
[55]	DAVIS346	346 × 260	10.0	74.8
[58]	Simulator	346 × 260	10.7	74.0
Ours	DAVIS346	346 × 260	7.5	66.5

A. Experimental Setup and Datasets Used

To evaluate the proposed stereo VO system we use sequences from publicly available datasets and simulators [21], [55], [58]. Data provided by [21] was collected with a hand-held stereo event camera in an indoor environment. Sequences used from [55] were collected using a stereo event camera mounted on a drone flying in a capacious indoor environment. The simulator [58] provides synthetic sequences with simple structure (*e.g.*, front-to-parallel planar structures, geometric primitives, etc.) and an "ideal" event camera model. Besides the above datasets, we collect several sequences using the stereo event-camera rig in Fig. 10. The stereo rig consists of two Dynamic and Active Pixel Vision Sensors (DAVIS 346) of 346×260 pixel resolution, which are calibrated intrinsically and extrinsically. The DAVIS comprises a frame camera and an event sensor (DVS) on the same pixel array, thus calibration can be done using standard methods on the intensity frames and applied to the events. Our algorithm works on undistorted and stereo-rectified coordinates, which are precomputed given the camera calibration. The parameters of the stereo event-camera setup in each dataset used are listed in Table II.

B. Comparison of Mapping Optimization Criteria: IRLS vs LS

With this experiment we briefly justify the probabilistic inverse depth model derived from empirical observations of the



(a) Standard LS solver.

(b) *Student's t* distribution based IRLS solver.

Fig. 11: *Mapping*. Qualitative comparison between standard least-squares (LS) solver and *Student's t* distribution-based iteratively reweighted LS (IRLS) solver. Regions highlighted with dashes are zoomed in for better visualization of details.

TABLE III: Comparison between standard least-squares (LS) solver and *Student's t* distribution-based IRLS solver.

	#Fusions ↑	Mean error ↓	Std. ↓
L_2 norm	$3.33{\cdot}10^5$	2.76 cm	2.94 cm
Student's t	$\mathbf{5.07}{\cdot}10^5$	**2.15** cm	**1.29** cm

distribution of time-surface residuals (Fig. 6); two very different but related quantities (10). Using synthetic data from [58], Fig. 11 and Table III show that the proposed probabilistic approach leads to more accurate 3D reconstructions than the standard least-squares (LS) objective criterion. The synthetic scene in Fig. 11 consists of three planes parallel to the image planes of the cameras at different depths. The reconstruction results in Fig. 11(b) shows more accurate planar structures than those in Fig. 11(a). As quantified in Table III, the depth error's standard deviation of the *Student's t* distribution-based objective is 2-3 times smaller than that of the standard LS objective, which explains the more compact planar reconstructions in Fig. 11(b) over Fig. 11(a).

C. Comparison of Stereo 3D Reconstruction Methods

To prove the effectiveness of the proposed mapping method, we compare against three stereo methods and ground truth depth when available. The baseline methods are abbreviated by GTS [26], SGM [45] and CopNet [62].

a) Description of Baseline Methods: The method in [26] proposes to match events by using a per-event time-based consistency criterion that also works on grayscale events from the ATIS [27] camera; after that, classical triangulation provides the 3D point location. Since the code for this method is not available, we implement an abridged version of it, without the term for grayscale events because they are not available with the DAVIS. The semiglobal matching (SGM) algorithm [45], available in OpenCV, is originally designed to solve the stereo matching problem densely on frame-based inputs. We adapt it to our problem by running it on the stereo time surfaces and masking the produced depth map so that depth estimates are only given at pixels where recent events happened. The method in [62] (CopNet) applies a cooperative stereo strategy [31] in an asynchronous fashion. We use the implementation in [63], where identical parameters are applied.

For a fair comparison against our method, which incrementally fuses successive depth estimates, we also propagate the depth estimates produced by GTS and SGM. Since the baselines do not provide uncertainty estimates, we simply warp depth estimates from the past to the present time (i.e., the time where fusion is triggered in our method). All methods start and terminate at the same time, and use ground truth poses to propagate depth estimates in time so that the evaluation does not depend on the tracking module. Due to software incompatibility, propagation was not applied to CopNet. Therefore, CopNet is called only at the evaluation time; however, the density of its resulting inverse depth map is satisfactory when fed with enough number of events (15 000 events [63]).

b) Results: Fig. 12 compares the inverse depth maps produced by the above stereo methods. The first column shows the raw grayscale frames from the DAVIS [64], which only illustrate the appearance of the scenes because the methods do not use intensity information. The second to the last columns show inverse depth maps produced by GTS, SGM, CopNet and our method, respectively. As expected because event cameras respond to the apparent motion of edges, the methods produce semi-dense depth maps that represent the 3D scene edges. This is more apparent in GTS, CopNet and our method than in SGM because the regularizer in SGM helps to hallucinate depth estimates in regions where the spatio-temporal consistency is ambiguous, thus leading to the most dense depth maps. Though CopNet produces satisfactory density results, it performs worse than our method in terms of depth accuracy. This may be due to the fact that CopNet's output disparity is quantized to pixel accuracy. In addition, the relatively large neighborhood size used (suggested by its creators [62]) introduces over-smoothing effects. Finally, it can be observed that our method gives the best results in terms of compactness and signal-to-noise ratio. This is due to the fact that we model both (inverse) depth and its uncertainty, which enables a principled multi-view depth fusion and pruning of unreliable estimates. Since our method incrementally fuses successive depth estimates, the density of the resulting depth maps remains stable even though the streaming rate of events may vary, as is noticeable in the accompanying video.

An interesting phenomenon regarding the GTS method is found: the density of the GTS's results on *upenn* sequences are considerably lower than in *rpg* sequences. *upenn* sequences differ from *rpg* sequences in two aspects: (i) they have larger depth range, and (ii) the motion is different (*upenn* cameras are mounted on a drone which moves in a dominantly translating manner, while *rpg* sequences are acquired with hand-held cameras performing general motions in 3D space).

