

## Star Tracking using an Event Camera

Tat-Jun Chin<sup>†</sup> Samya Bagchi<sup>†</sup> Anders Eriksson<sup>\*</sup> André van Schaik<sup>◇</sup>  
 The University of Adelaide<sup>†</sup> University of Queensland<sup>\*</sup> Western Sydney University<sup>◇</sup>

### Abstract

Star trackers are primarily optical devices that are used to estimate the attitude of a spacecraft by recognising and tracking star patterns. Currently, most star trackers use conventional optical sensors. In this application paper, we propose the usage of event sensors for star tracking. There are potentially two benefits of using event sensors for star tracking: lower power consumption and higher operating speeds. Our main contribution is to formulate an algorithmic pipeline for star tracking from event data that includes novel formulations of rotation averaging and bundle adjustment. In addition, we also release with this paper a dataset for star tracking using event cameras<sup>1</sup>. With this work, we introduce the problem of star tracking using event cameras to the computer vision community, whose expertise in SLAM and geometric optimisation can be brought to bear on this commercially important application.

### 1. Introduction

The attitude of a spacecraft is the 3DOF orientation (roll, pitch, yaw) of its body frame with respect to an inertial frame, such as the celestial reference frame [25, Sec. 2.6]. Attitude control is a basic functionality in space flight. The subsystem for attitude control is called the Attitude Determination and Control System (ADCS). As the name suggests, there are two main components in an ADCS: estimating the current attitude, and executing a sequence of appropriate signals to the actuators (reaction wheels, thrusters, etc.) to achieve the desired body orientation.

Our work focusses on the attitude determination problem. A number of sensors are in use for estimating spacecraft attitude, such as sun sensors and magnetometers. It has been established, however, that star trackers are state-of-the-art in spacecraft attitude estimation [24], especially to support *high precision* orientation control. As opposed to rough attitude estimation that is used, e.g., in the detumbling process of a satellite, star trackers play a crucial role during stable flight to deliver *fine* attitude estimation to sup-

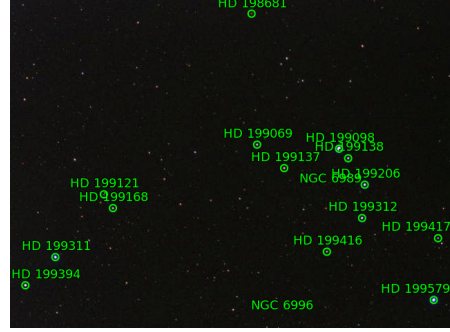


Figure 1. Star identification result using the method of [22, 2]. The labels correspond to identified stars from a star catalogue.

port the mission objectives, e.g., precisely aiming on-board instruments at a target region in space or on Earth.

A star tracker is essentially a camera with an image processing algorithm to estimate spacecraft attitude by recognising star patterns [25, Chap. 4]. Underpinning star tracking is the ability to perform *star identification* [35] from an image; see Fig. 1. In computer vision terms, this means to extract a set of 2D-3D correspondences  $\{(\mathbf{x}_p, \mathbf{X}_p)\}_{p=1}^N$  between the input image and a star chart, where  $\mathbf{x}_p \in \mathbb{R}^2$  are the 2D coordinates of an observed star in the image, and  $\mathbf{X}_p \in \mathbb{R}^3$  is a unit vector that represents the direction of the same star in the inertial frame<sup>2</sup>. In fact, the matching is often accomplished by comparing local descriptors (e.g., the geometric hash code of [22]) that encode the spatial configuration of stars in local image regions.

Given the correspondences, the camera orientation (defined by rotation  $\mathbf{R}$ ) is computed via Wahba’s problem [5]

$$\arg \min_{\mathbf{R} \in SO(3)} \sum_{p=1}^N \|\vec{\mathbf{x}}_p - \mathbf{R}\mathbf{X}_p\|_2^2, \quad (1)$$

where  $\vec{\mathbf{x}}_p$  is the backprojected unit ray of  $\mathbf{x}_p$ , i.e.,

$$\vec{\mathbf{x}}_p = \frac{\mathbf{K}^{-1}\bar{\mathbf{x}}_p}{\|\mathbf{K}^{-1}\bar{\mathbf{x}}_p\|_2}, \quad (2)$$

$\bar{\mathbf{x}}_p = [\mathbf{x}_p^T \ 1]^T$ , and  $\mathbf{K} \in \mathbb{R}^{3 \times 3}$  is the camera intrinsic matrix. Many algorithms exist for solving (1), such as the SVD

<sup>1</sup>Visit project website [1] for the data. This work was supported by AML and ARC grants LP160100495 and FT170100072.

<sup>2</sup>Since stars are effectively at infinity, only their directions matter.

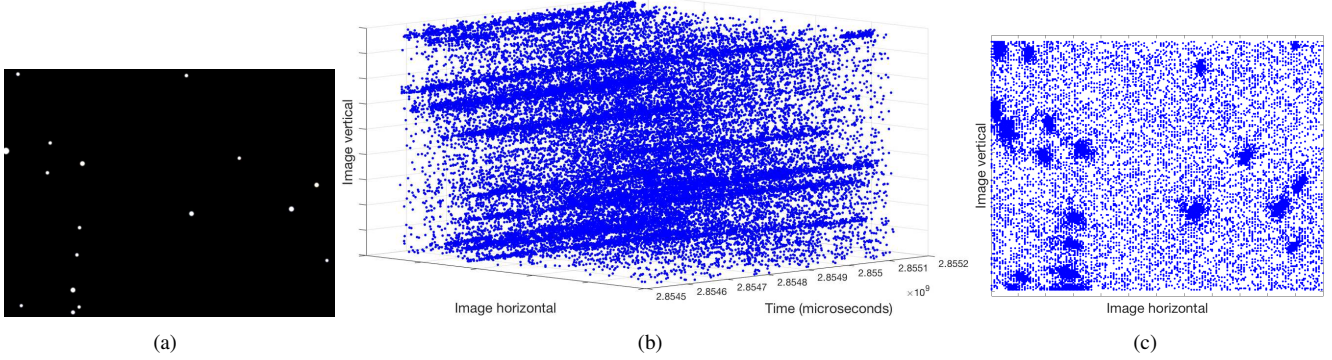


Figure 2. (a) A star field under the field of view (FOV) of a telescope. (b) Events recorded using an iniVation Davis 240C event camera as the FOV moves away from panel a (for clarity, the event polarities are not displayed here). (c) The event data collapsed onto the image frame - the star patterns in panel a are clearly observed here (though the stars are blurred due to the motion of the camera).

method [25, Chap. 5]. Robust versions of (1) also exist to deal with false correspondences [12]. Also critical to star tracking is a filtering step [25, Chap. 6] that takes a sequence of attitude estimates to produce more refined results.

