This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

EBS-EKF: Accurate and High Frequency Event-based Star Tracking

Albert W. Reed*, Connor Hashemi*, Dennis Melamed*, Nitesh Menon, Keigo Hirakawa†, and Scott McCloskey
Kitware
connor.hashemi@kitware.com
University of Dayton†
keigo.hirakawa@udayton.edu

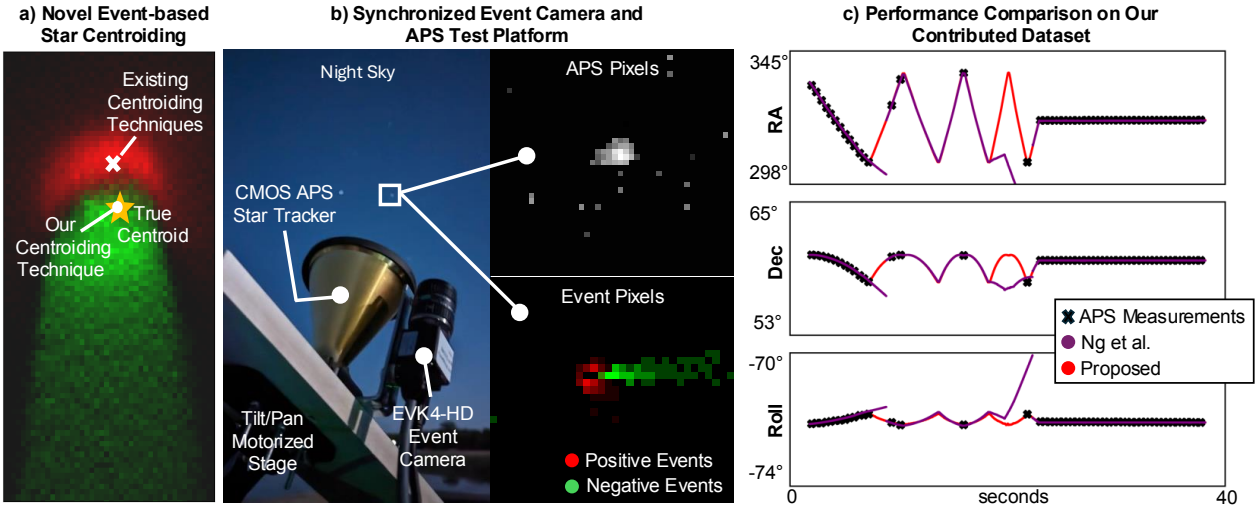


Figure 1. We present EBS-EKF, an event-based star tracking algorithm that combines a novel centroiding technique with an extended Kalman filter and is validated using the first ground-truthed dataset of event streams from real stars. (a) Our centroiding technique accounts for event camera behavior in low light, enabling more accurate tracking. (b) We develop a data collection setup of a EVK4-HD event camera that is rigidly mounted and synchronized with a space-ready APS star tracker. The insets show APS (top) and event camera (bottom) pixels for 8 Cygni, a subgiant star in the constellation Cygnus. (c) We demonstrate that our attitude estimates (red) are more accurate than existing methods (purple) [29], and that we can operate above the 3 deg/sec cutoff of the APS star tracker, highlighting the utility of our method for high-frequency star tracking.

Abstract

Event-based sensors (EBS) are a promising new technology for star tracking due to their low latency and power efficiency, but prior work has thus far been evaluated exclusively in simulation with simplified signal models. We propose a novel algorithm for event-based star tracking, grounded in an analysis of the EBS circuit and an extended Kalman filter (EKF). We quantitatively evaluate our method using real night sky data, comparing its results with those from a space-ready active-pixel sensor (APS) star tracker. We demonstrate that our method is an order-of-magnitude more accurate than existing methods due to improved signal modeling and state estimation, while providing more frequent updates and greater motion tolerance than conventional APS trackers. We provide all code and the first dataset of events synchronized with APS solutions.*

1. Introduction

Star trackers estimate the precise attitude — right ascension (RA), declination (Dec), and roll — of a camera with respect to the celestial coordinate frame, making them a standard sensor for space navigation. Compared to other sensors, star trackers provide highly accurate measurements,

*Denotes equal contribution.