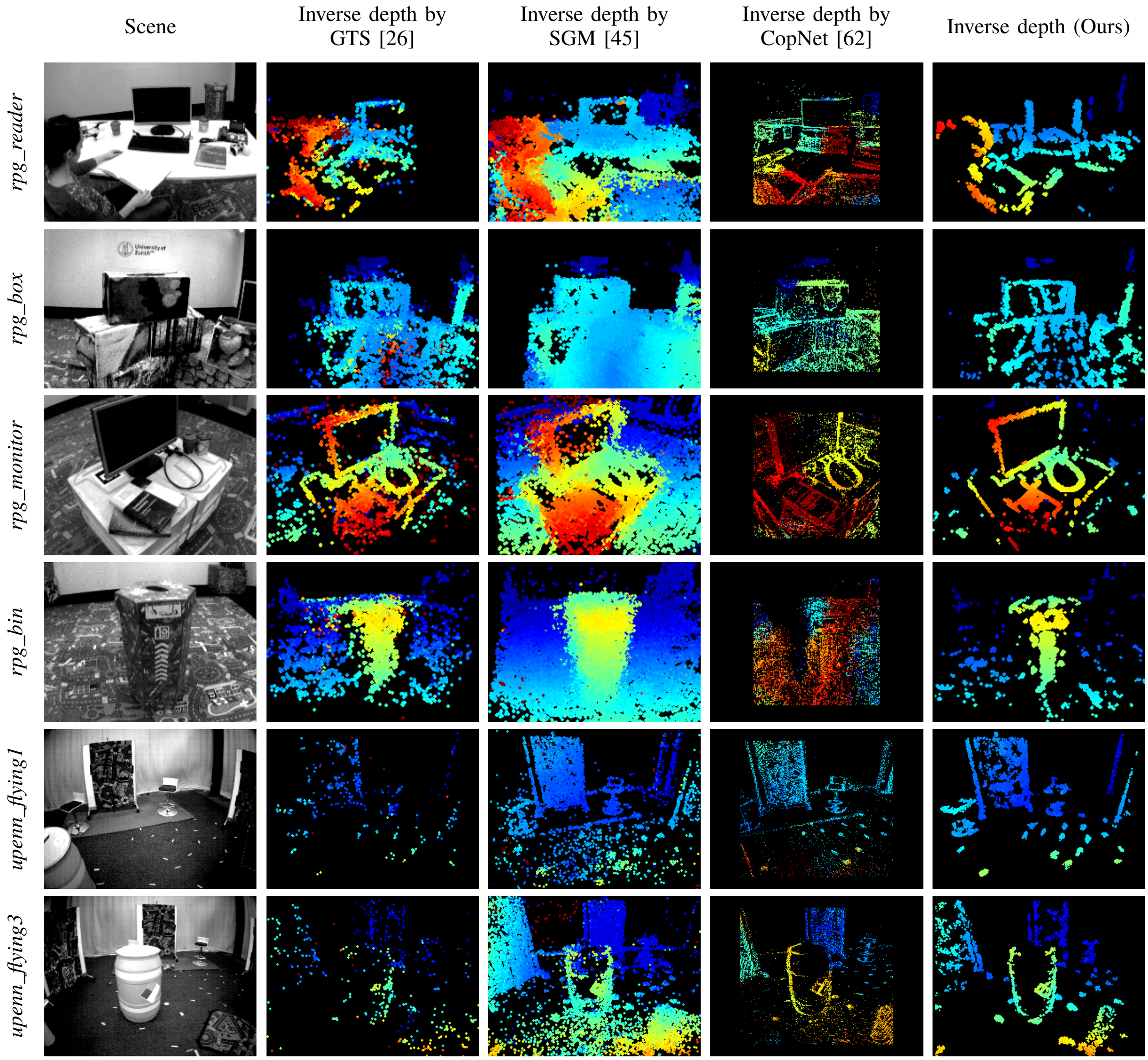


Fig. 12: *Mapping*. Qualitative comparison of mapping results (depth estimation) on several sequences using various stereo algorithms. The first column shows intensity frames from the DAVIS camera (not used, just for visualization). Columns 2 to 5 show inverse depth estimation results of GTS [26], SGM [45], CopNet [62] and our method, respectively. Depth maps are color coded, from red (close) to blue (far) over a black background, in the range 0.55–$6.25\,\mathrm{m}$ for the top four rows (sequences from [21]) and the range 1–$6.25\,\mathrm{m}$ for the bottom two rows (sequences from [55]).

The combination of both factors yields a smaller apparent motion of edges on the image plane in *upenn* sequences; this may produce large times between corresponding events (originated by the same 3D edge). To improve the density of the GTS's result, one may relax the maximum time distance used for event matching, which, however, would lead to less accurate and nosier depth estimation results.

We observe that the results of *rpg_reader* and *rpg_bin* are less sharp compared to those of *rpg_box* and *rpg_monitor*. This is due to the different quality of the ground truth poses provided; we found that poses provided in *rpg_reader* and *rpg_bin* are less globally consistent than in other sequences.

Finally, table IV quantifies the depth errors for the last two sequences of Fig. 12, which are the ones where ground truth depth is available (acquired using a LiDAR [55]). Our method outperforms the baseline methods in all criteria: mean, median and relative error (with respect to the depth range).

D. Full System Evaluation

To show the performance of the full VO system, we report ego-motion estimation results using two standard metrics: relative pose error (RPE) and absolute trajectory error (ATE) [65]. Since no open-source event-based VO/SLAM projects is yet available, we implement a baseline that leverages commonly applied methods of depth and rigid-motion estimation in computer vision. Additionally, we compare against a state-of-the-art frame-based SLAM pipeline (ORB-SLAM2 [66]) running on the grayscale frames acquired by the stereo DAVIS.

TABLE IV: Quantitative evaluation of mapping on sequences with ground truth depth.