### 1.1. Our contributions

We investigate the usage of *event cameras* for attitude estimation based on star tracking. Unlike a conventional optical sensor (e.g., CCD), an event sensor detects intensity changes asynchronously [23]. The output of an event camera is a set of events  $\{(\mathbf{x}, t, p)\}$ , where  $\mathbf{x}$  are the 2D coordinates of an event on the image plane,  $t$  is the time of the event, and  $p \in \{+, -\}$  is the polarity of the event. Fig. 2(b) illustrates event data from observing a star field.

There are two potential benefits of using event cameras for star tracking. First, due to the paucity of the scene (relatively few bright spots against a black background), the number of events generated tends to be small relative to the number of pixel positions. Hence, an event camera may consume less power. Second, event sensors have high temporal resolution (e.g., iniVation Davis 240C has  $\mu s$  resolution), which could enable higher-speed star tracking. This may be useful for ultra-fine attitude control.

In this work, we do not focus on demonstrating the above benefits, since the gains from using the sensor must be put in the context of other systems or processing requirements - a complex issue that is beyond the scope of this paper. Instead, our focus is on developing an algorithmic pipeline for star tracking using event cameras, so as to establish the feasibility and promise of the paradigm.

Due to the different sensing principle, existing methods [35, 25] cannot be directly applied. Moreover, event data can be significantly noisier than conventional cameras; see Fig. 2(b). To deal with this issue, we develop a novel processing pipeline that includes new formulations of *rotation averaging* and *bundle adjustment* for star tracking. We also release our event data to spark further research.

### 1.2. Previous work

Event cameras and more generally event-based processing are receiving significant attention in robotics and computer vision [4]. Many core capabilities, such as optic flow computation, 3D reconstruction, SLAM and visual servoing (details in [4]), have proven to be feasible with event cameras. In particular, in problems where high-speed operation is essential, techniques using event cameras outperform equivalent methods that use conventional cameras [30].

Recently, there have been a few works that applied event cameras in space. In [13], the feasibility of using event cameras (aided by a telescope) to observe objects in space was established. This was followed by [11], where a probabilistic multiple hypothesis tracker (PMHT) was used to track the objects through time. These works have not described event-based star tracking for high-speed attitude estimation.

## 2. Problem setting

Consider event data  $\mathcal{S} = \{(\mathbf{x}, t, p)\}$  that was generated by observing a star field over a contiguous period of time  $t \in [t_{\text{start}}, t_{\text{end}}]$  under camera motion. Our overall goal is to estimate the attitude of the camera over the period of time. Many previous works on motion estimation using event cameras conduct (either implicitly or explicitly) some form of temporal aggregation on event data to elicit the geometric structures of the scene [14, 20, 30, 29, 21, 34, 17]. Following these works, we generate event images

$$I_1, \dots, I_i, \dots, I_M \quad (3)$$

from  $\mathcal{S}$ , where each image

$$I_i(\mathbf{x}) = \sum_{t \in [t_{\text{start}}^{(i)}, t_{\text{end}}^{(i)}]} \delta(\mathbf{x}, t), \quad (4)$$

$$\delta(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \exists (\mathbf{x}, t, p) \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

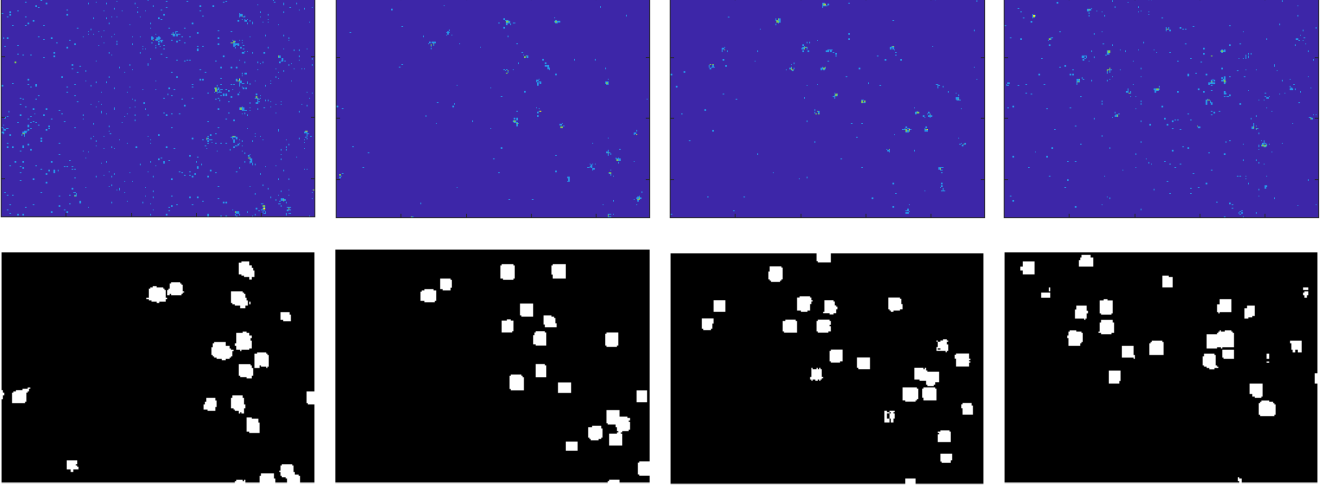


Figure 3. Top row: Event images generated from star field events according to Sec. 2. The integration time ( $t_{\text{start}}^{(i)} - t_{\text{end}}^{(i)}$ ) used here is 10 ms. Brighter pixels indicate higher  $I_i(\mathbf{x})$  values. Bottom row: Discrete point sets (each white pixel is a point) corresponding to the top row.

is obtained by collapsing all the events in a time window

$$[t_{\text{start}}^{(i)}, t_{\text{end}}^{(i)}] \subset [t_{\text{start}}, t_{\text{end}}] \quad (6)$$

onto the image domain. In our work, the time blocks do not overlap, i.e.,

$$[t_{\text{start}}^{(i)}, t_{\text{end}}^{(i)}] \cap [t_{\text{start}}^{(j)}, t_{\text{end}}^{(j)}] = \emptyset \quad \forall i \neq j, \quad (7)$$

and they uniformly partition the recording duration

$$[t_{\text{start}}^{(1)}, t_{\text{end}}^{(1)}] \cup \dots \cup [t_{\text{start}}^{(M)}, t_{\text{end}}^{(M)}] = [t_{\text{start}}, t_{\text{end}}]. \quad (8)$$

Fig. 3 (top row) illustrates.

