Event-based Structure-from-Orbit

Ethan Elms[†] Yasir Latif[†] Tae Ha Park[‡] Tat-Jun Chin[†*]
[†]The University of Adelaide [‡]Stanford University
{ethan.elms,yasir.latif,tat-jun.chin}@adelaide.edu.au[†],tpark94@stanford.edu[‡]

Abstract

Event sensors offer high temporal resolution visual sensing, which makes them ideal for perceiving fast visual phenomena without suffering from motion blur. Certain applications in robotics and vision-based navigation require 3D perception of an object undergoing circular or spinning motion in front of a static camera, such as recovering the angular velocity and shape of the object. The setting is equivalent to observing a static object with an orbiting camera. In this paper, we propose event-based structure-from-orbit (eSfO), where the aim is to simultaneously reconstruct the 3D structure of a fast spinning object observed from a static event camera, and recover the equivalent orbital motion of the camera. Our contributions are threefold: since state-of-the-art event feature trackers cannot handle periodic self-occlusion due to the spinning motion, we develop a novel event feature tracker based on spatio-temporal clustering and data association that can better track the helical trajectories of valid features in the event data. The feature tracks are then fed to our novel factor graph-based structure-from-orbit back-end that calculates the orbital motion parameters (e.g.*, spin rate, relative rotational axis) that minimize the reprojection error. For evaluation, we produce a new event dataset of objects under spinning motion. Comparisons against ground truth indicate the efficacy of eSfO.*

1. Introduction

Three-dimensional (3D) perception is a fundamental vision capability [37, 45]. Recent works have focused on the use of neuromorphic event sensors [19] for 3D perception tasks, such as visual odometry (VO) [31, 48], structure-from-motion (SfM) and simultaneous localization and mapping (SLAM) [35, 47]. Event sensors offer several advantages over conventional cameras, such as high temporal resolution, low power and low data rate asynchronous sensing. Event sensors also offer a higher dynamic range, enabling them to see more details in difficult lighting conditions.

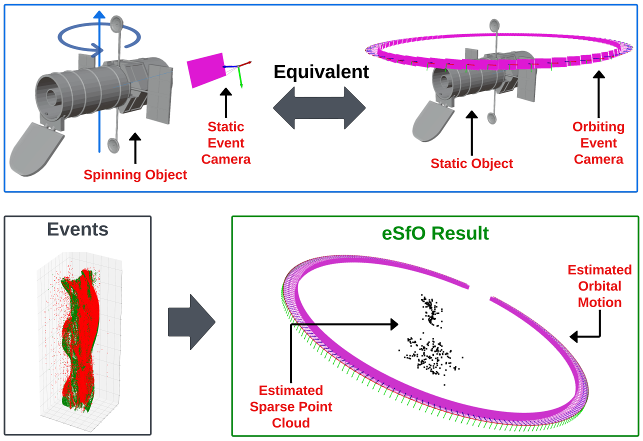


Figure 1. In eSfO, we exploit the equivalence between a static event camera observing a spinning object, and an orbiting event camera observing a static object. This enables us to jointly estimate the motion parameters (*e.g.*, spin rate, rotational axis relative to the camera), as well as the sparse structure of the object.

However, the advantages of event sensors come with certain limitations. Firstly, due to the asynchronous nature of events, the idea of a consistent local neighborhood for a pixel across multiple views no longer holds. Moreover, image gradients – a fundamental invariant in intensity images often used as basis for feature detection, description, and tracking – are not observable in event data. This makes feature detection and tracking challenging, which represents a major obstacle towards event-only SLAM systems.

In this work, we formulate the task of *event-based Structure-from-Orbit (eSfO)*, which entails the reconstruction of an object undergoing circular motion while being observed by a static event camera, and jointly estimating the motion of the object; see Fig. 1. An object is undergoing a circular motion if it is rotating about a fixed axis, *a.k.a.* spinning. Since observing a spinning object from a static camera is mathematically equivalent to observing a static object from an orbiting camera, the motion of the spinning object can be recovered as the motion of the orbiting camera. Using an event sensor to observe the spinning

[*] SmartSat CRC Professorial Chair of Sentient Satellites.

target is compelling, since the effect of motion blur is alleviated, especially in cases with high angular velocity.

SfM of a spinning object, usually placed on a turntable, has been studied previously [17, 18]. However, these early works assume known angular rates and solve the task of structure recovery only. Nonetheless, various e-commerce, multimedia and augmented reality applications can benefit from such a 3D reconstruction approach [1, 39, 40].

Outside of turntable-induced rotations, spinning objects can be observed naturally in the world, such as race-cars spinning in place [8] allowing opportunistic observation of the complete structure. Space is another domain abundant with naturally spinning objects, from as large as celestial bodies such as planets and asteroids to as small as satellites and space debris. For instance, as most small bodies such as asteroids tend to spin about the major principal axis, many works leveraged the single-axis spin assumption for 3D reconstruction purposes [4, 13, 14]. Such motion characteristics are also observed on some man-made objects freely rotating in inertial space [46]. Indeed, Kaplan et al. [30] reported in 2010 that there are over 100 expired satellites in geosynchronous orbits (GEO) spinning at high angular rates (tens of RPM) as they retain the angular momentum from spin-stabilization during service. Extracting the shape and motion parameters of such spinning objects in space can facilitate vision-based spacecraft navigation [13, 14] and formation flying [23].

Contributions Monocular eSfO is challenging since current event-based feature detection and tracking methods are not reliable on spinning objects that periodically self-occlude. Moreover, the underlying structure-from-orbit (SfO) problem imposes more constraints than SfM, and SfO has not been satisfactorily tackled in the literature. Our work addresses the difficulties by contributing:

- A novel eSfO formulation that takes into account the problem structure induced by a spinning object (Sec. 3).
- A novel event feature detection and tracking mechanism designed for the spinning object case (Sec. 4).
- A novel factor graph-based optimization back-end that efficiently estimates sparse structure and orbit motion parameters from event feature tracks (Sec. 5).
- A monocular event sensor dataset and benchmark for the eSfO problem (Sec. 6).