	Sequence [55]	*upenn_flying1*	*upenn_flying3*
	Depth range [m]	5.48 m	6.03 m
GTS [26]	Mean error	0.31 m	0.44 m
	Median error	0.18 m	0.21 m
	Relative error	5.64 %	7.26 %
SGM [45]	Mean error	0.31 m	0.20 m
	Median error	0.15 m	0.10 m
	Relative error	5.58 %	3.28 %
CopNet [62]	Mean error	0.59 m	0.53 m
	Median error	0.49 m	0.44 m
	Relative error	10.93 %	8.87 %
Our Method	Mean error	**0.16** m	**0.19** m
	Median error	**0.12** m	**0.09** m
	Relative error	**3.05** %	**3.13** %

TABLE V: Relative Pose Error (RMS) [$\mathbf{R}$: $^\circ$/s, $\mathbf{t}$: cm/s]

	ORB_SLAM2 (Stereo)		SGM + ICP		Our Method	
Sequence	$\mathbf{R}$	$\mathbf{t}$	$\mathbf{R}$	$\mathbf{t}$	$\mathbf{R}$	$\mathbf{t}$
rpg_bin	**0.6** (0.5)	**1.5** (1.2)	7.6	13.3	1.2	3.1
rpg_box	**1.8** (1.7)	**5.1** (2.7)	7.9	15.5	3.4	7.2
rpg_desk	**2.4** (1.7)	**3.3** (2.8)	10.1	14.6	3.1	4.5
rpg_monitor	**1.0** (0.6)	**1.8** (1.0)	8.1	10.7	1.7	3.2
upenn_flying1	5.4 (5.8)	20.4 (16.2)	4.8	31.6	**1.0**	**6.5**
upenn_flying3	5.6 (3.0)	22.0 (20.1)	7.3	26.3	**1.2**	**7.1**

The numbers in parentheses in ORB_SLAM2 represent the RMS errors with bundle adjustment enabled.

TABLE VI: Absolute Trajectory Error (RMS) [$\mathbf{t}$: cm]

	ORB_SLAM2	SGM + ICP	Our Method
rpg_bin	**0.9** (0.7)	13.8	2.8
rpg_box	**2.9** (1.6)	19.8	5.8
rpg_desk	7.7 (1.8)	8.5	**3.2**
rpg_monitor	**2.5** (0.8)	29.5	3.3
upenn_flying1	49.8 (41.7)	95.8	**13.9**
upenn_flying3	50.2 (36.5)	55.7	**11.1**

More specifically, the baseline solution, called "SGM+ICP", consists of combining the SGM method [45] for dense depth estimation and the iterative closest point (ICP) method [67] for estimating the relative pose between successive depth maps (i.e., point clouds). The whole trajectory is obtained by sequentially concatenating relative poses.

The evaluation is performed on six sequences with ground truth trajectories and the evaluation results can be found in Tables V and VI. The best results per sequence are highlighted in bold. It is clear that our method outperforms the event-based baseline solution on all sequences. To make the comparison against ORB-SLAM2 fair, global bundle adjustment (BA) was disabled; nevertheless, the results with global BA enabled are also reported in the tables, for reference. Our system is slightly less accurate than ORB-SLAM2 on *rpg* dataset, while shows a better performance on *upenn_indoor_flying* dataset. This is due to a flickering effect in the *rpg* dataset induced by the motion capture system, which slightly deteriorates the performance of our method but does not appear on the grayscale frames used by ORB-SLAM2.

The trajectories produced by event-based methods are compared in Fig. 13. Our method significantly outperforms the event-based baseline SGM+ICP. The evaluation of the full VO system using Fig. 13 assesses whether the mapping and tracking remain consistent with each other. This requires the mapping module to be robust to the errors induced by the tracking module, and vice versa. Our system does a remarkable job in this respect.

As a result of the above-mentioned flickering phenomena in *rpg* datasets, the spatio-temporal consistency across stereo time-surface maps may not hold well all the time. We find that our system performs robustly under this challenging scenario as long as it does not occur during initialization. Readers can get a better understanding of the flickering phenomena by watching the accompanying video.

The VO results on the *upenn* dataset show worse accuracy compared to those on the *rpg* dataset. This may be attributed to the following two reasons. First, the motion pattern (dominant translation with slight rotation) determines that no structures parallel to the baseline of the stereo rig are reconstructed (as will be discussed in Fig. 17(d)). These missing structures may lead to less accurate motion estimation in the corresponding degree of freedom. Second, the accuracy of the system (tracking and mapping) is limited by the relatively small spatial resolution of the sensor. Using event cameras with higher resolution (*e.g.*, VGA [68]) would improve the accuracy of the system. Finally, note that when the drone stops and hovers few events are generated and, thus, time surfaces triggered at constant rate become unreliable. This would cause our system to reinitialize. It could be mitigated by using more complex strategies to signal the creation of time surfaces, such as a constant or adaptive number of events [69]. However this is out of scope of the present work. Thus, we only evaluate on the dynamic section of the dataset.

We also evaluate the proposed system on the *hkust_lab* sequence collected using our stereo event-camera rig. The scene represents a cluttered environment which consists of various machine facilities. The stereo rig was hand-held and moved from left to right under a locally loopy behavior. The 3D point cloud together with the trajectory of the sensor are displayed in Fig. 14. Additionally, the estimated inverse depth maps at selected views are visualized. The live demonstration can be found in the supplemental video.

E. Experiments in Low Light and HDR Environments

In addition to the evaluation under normal illumination conditions, we test the VO system in difficult conditions for frame-based cameras. To this end, we run the algorithm on two sequences collected in a dark room. One of them is lit with a lamp to increase the range of scene brightness variations, creating high dynamic range conditions. Results are shown in Fig. 15. Under such conditions, the frame-based sensor of the DAVIS (with $55\,\mathrm{dB}$ dynamic range) can barely see anything in the dark regions using its built-in auto-exposure, which would lead to failure of VO pipelines working on this visual modality. By contrast, our event-based method is able to work robustly in these challenging illumination conditions due to the natural HDR properties of event cameras ($120\,\mathrm{dB}$ range).