*https://gitlab.kitware.com/nest-public/kw_ebs_star_tracking#

Table 1. Overview of related works. Color indicates high (green), medium (yellow) and low (red) performance. "s" denotes seconds.

Star Tracking Algorithm	Method Design		Key Features				
	State Estimation	Centroiding Method	Quantitative Evaluation	Camera State	Absolute Orientation Updates	Centroiding Accuracy	Update Rate
Chin et al. [8]	iterative closest point (ICP)	mean of all events	LCD screen	3D rotation	required often (e.g., every 5 s)	∼1.8 pixels	20-100 Hz
Bagchi et al. [3]	multi-resolution hough transform (HT)	mean of positive events	LCD screen	3D rotation	required often (e.g., every 5 s)	∼2.7 pixels	20-100 Hz
Ng et al. [29]	2D Kalman filter (2D-KF)	high-pass filtered intensity	LCD screen	2D position + 2D velocity	required intermittently (e.g., every 30 s)	∼3.0 pixels	500-1000 Hz
Ours	**3D extended Kalman filter (EBS-EKF)**	**mean of positive events with magnitude offset**	**night sky and LCD screen**	**3D rotation + 3D velocity**	**only on initialization**	**∼0.4 pixels**	**500-1000 Hz**

with attitude estimate errors on the order of arcseconds (1 arcsecond = 1/3600 degrees). Modern trackers generally employ active-pixel sensor (APS) cameras to identify star centroids and determine camera attitude. While APS trackers perform well when stationary, their finite exposure times and frame processing overhead usually limit update rates to 2-10 Hz, constraining performance during rapid attitude adjustments [27].

Event-based cameras have recently emerged as a promising technology for enhancing star tracker performance. Also known as event-based sensors (EBS), event cameras mimic human vision by capturing microsecond-resolution "events" in each pixel, indicating whether the light reaching a pixel has increased or decreased by a specified threshold. Unlike APS cameras, EBS measurements are asynchronous and can reach temporal resolutions down to the microsecond, suggesting that event-based star trackers could achieve significantly faster update rates. Furthermore, because events are triggered only by changes in the scene, most pixels remain inactive when imaging stars in the night sky, which could lead to lower power consumption.

Given these potential advantages, several recent studies have investigated the utility of EBS-based star tracking [3, 8, 23, 29]. While these studies have introduced new algorithms, many questions remain before EBS star trackers can potentially replace APS trackers. First, all quantitative evaluations thus far have been conducted in simulation, where star fields are displayed on a screen. To date, the performance of EBS cameras on real-world star fields, where noise and low-light imaging pose significant challenges, has not been quantified. Second, existing techniques from literature yield considerable variability in EBS star centroiding, leading to suboptimal attitude estimation. Finally, no real-world EBS datasets or open-source EBS trackers are currently available, making it challenging to benchmark and improve existing methods.

In this work, we present a novel event-based star centroiding technique by modeling the dynamics of a star imaged with an EBS circuit. Leveraging this model, we design an event-based star tracker based on an extended Kalman filter (EKF). We evaluate our algorithm in both lab settings and real night sky experiments. Our specific contributions are as follows:

1. A novel method for event-based star centroiding, based on rigorous analysis of event camera circuit behavior in low-light conditions.
2. An extended Kalman filter tailored for EBS-based star tracking, providing 3D attitude and velocity measurements with update rates up to 1 KHz and enhanced accuracy.
3. The first dataset of events paired with APS tracker quaternion solutions, collected from real starfields, enabling quantitative evaluation of our method against an APS tracker and existing EBS techniques.
4. Our code and dataset are released as open-source to support reproducibility of research.

2. Related Work

EBS Imaging: An EBS detects changes in pixel brightness by generating measurement tuples termed "events". Events indicate when a pixel's intensity rises or falls past a set threshold; each event includes the pixel's location, a microsecond timestamp, and polarity (i.e. "positive" if intensity increased and "negative" if it decreased) [13, 26]. Compared to APS, EBS offers lower power consumption, faster pixel response, and a larger dynamic range. EBS imaging has proven useful for many applications, including autonomous driving [14], robotics [37], high-FPS cameras [21, 31], wavefront imaging [16], and photogrammetry [38].