Due to the nature of our scene (i.e., star fields), the simple temporal aggregation method above is sufficient to elicit the scene structure (cf. the technique in [28] that is targeted at more complex scenes). However, by comparing Figs. 2(a) and Fig. 3 (top row), it is evident that event images are considerably noisier than conventional images. In Sec. 3, we will describe the proposed algorithm that takes as input a sequence of noisy event images  $I_1, \dots, I_i, \dots, I_M$  to compute accurate camera orientations  $\hat{\mathbf{R}}_1, \dots, \hat{\mathbf{R}}_i, \dots, \hat{\mathbf{R}}_M$ .

## 2.1. Why use event images?

By generating event images, we have effectively converted  $\mathcal{S}$  into a set of image frames. We emphasise that this does not defeat the purpose of using an event camera *in our target problem*, since the time windows  $[t_{\text{start}}^{(i)}, t_{\text{end}}^{(i)}]$  are small (only 40ms each), which still enables high-speed (25Hz) attitude estimation. In effect, we are using the event camera primarily as a high-speed low-power optical sensor.

Ideally, event data should be processed using asynchronous or event-based algorithms (e.g., the contrast maximisation framework [17]) to realise the full benefit of asynchronous sensors. We leave this as future work.

## 3. Attitude estimation from event images

Fig. 4 shows the proposed processing pipeline. Details will be described in the rest of this section.

### 3.1. Camera calibration

Notwithstanding the fundamentally different sensing technology, the pinhole imaging model applies to event cameras [15]. Hence a pixel position  $\mathbf{x}$  in an event image  $I_i$  can be backprojected to form a 3D ray by  $\mathbf{K}^{-1}\mathbf{x}$ . For better flow, we will discuss the estimation of  $\mathbf{K}$  in Sec. 4.2. For now we assume  $\mathbf{K}$  is known without loss of generality.

### 3.2. Rotation measurements

The proposed pipeline generates and uses two types of rotation measurements from the event images: absolute rotations and relative rotations.

#### 3.2.1 Absolute rotation estimation

Ideally, the processing could be simplified if we are able to estimate the attitude of each  $I_i$ , by performing star identification [35] followed by solving Wahba's problem (1). However, this is infeasible for two reasons:

- Since most of the event images are noisy, the accuracy of star identification and attitude estimation will be poor.
- Star identification is a relatively costly process. For example, the desktop version of [22, 2] requires seconds to process an image of size  $240 \times 180$ . It will thus be infeasible to execute star identification at event frame-rate, especially on a resource-constrained space platform.

To solve the above two difficulties, a simple heuristic (which we call *active pixel count* or APC) is conducted to select only high-quality event images for star identification.

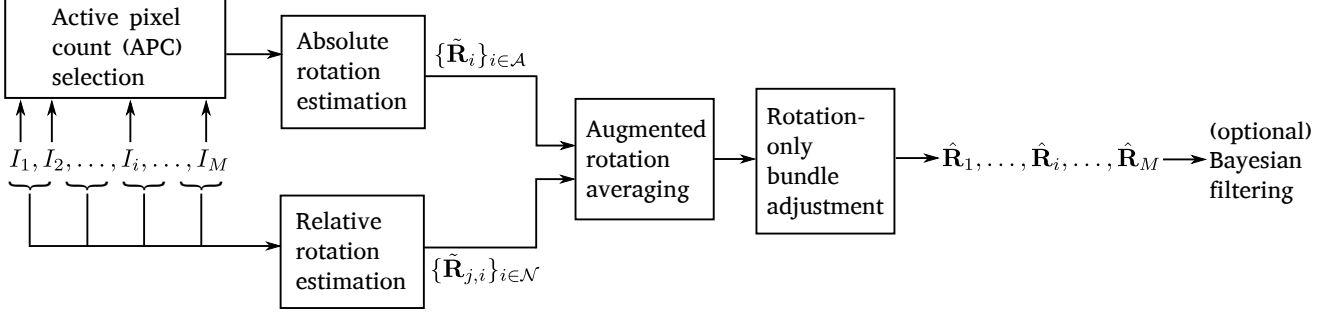


Figure 4. Pipeline of proposed star tracking algorithm: given a sequence of noisy event images  $I_1, \dots, I_i, \dots, I_M$ , a sequence of attitude estimates  $\hat{\mathbf{R}}_1, \dots, \hat{\mathbf{R}}_i, \dots, \hat{\mathbf{R}}_M$  are produced. The output sequence can optionally be postprocessed using Bayesian filtering [25, Chap. 6].

On each event image  $I_i$ , mean filtering by convolution with a  $3 \times 3$  averaging kernel is first conducted. Then the APC of  $I_i$  (using the mean filtered version) is calculated as

$$APC(I_i) = \sum_{\mathbf{x}} \mathbb{I}(I_i(\mathbf{x}) \geq \epsilon_1), \quad (9)$$

where  $\epsilon_1$  is a constant threshold. Basically,  $APC(I_i)$  gives the number of pixels that are “active” (i.e., have produced a sufficient number of events) in the time slab  $[t_{\text{start}}^{(i)}, t_{\text{end}}^{(i)}]$ . If

$$APC(I_i) \geq \epsilon_2, \quad (10)$$

where  $\epsilon_2$  is another constant threshold, then we regard  $I_i$  as suitable for direct attitude estimation. The heuristic is based on the observation that brighter and more well-defined stars tend to yield event data with more active pixels.

Let  $\mathcal{A} \subset \{1, \dots, M\}$  index the event images that are selected via the APC heuristic. To ensure accurate attitude estimates from this set, we use relatively high values for  $\epsilon_1$  and  $\epsilon_2$  (in our experiments,  $\epsilon_1 = 2$  and  $\epsilon_2 = 50$ ), thus,  $\mathcal{A}$  tends to be small, i.e.,  $|\mathcal{A}| \ll M$ . We subject set  $\mathcal{A}$  to star identification (we applied [22] as implemented in [2] in our pipeline), followed by the SVD technique for Wahba’s problem [5], to yield a set of attitude estimates

$$\{\tilde{\mathbf{R}}_i\}_{i \in \mathcal{A}}, \quad (11)$$

We call these rotation measurements the *absolute rotations*.