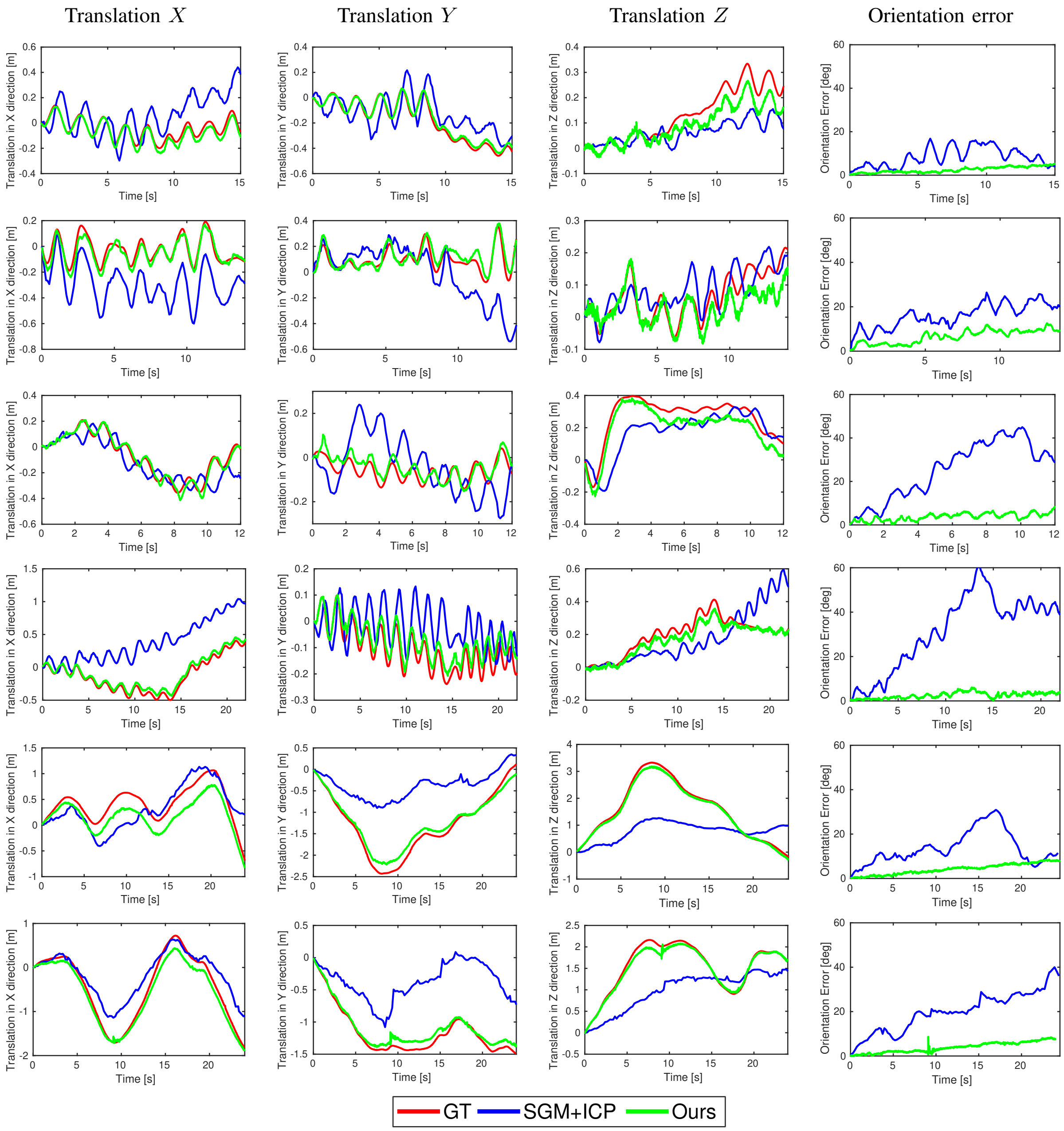


Fig. 13: *Tracking - DoF plots*. Comparison of two tracking methods against the ground truth camera trajectory provided by the motion capture system. Columns 1 to 3 show the translational degrees of freedom (in meters). The last column shows rotational error in terms of the geodesic distance in *SO(3)* (the angle of the relative rotation between the ground truth rotation and the estimated one). Each row corresponds to a different sequence: *rpg_bin*, *rpg_box*, *rpg_desk*, *rpg_monitor*, *upenn_flying1* and *upenn_flying3*, respectively. The ground truth is depicted with red color (—), the "SGM+ICP" method with blue (—) and our method with green (—). In the error plots the ground truth corresponds to the reference, i.e., zero. The *rpg* sequences [21] are captured with a hand-held stereo rig moving under a locally loopy behavior (top four rows). In contrast, the *upenn_flying* sequences [55] are acquired using a stereo rig mounted on a drone which switches between hovering and moving dominantly in a translating manner (bottom two rows).

F. Computational Performance

The proposed stereo visual odometry system is implemented in C++ on ROS and runs in real-time on a laptop with an Intel Core i7-8750H CPU. Its computational performance is summarized in Table VII. To accelerate processing, some nodes (mapping and tracking) are implemented with hyper-threading technology. The number of threads used by each node is indicated in parentheses next to the name of the node.

The creation of the time-surface maps takes about 5–10 ms, depending on the sensor resolution. The initialization node, active only while bootstrapping, takes 12–17 ms (up to sensor resolution) to produce the first local map (depth map given by