2. Related Work

2.1. Event-based Vision

Event sensors have been incorporated into various SLAM and VO pipelines, due to their low data rate and high dynamic range [26]. They are normally paired with other sensors such as image sensors and IMUs [26, 47] for fast motion scenarios. The primary benefit of such an approach is to allow feature detection in the image space and feature tracking during the blind-time of the image sensor. To resolve feature associations, EVO [43] combines two event sensors in a stereo configuration to enable feature matching across the two sensors. Similarly, EDS [24] pairs a monocular camera with an event sensor for event-aided direct sparse odometry. Various learning based approaches also use event cameras for optical flow estimation [5, 38, 49].

Despite its usefulness in robotics and vision-based navigation, *event-only* SfO has not received a lot of attention in the literature. Rebecq et al. [44] demonstrated visual-inertial odometry (VIO) with an event camera on a "spinning leash" sequence, which was obtained by spinning an event camera attached to a leash. However, the single result was evaluated only qualitatively [44, Fig. 7]. Moreover, being VIO, their method requires an onboard IMU.

2.2. Event-based Feature Tracking

The first step in many VO pipelines is the feature detection and subsequent tracking across frames, which has proven to be difficult for the event-only case. Methods for corner detection [25, 32, 34] have repurposed image feature detection methods to the event sensor. For feature tracking, methods either assume that known detections in the form of templates [3, 21] or use other sensors (such as images) for feature detection [20]. Several methods have also explored learning based feature detection and tracking [9, 20, 50] mechanisms to compensate for the lack of image gradients and consistent pixel neighborhood across multiple view.

A serious challenge posed by a spinning object is the periodic self-occlusion unavoidably affecting features on the surface of the object. As we will show in Sec. 7, this leads to poorer quality tracks by existing methods. By reasoning over the spatio-temporal space of the event sensor to detect clusters of corner events that occur together and connect them over time to arrive at meaningful feature tracks, our method produces more accurate tracks.

2.3. SfM for Single-axis Rotation

The SfO problem explored in this work finds its roots in the earlier days of research into the SfM problem of "single-axis rotation" – in which different methods explored reconstruction of objects placed on a turntable with a known rotational rate [17, 18]. These were further extended to include auto-calibration and recovery of camera poses along with the object reconstruction [15]. The problem has made a reappearance recently in an "in the wild" context [8] where opportunistic turntable sequences allow object coverage due to the rotational motion using frame-based cameras. While their approach relies heavily on learning based techniques to form an implicit model of the object, our focus is more on geometric methods that require no prior training. The method closest in application to ours is that of

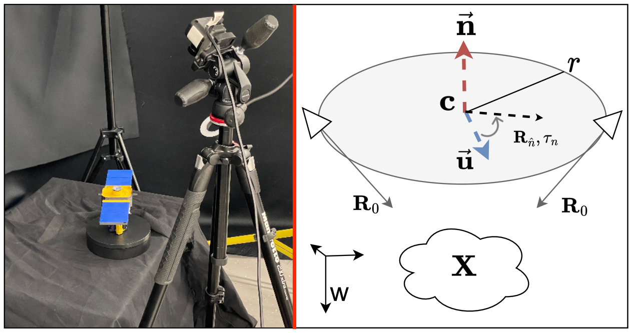


Figure 2. **L**: Problem setup. **R**: Visualization of eSfO parameters in the orbit view of the problem. w is an arbitrary world frame.

Chen et al. [7], who presented a dense reconstruction system using simulated event input, similar to a turntable sequence. The method generates dense object representations using a learning based framework, however, our focus is on estimating the rotational rate and axis as opposed to reconstructing a detailed model of the object. Similarly, the work of Baudron et al. [6] follows a shape from silhouette approach using an event sensor, mainly reconstructing the object shape. Although Tweddle et al. [46] formulate a similar problem in a similar setting, their work differs in that it focuses on modelling the motion of the object observed with stereo RGB cameras. To our best knowledge, our work is the first to explore an event-only solution to recover the camera's orbital motion parameters over time jointly with a sparse 3D structure.

2.4. Event Sensors for Space Applications

In addition to high effective frame rate and high dynamic range, power consumption in milli-Watt levels makes event sensors attractive as an onboard sensor. Event sensors have already been applied to problems such as star tracking [2, 10, 36], satellite material characterization [29] as well as the general context of space situational awareness [11]. In addition, there are several datasets for the application of event-based satellite pose estimation in space [27, 41, 42].

3. Event-based Structure-from-Orbit

eSfO represents a special case of event-based SfM in which additional constraints are placed on the poses of the cameras. As alluded to in Sec. 1, observing a spinning object with a static camera is equivalent to observing a static object with an orbiting camera; see supp. material for the proof.

The raw input stream from the event camera $\mathcal{E} = \{e_i\}_{i=1}^{N_E}$, where each $e_i = (t_i, x_i, y_i, p_i)$, consists of the event pixel-locations (x_i,y_i), timestamp (t_i) and binary polarity (p_i). SfO recovers a sparse point cloud representation of the object $\mathbf{X} = \{\mathbf{x}_p^w \in \mathbb{R}^3 | p = 1 \ldots N_P\}$ along with the following orbital parameters (see Fig. 2):

- $r \in \mathbb{R}$, the radius of the orbit.
- $f \in \mathbb{R}$, the rotational rate of the object (Hz).
- $\mathbf{R}_0 \in \mathcal{SO}(3)$, rotation with respect to the orbital plane that points the camera towards the object center
- $\vec{\mathbf{n}} \in \mathbb{R}^3$, unit vector representing the normal to the orbital plane pointing along the axis of rotation.
- $\vec{\mathbf{u}} \in \mathbb{R}^3$ lies in the plane, orthogonal to $\vec{\mathbf{n}}$.
- $\mathbf{c}^w \in \mathbb{R}^3$ which is the center of the 2D circle in 3D space.

The center of the camera lies on the circumference of the circle and its position in the world frame at time τ is

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

where $\vec{\mathbf{v}} = \vec{\mathbf{n}} \times \vec{\mathbf{u}}$ lies within the plane and is mutually perpendicular to both $\vec{\mathbf{n}}$ and $\vec{\mathbf{u}}$. θ denotes the rest of the parameters. The orientation of the camera can be decomposed into two rotations: $\mathbf{R}_0$ which is a constant rotation with respect to the orbital plane that allows the camera to look at the object center, and $\mathbf{R}_{\vec{\mathbf{n}},\tau}$ is the in-plane rotation around $\vec{\mathbf{n}}$ at time τ induced by the object's rotation. This fully characterizes the orbital motion of the sensor. Compared to a general SfM problem involving N cameras, where each camera has 6 DoF resulting in $6N$ parameters for the camera poses alone, the proposed formulation has a fixed number of motion parameters in this minimal representation (14 in total; see above) regardless of the number of cameras, due to the orbital constraint. Additionally, there are $3N_P$ parameters required for the estimated landmarks.

