

Event-based Spinning Object SLAM

Ethan Elms* Yasir Latif Tat-Jun Chin[†]
The University of Adelaide

{ethan.elms,yasir.latif,tat-jun.chin}@adelaide.edu.au

Abstract

Event sensors offer high temporal resolution and high dynamic range visual sensing which makes them appealing for visual simultaneous localization and mapping (VSLAM). In this work, we investigate event-based VSLAM for the scenario where a static event camera observes an object that is undergoing an unknown spinning motion. Geometrically, the setting is equivalent to a static object that is observed by an event camera that is orbiting around the object with unknown orbital parameters. We exploit this duality to develop an algorithm called *eSpinSLAM*. Key components of *eSpinSLAM* are an online event-only feature detection and tracking mechanism and a continuous-time back-end that can incrementally reconstruct the object and estimate the orbital motion. The problem geometry not only permits a camera state representation with a bounded number of parameters that can support infinite time horizon operation, but also enables effective loop closure detection and drift mitigation via spinning frequency estimation. Results on a real event dataset validate the improved feature tracking, higher reconstruction accuracy and greater throughput of *eSpinSLAM* over existing event-based 3D vision methods.

1. Introduction

Visual simultaneous localization and mapping (VSLAM) aims to concurrently recover the scene structure and poses of a camera that observes a scene. Emphasis is placed on online incremental operation such that the results can contribute to downstream tasks such as robotic interaction and navigation. Recently, event cameras have been used for VSLAM due to several advantages such as high temporal resolution and dynamic range [18]. However, event-based VSLAM is generally less mature than frame-based VSLAM, thus the former is very much an active research area.

In this paper, we investigate event-based VSLAM for the scenario where a static event camera observes a target ob-

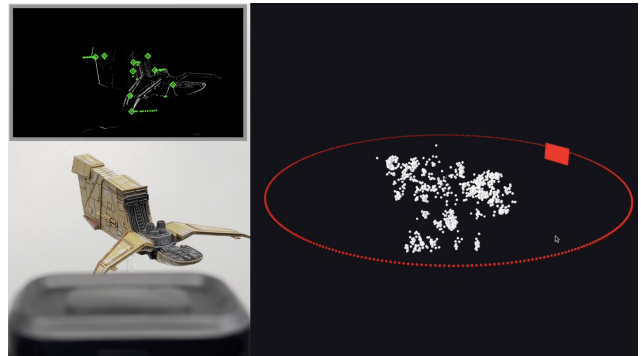


Figure 1. Live system operation: a static event camera observes a spinning object (bottom left), the front-end tracks features (top left) which the back-end uses to estimate both the structure of the object and the motion parameters (right). See supplementary video `eSpinSLAM.Demo.mp4` for a demonstration.

ject spinning at a constant angular velocity; see Fig. 1. We assume that the spinning motion of the object is due to a phenomenon not under the control of the observer, such as low-gravity tumble [46] and accidental spin [8]. The fact that the *unknown* spinning motion *and* object structure need to be estimated makes this a bona fide VSLAM problem.

While of fundamental interest in its own right [8, 28], VSLAM of spinning objects is relevant to robotic perception applications such as visual servoing and grasping [31, 34, 49], and spacecraft rendezvous and docking [5, 17, 46]. The usage of an event camera is motivated by the fact that the recurrent spinning motion continuously triggers edge features that are useful to VSLAM at a high temporal rate [19], thereby supporting responsive interactions. If the background is static w.r.t. the camera, no events are triggered by the background which greatly simplifies data association. Moreover, the energy efficiency of event sensors makes them attractive for space applications [25].

Geometrically, observing a spinning object using a static camera is equivalent to observing a static object using a camera that orbits around the object [16]. We exploit this duality to formulate a continuous-time orbital motion model with a bounded number of motion parameters, which then leads to a novel VSLAM algorithm called *eSpinSLAM*. Our

*This work has been supported by the AIML and the SmartSat CRC, whose activities are funded by the Australian Government's CRC Program.

[†]SmartSat CRC Professorial Chair of Sentient Satellites.

algorithm can process the live event stream at high throughput to estimate on-the-fly a semi-dense point cloud of the object and the relative orbital motion. Key components of eSpinSLAM include:

- An online event-only feature tracking and data association front-end, adapted for asynchronous event streams.
- A continuous-time back-end optimization framework for real-time incremental reconstruction and pose estimation.
- A novel loop closure mechanism via frequency estimation that exploits the recurring nature of spinning objects to minimize drift in long-term operation.
- Robust outlier management, landmark association, and track management strategies to enable scalable, indefinite runtime without memory blow-up.

When executed over a long term, eSpinSLAM incrementally reconstructs and refines the output structure; see supplementary video `eSpinSLAM_Demo.mp4` for a demo.

The closest work to ours is event-based structure-from-orbit (eSfO) [16]. Due to using an over-parameterized motion model and computationally costly subroutines (*e.g.*, feature detection and tracking, COLMAP [43] for initialization), eSfO is fundamentally incapable of online operation. The key components of eSpinSLAM listed above not only enable its throughput to be orders of magnitude higher than that of eSfO, the reconstruction accuracy and long-term stability of eSpinSLAM are also greater than eSfO’s, based on results on the TOPSPIN dataset [16] (see Sec. 5).

2. Related work

2.1. Event-based VSLAM and visual odometry

Event cameras provide high temporal resolution sensing with higher dynamic range. This has made them useful for VSLAM applications where conventional frame-based cameras struggle. Many works have used an event camera in conjunction with other sensing modalities to provide robust VSLAM solutions. EDS [23] paired an event-camera with a frame-camera to enable camera motion estimation in the blind time between frames for direct sparse odometry. Similarly, Ultimate SLAM [47] combined frames and IMU with event sensing for applications in high-speed scenarios. Finally, since many event cameras incorporate IMUs, several works have also developed paired event and IMU VSLAM systems [7, 20, 48].

Recent works have also explored methods for event-only motion estimation and structure recovery. DEVO [30] is a monocular, event-only Visual Odometry (VO) system that leverages a novel deep learning-based patch selection mechanism to accurately track camera motion. An earlier work, EVO [42], recovers structure and motion from a monocular event camera via edge map alignment. Event-only rotation estimation was explored in CMax-SLAM [21], which leverages contrast maximization to build a SLAM system. Using

lines as the geometric primitive, [4] demonstrated an event-only parallel tracking and mapping system using a Kalman filter that can reconstruct a sparse line-based representation of a scene. eSfO [16] recovers the sparse structure and motion of a spinning object using a single event camera, but can only provide estimates offline. Despite all this, monocular event-only VSLAM is still a topic of active research.