### 3.2.2 Relative rotation estimation

To fully make use of available data, we estimate *relative rotations* from the event images. Let  $I_i$  and  $I_j$  be two event images that overlap, i.e., they observe some common stars in their respective FOVs. The relative rotation  $\mathbf{R}_{j,i}$  between the images aligns the noisy coordinates  $\mathbf{x}$  and  $\mathbf{x}'$  of a star that is observed simultaneously in  $I_i$  and  $I_j$ , i.e.,

$$\tilde{\mathbf{x}}' \approx \mathbf{R}_{j,i} \tilde{\mathbf{x}}. \quad (12)$$

Note that since stars are at infinity, there is no parallax between  $I_i$  and  $I_j$ , which justifies rotational alignment [25].

To estimate  $\mathbf{R}_{j,i}$ , we begin by thresholding on the mean filtered versions of  $I_i$  and  $I_j$  to create discrete point sets

$$\{\mathbf{x}_{i,p}\}_{p=1}^P \quad \text{and} \quad \{\mathbf{x}_{j,q}\}_{q=1}^Q. \quad (13)$$

See Fig. 3 (bottom row) for example point sets generated. Note that at this stage, data association has not been performed and  $P \neq Q$  in general.

Given the point sets (13), to obtain the relative rotation measurement  $\tilde{\mathbf{R}}_{j,i}$  we solve the registration problem

$$\tilde{\mathbf{R}}_{j,i} = \arg \min_{\mathbf{R} \in SO(3)} \sum_{p=1}^L r_{(p)}(\mathbf{R}), \quad (14)$$

where  $L < P$ . The per-point error is computed as

$$r_p(\mathbf{R}) = \min_q \|\mathbf{R} \tilde{\mathbf{x}}_{i,p} - \tilde{\mathbf{x}}_{j,q}\|_2, \quad (15)$$

and  $r_{(p)}(\mathbf{R})$  is the  $p$ -th largest value of the set

$$\{r_p(\mathbf{R})\}_{p=1}^P \quad (16)$$

By setting  $L < P$ , we obtain a robust solution since only the  $L$ -smallest residuals are minimised. In other words, points from  $I_i$  due to spurious events that have no correspondences in  $I_j$  (i.e., the outliers) have no effect on the solution. Problem (14) can be solved efficiently by a variant of ICP called *trimmed ICP*; see [10] for details of the algorithm.

It is computationally wasteful to attempt (14) on all pairs of event images, hence, we only conduct trimmed ICP on image pairs that are within a time window of fixed size  $W$  (we set  $W = 5$  in our experiments). This creates a set of relative rotation measurements

$$\{\tilde{\mathbf{R}}_{j,i}\}_{\langle j,i \rangle \in \mathcal{N}}, \quad (17)$$

where  $\mathcal{N}$  is the adjacency graph such that  $\langle j,i \rangle \in \mathcal{N}$  if and only if  $|i - j| \leq W$ .

### 3.3. Optimisation

The aim of optimisation is to denoise and fuse the different rotation measurements to yield accurate attitude estimates. This is accomplished efficiently using two novel formulations of rotation averaging and bundle adjustment.



### 3.3.1 Augmented rotation averaging

The original form of rotation averaging [18] takes as input a set of relative rotations and computes a set of absolute rotations. Put in our context, this is the problem

$$\min_{\{\mathbf{R}_i\}_{i=1}^M} \sum_{\langle j,i \rangle \in \mathcal{N}} \left\| \mathbf{R}_j - \tilde{\mathbf{R}}_{j,i} \mathbf{R}_i \right\|_F, \quad (18)$$

where  $\|\cdot\|_F$  is the Frobenius norm. However, this formulation is unsuitable since  $\mathcal{N}$  is a chain with no loops, and optimising the attitudes using (18) will lead to drift errors.

To mitigate drift error, the absolute rotations (11) must be factored into rotation averaging. To this end, we formulate the *augmented rotation averaging* problem

$$\begin{aligned} \min_{\{\mathbf{R}_i\}_{i=1}^M, \mathbf{R}_{M+1}} & \sum_{\langle j,i \rangle \in \mathcal{N}} \left\| \mathbf{R}_j - \tilde{\mathbf{R}}_{j,i} \mathbf{R}_i \right\|_F \\ & + \alpha \sum_{k \in \mathcal{A}} \left\| \mathbf{R}_k - \tilde{\mathbf{R}}_k \mathbf{R}_{M+1} \right\|_F \\ \text{s.t.} & \quad \mathbf{R}_{M+1} = \mathbf{I}, \end{aligned} \quad (19)$$

where  $\mathbf{R}_{M+1}$  is a “dummy” attitude variable,  $\mathbf{I}$  is the identity matrix, and  $\alpha$  is a positive constant that defines the relative importance of the relative and absolute rotations. Intuitively, adding error terms of the form

$$\left\| \mathbf{R}_k - \tilde{\mathbf{R}}_k \mathbf{R}_{M+1} \right\|_F = \left\| \mathbf{R}_k - \tilde{\mathbf{R}}_k \right\|_F, \quad k \in \mathcal{A} \quad (20)$$

encourage consistency between some of the attitude estimates and the measured absolute rotations, which is then propagated to the rest of the sequence.

Despite having the same form as (18) except for the constraint  $\mathbf{R}_{M+1} = \mathbf{I}$ , existing rotation averaging algorithms [26, 27, 18, 9, 7, 8] (which are tailored for (18)) cannot be directly applied to (19). A simple workaround is as follows: temporarily ignore the constraint  $\mathbf{R}_{M+1} = \mathbf{I}$  in (19) and optimise the attitudes using an existing rotation averaging algorithm (we used [9] in our work). Then, right multiply each of the estimated attitude  $\hat{\mathbf{R}}_i$  with  $(\hat{\mathbf{R}}_{M+1})^{-1} = (\hat{\mathbf{R}}_{M+1})^T$ .

It has been shown that rotation averaging is quite insensitive to initialisations [32, 16], thus, when solving (19) we simply initialise all rotation variables as the identity matrix.