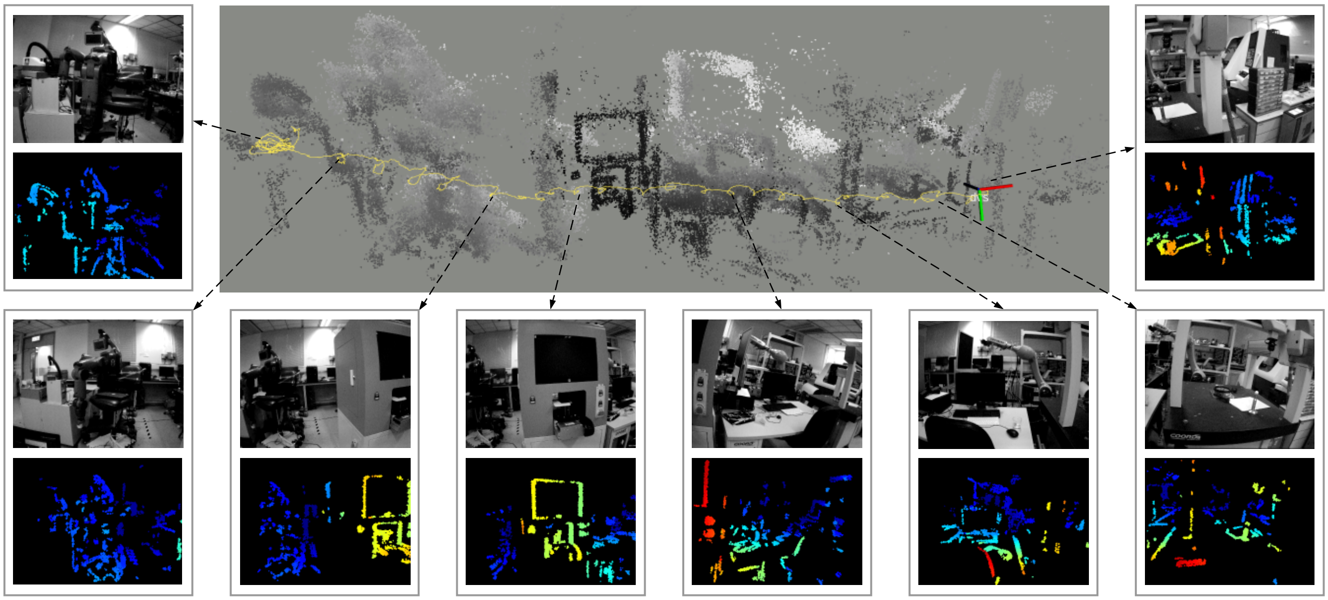


Fig. 14: Estimated camera trajectory and 3D reconstruction of *hkust_lab* sequence. Computed inverse depth maps at selected viewpoints are visualized sequentially, from left to right. Intensity frames are shown for visualization purpose only.

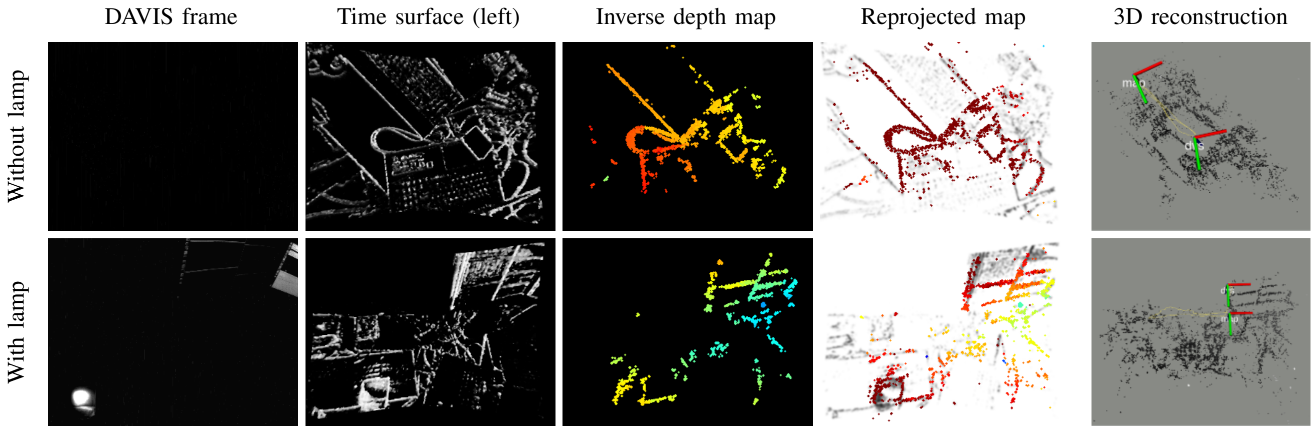


Fig. 15: *Low light and HDR scenes.* Top row: results in a dark room; Bottom row: results in a dark room with a directional lamp. From left to right: grayscale frames (for visualization purpose only), time surfaces, estimated depth maps, reprojected maps on time surface negatives (tracking), and 3D reconstruction with overlaid camera trajectory estimates, respectively.

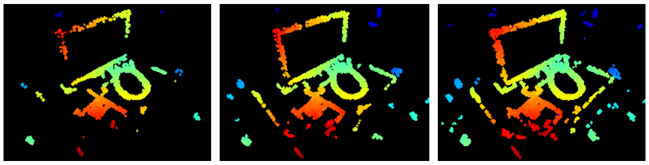
TABLE VII: Computational performance

Node (#Threads)	Function	Time (ms)
Time surfaces (1)	Exponential decay	5 – 10
Initialize depth (1)	SGM & masking	12 – 17
Mapping (4)	Event matching	6 ($\sim$ 1000 events)
	Depth optimization	15 ($\sim$ 500 events)
	Depth fusion	20 ($\sim$ 60000 fusions)
Tracking (2)	Non-linear solver	10 (300 points $\times$ 5 iterations)

the SGM method and masked with an event map).

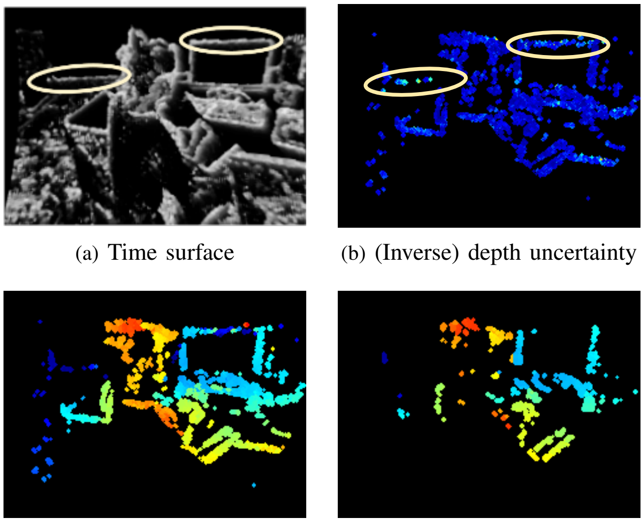
The mapping node uses 4 threads and takes about $41\,\mathrm{ms}$, spent in three major functions. (i) The matching function takes $\approx 6\,\mathrm{ms}$ to search for 1000 corresponding patches across a pair of time surfaces. The matching success rate is $\approx$ 40–50 %, depending on how well the spatio-temporal consistency holds in the data. (ii) The depth refinement function returns 500 inverse depth estimates in $15\,\mathrm{ms}$. (iii) The fusion function (propagation and update steps) does 60000 operations in $20\,\mathrm{ms}$. Thus, the mapping node runs at $20\,\mathrm{Hz}$ typically.