2.2. Vision-based navigation in space

The success of ground-based SLAM and pose estimation methods have motivated their application to space-based environments and spacecraft relative navigation. Methods for autonomous mapping of asteroids [14, 39], spacecraft rendezvous and close-proximity operations [13, 38, 40, 45], and pose estimation [6, 29, 37] have been explored using conventional frame-based cameras. As the number of objects in orbit grows, there is a greater need for in-orbit perception for vision-based navigation.

Event cameras provide low-powered high fidelity on-board sensing and have been utilized for various space applications including star trackers [9, 36], spacecraft landing [33, 44] and space situational awareness [1, 11]. Methods for event-based satellite pose estimation [26, 27, 41] have shown better perception in the harsh space conditions.

By tackling a novel problem setting with a compelling application, our work pushes the boundaries of vision-based navigation for space. Our eSpinSLAM algorithm, which can achieve monocular event-only long-term spinning object reconstruction, also represents a major innovation in event-based 3D perception.

3. Problem statement

In eSpinSLAM, a static event camera observes an object χ that is spinning at a constant angular velocity, taking t_{rev} seconds to complete one revolution. This is mathematically equivalent to the dual formulation (see [16] and proof therein) of a static target observed by an orbiting camera. Under this dual view, the executed orbit of the event camera traces out a circle during the time t_{rev} which lies within the orbital plane Δ ; see Fig. 2. Let the data stream from the event camera be $\mathcal{E} = \{e_i\}_{i=1}^{N_E}$, where each $e_i = (t_i, x_i, y_i, p_i)$ consists of the event pixel-location (x_i, y_i) , the timestamp (t_i) of when the event occurred and a binary polarity $(p_i \in \{0, 1\})$ indicating the direction of intensity change. Note that the length N_E of \mathcal{E} could be ∞ , corresponding to the case where the object χ spins perpetually. The aim of eSpinSLAM is to recover a sparse point cloud $\mathbf{X}^w = \{\mathbf{x}_j^w \in \mathbb{R}^3\}_{j=1}^{N_X}$ in the world reference frame w representing χ , and the orbit parameters (defined below).

3.1. Orbital motion model

As the camera executes orbital motion around the object, the task of the motion model is to provide a pose generation

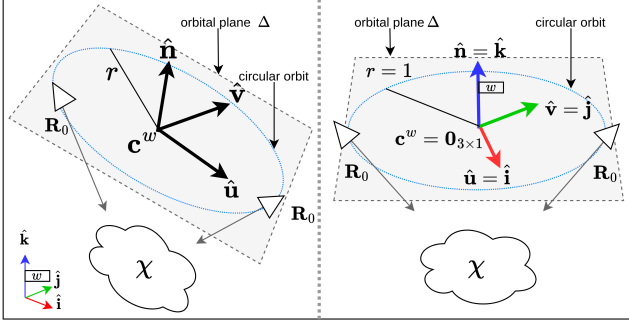


Figure 2. **L**: eSfO [16] parameterization where the orbital parameters are \mathbf{c}^w , r , \mathbf{R}_0 , $\hat{\mathbf{n}}$, $\hat{\mathbf{u}}$, and f (not shown). **R**: eSpinSLAM parameterization where the world frame w is at the center of the circular orbit and $\hat{\mathbf{u}}$ and $\hat{\mathbf{n}}$ coincide with the x and z direction of w , i.e., $\hat{\mathbf{i}}$ and $\hat{\mathbf{k}}$, respectively. The orbital parameters are just \mathbf{R}_0 and f (not shown).

function which can be evaluated at a continuous-time t to obtain the corresponding full 6-DoF pose of the camera at t . As opposed to discrete poses used in factor graph formulation [46], the pose generation function approach aligns with a continuous-time SLAM formulation [10].

3.1.1. eSfO motion model

The orbital motion model presented in eSfO [16] (see Fig. 2L) includes the following parameters:

- $\mathbf{c}^w \in \mathbb{R}^3$: center of the circular orbit in w .
- $r \in \mathbb{R}$: the radius of the orbit.
- $f \in \mathbb{R}$: the frequency at which the object rotates (Hz), i.e., $f = 1/t_{rev}$.
- $\mathbf{R}_0 \in \mathcal{SO}(3)$: orientation of the camera relative to Δ .
- $\hat{\mathbf{n}} \in \mathbb{R}^3$: the orbital axis, unit vector normal to Δ
- $\hat{\mathbf{u}} \in \mathbb{R}^3$ unit vector in Δ pointing towards the position of the camera at the start of recording, i.e., at time t_1 .

The position of the camera at any given time t is

$$\mathbf{t}^w(t; \theta) = r \cos(2\pi ft) \hat{\mathbf{u}} + r \sin(2\pi ft) \hat{\mathbf{v}} + \mathbf{c}^w, \quad (1)$$

where $\hat{\mathbf{v}} = \hat{\mathbf{n}} \times \hat{\mathbf{u}}$ and $\theta = \{\mathbf{c}^w, r, f, \hat{\mathbf{n}}, \hat{\mathbf{u}}\}$. Note that t is a *continuous* time value in the range $[t_1, t_{N_E}]$.

3.1.2. eSpinSLAM motion model

The eSfO motion model can be greatly simplified without loss of generality (see Fig. 2R) as follows:

- Center the orbit at the origin of w , i.e., set $\mathbf{c}^w = \mathbf{0}_{3 \times 1}$.
- Align $\hat{\mathbf{n}}$ and $\hat{\mathbf{u}}$ with the z and x axes of w , i.e., set $\hat{\mathbf{n}} = \hat{\mathbf{k}}$ and $\hat{\mathbf{u}} = \hat{\mathbf{i}}$.
- Recognizing that r is unobservable in the monocular case, fix r to an arbitrary value; see Table 1.