### 3.3.2 Rotation-only bundle adjustment

The solutions from rotation averaging are then refined using *rotation-only bundle adjustment*. First, as a by-product of computing the relative rotations (14), we associate points across event images to form *star tracks*  $\{\mathcal{V}_s\}_{s=1}^S$ , where

$$\mathcal{V}_s = \{\mathbf{y}_{i,s} \in \mathbb{R}^2 \mid \eta(i, s) = 1\}, \quad (21)$$

$$\eta(i, s) = \begin{cases} 1 & \text{if the } s\text{-th star is seen in } I_i, \\ 0 & \text{otherwise,} \end{cases} \quad (22)$$

and  $\mathbf{y}_{i,s}$  are the 2D coordinates of the  $s$ -th observed star as viewed in event image  $I_i$ . We then define the nonlinear least squares (NLS) problem

$$\begin{aligned} \min_{\{\mathbf{R}_i\}, \{\mathbf{X}_s\}} & \sum_{i,s} \eta(i, s) \left\| \vec{\mathbf{y}}_{i,s} - \mathbf{R}_i \mathbf{X}_s \right\|_2^2 \\ \text{s.t.} & \quad \mathbf{X}_s \in \mathbb{R}^3 \text{ and } \|\mathbf{X}_s\|_2 = 1, \forall s, \end{aligned} \quad (23)$$

where  $\mathbf{X}_s$  defines the 3D direction of the  $s$ -th observed star as seen from the inertial frame. Intuitively, (23) is a special case of bundle adjustment [36] where only rotational motion and 3D directions are optimised.

Compared to (19), problem (23) takes into account the observed star coordinates in the estimation. To use (23) to refine the solutions of (19), we simply initialise the rotation variables in (23) using the output of (19), then apply a standard NLS solver (i.e., Ceres [6]) to carry out bundle adjustment. As with most bundle adjustment solvers, Ceres can take into account problem sparsity (i.e., not all points are seen in every frame) to speed up convergence.

To initialise the star directions  $\{\mathbf{X}_s\}$ , we compute

$$\mathbf{X}_s = \frac{1}{|\mathcal{V}_s|} \sum_i \eta(i, s) \hat{\mathbf{R}}_i^{-1} \vec{\mathbf{y}}_{i,s}, \quad (24)$$

i.e., the mean direction of the separate observations of the  $s$ -th star as seen in the inertial frame, using initial estimates of the attitudes  $\{\hat{\mathbf{R}}_i\}$  from rotation averaging (19).

Secs. 4 and 5 will describe the testing methodology and results of the proposed star tracking approach.

## 4. Generating testing data

Conducting a space mission for the purpose of our paper is beyond our budget. Following other works in astronautics research (e.g., [33]), we use simulation data to test our method. While it is possible to use the event data simulator by Mueggler et al. [31], for our problem it generates unrealistically clean data due to the simplicity of the scene (points at infinity with no other structures or occlusions).

The approach we have taken is to use the planetarium software Stellarium<sup>3</sup> to render real star fields on a screen, then capture the screen using an event camera, specifically the iniVation Davis 240C. Fig. 5(a) shows our setup. Since our target operating environment is space, where the atmosphere is much thinner, stars imaged in space will have constant brightness. This supports the usage of Stellarium which does not simulate atmospheric effects.

Given a fixed FOV and an arbitrary initial attitude, a rotational motion of a fixed angular velocity is executed in Stellarium to continuously change the attitude and generate event data. Since the motion is controlled, the ground truth

<sup>3</sup><https://stellarium.org>

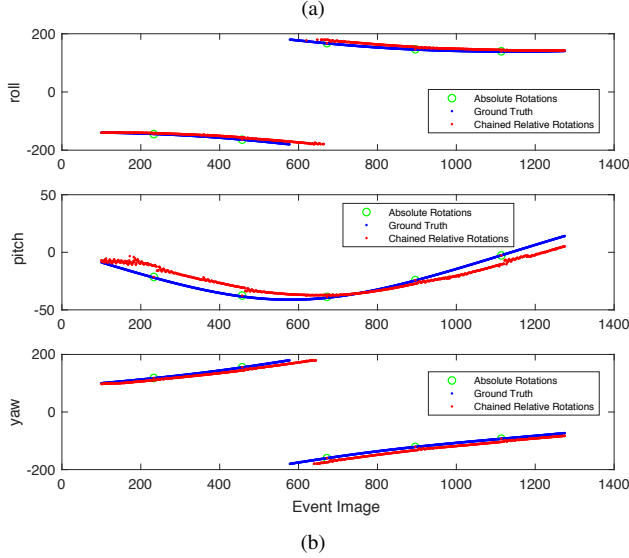


Figure 5. (a) Testing data generation rig. (b) Ground truth attitudes and initial attitude estimates (both sets expressed as Euler angles [3]) for a sample event data sequence generated according to Sec. 4. See Sec. 5.1 on deriving the initial attitudes from the rotation measurements computed according to Sec. 3.2.

attitudes throughout the recording duration can be analytically calculated; Fig. 5(b) illustrates ground truth attitudes for such a sequence. In the above way, we can generate event data sequences for objective testing. Before further describing the experiments and results in Sec. 5, we first explain other preprocessing and calibration steps.

#### 4.1. Removing spurious events

In our setup, the events are mainly triggered by two sources of intensity changes: the star motions and the screen refresh. Events due to the latter are considered spurious in our setting, thus we removed them using the Spatial Band-pass Filter tool in the jAER suite<sup>4</sup>. Of course there remain other noisy events due to inherent flaws of the sensor.

#### 4.2. Calibrating the virtual telescope

Mathematically, Stellarium acts as a “virtual telescope”. By imaging the screen output of Stellarium with an event

<sup>4</sup><https://github.com/SensorsINI/jaer>

camera, we can relate a pixel position  $\mathbf{x}$  on the event camera and the corresponding star direction  $\mathbf{X}$  with the equation

$$\bar{\mathbf{x}} = \mathbf{K}_{\text{ev}} \mathbf{H}_{\text{sc}} \mathbf{K}_{\text{te}} \mathbf{R} \mathbf{X} \equiv \mathbf{K} \mathbf{R} \mathbf{X}, \quad (25)$$

where  $\mathbf{K}_{\text{ev}}$ ,  $\mathbf{H}_{\text{sc}}$ ,  $\mathbf{K}_{\text{te}}$  and  $\mathbf{R}$  (all  $3 \times 3$ ) are defined as follows:

- $\mathbf{K}_{\text{ev}}$  is the intrinsic matrix of the event camera.
- $\mathbf{H}_{\text{sc}}$  is the homography that accounts for the non-fronto-parallel viewpoint of the event camera to the screen. A homography is mathematically valid since there is no parallax in star field images, and the screen and event camera image plane differ by a planar perspective transform.
- $\mathbf{K}_{\text{te}}$  is the intrinsic matrix of the virtual telescope (determines the magnification of the instrument etc.).
- $\mathbf{R}$  is the attitude of the telescope (the quantity of interest).