Regarding the choice for the number of events being processed in the inverse depth estimation (*i.e.*, 1000 as mentioned above), we justify it by showing its influence on the reconstruction density of the estimated depth maps. Fig. 16 shows mapping results using 500, 1000 and 2000 events for inverse depth estimation. We randomly pick these events out of the latest 10000 events. For a fair comparison, the number of fusion steps remains constant. As it is observed, the more events are used the more dense the inverse depth map becomes. The map obtained using 500 events is the sparsest. We notice that using 1000 or 2000 events produces nearly the same reconstruction density. However the latter (2000



(a) 500 events.
(b) 1000 events.
(c) 2000 events.

Fig. 16: Influence of the number of events used for (inverse) depth estimation on the density of the fused depth map.



(a) Time surface
(b) (Inverse) depth uncertainty
(c) Depth map before pruning estimates with low uncertainty.
(d) Depth map after pruning estimates with low uncertainty.

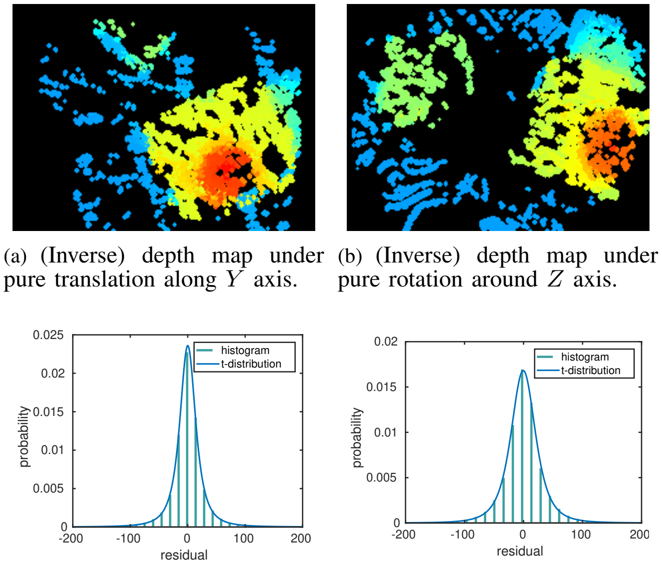
Fig. 17: Depth uncertainty allows to filter unreliable estimates.

events) is computationally more expensive (computation time is approximately proportional to the number of events); hence, for real-time performance opt for 1000 events.

The tracking node uses 2 threads and takes ≈10 ms to solve the pose estimation problem using an IRLS solver (a batch of 300 points are randomly sampled in each iteration and at most five iterations are performed). Hence, it can run up to 100 Hz.

G. Discussion: Missing Edges in Reconstructions

Here we note an effect that appears in some reconstructions, even when computed using ground truth poses (Section VI-C). We observe that edges that are parallel to the baseline of the stereo rig, such as the upper edge of the monitor in *rpg_reader* and the hoops on the barrel in *upenn_flying3* (Fig. 12), are difficult to recover regardless of the motion. All stereo methods suffer from this: although GTS, SGM and CopNet can return depth estimates for those parallel structures, they are typically unreliable; our method is able to reason about uncertainty and therefore rejects such estimates. In this respect, Fig. 17 shows two horizontal patterns (highlighted with yellow ellipses in Fig. 17(a)) and their corresponding uncertainties (Fig. 17(b)), which are larger than those of other edges. By thresholding on the depth uncertainty map (Fig. 17(c)), we obtain a more reliable albeit sparser depth map (Fig. 17(d)). Improving the completeness of reconstructions suffering from the above effect is left as future work.



(a) (Inverse) depth map under pure translation along Y axis.
(b) (Inverse) depth map under pure rotation around Z axis.

(c) Distribution of residuals for pure translation along Y axis.
(d) Distribution of residuals for pure rotation around Z axis.

Fig. 18: Analysis of spatio-temporal consistency. (a)-(b): Inverse depth estimates under two different types of motion. (c)-(d): Corresponding histograms of temporal residuals. Scene: *toy_room* in [9]. The corresponding videos can be found at https://youtu.be/QY82AcX1LDo (translation along Y axis); and https://youtu.be/RkxBn304gJI (rotation around Z axis).

H. Dependency of Spatio-Temporal Consistency on Motion

Time surfaces are motion dependent, and consequently, even in the noise-free case the proposed spatio-temporal consistency criterion may not hold perfectly when the stereo rig undergoes some specific motions. One extreme case could be a pure rotation of the left camera around its optical axis; thus the right camera would rotate and translate. Intuitively, the additional translation component of the right camera would produce spatio-temporal inconsistency between the left-right time surfaces such that the mapping module would suffer. To analyze the sensitivity of the mapping module with respect to the spatio-temporal consistency, we carried out the following experiment (Fig. 18). We used an event camera simulator [70] to generate sequences with perfect control over the motion. Specifically, we generated sequences with pure rotation of the left camera around the optical axis (Z axis), and compared the mapping results against those of pure translational motion along the X or Y axis of the camera. The fused depth maps were slightly worse in the former case (partly because there are fewer events triggered around the centre of the image plane), but they were still accurate in most pixels (see Fig. 18(a) and 18(b)). Additionally, we analyzed the temporal inconsistency through the histogram of temporal residuals (like in Fig. 6). The histogram of residuals for the rotation around the Z axis (Fig. 18(d)) is broader than the one for translation around the X/Y axes (Fig. 18(c)). Numerically, the scale values of the t-distributions are $s_{\text{trans\_Y}} = 14.995$ and $s_{\text{rot\_Z}} = 21.838$. Compared to those in Fig. 6, the residuals in Fig. 18(d) are similar to those of the *upenn_flying1* sequence.

We conclude that, in spite of time surfaces being motion dependent, we did not observe a significant temporal inconsistency that would break down the system in a priori difficult motions for stereo. Actually the proposed method performed well in practice, as shown in all previous experiments with real data. We leave a more theoretical and detailed analysis of such motions for future research since we consider this work to address the most general motion case.