With the above simplification, (1) can be rewritten as

$$\mathbf{t}^w(t; f) = r \cos(2\pi ft) \hat{\mathbf{i}} + r \sin(2\pi ft) \hat{\mathbf{j}}. \quad (2)$$

The unknown motion parameters thus reduce to f and \mathbf{R}_0 . The continuous-time pose of the camera in the manifold of

rigid transformations $\mathbb{SE}(3)$ is given by

$$\mathbf{T}_w^t = \mathbf{T}_b^t \mathbf{T}_w^b, \quad (3)$$

which takes a point \mathbf{x}^w in the world frame w and projects it to the image plane at time t , and

$$\mathbf{T}_b^t = \begin{bmatrix} \hat{\mathbf{r}}^x & \hat{\mathbf{r}}^y & \hat{\mathbf{r}}^z & \mathbf{t}^w(t; f) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

positions the camera center on the circle and orients the camera's principal axis to point at \mathbf{c}^w , with $\mathbf{r}^z = \mathbf{c}^w - \mathbf{t}^w(t; f)$, $\hat{\mathbf{r}}^x = \hat{\mathbf{r}}^z \times \hat{\mathbf{k}}$ and $\hat{\mathbf{r}}^y = \hat{\mathbf{r}}^z \times \hat{\mathbf{r}}^x$. The rigid transformation

$$\mathbf{T}_w^b = \begin{bmatrix} \mathbf{R}_0 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (5)$$

rotates the camera relative to the orbital plane Δ to point towards the target and is not time-dependent.

3.2. Strengths and weaknesses of the motion model

The assumption of a circular orbit bounds the total number of parameters in eSpinSLAM to $4 + 3N_X$ (Sec. 3.1.2), even if the camera undergoes a perpetual orbit that generates an infinitely long event stream \mathcal{E} . The motion propagator (2) also enables continuous-time SLAM [10], which aligns well with the usage of an event sensor.

On the other hand, the apparent camera motion could be more complex than a stationary circular orbit, e.g., the orbit could drift, wobble, or precess. Precession of the objects orbital axis $\hat{\mathbf{n}}$, for example, leads to the rotation of the orbital plane Δ . The ability to execute eSpinSLAM (Sec. 4) online and long-term allows to track the orbit based on the most recent observations and account for transients in the orbit. Dealing with more complex motions (e.g., precession, change in angular velocity) is left as future work.

4. eSpinSLAM algorithm

The eSpinSLAM pipeline (Fig. 3) consists of two parts: the front-end, that abstracts the sensor stream \mathcal{E} into a set of feature track hypothesis, and the back-end, that operates on feature tracks to estimate the scene structure and motion parameters.

4.1. Front-end

Instead of considering the streaming nature of the event input, eSfO [16] assumes that the entire event stream is available for processing from the start, i.e. its a offline method. Such an approach is infeasible for an online incremental method such as eSpinSLAM. Therefore, we restructure the eSfO front-end to operate at the ‘‘event chunk’’ $\mathcal{C} \subset \mathcal{E}$ granularity. The eSpinSLAM front-end is organized into three event-chunk-driven tasks; see Fig. 3. The process begins with the camera driver pushing an event chunk to the

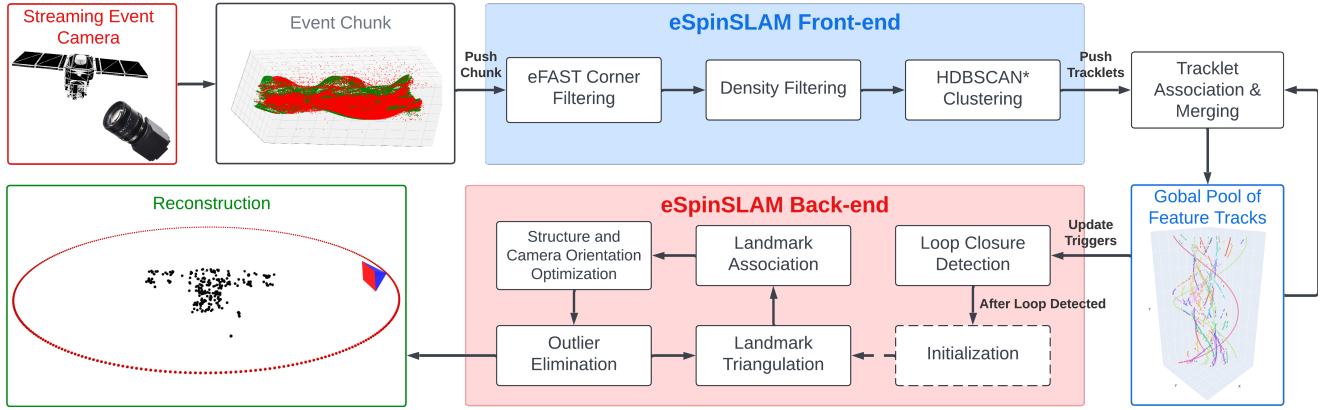


Figure 3. The eSpinSLAM pipeline: event chunks (\mathcal{C}) are taken as input to the method, which estimates a sparse point cloud of the object (\mathbf{X}^w) and its motion parameters (f, \mathbf{R}_0).

front-end after a set number of events have been triggered ($|\mathcal{C}| \approx 1,000$ for Prophesee’s Metavision SDK driver). The first stage applies eFAST corner detection [35] on \mathcal{C} to extract a set of corner events $\mathcal{F} \subseteq \mathcal{C}$. The second stage performs density filtering on \mathcal{F} [16] to generate a subset $\mathcal{G} \subseteq \mathcal{F}$. Density is defined as [16]

$$D(e_i) := \frac{1}{\lambda} \sum_{e_q \in \mathcal{F}_i} \mathbb{I}(p_i = p_q), \quad (6)$$

where $\mathcal{F}_i \subset \mathcal{F}$ consists of events within distance λ from e_i , p_q is the polarity of a neighboring event $e_q = \{t_q, x_q, y_q, p_q\}$, and \mathbb{I} returns 1 for matching polarities, zero otherwise. Density based filtering is required to normalize the variation in number of events generated for each polarity as the sensitivity for positive and negative events is different for the same intensity change [19]. A corner e_i with density score $D(e_i)$ less than the mean μ_D is removed. This enables identification of event corner clusters that can be reliably grouped together over time and space. Finally, HDBSCAN [32] is employed to spatio-temporally cluster the events in \mathcal{G} to generate a set of candidate tracklets $\mathcal{M} = \{\mathcal{M}_c \subseteq \mathcal{G}\}_{c=1}^{N_C}$, where each tracklet \mathcal{M}_c contains spatio-temporally adjacent events.

In eSpinSLAM, the tracklets \mathcal{M} from the latest event chunk are accumulated into and merged with a global pool of tracks $\mathcal{H} = \{\mathcal{H}_u\}_{u=1}^{N_H}$, where each $\mathcal{H}_u = \{e_v \in \mathcal{E}\}$ contains spatio-temporally adjacent events, potentially across multiple event chunks. To avoid the unbounded increase of \mathcal{H} , subsequent landmark association and track forgetting steps (Secs. 4.2.6 and 4.2.7) will be performed.