APS Star Tracking: In APS tracking, stationary stars are typically modeled as 2D Gaussians on the imaging plane, then localized using center-of-gravity (COG) [2, 33] or Gaussian fitting [9, 36] techniques. Once stars are localized on the image plane, various techniques can translate these positions into attitude estimates. A common approach, and the one employed by our reference APS tracker, estimates an *absolute* rotation by cross-referencing identified stars with a star catalog (we use *Astrometry.net* [22]). Other methods derive relative rotations from the frame-to-

frame optical flow [25, 34].

EBS Star Tracking: Chin et al. [8] were the first to address event-based star tracking, proposing a method that integrates events into synchronous frames and uses the iterative closest point (ICP) algorithm to estimate frame-to-frame rotations; we shorthand the algorithm from [8] as "ICP" throughout this paper. They observed that this technique is prone to noisy attitude estimates and drift, necessitating periodic absolute measurements from a reference star catalog [22]. The authors proposed a follow-up work that used the Hough transform to asynchronously update the tracker rotation [3] from streaks in the event stream — we refer to this method as "Hough". While this approach demonstrated asynchronous EBS star tracking, it achieved higher error rates compared to ICP [8].

More recent works propose an asynchronous 2D Kalman filter for EBS star tracking [23, 29], which we call "2D KF" throughout this paper. The Kalman filter models star position and velocity as 2D linear translations and ignores the non-linear effects of roll. Their simulation results demonstrated superior update rates and accuracy to prior works [3, 8].

The asynchronous Kalman filter works are most related to ours, but we introduce several key advancements. First, previous works either implicitly [3, 8, 23] or explicitly [29] model the likelihood of events using a Gaussian distribution centered on the star, whereas we model it as a Gaussian whose offset from the star's location depends on the star's intensity. Secondly, we employ an asynchronous extended Kalman filter that estimates 3D rotations and angular velocities, and demonstrate that 3D modeling is essential for effective tracking on real data.

As summarized in Table 1, existing methods quantify tracking performance on simulated starfields. Ng et al. [29] demonstrate an ability to track stars in the night sky but do not quantify tracking performance due to lack of a ground truth. Our work is the first to quantify the performance of event-based tracking algorithms on real data.

3. EBS-EKF Method

We first describe our problem setting and camera geometry. We then define a signal model that relates the likelihood of events to star positions. We use this signal model in conjunction with an extended Kalman filter to perform high-frequency and accurate event-based star tracking.

3.1. Measurement Geometry

As illustrated in Fig. 2, we aim to use EBS measurements of stars to estimate the camera state $\mathbb{S} = [\mathbf{q}, \omega]$, where $\mathbf{q} \in SO(3)$ is a quaternion in the 3D rotation group, and $\omega \in \mathbb{R}^3$ denotes the 3D angular velocity. We measure the set of N stars $\mathbf{S} \in \mathbb{R}^{N\times 3}$; each star is 3D point normalized to the unit sphere with negligible parallax effects [12]. Each

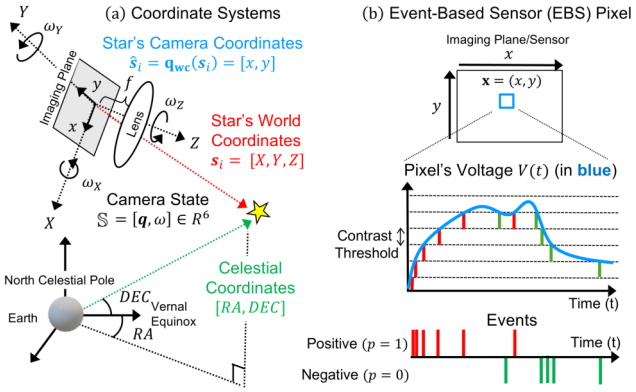


Figure 2. Measurement model for event-based star tracking. (a) depicts the coordinate systems used; (b) depicts an example operation of an EBS pixel as a star's projection passes over it.

star has an apparent magnitude m_{s_i}, which is a measure of the star brightness on a reverse logarithmic scale (i.e., smaller values indicate brighter stars)[†]. A single star $\mathbf{s}_i \in \mathbf{S}$ with 3D coordinates (X, Y, Z) is rotated into the camera's frame using a world-to-camera quaternion, $\hat{\mathbf{s}}_i = \mathbf{q}_{\text{wc}}(\mathbf{s}_i)$[‡], where $\mathbf{q}_{\text{wc}}(\cdot)$ denotes the quaternion's rotation of a 3D vector. After rotating, the star is projected onto the 2D imaging plane using the pinhole camera model:

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z}, \tag{1}$$

where f is the focal length. The EBS indicates a change of intensity at a pixel by triggering an event $\mathbf{e}_i = (\mathbf{x}, p, t)$, where $\mathbf{x} = (x, y)$ denotes the pixel, p denotes the event polarity, and t denotes the timestamp.