VII. Conclusion

This paper has presented a complete event-based stereo visual odometry system for a pair of calibrated and synchronized event cameras in stereo configuration. To the best of our knowledge, this is the first published work that tackles this problem. The proposed mapping method is based on the optimization of an objective function designed to measure spatio-temporal consistency across stereo event streams. To improve density and accuracy of the recovered 3D structure, a fusion strategy based on the learned probabilistic characteristics of the estimated inverse depth has been carried out. The tracking method is based on $3\text{D} - 2\text{D}$ registration that leverages the inherent distance field nature of a compact and efficient event representation (time surfaces). Extensive experimental evaluation, on publicly available datasets and our own, has demonstrated the versatility of our system. Its performance is comparable with mature, state-of-the-art VO methods for frame-based cameras in normal conditions. We have also demonstrated the potential advantages that event cameras bring to stereo SLAM in difficult illumination conditions. The system is computationally efficient and runs in real-time on a standard CPU. The software, design of the stereo rig and datasets used for evaluation have been open sourced. Future work may include fusing the proposed method with inertial observations (i.e., Event-based Stereo Visual-Inertial Odometry) and investigating novel methods for finding correspondences in time on each event camera (i.e., "temporal" event-based stereo). These are closely related topics to the problem here addressed of Event-based Stereo VO.

Acknowledgment

The authors would like to thank Ms. Siqi Liu for the help in data collection, and Dr. Alex Zhu for providing the CopNet baseline [21], [62] and assistance in using the dataset [55]. We also thank the editors and anonymous reviewers of IEEE TRO for their suggestions, which led us to improve the paper.

References

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[2] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[3] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1710–1720, Aug. 2015.

[4] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based feature tracking with probabilistic data association," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 4465–4470.

[5] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "EKLT: Asynchronous photometric feature tracking using events and frames," *Int. J. Comput. Vis.*, vol. 128, pp. 601–618, 2020.

[6] E. Mueggler, G. Gallego, and D. Scaramuzza, "Continuous-time trajectory estimation for event-based vision sensors," in *Robotics: Science and Systems (RSS)*, 2015.

[7] G. Gallego, J. E. A. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, "Event-based, 6-DOF camera tracking from photometric depth maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2402–2412, Oct. 2018.

[8] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 632–639, 2017.

[9] S. Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza, "Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 325–331.

[10] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbruck, "A pencil balancing robot using a pair of AER dynamic vision sensors," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2009, pp. 781–784.

[11] T. Delbruck and M. Lang, "Robotic goalie with 3ms reaction time at 4% CPU load using event-based dynamic vision sensor," *Front. Neurosci.*, vol. 7, p. 223, 2013.

[12] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, no. 40, p. eaaz9712, Mar. 2020.

[13] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in *Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 349–364.

[14] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 593–600, 2017.

[15] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[16] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, Dec. 2018.

[17] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt, "Event-based 3D SLAM with a depth-augmented dynamic vision sensor," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 359–364.

[18] A. Censi and D. Scaramuzza, "Low-latency event-based visual odometry," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 703–710.

[19] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2016, pp. 16–23.

[20] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE ACM Int. Sym. Mixed and Augmented Reality (ISMAR)*, Nara, Japan, Nov. 2007, pp. 225–234.

[21] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza, "Semi-dense 3D reconstruction with a stereo event camera," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 242–258.

[22] J. Kogler, M. Humenberger, and C. Sulzbachner, "Event-based stereo matching approaches for frameless address event stereo data," in *Int. Symp. Adv. Vis. Comput. (ISVC)*, 2011, pp. 674–685.

[23] P. Rogister, R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbruck, "Asynchronous event-based binocular stereo matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 347–353, 2012.

[24] L. A. Camunas-Mesa, T. Serrano-Gotarredona, S. H. Ieng, R. B. Benosman, and B. Linares-Barranco, "On the use of orientation filters for 3D reconstruction in event-driven stereo vision," *Front. Neurosci.*, vol. 8, p. 48, 2014.

[25] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003, 2nd Edition.

[26] S.-H. Ieng, J. Carneiro, M. Osswald, and R. Benosman, "Neuromorphic event-based generalized time-based stereovision," *Front. Neurosci.*, vol. 12, p. 442, 2018.

[27] C. Posch, D. Matolin, and R. Wohlgenannt, “A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS,” *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[28] E. Piatkowska, A. N. Belbachir, and M. Gelautz, “Cooperative and asynchronous stereo vision for dynamic vision sensors,” *Meas. Sci. Technol.*, vol. 25, no. 5, p. 055108, Apr. 2014.

[29] M. Firouzi and J. Conradt, “Asynchronous event-based cooperative stereo matching using neuromorphic silicon retinas,” *Neural Proc. Lett.*, vol. 43, no. 2, pp. 311–326, 2016.

[30] M. Osswald, S.-H. Ieng, R. Benosman, and G. Indiveri, “A spiking neural network model of 3D perception for event-based neuromorphic stereo vision systems,” *Sci. Rep.*, vol. 7, no. 1, Jan. 2017.

[31] D. Marr and T. Poggio, “Cooperative computation of stereo disparity,” *Science*, vol. 194, no. 4262, pp. 283–287, 1976.

[32] L. Steffen, D. Reichard, J. Weinland, J. Kaiser, A. Rönnau, and R. Dillmann, “Neuromorphic stereo vision: A survey of bio-inspired sensors and algorithms,” *Front. Neurorobot.*, vol. 13, p. 28, 2019.

[33] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, “EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time,” *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1394–1414, Dec. 2018.

[34] G. Gallego, H. Rebecq, and D. Scaramuzza, “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)*, 2018, pp. 3867–3876.

[35] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger, “Interacting maps for fast visual interpretation,” in *Int. Joint Conf. Neural Netw. (IJCNN)*, 2011, pp. 770–776.

[36] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison, “Simultaneous mosaicing and tracking with an event camera,” in *British Mach. Vis. Conf. (BMVC)*, 2014.

[37] C. Reinbacher, G. Munda, and T. Pock, “Real-time panoramic tracking for event cameras,” in *IEEE Int. Conf. Comput. Photography (ICCP)*, 2017, pp. 1–9.

[38] D. Weikersdorfer and J. Conradt, “Event-based particle filtering for robot self-localization,” in *IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, 2012, pp. 866–870.