4.2. Back-end

As soon as the global pool of tracks \mathcal{H} is updated, the back-end of eSpinSLAM is triggered to re-estimate the orbit parameters (f, \mathbf{R}_0) and the sparse structure \mathbf{X}^w . f is required

for system initialization and is closely related to loop closure detection, as explained below.

4.2.1. Frequency estimation and loop closure

VSLAM systems accumulate drift over time which can only be corrected by detecting a revisit to a previously observed part of the world, *aka*, “loop closure detection”. It is essential for any long term VSLAM system to correctly close loops to prevent the growth of otherwise unbounded drift.

Contrary to the general VSLAM setting where loop closure detection is opportunistic, eSpinSLAM enjoys ample opportunities to close loops due to the cyclical nature of the problem. Indeed, the frequency f not only influences the propagation of the camera motion (2) and by extension structure estimation, it also informs when the camera completes a revolution and previous object features are re-observed. Determining f is equivalent to loop closure detection in eSpinSLAM.

We estimate the period $t_{rev} = 1/f$ by finding the temporal shift that leads to maximal spatial alignment. The cumulative events that exist in the current global pool of tracks \mathcal{H} are

$$\mathcal{A} = \cup_{u=1}^{N_H} \mathcal{H}_u = \{e_k \in \mathcal{E} \mid \exists u \text{ s.t. } e_k \in \mathcal{H}_u\}. \quad (7)$$

Defining the “midlife” of \mathcal{H} as

$$\hat{\tau} = 0.5 \left[\left(\min_{k=1}^{|\mathcal{A}|} t_k \right) + \left(\max_{k=1}^{|\mathcal{A}|} t_k \right) \right], \quad (8)$$

we partition \mathcal{A} into two subsets

$$\mathcal{A}_\ell = \{e_m \in \mathcal{A} \mid t_m \leq \hat{\tau}\}, \quad \mathcal{A}_r = \{e_n \in \mathcal{A} \mid t_n > \hat{\tau}\},$$

based on whether the events were triggered before or after the midlife. The orbital period t_{rev} is estimated as

$$t_{rev}^* = \arg \min_{t_{rev}} \sum_{m=1}^{|\mathcal{A}_\ell|} \min_{n=1}^{|\mathcal{A}_r|} \left\| \begin{bmatrix} x_m \\ y_m \\ t_m + t_{rev} \end{bmatrix} - \begin{bmatrix} x_n \\ y_n \\ t_n \end{bmatrix} \right\|_2 \quad (9)$$

using a 1D variant of the iterative closest point (ICP) [3] algorithm. The estimated frequency is thus $f^* = 1/t_{rev}^*$.

This process is executed whenever \mathcal{H} is updated. Loop detection is deemed successful when the last \aleph frequency estimations have a sample standard deviation of less than σ .

4.2.2. Structure and camera orientation optimization

The continuous-time formulation of eSpinSLAM obviates the need to accumulate event frames, and allows each event $e_v \in \mathcal{H}_u$ in each track \mathcal{H}_u to contribute to the estimation. For each $e_v = (t_v, x_v, y_v, p_v)$, a *virtual* camera with pose

$$\mathbf{T}_w^{t_v} = \begin{bmatrix} \mathbf{R}_w^{t_v} & \mathbf{t}_w^{t_v} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (10)$$

is defined based on the motion model (3) with the frequency fixed to f^* . Recall that (10) also depends on \mathbf{R}_0 . Then, by regarding each track \mathcal{H}_u as events generated by the same world point \mathbf{x}_u^w , the sum of reprojection errors is minimized to estimate the structure and camera orientation

$$\min_{\mathbf{R}_0, \{\mathbf{x}_u^w\}} \sum_{u=1}^{N_H} \sum_{v=1}^{|\mathcal{H}_u|} \rho \left(\left\| \pi(\mathbf{K}\mathbf{R}_w^{t_v} \mathbf{x}_u^w + \mathbf{K}\mathbf{t}_w^{t_v}) - \begin{bmatrix} x_v \\ y_v \end{bmatrix} \right\|_2 \right), \quad (11)$$

where $\pi(\cdot)$ is the perspective projection function, \mathbf{K} is the intrinsic calibration matrix, and ρ is an M-estimator to provide robustness against outliers. In total, $3 + 3N_H$ variables are optimized, which is solved using GTSAM [12]. Optimization runs every ϵ/f seconds to ensure sufficient new information is available for state update.

4.2.3. Initialization

Unlike eSfO [16] which relies on COLMAP [43] for bootstrapping, eSpinSLAM does not require external routines for initialization. The frequency can be estimated as in Sec. 4.2.1 from the tracks alone. In the very first run of (11), the world points are initialized to zero, *i.e.*, $\{\mathbf{x}_u^w = \mathbf{0}_{3 \times 1}\}_{u=1}^{N_H}$, and \mathbf{R}_0 is set to $\mathbf{I}_{3 \times 3}$, while f is fixed to f^* .

4.2.4. Landmark triangulation

In subsequent runs of the back-end, triggered by updates to \mathcal{H} , the world points for newly acquired or updated tracks are initialized by multiview triangulation [22], based on virtual camera poses that are propagated forward using the motion model (3), f^* and the most recently estimated \mathbf{R}_0 .

4.2.5. Outlier elimination

Tracks with fewer than N_{match} events are considered uninformative and discarded. We also eliminate tracks whose landmarks have a mean reprojection error greater than γ . Assuming that the scene contains a single object with no background, points that are beyond a radius of r from $\mathbf{c}^w = \mathbf{0}_{3 \times 1}$ are also discarded as outliers. Note that the latter assumption is due to the data collection routine (see Sec. 5) and not a weakness of our method.

4.2.6. Landmark Association

Tracks originating from a single world point might be disconnected due to self-occlusion, leading to the reconstruction of multiple world points. To reduce drift, such world points need to be fused. When two reconstructed world points $\mathbf{x}_{u_1}^w$ and $\mathbf{x}_{u_2}^w$ are within a predefined threshold δ_{fuse} , the corresponding tracks are merged such that $\mathcal{H}_{u_2} = \mathcal{H}_{u_1} \cup \mathcal{H}_{u_2}$. This fuses the two world points in the next optimization.