Calibrating for the overall intrinsic matrix  $\mathbf{K}$  (cf. Sec. 3.1) is thus achieved by calculating  $\mathbf{K}_{\text{ev}}$ ,  $\mathbf{H}_{\text{sc}}$  and  $\mathbf{K}_{\text{te}}$ .

The intrinsic matrix  $\mathbf{K}_{\text{ev}}$  can be estimated using existing techniques; see [4, Calibration]. The homography  $\mathbf{H}_{\text{sc}}$  can also be estimated using standard methods [19] given sufficient 2D-2D correspondences (manually extracted) between an event image and the screen output of Stellarium.

To compute  $\mathbf{K}_{\text{te}}$ , define the pinhole projection

$$\bar{\mathbf{z}} = \mathbf{K}_{\text{te}} \mathbf{R} \mathbf{X} \equiv \mathbf{P} \mathbf{X} \quad (26)$$

performed by Stellarium at a particular magnification, where  $\mathbf{z}$  are the image coordinates of a star direction  $\mathbf{X}$ . Given 2D-3D correspondences  $\{(\mathbf{z}_i, \mathbf{X}_i)\}_{i=1}^N$  between a set of observed stars in the Stellarium image and their corresponding star directions (the latter can be easily retrieved via the software interface), we set up the linear system

$$\begin{bmatrix} \bar{\mathbf{z}}_1 \\ \vdots \\ \bar{\mathbf{z}}_N \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_N \end{bmatrix} \quad (27)$$

and solve for  $\mathbf{P}$  using standard linear least squares. Then,  $\mathbf{K}_{\text{te}}$  and  $\mathbf{R}$  can be obtained from  $\mathbf{P}$  via QR decomposition.

## 5. Results

Using the methodology for generating testing data in Sec. 4, we generated a number of event data sequences to test our star tracking pipeline. In all sequences, we fixed the FOV on Stellarium to 20 degrees, the recording duration to 45 seconds, and the angular velocity to  $4^\circ/\text{s}$ . The integration time  $t_{\text{end}}^{(i)} - t_{\text{start}}^{(i)}$  for event image generation was fixed at  $40\text{ms}$  for all sequences, hence there are 1125 event images per sequence. In all the testing sequences, the initial attitude and rotational motion were chosen arbitrarily.

In this section, we show the results on six of the generated sequences; see the supplementary material for videos of the corresponding event images (results on more sequences are also available there), and [1] to obtain the data.

| Angular error        | $< 1^\circ$ | $< 10^\circ$ | $> 10^\circ$ |
|----------------------|-------------|--------------|--------------|
| # absolute rotations | 30          | 1            | 2            |

Table 1. Angular error of absolute rotations  $\{\tilde{\mathbf{R}}_i\}_{i \in \mathcal{A}}$ .

### 5.1. Error analysis of input rotations

On each testing sequence, via Sec. 3.2, we obtained absolute rotations  $\{\tilde{\mathbf{R}}_i\}_{i \in \mathcal{A}}$  and relative rotations  $\{\tilde{\mathbf{R}}_{j,i}\}_{(j,i) \in \mathcal{N}}$ . Let  $\{\mathbf{R}_i^*\}_{i=1}^M$  be the ground truth attitudes. The angular error between two rotations  $\mathbf{R}_1$  and  $\mathbf{R}_2$  [18] is

$$\angle(\mathbf{R}_1, \mathbf{R}_2) = 2 \arcsin \left( (2\sqrt{2})^{-1} \|\mathbf{R}_1 - \mathbf{R}_2\|_F \right). \quad (28)$$

To analyse the quality of the rotation measurements:

- Table 1 summarises the angular error between absolute rotations  $\{\tilde{\mathbf{R}}_i\}_{i \in \mathcal{A}}$  and corresponding ground truth rotations  $\{\mathbf{R}_i^*\}_{i \in \mathcal{A}}$  over the six sequences, where there are cumulatively 33 absolute rotations. It can be seen that most of the absolute rotations are reasonably accurate, indicating the effectiveness of the APC heuristic in selecting event images for star identification.
- The ground truth relative rotation between  $I_i$  and  $I_j$  is

$$\mathbf{R}_{j,i}^* = \mathbf{R}_i^* (\mathbf{R}_j^*)^T. \quad (29)$$

Table 2 displays the RMSE and standard deviation (SD) of the relative angular error  $\angle(\tilde{\mathbf{R}}_{j,i}, \mathbf{R}_{j,i}^*)$  in each sequence. It is evident that the relative rotations are very accurate ( $\leq 1^\circ$  RMSE), which supports our idea of using absolute rotations sparingly (since they are costly to compute), and relative rotations extensively (see Sec. 3.2).

To visualise the input rotations, in Fig. 5(b) we plot the attitudes that were obtained by chaining the absolute and relative rotations of the particular sequence. Note that there are multiple ways to chain the rotations, and an arbitrary chaining order was used in the figure - since chaining was done mainly for illustration (NB: our processing pipeline does not require it as initialisation), the choice is sufficient. The more important insight is that, while the *individual* input rotations were accurate, simply chaining them without further optimisation will lead to significant drift error.

### 5.2. Qualitative results

Fig. 6 plots the attitudes (in Euler angles) resulting from rotation averaging and bundle adjustment, and the ground truth attitudes, for the sequence in Fig. 5(b). It is evident that the proposed optimisation routines have reduced the errors significantly and accurately estimated the attitudes.

### 5.3. Quantitative results

Fig. 7 shows quantitative results for six sequences (more results in supplementary material). For each sequence, we plot the angular errors (as well as their RMSE and SD) between the ground truth attitudes and

| Seq # | 1      | 2       | 3      | 4      | 5      | 6      |
|-------|--------|---------|--------|--------|--------|--------|
| RMSE  | 0.0166 | 0.01276 | 0.2175 | 0.2345 | 0.3185 | 0.2234 |
| SD    | 0.1239 | 0.155   | 0.2525 | 0.3423 | 0.1421 | 0.3121 |

Table 2. Angular error of relative rotations  $\{\tilde{\mathbf{R}}_{j,i}\}_{(j,i) \in \mathcal{N}}$ .