The proposed eSfO pipeline is depicted in Fig. 3. A front-end takes in the raw event stream and performs feature detection and tracking (Sec. 4). In the back-end, the set of tracks is used to recover an initialization using a generalized SfM formulation (COLMAP [45]). This initialization is upgraded using the eSfO optimization to conform to the orbital model of the problem (Sec. 5).

4. Event Feature Detection & Tracking

Event-only feature detection and tracking is an inherently difficult problem due to the lack of consistent neighborhood structure and the asynchronous nature of the sensor. However, the temporal aspect of the event stream provides additional information to establish continuity and proximity over time. To address the problem of feature detection and tracking, we exploit densely clustered corner events in the spatio-temporal space. Our approach, **Event Tracking by Clustering (ETC)**, hinges on two key observations: **(1)** The spatio-temporal density of corner events serves as a dependable metric for feature detection, as it is induced by the 3D structure of the scene; and **(2)** The spatio-temporal proximity of corner events indicates whether these events originate from the same 3D point in space.

Based on these observations, we propose a feature detection and tracking mechanism taking advantage of the spatio-temporal nature of the event stream.

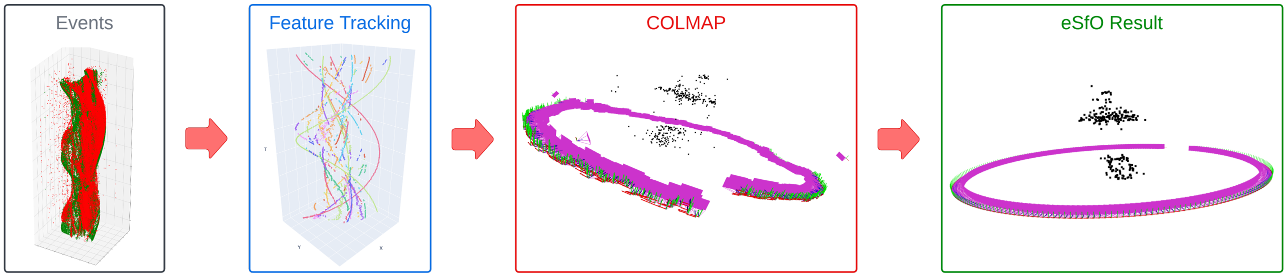


Figure 3. The proposed eSfO pipeline.

4.1. Spatio-temporal Feature Detection

Given $\mathcal{E}$, the first step in ETC is to detect "corner" events using the eFAST corner event detector [34]. eFAST leverages the Surface of Active Events (SAE) representation of the event stream. An event e_i is a corner when a certain number of neighboring pixels, located along a circular path around (x_i, y_i), exhibit either consistently darker or brighter intensities in the SAE image; see [34] for details.

In our method, the detected corners are further filtered based on their local neighborhood density score, defined as

$$D(e_i) := \frac{1}{\lambda} \sum_{e_q \in \mathcal{E}_i} \mathbb{I}(p_i = p_q), \tag{2}$$

where $\mathcal{E}_i \subset \mathcal{E}$ is the set 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 $\mathbb{I}$ is the indicator function that returns 1 if its input condition is true and 0 otherwise.

Event cameras have different positive and negative polarity bias settings, leading to differing sensitivities for positive and negative polarity events [22] – and thus differing densities of corner events for the same brightness intensity edge. We therefore compare the density score $D(e_i)$ of a corner e_i with the mean density score μ_D of all detected corners of the same polarity as e_i, and remove e_i if $D(e_i) < \mu_D$. This helps identify corner event clusters, which subsequently serve as candidate feature tracks.

4.2. Spatio-temporal Feature Clustering

We utilize our second observation to refine feature detections and track them by employing HDBSCAN [33], a non-parametric, density-based clustering algorithm which identifies clusters by discovering regions of high data point density. HDBSCAN is applied to the remaining corner events in the spatio-temporal space, disregarding polarity. This generates N_H clusters of corner events, $C_i, i = \{1 \dots N_H\}$ that lay the foundation for our feature tracks. Due to noise and the speed of motion in the scene, the tracks belonging to the same feature may be identified as disjoint clusters. To join clusters that are spatio-temporally close to each other, we apply nearest neighbor head and tail matching within a spatio-temporal hemisphere (since time can only move forward). The "head" represents the start of a cluster and "tail" represents the end. To connect clusters C_i and C_j, the mean of the last and respectively first N_σ events' spatio-temporal locations is computed and used as the descriptors for the clusters. Let these spatio-temporal descriptors be given by $D_\zeta = (t_\zeta, x_\zeta, y_\zeta)$ and $D_\eta = (t_\eta, x_\eta, y_\eta)$ for the tail and head of clusters C_i and C_j respectively. The cluster C_i is connected to the cluster C_j with the smallest euclidean distance between D_ζ and D_η if it lies within the spatio-temporal hemisphere of radius ϕ defined by D_ζ:

$$(x_\eta - x_\zeta)^2 + (y_\eta - y_\zeta)^2 + (t_\eta - t_\zeta)^2 < \phi^2 \tag{3}$$

The two cluster are merged to form a single cluster, $C_i \leftarrow C_i \cup C_j$, reducing the final number of clusters to N_P, corresponding to the number of tracks as well as the number of points in the reconstruction.