4.2.7. Enabling infinite lifetime

To avoid an unbounded pool of tracks \mathcal{H} during infinite operation and reduce the influence of stale measurements, old tracks are progressively removed from \mathcal{H} . We calculate the age of a track \mathcal{H}_u as the time elapsed since the most recent event in the track, *i.e.*,

$$\text{age}(\mathcal{H}_u) = \left(\max_{k=1}^{|\mathcal{A}|} t_k \right) - \left(\max_{v=1}^{|\mathcal{H}_u|} t_v \right). \quad (12)$$

Tracks with age older than $\sigma_{forget} t_{rev}$ are discarded where σ_{forget} is a system hyper-parameter. Some tracks might continuously be observed and could never become stale. We additionally restrict the number of events from a particular track that can be retained. A circular buffer of a pre-defined length σ_{retain} per track is maintained to which new events are added. σ_{retain} is a hyper-parameter of the system.

4.3. Degeneracy

Degeneracy occurs in eSpinSLAM when the optical axis of the camera is parallel to the axis of rotation $\hat{\mathbf{n}} = \hat{\mathbf{k}}$, *e.g.*, vertically downward pointing camera. This is analogous to a purely rotating camera in SfM. For brevity, we assume no degeneracy in our application scenario.

5. Experiments

Experiments are presented to demonstrate the efficacy of eSpinSLAM in frequency estimation, throughput, point cloud reconstruction, and long-term operation. All experiments were run on a machine with an Intel i5-8400 CPU (which has six cores) and 32GB of RAM. The live demonstration (eSpinSLAM_Demo.mp4) used a Prophesee EVK4 event camera.

5.1. Dataset

eSpinSLAM is formulated to recover the structure and orbital parameters for a spinning object viewed by a static camera. Therefore, we evaluate eSpinSLAM on the TOPSPIN [15] dataset which contains six spinning objects on a turntable, at three rotational speeds, observed from a static event camera.

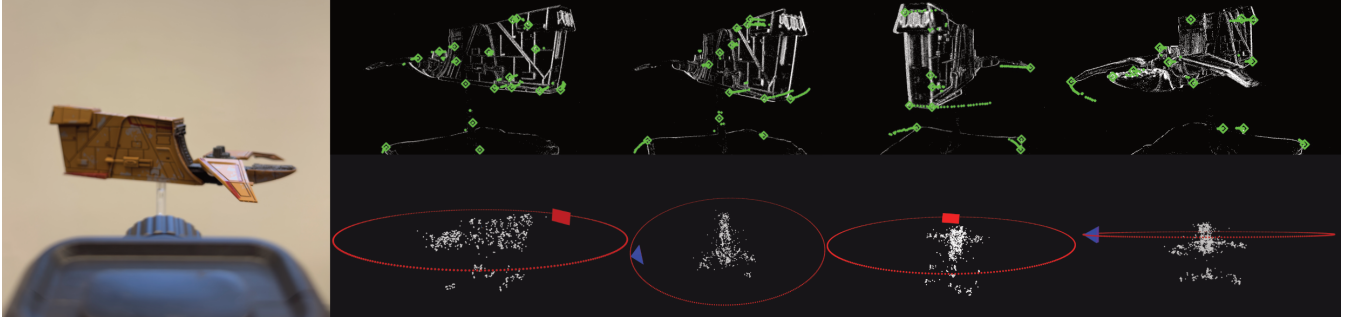


Figure 4. Sample images of eSpinSLAM online operation: while the camera observes the spinning object (**left**), which is equivalent to the camera orbiting the static object [16], features are tracked by the front-end (**top row**) which are then used by the back-end to estimate the orbit and reconstruct the point cloud (**bottom row**). See attached supplementary video (eSpinSLAM_Demo.mp4).

Standard deviation f estimation σ	10^{-3}
Frequency estimation samples \aleph	20
Inlier threshold for ρ	2
Forgetting threshold σ_{forget}	3
Circular buffer length σ_{retain}	10^3
Track length N_{match}	3
Orbit radius r	10
Maximum landmark mean reprojection error γ	10
Landmark merge radius δ_{fuse}	0.01
Cadence of optimization ϵ	0.1

Table 1. Hyperparameters of eSpinSLAM.

5.2. Hyperparameter settings

Table 1 lists the hyperparameters and their values used in the experiments.

5.3. System throughput

Event-based sensing is asynchronous by design and lacks the notion of a “frame rate”. Moreover, event rate varies with scene motion, texture, lighting conditions and camera resolution – making performance characterization difficult. Previous methods report metrics such as a “real time factor” (compute time divided by the duration of the events) to quantify performance, which is highly dependent on the aforementioned factors. Instead, we argue that throughput – the number of events processed per unit time – being invariant to these factors, is a better performance metric. Throughput measures a pipeline’s event processing capacity enabling comparison between methods and complexity between dataset scenes.

Fig. 5 depicts the throughput for various algorithms (without any downstream processing) and datasets. The higher throughput of both ETC and eSpinSLAM permits high speed processing on the TOPSPIN dataset, whilst HASTE struggles due to low throughput. eSpinSLAM achieves consistent high throughput on TOPSPIN scenes (Fig. 7).

5.4. Frequency estimation

Frequency plays a critical role in eSpinSLAM and underpins reconstruction quality. We compare against FFT-based frequency estimation presented in eSfO [16]. Fig. 6 demonstrates the advantage of loop closure based f estimation where the frequency is recovered with high precision and high accuracy. Since f directly affects the reconstruction quality, we demonstrate its effects on reprojection and model-alignment error in Table 3.

5.5. Comparison against other methods

Table 2 presents extensive comparisons against the closest approach (eSfO [16]) and demonstrated consistent improvement. eSfO operates offline, while eSpinSLAM operates incrementally online and achieves about an order of magnitude improvement in CAD model alignment accuracy. We also compare against the original baseline (COLMAP + ETC Tracks) from [16]. Comparison against other event-based methods such as EVO [42] was attempted but failed due to the restricted motion setting of eSpinSLAM: EVO needs a planar fronto-parallel scene for initializa-

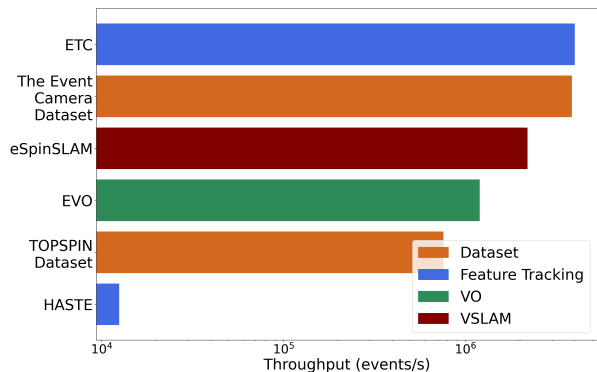


Figure 5. Average throughput (**log scale**): event rate for various algorithms; our ETC tracker front-end, eSpinSLAM, EVO [42], HASTE [2], and datasets; Event Camera Dataset [24] and TOPSPIN [15].