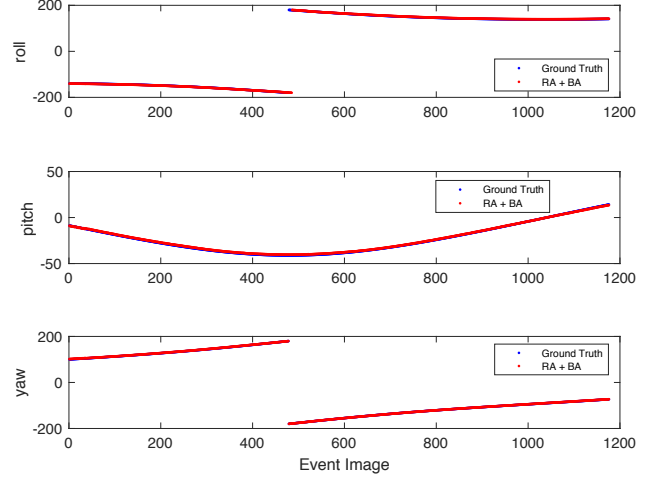


Figure 6. Visualisation of optimised attitudes and ground truth attitudes for a testing sequence. Both sets of values are very close.

- The initial attitudes obtained by an arbitrary way of chaining the input rotations (again, this is done only for illustration and comparison purposes, since our algorithms do not need to be initialised in this manner);
- The estimates after augmented rotation averaging; and
- The estimates after augmented rotation averaging and rotation-only bundle adjustment.

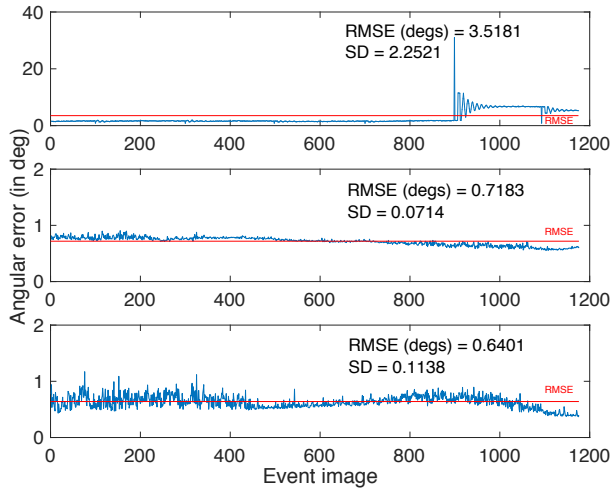
In all the sequences, our pipeline achieved an error of  $\leq 1^\circ$  RMSE, which is on par with commercial star trackers [33].

### 5.4. Runtime

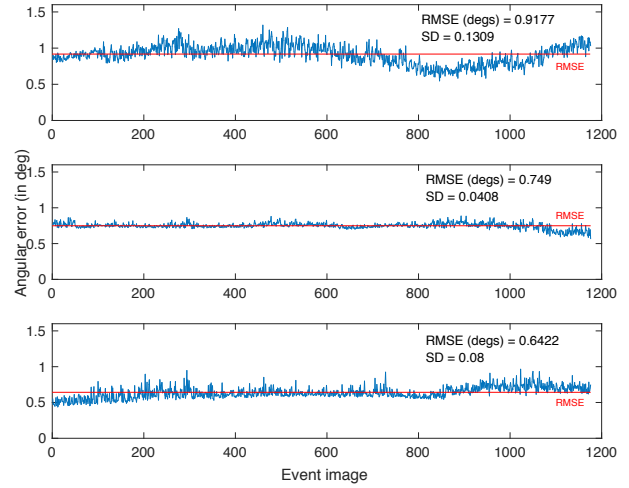
Executed on a 2.9GHz Intel i7 machine, the average total runtime per sequence is 58.8s, which includes time for event image generation (8.1s), extracting rotation measurements (41.3s) and optimisation (9.4s). A simple apportionment over the 1125 event images in each sequence gives  $\approx 19$  FPS. Note that this result is mainly to indicate the efficiency of the proposed algorithms - not the speed of the star tracker. As mentioned in Sec. 1.1, the actual speed will likely depend on overall system design.

## 6. Conclusion

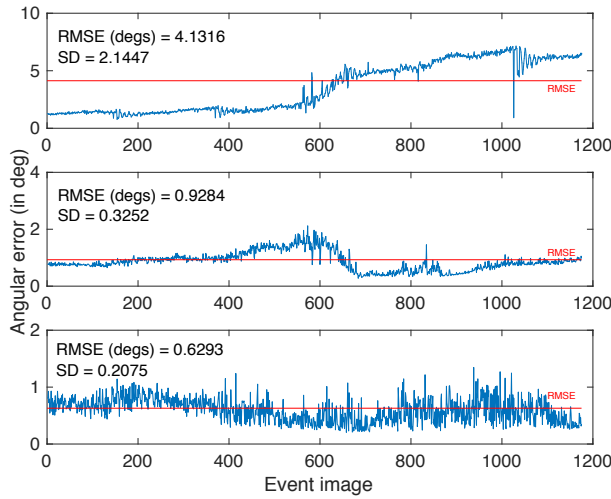
In this paper, we investigated event cameras for star tracking. The main components in our processing pipeline are novel formulations of rotation averaging and bundle adjustment. We also developed a simulation technique for testing our method. Our results suggest that star tracking using event cameras is feasible and promising.