4.3. Feature Track Extraction

To extract feature tracks from the feature clusters, we divide the duration of the event stream, $T = t_{N_E} - t_1$, into $K = (T/\delta t) - 1$ mutually disjoint temporal windows $\mathcal{W}_k, k = \{1 \dots K\}$ of duration δt each such that the k-th window spans the time duration $(k\delta t, (k+1)\delta t]$. For each cluster $C_p, p = \{1 \dots N_P\}$, we identify the events that fall within the time window $\mathcal{W}_k$: $\mathcal{E}_{p,k} = \{e_p | e_p \in C_p \text{ and } t_p \in \mathcal{W}_k\}$ and compute the mean pixel location within the window, giving us the position of the p-th feature track at time $t_k = k\delta t$:

$$\mathbf{f}^p_{t_k} = \frac{\sum_{e_i \in \mathcal{E}_{p,k}} (x_i, y_i)}{|\mathcal{E}_{p,k}|} \tag{4}$$

This approach has a few benefits: (1) the mean operation reduces noise of the corner events; and (2) we can choose the δt to generate arbitrarily large number of camera views. However, determining the appropriate window size δt is application-specific, as it has an inverse relation with the tracking accuracy.

5. Optimization for eSfO

5.1. Initialization

The N_P feature tracks generated using feature tracking provide the required information needed to run a general SfM pipeline. Each of the tracks correspond to a 3D point in the world frame $\mathbf{x}_p^w$, who's projection at time t_k is the observed feature $\mathbf{f}_{t_k}^p$. Each of the K time windows correspond to a camera. Using this information, we generate an initial set of cameras and a point cloud using COLMAP – which minimizes the reprojection error for each 3D-2D correspondence $(\mathbf{x}_p^w, \mathbf{f}_{t_k}^p)$ over the K cameras and N_P world points:

$$\sum_{k=1}^{K}\sum_{p=1}^{N_P} ||\pi(\mathbf{K}\mathbf{R}_w^{t_k}\mathbf{x}_p^w + \mathbf{K}\mathbf{t}_w^{t_k}) - \mathbf{f}_{t_k}^p|| \tag{5}$$

where $t_k = k\delta t$, $\pi(.)$ is the pin-hole projection and $\mathbf{K}$ is the intrinsic calibration matrix. COLMAP provides an initial set of camera positions and 3D points, but since no constraints about the orbital trajectory were enforced in this general SfM pipeline, the camera trajectory deviates significantly from a circular path, see Fig. 3 (COLMAP). However, initial values for the SfO parameters can be computed from this intermediate solution.

We compute the best fitting circle to the camera centers $\mathbf{t}_{t_k}^w, k = \{1 \dots K\}$ generated by COLMAP by estimating $\vec{\mathbf{n}}, \mathbf{c}, \vec{\mathbf{u}}$ and the radius r. To compute the normal vector $\vec{\mathbf{n}}$, we compute the mean of the camera centers $\mathbf{t_c} = \frac{1}{K}\sum_k \mathbf{t}_{t_k}^w$ and find the least squares fit for the plane using the $K \times 3$ mean normalized camera-center matrix $\mathbf{T}$.

$$\mathbf{T} = \begin{bmatrix}\mathbf{t}_{t_1}^w - \mathbf{t}_c, & \mathbf{t}_{t_2}^w - \mathbf{t}_c, & \dots & \mathbf{t}_{t_k}^w - \mathbf{t}_c\end{bmatrix}^T \tag{6}$$

We solve $[\mathbf{T}_{[:,1:2]}\ \mathbf{1}_{K\times 1}]\mathbf{n} = \mathbf{T}_{[:,3]}$ using least squares and normalize the resulting vector to obtain $\vec{\mathbf{n}}$. The notation $T_{[:,p:q]}$ selects all the rows and the p to q columns from the matrix $\mathbf{T}$ and $\mathbf{1}_{K\times 1}$ is a $K \times 1$ matrix of ones.

Camera centers are then projected to the computed plane to estimate the circle. The projection of the camera center $\mathbf{t}_{t_k}^w$ onto the plane is denoted as $\tilde{\mathbf{t}}_{\mathbf{t_k}}^{\mathbf{w}} = [Rod(\vec{\mathbf{n}}, \vec{\mathbf{z}})\mathbf{t}_{t_k}^w]_{[1:2]}$, where $Rod(\vec{\mathbf{a}}, \vec{\mathbf{b}})$ denotes the Rodriguez rotation matrix that rotates points about an axis $\vec{\mathbf{k}} = \vec{\mathbf{a}} \times \vec{\mathbf{b}}$ by an angle $\theta = \arccos(\vec{\mathbf{a}}^T\vec{\mathbf{b}})$. The least square fit is found for the parameters of the circle $\mathbf{\Theta}_c \in \mathbb{R}^3$ using the implicit equation of a circle. Using

$$\tilde{\mathbf{T}} = \begin{bmatrix}\tilde{\mathbf{t}}_{t_1}^w, & \tilde{\mathbf{t}}_{t_2}^w, & \dots & \tilde{\mathbf{t}}_{t_k}^w\end{bmatrix}^T, \tag{7}$$

we find the least square fit in the solution to the equation

$$[\tilde{\mathbf{T}}\ \mathbf{1}_{K\times 1}]\mathbf{\Theta}_c = \begin{bmatrix}||\tilde{\mathbf{t}}_{t_1}^w||^2, & ||\tilde{\mathbf{t}}_{t_2}^w||^2, & \dots & ||\tilde{\mathbf{t}}_{t_k}^w||^2\end{bmatrix}^T. \tag{8}$$

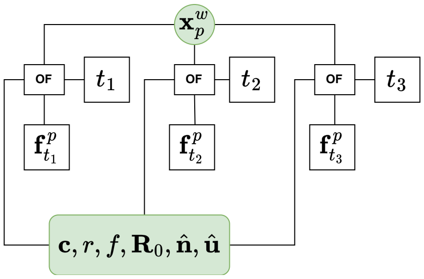


Figure 4. The formulation of the orbit factor (OF) illustrated for a single 3D point $\mathbf{x}_i$ and its corresponding feature positions for three different timestamps. Global parameters are highlighted as a group at the bottom of the figure. Estimated quantities are in green and inputs are in white squares.

The remaining orbit parameters are computed as:

$$r = \sqrt{\Theta_c(3) + (\frac{\Theta_c(1)}{2})^2 + (\frac{\Theta_c(2)^\star}{2})^2}$$

$$\mathbf{c}^w = Rod(\vec{\mathbf{z}}, \vec{\mathbf{n}})\begin{pmatrix}\frac{\Theta_c(1)}{2}\\ \frac{\Theta_c(2)}{2}\\ 0\end{pmatrix} + \mathbf{t}_c$$

$$\mathbf{u} = \mathbf{t}_{t_1}^w - \mathbf{c}^w$$

where the $(.)$ notation is used to access elements in the vector and $\vec{\mathbf{z}} = (0,0,1)^T$.