Method	Hubble	SOHO	TDRS	Camera	Dualshock	Switch
	Reprojection Error (px) ↓					
COLMAP + ETC Tracks	2.12 ± 1.51	2.32 ± 0.59	2.22 ± 0.59	2.70 ± 1.79	2.75 ± 0.82	2.25 ± 1.21
eSfO [16]	8.70 ± 8.31	18.25 ± 10.23	11.02 ± 8.21	5.78 ± 3.01	5.12 ± 4.266	23.91 ± 17.03
EVO [42]	–	–	–	–	–	–
eSpinSLAM	3.25 ± 3.01	2.99 ± 3.17	1.54 ± 0.20	1.98 ± 0.15	2.19 ± 0.91	1.94 ± 0.35
	ICP Alignment RMSE (mm) ↓					
COLMAP + ETC Tracks	1.43 ± 0.59	7.52 ± 4.67	7.61 ± 5.72	1.89 ± 0.57	5.71 ± 5.98	6.42 ± 5.45
eSfO [16]	2.08 ± 0.411	4.20 ± 3.21	3.95 ± 2.39	0.51 ± 0.34	2.97 ± 2.88	5.52 ± 5.56
EVO [42]	–	–	–	–	–	–
eSpinSLAM	0.92 ± 0.71	0.56 ± 0.41	0.66 ± 0.76	0.75 ± 0.57	0.93 ± 1.02	0.83 ± 0.66

Table 2. Comparison of Reprojection Error and ICP Alignment RMSE for various methods on the TOPSPIN [16] dataset. The results are collated for each model present in TOPSPIN, where the mean and standard deviation for each metric are reported. EVO [42] failed to initialize for all sequences.

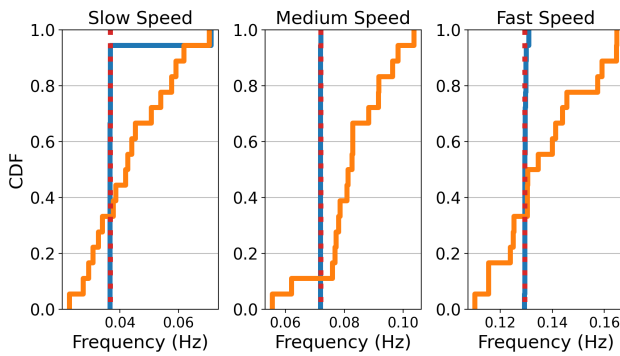


Figure 6. Cumulative distribution function (CDF) plots of frequency estimation using loop closure (eSpinSLAM) in blue and the FFT-based method from eSfO [16] in orange, for scenes in the TOPSPIN dataset [16]. Ground truth f is presented in red. Loop closure based estimates are both accurate and precise, with nearly all estimates virtually identical to the ground truth.

tion, which is not compatible with eSpinSLAM’s problem setting. State-of-the-art event-based VO methods such as DEVO [30]—which only estimate the camera’s trajectory—are also not viable as there is no common basis for comparison: there is no reconstructed structure or estimated object motion with which to compare.

Reprojection Error Reprojection error quantifies the alignment between the set of tracks and the reconstructed world points. In Table 2, eSpinSLAM leads to the least reprojection error; however, it is followed closely by tracks from ETC tracker used as input to a state-of-the-art bundle adjustment pipeline (COLMAP [43]), which is a general purpose SLAM backend and unaware of orbital motion. As expected, loop closure reduces reprojection error for eSpinSLAM – indicating accurate frequency estimation enhances consistency between observations and world points. Finally, eSfO [16] accumulates the largest reprojection error, indicating potential weakness in its FFT-based

frequency estimation and more complex parameterization.

CAD model alignment error To measure the accuracy of the recovered structure, estimated world points are aligned using the ICP [3] algorithm to the corresponding CAD models and alignment error is computed (Table 2). eSpinSLAM demonstrates better accuracy than eSfO thanks to precise frequency estimation and updated formulation. Furthermore, the low reprojection error of COLMAP + ETC tracks does not translate to a higher-quality reconstruction. The reconstructions are internally consistent but the recovered structure deviates from the real-world. Despite the higher reprojection error for eSfO, the reconstructions are of generally close to the CAD models.

5.6. Incremental reconstruction

The incremental operation of eSpinSLAM is demonstrated using the evolution of reprojection error over time for a sample scene (Fig. 8). Reprojection error decreases as the loop

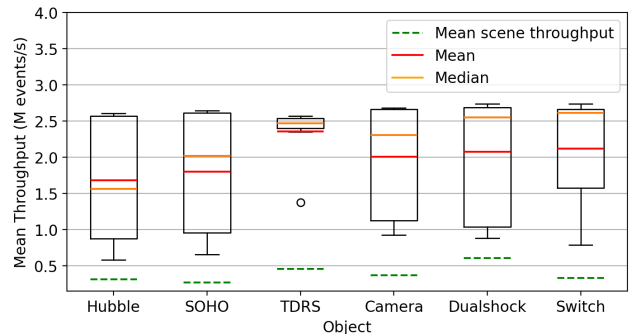


Figure 7. eSpinSLAM throughput on TOPSPIN (million events/s). Mean scene throughput indicates the mean throughput of all the scenes for the object. System throughput is consistently above the mean scene throughput for all objects—indicating real-time performance.

Setting	Hubble	SOHO	TDRS	Camera	Dualshock	Switch
	Reprojection Error (px) ↓					
FFT f estimation	5.61 ± 7.93	5.81 ± 5.58	3.41 ± 3.56	2.67 ± 1.34	2.99 ± 2.35	4.39 ± 5.11
No factor graph f constraint	6.01 ± 3.79	17.16 ± 9.03	4.29 ± 2.33	2.89 ± 3.11	5.27 ± 7.75	12.38 ± 6.91
No landmark association	2.11 ± 1.93	3.68 ± 0.71	1.18 ± 0.71	0.42 ± 0.80	1.67 ± 1.86	1.09 ± 1.50
eSpinSLAM	3.25 ± 3.01	2.99 ± 3.17	1.54 ± 0.20	1.98 ± 0.15	2.19 ± 0.91	1.94 ± 0.35
	ICP Alignment RMSE (mm) ↓					
FFT f estimation	3.17 ± 1.82	3.51 ± 2.44	5.95 ± 1.62	2.13 ± 1.05	4.21 ± 1.33	3.16 ± 2.03
No factor graph f constraint	8.14 ± 7.29	15.49 ± 3.60	5.11 ± 2.49	1.12 ± 1.27	4.35 ± 0.74	9.38 ± 3.29
No landmark association	13.74 ± 3.92	12.47 ± 11.16	19.61 ± 8.31	9.19 ± 5.10	7.72 ± 3.27	13.11 ± 3.30
eSpinSLAM	1.92 ± 0.71	0.56 ± 0.41	0.66 ± 0.76	0.75 ± 0.57	0.93 ± 1.02	1.83 ± 0.66

Table 3. Ablation study on the effect of method components within eSpinSLAM. The experiments were conducted on the TOPSPIN [16] dataset; results are collated by each model present in the dataset, where the mean and standard deviation for each metric are reported.