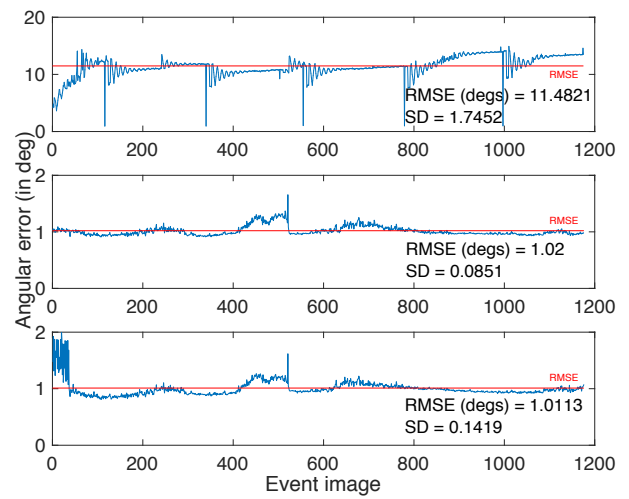
(a) Sequence 1



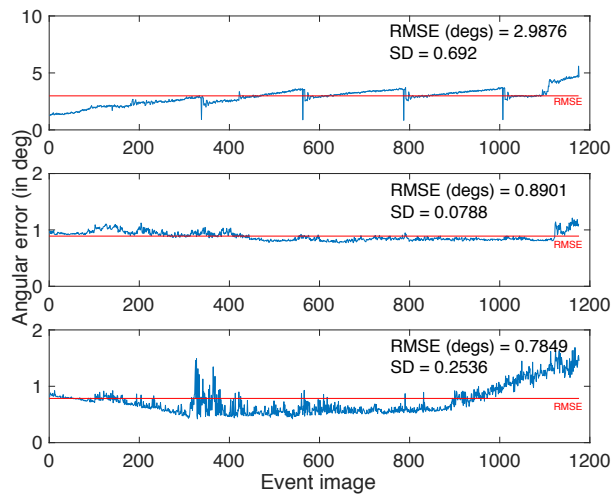
(b) Sequence 2



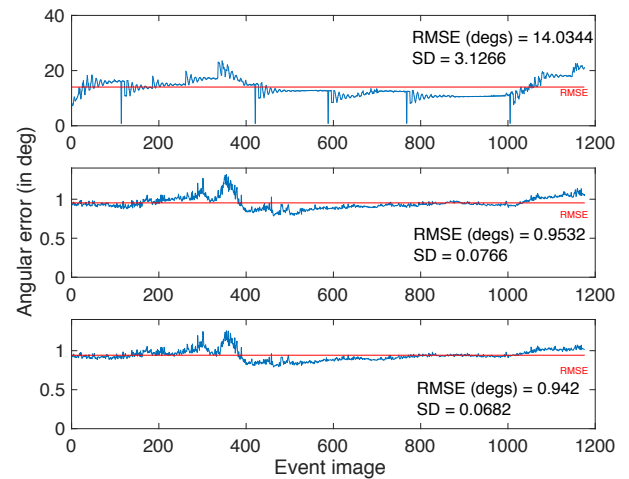
(c) Sequence 3



(d) Sequence 4



(e) Sequence 5



(f) Sequence 6

Figure 7. Quantitative results on six sequences (more results in supplementary material).



## References

- [1] <https://cs.adelaide.edu.au/~tjchin/startracking/>. 1, 7
- [2] <http://nova.astrometry.net/>. 1, 4
- [3] [https://en.wikipedia.org/wiki/Euler\\_angles](https://en.wikipedia.org/wiki/Euler_angles). 6
- [4] Event-based vision resources. [https://github.com/uzh-rpg/event-based\\_vision\\_resources](https://github.com/uzh-rpg/event-based_vision_resources). 2, 6
- [5] Wahba's problem. [https://en.wikipedia.org/wiki/Wahba's\\_problem](https://en.wikipedia.org/wiki/Wahba's_problem). 1, 4
- [6] S. Agarwal, K. Mierle, and Others. Ceres solver. <http://ceres-solver.org>. 5
- [7] F. Arrigoni, L. Magri, B. Rossi, P. Fragneto, and A. Fusiello. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *International Conference on 3D Vision (3DV)*, 2014. 5
- [8] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3d slam: A survey on rotation estimation and its use in pose graph optimization. In *International Conference on Robotics and Automation (ICRA)*, 2015. 5
- [9] A. Chatterjee and V. Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–528, 2013. 5
- [10] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *International Conference on Pattern Recognition (ICPR)*, 2002. 4
- [11] B. Cheung, M. Rutten, S. Davey, G. Cohen, and N. Gordon. Probabilistic multi hypothesis tracker for an event based sensor. In *International Conference on Information Fusion (FUSION)*, 2018. 2
- [12] T.-J. Chin and M. Tetlow. Robust attitude estimation to support space monitoring using nano-satellites. In *AIAA Space Forum*, 2014. 2
- [13] G. Cohen, S. Afshar, A. van Schaik, A. Wabnitz, T. Bessell, M. Rutten, and B. Morreale. Event-based sensing for space situational awareness. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, 2017. 2
- [14] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger. Interacting maps for fast visual interpretation. In *International Joint Conference on Neural Networks (IJCNN)*, 2011. 2
- [15] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch. Activity-driven, event-based vision sensors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2426–2429, 2010. 3
- [16] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin. Rotation averaging and strong duality. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5
- [17] G. Gallego, H. Rebecq, and D. Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3
- [18] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision (IJCV)*, 103(3):267–305, 2013. 5, 7
- [19] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2004. 6
- [20] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. Davison. Simultaneous mosaicing and tracking with an event camera. In *British Machine Vision Conference (BMVC)*, 2014. 2
- [21] H. Kim, S. Leutenegger, and A. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision (ECCV)*, 2016. 2
- [22] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis. Astrometry.net: blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal*, 139:1782–1800, 2010. 1, 4
- [23] P. Lichtsteiner, C. Posch, and T. Delbruck. A  $128 \times 128$  120 dB  $15 \mu\text{s}$  latency asynchronous temporal contrast vision sensor. In *IEEE Journal of Solid-State Circuits*, volume 43, pages 566–576, 2008. 2
- [24] C. C. Liebe. Accuracy performance of star trackers - a tutorial. *IEEE Trans. Aero. Electron. Syst.*, 38(2):587–599, 2002. 1
- [25] F. L. Markley and J. L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*, volume 33 of *Space Technology Library*. Springer-Verlag New York, 2014. 1, 2, 4
- [26] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 5
- [27] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *International Conference on Computer Vision (ICCV)*, 2013. 5
- [28] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza. Lifetime estimation of events from dynamic vision sensors. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 3
- [29] E. Mueggler, G. Gallego, and D. Scaramuzza. Continuous-time trajectory estimation for event-based vision sensors. In *Robotics: Science and Systems*, 2015. 2
- [30] E. Mueggler, B. Huber, and D. Scaramuzza. Event-based, 6-DOF pose tracking for high-speed maneuvers. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2014. 2
- [31] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *Int. J. Robotics Research (IJRR)*, 36(2):142–149, 2017. 6
- [32] C. Olsson and O. Enqvist. Stable structure from motion for unordered image collections. In *Scandinavian Conference on Image Analysis (SCIA)*, 2011. 5
- [33] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, C. Pernechele, and C. Dionisio. A new star tracker concept for satellite attitude determination based on a multi-purpose panoramic camera. *Acta Astronautica*, 140:166–175, 2017. 5, 7
- [34] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza. Evo: a geometric approach to event-based 6-DOF

parallel tracking and mapping in real-time. *IEEE Robotics and Automation Letters (RA-L)*, 2(2):593–600, 2017. [2](#)

- [35] B. B. Spratling and D. Mortari. A survey on star identification algorithm. *Algorithms*, 2(1):93–107, 2009. [1](#), [2](#), [3](#)
- [36] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372. Springer-Verlag, 2000. [5](#)