Celestial Coordinates: As illustrated in Figure 2, a camera's attitude can be defined as the optical axis' orientation to the celestial coordinate frame, in degrees of right-ascension (RA), declination (Dec), and roll (about the z-axis). In alignment with star tracking literature, we use *across* orientation (i.e., pointing direction) to refer to Ra and Dec, and *about* to refer to roll orientation.

3.2. Event Likelihood Model

Continuous-time Event Likelihood: We now derive the likelihood of an event triggering based on a star's brightness and dynamics on the imaging plane. A model for EBS operation is shown in Fig 2b, where the EBS pixel's photoreceptor converts irradiance to pixel voltage and triggers events [26].

In this work, we define the *event likelihood* $E(t)$, which is a model that defines the event rate of a EBS pixel $\mathbf{x}$ at each time t. Because there are two polarities of events, $E(t) > 0$

[†]This works considers the set of stars with apparent magnitude ≤ 7.

[‡]We typically omit the wc subscript for brevity.

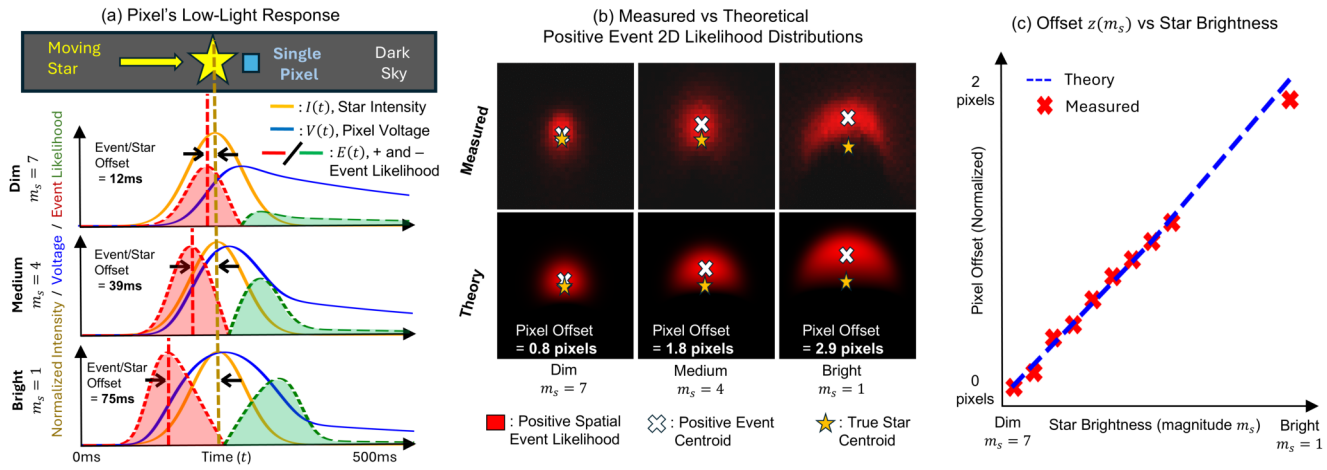


Figure 3. In low-light conditions, a star's brightness introduces a variable delay between the observed peak of positive events and its true locations. (a) depicts the numerical solution of event likelihood $E_{\text{LL}}(t)$ in Eq 4 compared to star intensity $I(t)$ for a single pixel. Bright stars produce a distribution of positive and negative events that follows the derivative of the intensity. Dim stars, however, yield a distribution of positive events that peaks nearer to the peak intensity, and the negative events fall to the tail of the distribution. (b) compares the theoretical spatial positive event likelihood distribution (calculated by solving Eq 6) to those measured in our night sky dataset. (c) depicts the offset $z(m_s)$ between the centroid of the positive events and the star's true centroid as a function of the star's brightness (i.e. relative magnitude m_s). The offsets are normalized to have a minimum offset of 0.