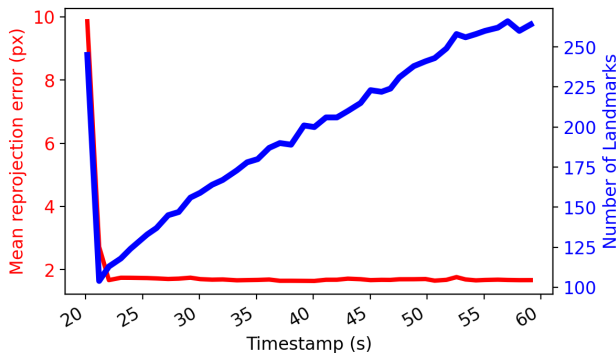


Figure 8. Mean reprojection error and number of world points over time for the soho-perpendicular-fast scene.

closure is detected and f is determined. Concurrently, the number of world points decreases as duplicates are fused together. Whilst the number of reconstructed points steadily increases over time, reprojection error stays consistently low – demonstrating the ability of eSpinSLAM to operate incrementally online. It should be noted that while initially some features are rejected due to their bad quality, as the motion parameters are better estimated, an increasing number of world points (Fig. 8) are incorporated into the estimation over time, all whilst the reprojection errors remain low. This demonstrates the incremental reconstruction capabilities of the eSpinSLAM algorithm.

6. Ablation study

To quantify the contribution of key components in eSpinSLAM’s design, we conducted an ablation study using the TOPSPIN dataset [16]. Specifically, we investigated the effects of (i) estimating frequency via FFT instead of Sec. 4.2.1; (ii) disabling the factor graph constraint on frequency during optimization, so that f is not fixed to f^* and can be optimized; and (iii) omitting landmark association during long-term operation. Table 3 presents a com-

parative summary of reprojection errors and ICP alignment RMSE for each setting. The complete eSpinSLAM pipeline achieves the best performance, validating the necessity of each individual component.

Replacing the loop closure-based frequency estimation with FFT (as in *eSfo* [16]) leads to a notable increase in both reprojection and alignment errors. The FFT estimation method is sensitive to noise and the resolution of the frequency estimates is limited by the Nyquist–Shannon sampling theorem. These are limitations that are not present in our loop closure method.

Disabling the factor graph constraint on frequency during optimization results in the largest degradation. Both reprojection and alignment errors increase significantly, particularly for objects with multiple axes of symmetry (SOHO and Switch). This stems from ambiguities in motion interpretation — without a frequency bound, multiple local minima exist in the frequency space due to frequency harmonics and object symmetry presenting locally optimal solutions.

Eliminating landmark association severely increases alignment errors while maintaining only a modest increase in reprojection error. While image-space reprojection error remains locally accurate, duplicate world points accumulate due to track fragmentation caused by occlusion or sensor noise. Landmark association can mitigate this by merging landmarks from related but time separated tracks.

7. Conclusions

We presented a novel spinning object reconstruction formulation, eSpinSLAM, that enables high throughput, online recovery of the target’s structure and continuous time camera pose estimation utilizing a single event camera. eSpinSLAM incorporates loop closure detection, feature track management, and world point fusion to enable infinite-time operation. Experiments on real data demonstrate improved structure estimation. Future work will explore extending the formulation to more complex object motion *e.g.* variation of the spin rate and procession of the spin axis.