[39] D. Weikersdorfer, R. Hoffmann, and J. Conradt, “Simultaneous localization and mapping for event-based vision systems,” in *Int. Conf. Comput. Vis. Syst. (ICVS)*, 2013, pp. 133–142.

[40] E. Mueggler, B. Huber, and D. Scaramuzza, “Event-based, 6-DOF pose tracking for high-speed maneuvers,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2014, pp. 2761–2768.

[41] G. Gallego, M. Gehrig, and D. Scaramuzza, “Focus is all you need: Loss functions for event-based vision,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 12 272–12 281.

[42] D. Migliore (Prophesee), “Sensing the world with event-based cameras,” https://robotics.sydney.edu.au/icra-workshop/, Jun. 2020.

[43] X. Lagorce, G. Orchard, F. Gallupi, B. E. Shi, and R. Benosman, “HOTS: A hierarchy of event-based time-surfaces for pattern recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.

[44] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA Workshop Open Source Softw.*, vol. 3, no. 2, 2009, p. 5.

[45] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[46] S.-C. Liu and T. Delbruck, “Neuromorphic sensory systems,” *Current Opinion in Neurobiology*, vol. 20, no. 3, pp. 288–295, 2010.

[47] T. Delbruck, “Frame-free dynamic digital vision,” in *Proc. Int. Symp. Secure-Life Electron.*, 2008, pp. 21–26.

[48] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, “End-to-end learning of representations for asynchronous event-based data,” in *Int. Conf. Comput. Vis. (ICCV)*, 2019.

[49] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, “Event-based visual flow,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, 2014.

[50] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, “EV-FlowNet: Self-supervised optical flow estimation for event-based cameras,” in *Robotics: Science and Systems (RSS)*, 2018.

[51] J. Engel, J. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 834–849.

[52] Y. Zhou, H. Li, and L. Kneip, “Canny-VO: Visual odometry with RGB-D cameras based on geometric 3-D–2-D edge alignment,” *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 184–199, 2018.

[53] R. Benosman, S.-H. Ieng, P. Rogister, and C. Posch, “Asynchronous event-based Hebbian epipolar geometry,” *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1723–1734, 2011.

[54] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in *Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 2320–2327.

[55] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, “The multivehicle stereo event camera dataset: An event camera dataset for 3D perception,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2032–2039, Jul. 2018.

[56] S. Kotz and S. Nadarajah, *Multivariate t-distributions and their applications*. Cambridge University Press, 2004.

[57] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for rgb-d cameras,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2013.

[58] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, “The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM,” *Int. J. Robot. Research*, vol. 36, no. 2, pp. 142–149, 2017.

[59] M. Roth, T. Ardeshiri, E. Özkan, and F. Gustafsson, “Robust bayesian filtering and smoothing using student’s t distribution,” *arXiv preprint arXiv:1703.02428*, 2017.

[60] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[61] A. Cayley, “About the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic,” in *Reine Angewandte Mathematik*, 1846.

[62] E. Piatkowska, J. Kogler, N. Belbachir, and M. Gelautz, “Improved cooperative stereo matching for dynamic vision sensors with ground truth evaluation,” in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2017.

[63] A. Z. Zhu, Y. Chen, and K. Daniilidis, “Realtime time synchronized event-based stereo,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 438–452.

[64] C. Brandli, T. Mantel, M. Hutter, M. Höpflinger, R. Berner, R. Siegwart, and T. Delbruck, “Adaptive pulsed laser line extraction for terrain reconstruction using a dynamic vision sensor,” *Front. Neurosci.*, vol. 7, p. 275, 2014.

[65] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Oct. 2012.

[66] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[67] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[68] B. Son, Y. Suh, S. Kim, H. Jung, J.-S. Kim, C. Shin, K. Park, K. Lee, J. Park, J. Woo, Y. Roh, H. Lee, Y. Wang, I. Ovsiannikov, and H. Ryu, “A 640x480 dynamic vision sensor with a 9μm pixel and 300Meps address-event representation,” in *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2017.

[69] M. Liu and T. Delbruck, “Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors,” in *British Mach. Vis. Conf. (BMVC)*, 2018.

[70] H. Rebecq, D. Gehrig, and D. Scaramuzza, “ESIM: an open event camera simulator,” in *Conf. on Robotics Learning (CoRL)*, 2018.

Yi Zhou received the B.Sc. degree in aircraft manufacturing and engineering from Beijing University of Aeronautics and Astronautics, Beijing, China in 2012, and the Ph.D. degree with the Research School of Engineering, Australian National University, Canberra, ACT, Australia in 2018. Since 2019 he is a postdoctoral researcher at the Hong Kong University of Science and Technology, Hong Kong. His research interests include visual odometry / simultaneous localization and mapping, geometry problems in computer vision, and dynamic vision sensors. Dr. Zhou was awarded the NCCR Fellowship Award for the research on event based vision in 2017 by the Swiss National Science Foundation through the National Center of Competence in Research Robotics.

Guillermo Gallego (SM'19) is Associate Professor at the Technische Universität Berlin, in the Dept. of Electrical Engineering and Computer Science, and at the Einstein Center Digital Future, both in Berlin, Germany. He received the PhD degree in Electrical and Computer Engineering from the Georgia Institute of Technology, USA, in 2011, supported by a Fulbright Scholarship. From 2011 to 2014 he was a Marie Curie researcher with Universidad Politecnica de Madrid, Spain, and from 2014 to 2019 he was a postdoctoral researcher at the Robotics and Perception Group, University of Zurich, Switzerland. His research interests include robotics, computer vision, signal processing, optimization and geometry.

Shaojie Shen (Member, IEEE) received the B.Eng. degree in electronic engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2009, the M.S. degree in robotics and the Ph.D. degree in electrical and systems engineering, both from the University of Pennsylvania, Philadelphia, PA, USA, in 2011 and 2014, respectively. He joined the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology in September 2014 as an Assistant Professor, and was promoted to associate professor in 2020. His research interests are in the areas of robotics and unmanned aerial vehicles, with focus on state estimation, sensor fusion, computer vision, localization and mapping, and autonomous navigation in complex environments.