Finally, for an estimate of the frequency, we employ the Fourier Transform. We divide the duration of the event stream, $T = t_{N_E} - t_1$, into $F = (T/\delta t_f) - 1$ mutually disjoint temporal windows $\mathcal{W}_f, f = \{1 \dots F\}$ (each of duration δt_f) such that the f-th window spans the time duration $(f\delta t_f, (f+1)\delta t_f]$. Within each window, we compute the mean value of the x location of all the events.

$$\tilde{x}(t_f) = \frac{\sum_{e_i | t_i \in \mathcal{W}_f} x_i}{|\{e_i | t_i \in \mathcal{W}_f\}|} \tag{9}$$

We then compute the Fourier Transform $(\tilde{X}(f))$ of $\tilde{x}(t_f)$.

$$\tilde{X}(f) \triangleq \int_{-\infty}^{\infty} \tilde{x}(t_f)\, e^{-i2\pi f t}\, \mathrm{d}t \tag{10}$$

The frequency with the highest power is used as the initial estimate of the rate of rotation.

5.2. Solving for eSfO parameters

The eSfO parameterization consist of two sets of parameters: global parameters that capture the overall structure of the problem and local time-dependent observations of a 3D point in the world. The factor graph (See Fig. 4) therefore includes dependencies between each observation induced by these global parameters.

The formulation in its essence is a reprojection error minimization formulation similar to SfM, but differs in that the computation of the camera poses is instead governed by the orbit formulation. More concretely, we formulate the orbit-based camera transformation as

$$\mathbf{T}_w^{t_k} = \mathbf{T}_p^{t_k}\mathbf{T}_o^p\mathbf{T}_w^o, \tag{11}$$

which takes a point $\mathbf{x}^w$ in a world frame and projects it to the camera coordinate frame at time t_k, where

$$\mathbf{T}_w^o = \begin{bmatrix} \mathbf{R}_{\vec{\mathbf{n}}}(2\pi f t_k) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Rod(\vec{\mathbf{n}}, \vec{\mathbf{z}}) & t^w(t_k; \theta) \\ 0 & 1 \end{bmatrix} \tag{12}$$

positions and orients the camera in the correct position along the circumference of the circle;

$$\mathbf{T}_o^p = \begin{bmatrix} & \vec{\mathbf{d}} & & 0 \\ & -\vec{\mathbf{y}} \times \vec{\mathbf{d}} & & 0 \\ & \vec{\mathbf{d}} \times (-\vec{\mathbf{y}} \times \vec{\mathbf{d}}) & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

rotates the camera to look at the center ($\mathbf{c}^w$) of the orbit circle, with $\vec{\mathbf{d}} = \mathbf{T}_w^o\mathbf{c}^w/||\mathbf{T}_w^o\mathbf{c}^w||$, $\vec{\mathbf{y}} = [0, 1, 0]^T$; and

$$\mathbf{T}_p^{t_k} = \begin{bmatrix} \mathbf{R}_0 & 0 \\ 0 & 1 \end{bmatrix} \tag{14}$$

rotates the camera from the orbital plane to toward the object. Points in the world frame $\mathbf{x}_p^w$ can then be projected to cameras where they are visible to minimize reprojection error, similar to (5). This is used in a factor-graph method [12] to refine SfO parameters.

6. Dataset

For performance evaluation, we created a dataset of objects placed on a turntable and observed by a static event camera (Fig. 2). Our **daTaset Of sPinning objectS with neuromorPhic vIsioN (TOPSPIN)** [16] consists of six objects under three rotational speeds and four perspectives (Tab. 1). For each scene, one of the objects was placed on the turntable and rotated at a given speed, while the camera observed it from a given perspective. We exhaustively generated 72 scenes using the combinations in Tab. 1. Several sample event frames are displayed in Fig. 5.

Object	Turntable speed	Camera perspective
Hubble Satellite		
SOHO Satellite	Fast	Top Down
TDRS Satellite	Medium	Side On
PS4 Dualshock Controller	Slow	Perpendicular
Nintendo Switch Controller		Diagonal
Inivation DAVIS346 Camera		

Table 1. Taxonomy of settings for our event dataset.

7. Results

We present experiment results to evaluate the efficacy of the two main contributions separately. For the front-end, we compared the performance of our feature tracker against state-of-the-art methods. For the back-end, we compared our method's sparse reconstruction against the objects' CAD model and compared the estimated frequency against the ground truth. We also provide qualitative results for the estimation of the axis of rotation of the objects.

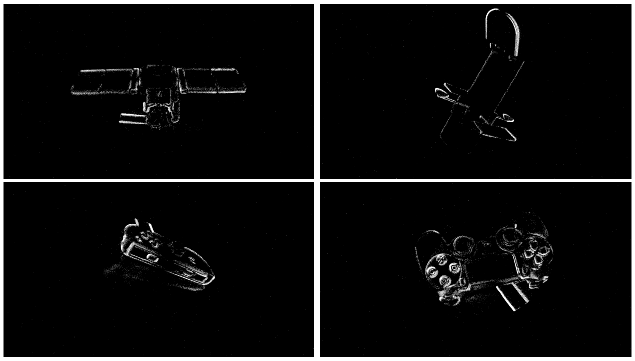


Figure 5. Example event frames from our dataset. `soho-sideon-fast` (top left), `hubble-diagonal-med` (top right), `switch-perpendicular-slow` (bottom left) and `dualshock-topdown-med` (bottom right).

7.1. Hyperparameter settings

The hyperparameters of the eSfO pipeline and their respective values are provided in Tab. 2. HDBSCAN has two hyperparameters $minPts$ and ϵ (see [33] for details).

Local event neighborhood distance (λ)	7
HDBSCAN minimum cluster size ($minPts$)	10
HDBSCAN cluster selection epsilon (ϵ)	5
Track association hemisphre radius (ϕ)	30
Cluster descriptor sample size (N_σ)	5
Feature track extraction window duration (δt)	30ms
FFT sampling window duration (δt_f)	20ms

Table 2. Parameter settings for eSfO.