defines the likelihood of a positive event while $E(t) < 0$ defines the likelihood of a negative event. In bright (i.e. ≥ 100 lux) conditions, the event likelihood can be modeled as the continuous change in the photoreceptor's voltage over time, or

$$E(t) = \frac{dV(t)}{dt} = \frac{d}{dt}\log(I(t)/I_0 + 1), \tag{2}$$

where $V(t)$ is the pixel's voltage, $I(t)$ is the pixel's image intensity, and I_0 is an intensity offset defined by the circuit and the dark current [15].

Following existing star imaging optics, the star's point-spread function on the sensor can be well-approximated by a multivariate, isotropic Gaussian [9]. Therefore, we model the star intensity $I(t)$ at pixel location $\mathbf{x}$ as

$$I(t) = \frac{1}{\sqrt{2\pi}}|\Sigma|^{-\frac{1}{2}}\exp\Big(-\frac{1}{2}(\mathbf{x}_0 - \mathbf{v}t)\Sigma^{-1}(\mathbf{x}_0 - \mathbf{v}t)^T\Big), \tag{3}$$

where $\mathbf{x}_0 = (x_0, y_0)$ is the star's offset from the pixel $\mathbf{x}$ at $t = 0$, $\mathbf{v} = (v_x, v_y)$ is the star's image-plane velocity, and $\Sigma = \sigma_s^2\mathbf{I}_{2x2}$, where σ_s^2 is the star's variance across pixels. Plugging the star intensity of Eq. 3 into the event rate of Eq. 2 defines the measurement model leveraged by existing works [29], where likelihood $E(t)$ is derived as the derivative of the Gaussian intensity.

However, ***our key insight is that the event likelihood's position and shape depends on star brightness***, which can be explained by low-light EBS circuit phenomenology. In low-light, the bandwidth of the EBS circuit is proportional to the photocurrent, making the pixel's voltage $V(t)$ behave like a single-order low-pass filter with an intensity-dependent cutoff frequency f_c [19, 26, 28, 30]. We incorporate this into a low-light (LL) event likelihood model using the first-order differential equation

$$E_{\text{LL}}(t) = \frac{dV(t)}{dt} = 2\pi \cdot f_c(\tilde{I}(t)) \cdot \Big[\tilde{I}(t) - V(t)\Big], \tag{4}$$

where $\tilde{I}(t) = \log(I(t)/I_0 + 1)$ is the pixel's photocurrent and $f_c(\tilde{I}(t))$ is the intensity-dependent cutoff frequency [19, 30]. We model the bandwidth dependence as a linear function of photocurrent $\tilde{I}(t)$:

$$f_c(\tilde{I}(t)) \approx b + a\tilde{I}(t), \tag{5}$$

where b is the cutoff frequency when the pixel is dark (i.e. $\tilde{I}(t) = 0$) and a is its linear dependence on photocurrent $\tilde{I}(t)$. Intuitively, the lower bandwidth associated with dim stars delays the peak of the event likelihood compared to the higher bandwidth of bright stars. Since the event likelihood follows the derivative of a Gaussian, this implies that the likelihood for bright stars generally leads the true centroid, while the likelihood for dim stars lags (or coincides) with it. We observe that this linear model facilitates accurate prediction of the pixel offsets observed in our real night sky data, as shown in the agreement of theory and measured offset curves in Figure 3c.

As shown in Figure 3a, the negative event likelihood exhibits high entropy for dim stars. Considering their significant latency in low-light and substantial processing overhead, we choose to process only positive events.

Spatial Event Likelihood: Using our continuous-time event likelihood, we can express the likelihood of a positive event $e_i = (\mathbf{x}_i, t_i)$ as:

$$\mathcal{L}(e_i|\mathbf{x_0}, \mathbf{v}) = \max\big(0, E_{\text{LL}}(t_i - (\mathbf{x}_i - \mathbf{x}_0)/\mathbf{v})\big), \quad (6)$$

given a star with constant linear velocity $\mathbf{v}$ and position $\mathbf{x_0}$.[§] Figure 3b illustrates several example spatial likelihoods, where the positive event mean is offset from the true centroid depending on the star brightness. Figure 3c plots this offset, and we observe agreement with real data when numerically solving our model with choice parameters of the dark current I_0, cutoff-frequency slope a, and offset b. We note the presence of a "bowshock" effect for bright stars (also observed by [5]), where the edges of the event likelihood are more delayed than the center — this is due to a lower cutoff frequency in the dimmer regions of the star, which induces more delay relative to the bright center.