References

- [1] Saeed Afshar, Andrew Peter Nicholson, Andre Van Schaik, and Gregory Cohen. Event-based object detection and tracking for space situational awareness. *IEEE Sensors Journal*, 20(24):15117–15132, 2020. 2
- [2] Ignacio Alzugaray and Margarita Chli. Haste: multi-hypothesis asynchronous speeded-up tracking of events. In *31st British Machine Vision Virtual Conference (BMVC 2020)*, page 744. ETH Zurich, Institute of Robotics and Intelligent Systems, 2020. 6
- [3] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 5, 7
- [4] William Chamorro, Joan Sola, and Juan Andrade-Cetto. Event-based line slam in real-time. *IEEE Robotics and Automation Letters*, 7(3):8146–8153, 2022. 2
- [5] Dejjia Che, Zixuan Zheng, and Jianping Yuan. An innovate filter for space robots to unfirmly capture tumbling targets. *International Journal of Adaptive Control and Signal Processing*, 36(2):282–299, 2022. 1
- [6] Bo Chen, Jiewei Cao, Alvaro Parra, and Tat-Jun Chin. Satellite pose estimation with deep landmark regression and non-linear pose refinement. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019. 2
- [7] Peiyu Chen, Weipeng Guan, and Peng Lu. Esvio: Event-based stereo visual inertial odometry. *IEEE Robotics and Automation Letters*, 8(6):3661–3668, 2023. 2
- [8] Zezhou Cheng, Matheus Gadelha, and Subhransu Maji. Accidental turntables: Learning 3d pose by watching objects turn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2113–2122, 2023. 1
- [9] Tat-Jun Chin, Samya Bagchi, Anders Eriksson, and Andre Van Schaik. Star tracking using an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2
- [10] Giovanni Cioffi, Titus Cieslewski, and Davide Scaramuzza. Continuous-time vs. discrete-time vision-based slam: A comparative study. *IEEE Robotics and Automation Letters*, 7(2):2399–2406, 2022. 3
- [11] Gregory Cohen, Saeed Afshar, Brittany Morreale, Travis Bessell, Andrew Wabnitz, Mark Rutten, and André van Schaik. Event-based sensing for space situational awareness. *The Journal of the Astronautical Sciences*, 66:125–141, 2019. 2
- [12] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. Foundations and Trends in Robotics, Vol. 6, 2017. 5
- [13] Mehregan Dor and Panagiotis Tsiotras. Orb-slam applied to spacecraft non-cooperative rendezvous. In *2018 Space Flight Mechanics Meeting*, page 1963, 2018. 2
- [14] Mehregan Dor, Katherine A Skinner, Travis Driver, and Panagiotis Tsiotras. Visual slam for asteroid relative navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2066–2075, 2021. 2
- [15] Ethan Elms. TOPSPIN: daTaset Of sPinning objectS with neuromorphic vIsioN, 2024. <https://doi.org/10.5281/zenodo.10884693>. 5, 6
- [16] Ethan Elms, Yasir Latif, Tae Ha Park, and Tat-Jun Chin. Event-based structure-from-orbit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19541–19550, 2024. 1, 2, 3, 4, 5, 6, 7, 8
- [17] Matthew A. Estrada, Benjamin Hockman, Andrew Bylard, Elliot W. Hawkes, Mark R. Cutkosky, and Marco Pavone. Free-flyer acquisition of spinning objects with gecko-inspired adhesives. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4907–4913, 2016. 1
- [18] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conrath, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1
- [19] Rui Graca, Brian McReynolds, and Tobi Delbruck. Shining light on the dvs pixel: A tutorial and discussion about biasing and optimization, 2023. 1, 4
- [20] Weipeng Guan and Peng Lu. Monocular event visual inertial odometry based on event-corner using sliding windows graph-based optimization. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2438–2445. IEEE, 2022. 2
- [21] Shuang Guo and Guillermo Gallego. CMax-SLAM: Event-based rotational-motion bundle adjustment and SLAM system using contrast maximization. *IEEE Transactions on Robotics*, 40:2442–2461, 2024. 2
- [22] Richard Hartley and Andrew Zisserman. *Multiple view geometry in Computer Vision*. Cambridge University Press, 2019. 5
- [23] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza. Event-aided direct sparse odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5781–5790, 2022. 2
- [24] Sumin Hu, Yeeun Kim, Hyungtae Lim, Alex Junho Lee, and Hyun Myung. ecdt: Event clustering for simultaneous feature detection and tracking. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3808–3815. IEEE, 2022. 6
- [25] Dario Izzo, Alexander Hadjiivanov, Domink Dold, Gabriele Meoni, and Emmanuel Blazquez. Neuromorphic computing and sensing in space. *arXiv preprint arXiv:2212.05236*, 2022. 1
- [26] Mohsi Jawaid, Ethan Elms, Yasir Latif, and Tat-Jun Chin. Towards bridging the space domain gap for satellite pose estimation using event sensing. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 2
- [27] Mohsi Jawaid, Rajat Talak, Luca Carlone, and Tat-Jun Chin. Test-time certifiable self-supervision to bridge the sim2real gap in event-based satellite pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. 2
- [28] Guang Jiang, Hung tat Tsui, Long Quan, and A. Zisserman. Geometry of single axis motions using conic fitting. *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1343–1348, 2003. [1](#)
- [29] Mate Kisantal, Sumant Sharma, Tae Ha Park, Dario Izzo, Marcus Märtens, and Simone D’Amico. Satellite pose estimation challenge: Dataset, competition design, and results. *IEEE Transactions on Aerospace and Electronic Systems*, 56(5):4083–4098, 2020. [2](#)
- [30] Simon Klenk, Marvin Motzet, Lukas Koestler, and Daniel Cremers. Deep event visual odometry. In *International Conference on 3D Vision, 3DV 2024, Davos, Switzerland, March 18-21, 2024*, pages 739–749. IEEE, 2024. [2](#), [7](#)
- [31] Roberto Lampariello and Gerd Hirzinger. Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5652–5659, 2013. [1](#)
- [32] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. [4](#)
- [33] S. McLeod, G. Meoni, D. Izzo, A. Mergy, D. Liu, Y. Latif, I. Reid, and T.-J. Chin. Globally Optimal Event-Based Divergence Estimation for Ventral Landing. In *Computer Vision – ECCV 2022 Workshops*, 2022. [2](#)
- [34] P Mithun, Harit Pandya, Ayush Gaud, Suril V. Shah, and K. Madhava Krishna. Image based visual servoing for tumbling objects. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2901–2908, 2018. [1](#)
- [35] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. Fast event-based corner detection. In *British Machine Vision Conference (BMVC)*, 2017. [4](#)
- [36] Yonhon Ng, Yasir Latif, Tat-Jun Chin, and Robert Mahony. Asynchronous kalman filter for event-based star tracking. In *Computer Vision – ECCV 2022 Workshops*, pages 66–79. Springer, 2022. [2](#)
- [37] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations. *Progress in Aerospace Sciences*, 93: 53–72, 2017. [2](#)
- [38] Tae Ha Park and Simone D’Amico. Online supervised training of spaceborne vision during proximity operations using adaptive kalman filtering. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11744–11752. IEEE, 2024. [2](#)
- [39] Vincenzo Pesce, Roberto Opromolla, Salvatore Sarno, Michèle Lavagna, and Michele Grassi. Autonomous relative navigation around uncooperative spacecraft based on a single camera. *Aerospace Science and Technology*, 84:1070–1080, 2019. [2](#)
- [40] Antoine Petit. *Robust visual detection and tracking of complex objects: applications to space autonomous rendez-vous and proximity operations*. PhD thesis, Université de Rennes, 2013. [2](#)
- [41] Arunkumar Rathinam, Haytam Qadadri, and Djamila Aouada. Spades: A realistic spacecraft pose estimation dataset using event sensing. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11760–11766, 2024. [2](#)
- [42] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2016. [2](#), [6](#), [7](#)
- [43] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. [2](#), [5](#), [7](#)
- [44] O. Sikorski, D. Izzo, and Meoni G. Event-Based Spacecraft Landing Using Time-To-Contact. pages 1941–1950. Proceedings of the IEEE/CVF Conference on Computer Vision, 2021. [2](#)
- [45] Jianing Song, Duarte Ronda, and Nabil Aouf. Deep learning-based spacecraft relative navigation methods: A survey. *Acta Astronautica*, 191:22–40, 2022. [2](#)
- [46] Brent E. Tweddle, Alvar Saenz-Otero, John J. Leonard, and David W. Miller. Factor graph modeling of rigid-body dynamics for localization, mapping, and parameter estimation of a spinning object in space. *Journal of Field Robotics*, 32(6):897–933, 2015. [1](#), [3](#)
- [47] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschäfer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018. [2](#)
- [48] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5816–5824, 2017. [2](#)
- [49] Yiman Zhu, Zhicai Dong, Meng Ding, and Yu Guo. Rotational dynamics and capture strategy for space tumbling targets. In *2024 43rd Chinese Control Conference (CCC)*, pages 4320–4325, 2024. [1](#)