7.2. Front-end evaluation

Event Camera Dataset To evaluate ETC's performance against other state-of-the-art methods, we used the Event Camera Dataset [35] and the corresponding benchmarking strategy outlined in [25]: using ground truth camera poses, we triangulated the 3D points using the estimated feature tracks. The estimated points were then projected to each ground truth camera and the reprojection error was computed against the tracked feature point. Due to lack of a public implementation, results for eCDT are reported from their work [25]. eCDT without the HT Matching module is denoted as "eCDT (w/o HT)". We initialized HASTE with the feature detections from our method, as it is only a tracking method. Metavision [9] used the pretrained weights from their original work. Each dataset was evaluated with an error threshold of 3, 5 and 7 pixels. Performance was then measured as the Root Mean Squared Error (RMSE) of the

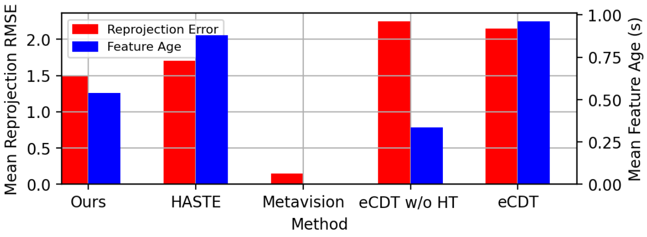


Figure 6. Mean feature age and mean reprojection error computed across `shapes`, `poster` and `boxes`.

projection error for each feature track and reported in Tab. 3. Results for the feature age, the amount of time a feature is successfully tracked, are reported in Tab. 4. eCDT had the longest mean feature age, but suffered from a high mean reprojection error: their feature tracks were longer but less accurate. Conversely, the Metavision feature tracker had extremely short feature tracks, which severely diminish its usefulness. The ideal method would produce tracks which have both low reprojection error and longer feature age – like the behavior seen for HASTE and ETC (Fig. 6).

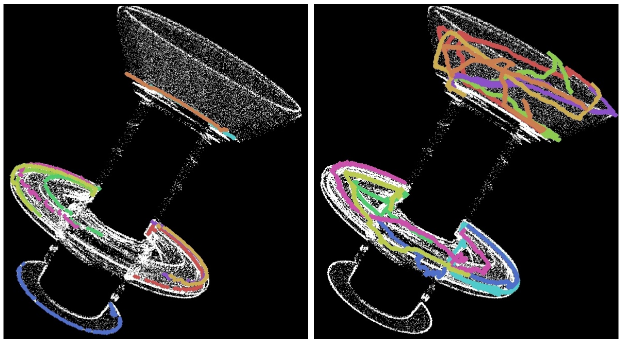
Event feature tracking for spinning objects However, the underlying assumption of HASTE about the fixed (planar) tracking template during tracking is easily violated in the SfO setting, as observed when HASTE is evaluated on our dataset. Fig. 7 shows feature tracks (colored) overlaid on accumulated corner events during a complete object revolution. Due to the object's rotation, the points on the object trace out an ellipse in the image [28]. HASTE tracks diverge from the elliptical arcs due to tracking failures. This is also reflected in the resulting COLMAP reconstruction, with a mean reprojection error of 348.48 pixels for the HASTE tracks. In contrast, our method exhibited more robust tracking performance under the same conditions, with a mean COLMAP reprojection error of 2.39 pixels. See supplementary material for the full quantitative results.

Note also that the feature age depends on the rotational velocity of the object (Fig. 8b). For faster motion, self-occlusion occurs faster than when observing slowly rotationing objects – leading to a shorter mean feature age.

The results for the front-end feature detection and tracking show that our method is well-suited for the eSfO problem and compares favorably to other state-of-the-art methods on traditional benchmarks.

7.3. Back-end

In this section, we present quantitative and qualitative results for the various quantities estimated using eSfO and its efficacy at recovering the orbital structure. Again, see the supp. material for the full quantitative results.



(a) ETC (b) HASTE

Figure 7. 10 longest feature tracks for both HASTE and ETC (our method) for the `hubble-diagonal-fast` scene.

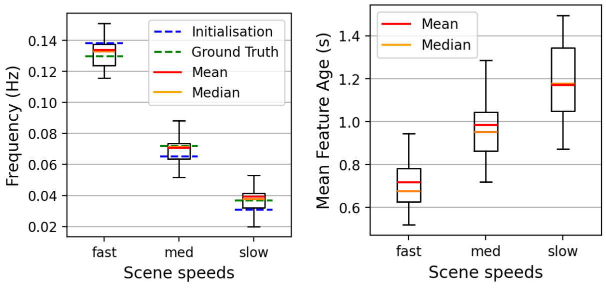
	Method	7	5	3
`shapes`	HASTE [3]	2.37 ± 1.87	1.73 ± 1.45	0.98 ± 0.85
	Metavision [9]	**0.17** ± 0.57	**0.17** ± 0.57	**0.17** ± 0.55
	eCDT(w/o HT)	3.10 ± 1.67	2.58 ± 1.25	1.76 ± 0.76
	eCDT	2.94 ± 1.61	2.47 ± 2.32	1.78 ± 0.67
	ETC	1.70 ± 1.68	1.26 ± 1.21	0.88 ± 0.80
`poster`	HASTE [3]	2.30 ± 1.65	1.78 ± 1.24	1.17 ± 0.77
	Metavision [9]	**0.13** ± 0.45	**0.13** ± 0.45	**0.13** ± 0.44
	eCDT(w/o HT)	2.95 ± 1.65	2.47 ± 1.22	1.75 ± 0.72
	eCDT	2.69 ± 1.55	2.32 ± 1.17	1.71 ± 0.69
	ETC	2.38 ± 2.09	1.66 ± 1.50	0.95 ± 0.92
`boxes`	HASTE [3]	2.21 ± 1.78	1.68 ± 1.31	1.10 ± 0.79
	Metavision [9]	**0.18** ± 0.52	**0.13** ± 0.45	**0.13** ± 0.44
	eCDT(w/o HT)	2.22 ± 1.54	1.94 ± 1.21	1.44 ± 0.74
	eCDT	2.12 ± 1.48	1.90 ± 1.20	1.42 ± 0.77
	ETC	2.23 ± 2.07	1.56 ± 1.50	0.88 ± 0.91

Table 3. RMSE (pixel) Reprojection errors. Reprojection errors above the threshold (7, 5, and 3) are considered outliers and removed.