Gaussian Approximation: We aim to use the likelihood in Eq. 6 for real-time estimation of the camera attitude, but it is a first-order differential equation that is computationally expensive to solve. Instead of using it directly, we propose approximating it with an isotropic Gaussian distribution:

$$\mathcal{L}(e_i|\mathbf{x_0}, \mathbf{v}) \sim \mathcal{N}\Big(\mathbf{x}_i - \big[\mathbf{x}_0 - \bar{\mathbf{v}} \cdot z(m_s)\big], \sigma_s^2 \mathbf{I}_{2x2}\Big), \quad (7)$$

where the function $z : \mathbb{R} \to \mathbb{R}$ corrects for the star's magnitude-dependent m_s offset (see Figure 3c) and $\bar{\mathbf{v}}$ denotes the normalized linear velocity. We quantify the accuracy of this approximation in Section 4.

Algorithm 1 EBS-EKF Star Tracking

Input: Positive event stream $\mathcal{E} = \{\mathbf{x}, t\}_{i=0:N}$,
Output: Camera State $\mathbb{S}$ (3D rotation and angular velocity)
Initialize: $\mathbb{S}_0 \leftarrow$ astrometry [22] with binned events
r $\leftarrow$ max search radius
for each event $e_i = (\mathbf{x}_i, t_i)$ **do**
  $\hat{\mathbb{S}}_i \leftarrow$ EKF predict at t_i
  $\mathbf{x}_s \leftarrow$ closest projected catalog star to $\mathbf{x}_i$
  if distance($\mathbf{x}_i$, $\mathbf{x}_s$) $\leq$ r **then**
    $\bar{\mathbf{v}}_s \leftarrow$ star's normalized linear velocity via $\hat{\mathbb{S}}_i$
    $m_s \leftarrow$ star's apparent magnitude
    $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i + \bar{\mathbf{v}}_s \cdot z(m_s)$ # apply offset correction
    $\mathbb{S}_i \leftarrow$ EKF update with $\hat{\mathbf{x}}_i$
  end if
end for

3.3. State Estimation and Tracking Algorithm

We now use the event likelihood in a Bayesian estimation framework to estimate the camera state. Our measurements are the set of N positive events $\mathcal{E}_N = \{(\mathbf{x}_i, t_i)\}_{i=0:N}$, where we denote each event as $\mathbf{e}_i \in \mathcal{E}$. Under a Markov assumption, the probability distribution over all states and measurements is,

$$p(\mathbb{S}_0,...\mathbb{S}_N; e_0,...e_N) = p(\mathbb{S}_0)\prod_{i=0}^{N}\mathcal{L}(e_i|\mathbb{S}_i)p(\mathbb{S}_i|\mathbb{S}_{i-1}). \quad (8)$$

Intuitively, this formulation estimates the states as an interpolation of the measurement likelihood $\mathcal{L}(e_i|\mathbb{S}_i)$ and prior $p(\mathbb{S}_i|\mathbb{S}_{i-1})$ on the state's evolution. Notably, we define the measurement likelihood $\mathcal{L}(e_i|\mathbb{S}_i)$ as conditioned on the state $\mathbb{S}$; the state defines the star position $\mathbf{x}_0$ and velocity $\mathbf{v}$ shown in Eq. 7.

Extended Kalman Filter: We propose using an EKF to recursively estimate the posterior distribution $p(\mathbb{S}_i|\mathcal{E}_i)$ derived from Eq. 8, which is the state conditioned on the measurements up to the current time step [4][¶]. The EKF estimates the state through *predict* and *update* phases, which require key matrices as input: the forward model Jacobian $\mathbf{F}$, processing noise $\mathbf{Q}$, and measurement matrix $\mathbf{H}$. In the supplemental material, we derive the measurement matrix based on the projection geometry of Section 3.1 and the forward model and process noise matrices for a general constant velocity model in 3D space. We find that the EKF enables more accurate state estimation compared to other methods (see results) and recognize the potential of many other state estimation techniques (e.g., [10, 35]) as avenues for future research.