	Method	7		5		3	
		Mean	Med.	Mean	Med.	Mean	Med.
`shapes`	HASTE [3]	0.646	0.200	0.671	0.270	0.861	0.430
	Metavision [9]	0.007	0.005	0.007	0.005	0.007	0.005
	eCDT(w/o HT)	0.417	0.240	0.427	0.235	0.446	0.227
	eCDT	**1.224**	**0.550**	**1.309**	**0.585**	**1.518**	**0.628**
	ETC	0.332	0.200	0.401	0.210	0.524	0.240
`poster`	HASTE [3]	**1.699**	**1.107**	**1.186**	**0.829**	0.705	0.481
	Metavision [9]	0.005	0.005	0.005	0.005	0.005	0.005
	eCDT(w/o HT)	0.297	0.205	0.298	0.200	0.304	0.200
	eCDT	0.795	0.480	0.814	0.488	**0.841**	**0.505**
	ETC	0.499	0.180	0.497	0.180	0.641	0.190
`boxes`	HASTE [3]	**0.904**	**0.685**	**0.721**	**0.554**	0.524	0.376
	Metavision [9]	0.005	0.005	0.005	0.005	0.005	0.005
	eCDT(w/o HT)	0.275	0.200	0.277	0.200	0.283	0.200
	eCDT	0.707	0.380	0.719	0.385	0.728	**0.385**
	ETC	0.525	0.180	0.702	0.200	**0.738**	0.200

Table 4. Comparison of feature age (seconds). Reprojection errors above the threshold (7, 5, and 3) are omitted as outliers.

Frequency estimation eSfO aims to recover the rotation rate (frequency) of the object being observed. We compare the estimated frequency against the ground-truth obtained from the turntable. The results are presented in Fig. 8a,



(a)

(b)

Figure 8. (a) Estimated frequency for the `fast`, `med` & `slow` speeds across all scenes. (b) Distribution of mean feature age for the feature tracks from our method for all scenes; scenes are grouped by their speed (`fast`, `med` & `slow`).

which depicts the average estimated frequency for the different speeds of the turntable. Observe that the initialization provided from the Fourier Transform is further refined by eSfO, bringing the estimate closer to the ground-truth.

Is eSfO effective? To demonstrate the efficacy of eSfO at resolving the camera poses and structure, we first look at how the eSfO optimization behaves in the constrained orbit estimation problem. We report the average reprojection error for the eSfO reconstruction in Fig. 10a, and those for the COLMAP reconstruction in Fig. 10b. These plots show increased error for eSfO compared to COLMAP. However, eSfO imposes further constraints on camera poses to align them to a circular trajectory. This leads to more residual error after optimization, as not all constraints can be fully resolved. However, when comparing the generated trajectories from both (Fig. 3), it is evident that eSfO recovers a better orbital trajectory compared to COLMAP.

To investigate the effect of eSfO on point reconstruction, we aligned the point clouds with their respective 3D CAD models, initially using a manual alignment strategy and further refining it using an Iterative Closest Point (ICP) based optimization scheme. The results (Fig. 9) show that the estimated point clouds are a highly accurate, albeit sparse, rendition of their respective 3D CAD model. Additional constructed trajectory and models are shown in the supplementary material.

Axis of rotation estimation We present qualitative results for the projection of the axis of rotation into the image plane, referred to as the “screw line”. This is the projection of the vector $\vec{\mathbf{n}}$ at the center $\mathbf{c}^w$ of the circle, onto the image plane. As seen in Fig. 11, the estimated screw line clearly marks out the axis of rotation of the object over the accumulated event frame – suggesting that the plane normal and center of the circle were recovered correctly.

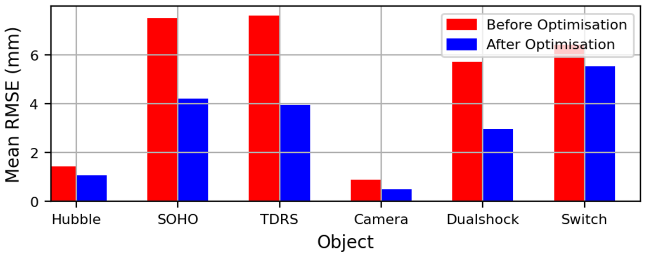
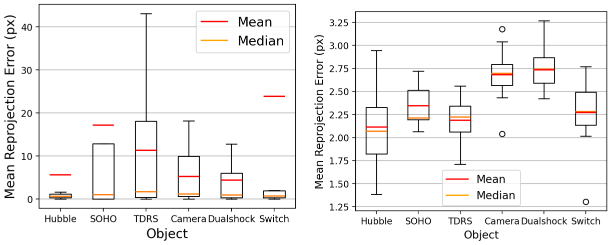


Figure 9. Mean sparse point cloud to CAD model registration error before and after orbit optimization (all objects).



(a)

(b)

Figure 10. (a) Distribution of mean reprojection error after eSfO optimization. (b) Distribution of mean reprojection error of COLMAP reconstruction.

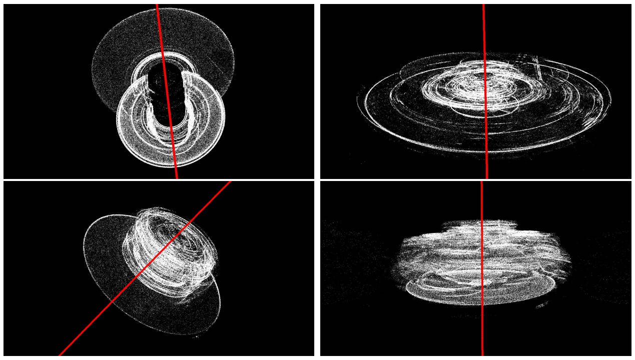


Figure 11. Estimated axis of rotation projected onto the accumulated event frame (red). `hubble-topdown-fast` (top left), `tdrs-sideon-med` (top right), `camera-diagonal-fast` (bottom left) and `dualshock-perpendicular-slow` (bottom right).

8. Conclusion

In this work, we have presented a new reconstruction problem, *eSfO*, which recovers a sparse representation of an object rotating about a fixed axis and observed by a static event camera. Through extensive experiments, we have demonstrated that the frequency, screw line, camera pose, and reliable sparse reconstruction can be recovered using the proposed pipeline. The dataset has been released publicly [16], and the code for eSfO can be found here:
`https://github.com/0thane/eSfO`.

References

[1] Automatic reconstruction of 3d objects using a mobile camera. 17(2):125–134, 1999. 2

[2] 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. 3

[3] 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. 2, 7

[4] Santarshi Bandyonadhyay, Issa Nesnas, Shvam Bhaskaran, Beniamin Hockman, and Benjamin Morrell. Silhouette-based 3d shape reconstruction of a small body from a spacecraft. In *2019 IEEE Aerospace Conference*, pages 1–13, 2019. 2

[5] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 884–892, 2016. 2

[6] Alexis Baudron, Zihao W. Wang, Oliver Cossairt, and Aggelos K. Katsaggelos. E3d: Event-based 3d shape reconstruction, 2020. 3

[7] Haodong Chen, Vera Chung, Li Tan, and Xiaoming Chen. Dense voxel 3d reconstruction using a monocular event camera. In *2023 9th International Conference on Virtual Reality (ICVR)*. IEEE, 2023. 3

[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. 2

[9] Philippe Chiberre, Etienne Perot, Amos Sironi, and Vincent Lepetit. Long-lived accurate keypoints in event streams, 2022. 2, 6, 7

[10] 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. 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. 3

[12] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. Foundations and Trends in Robotics, Vol. 6, 2017. 6

[13] Kaitlin Dennison, Nathan Stacey, and Simone D'Amico. Autonomous asteroid characterization through nanosatellite swarming. *IEEE Transactions on Aerospace and Electronic Systems*, 59(4):4604–4624, 2023. 2

[14] Mehregan Dor, Katherine A. Skinner, Panagiotis Tsiotras, and Travis Driver. Visual slam for asteroid relative navigation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2066–2075, 2021. 2

[15] P. Eisert, E. Steinbach, and B. Girod. Automatic reconstruction of stationary 3-d objects from multiple uncalibrated camera views. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(2):261–277, 2000. 2

[16] Ethan Elms. TOPSPIN: daTaset Of sPinning objectS with neuromorPhic vIsioN, 2024. `https://doi.org/10.5281/zenodo.10884693`. 6, 8

[17] Andrew W Fitzgibbon, Geoff Cross, and Andrew Zisserman. Automatic 3d model construction for turn-table sequences. In *3D Structure from Multiple Images of Large-Scale Environments: European Workshop, SMILE'98 Freiburg, Germany, June 6–7, 1998 Proceedings*, pages 155–170. Springer, 1998. 2

[18] Vincent Fremont and Ryad Chellali. Turntable-based 3d object reconstruction. In *IEEE Conference on Cybernetics and Intelligent Systems, 2004.*, pages 1277–1282. IEEE, 2004. 2

[19] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1

[20] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Asynchronous, photometric feature tracking using events and frames. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–765, 2018. 2

[21] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Eklt: Asynchronous photometric feature tracking using events and frames. *International Journal of Computer Vision*, 128(3):601–618, 2020. 2

[22] Rui Graca, Brian McReynolds, and Tobi Delbruck. Shining light on the dvs pixel: A tutorial and discussion about biasing and optimization, 2023. 4

[23] Tommaso Guffanti, Toby Bell, Samuel Y. W. Low, Mason Murray-Cooper, and Simone D'Amico. Autonomous guidance navigation and control of the visors formation-flying mission, 2023. 2

[24] 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

[25] 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. 2, 6

[26] Kunping Huang, Sen Zhang, Jing Zhang, and Dacheng Tao. Event-based simultaneous localization and mapping: A comprehensive survey, 2024. 2

[27] Mohsi Jawaid, Ethan Elms, Yasir Latif, and Tat-Jun Chin. Towards bridging the space domain gap for satellite pose estimation using event sensing, 2022. 3

[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. 7

[29] Andrew Jolley, Greg Cohen, Damien Joubert, and Andrew Lambert. Evaluation of event-based sensors for satellite material characterization. *Journal of Spacecraft and Rockets*, 59(2):627–636, 2022. 3

[30] Marshall Kaplan, Bradley Boone, Robert Brown, Thomas Criss, and Edward Tunstel. *Engineering Issues for All Major Modes of In Situ Space Debris Capture*. 2

[31] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 16–23. IEEE, 2016. 1

[32] Ruoxiang Li, Dianxi Shi, Yongjun Zhang, Kaiyue Li, and Ruihao Li. Fa-harris: A fast and asynchronous corner detector for event cameras. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6223–6229. IEEE, 2019. 2

[33] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. 4, 6

[34] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. Fast event-based corner detection. In *British Machine Vision Conference (BMVC)*, 2017. 2, 4

[35] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 1, 6

[36] Yonhon Ng, Yasir Latif, Tat-Jun Chin, and Robert Mahony. Asynchronous kalman filter for event-based star tracking. In *European Conference on Computer Vision*, pages 66–79. Springer, 2022. 3

[37] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017. 1

[38] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1669–1678. IEEE, 2020. 2

[39] Soon-Yong Park. *Stereo vision and range image techniques for generating 3D computer models of real objects*. PhD thesis, State University of New York at Stony Brook, 2003. 2

[40] Soon-Yong Park and Murali Subbarao. A multiview 3d modeling system based on stereo vision techniques. *Machine Vision and Applications*, 16:148–156, 2005. 2

[41] Tae Ha Park, Marcus Martens, Gurvan Lecuyer, Dario Izzo, and Simone D'Amico. Speed+: Next-generation dataset for spacecraft pose estimation across domain gap. In *2022 IEEE Aerospace Conference (AERO)*. IEEE, 2022. 3

[42] Arunkumar Rathinam, Haytam Qadadri, and Djamila Aouada. Spades: A realistic spacecraft pose estimation dataset using event sensing, 2023. 3

[43] 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

[44] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017. 2

[45] 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. 1, 3

[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. 2, 3

[47] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, 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. 1, 2

[48] Yi Zhou, Guillermo Gallego, and Shaojie Shen. Event-based stereo visual odometry. *IEEE Transactions on Robotics*, 37 (5):1433–1450, 2021. 1

[49] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018. 2

[50] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019. 2