Full Tracking Algorithm: Alg. 1 outlines our algorithm for estimating the camera state from events, for which we provide more details in the supplemental material. We initialize the camera attitude by binning events with a small time window (e.g., 30 ms), identifying event clusters with DBSCAN clustering [11], and passing the cluster centroids to astrometry [22] to determine an initial attitude. We then use events to update the state. Specifically, we use the event's timestamp to update the EKF prediction, and check if the event is within the radius of a catalog star, given the current attitude. If so, we use the star's magnitude and linear velocity to apply our intensity-dependent offset correction, and then use it to update the camera state using our EKF update equations (see supplemental). We note that this algorithm is asynchronous and operable from single positive events, facilitating the 1 KHz update rate used in this work.

4. Results

We now discuss validation of our event-based centroiding theory and star tracking algorithm.

4.1. Laboratory Centroiding Experiments

Our first set of experiments are performed in a laboratory setting where we can manually configure star brightness,

[§]The max operator effectively ignores negative events by ensuring the likelihood remains positive.

[¶]We provide more details on Eq. 8's Bayesian formulation of the Kalman filter in the supplementary material.

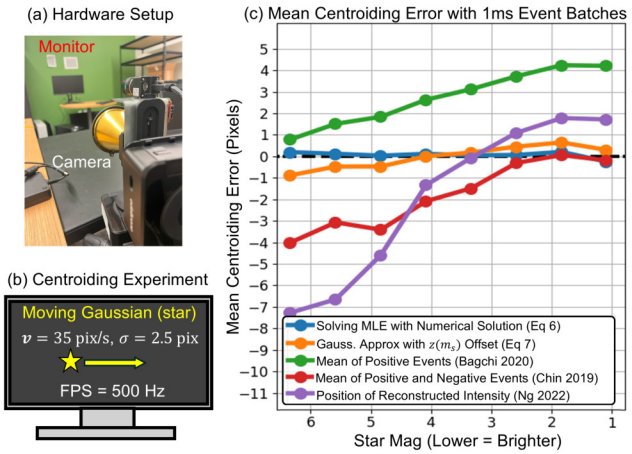


Figure 4. Testing the centroiding accuracy of Eqs 6 and 7 via an LCD monitor. (a) depicts the hardware setup. (b) depicts the centroiding experiment on the high-update rate monitor, where the star is moving at 35 pixels per second and has a 2.5 pixel standard deviation to emulate stars in our night sky dataset. (c) depicts the centroiding accuracies of our methods (blue and orange curve) compared to those of previous works. The black dotted line depicts zero centroiding error. Details about the synchronization and other aspects are included in supplementary.

position, and velocity. We use these experiments to quantify the accuracy of centroiding methods and to evaluate the accuracy of our Gaussian approximation (Eq. 7) for the analytical likelihood (Eq. 4).

Implementation: Our experimental setup shown in Fig 4(a-b) consists of an EVK4-HD event camera focused on a high-refresh rate monitor. We use the monitor to simulate stars by displaying gamma-corrected, isotropic Gaussians that move at a constant speed. We set the apparent magnitude of the stars on the monitor with a combination of placing neutral-density (ND) filters in front of the camera and adjusting the brightness of display pixels. We use the transformation matrix between camera and monitor pixels to compute an error between estimated and ground truth star positions. We provide more implementation details in the supplemental.

Results: Fig. 4c displays the centroiding results for all methods against different star brightness levels. We use 1 ms batches to estimate the centroid using: (1) maximum likelihood estimation (MLE) of our analytical model in Eq. 6 (blue), (2) our Gaussian approximation with intensity-dependent offset of Eq 7 (orange), (3) the mean of the positive events [3] (green), (4) the mean of positive and negative events [8, 23] (red), and (5) the position of the reconstructed intensity [29] (purple). Our analytical model is the most accurate of the set (error of 0.16 ± 0.18 pixels), but requires numerically solving the differential equation in Eq 4. Our Gaussian approximation has an error of 0.42 ± 0.48 pixels, and is orders of magnitudes faster to calculate, taking less

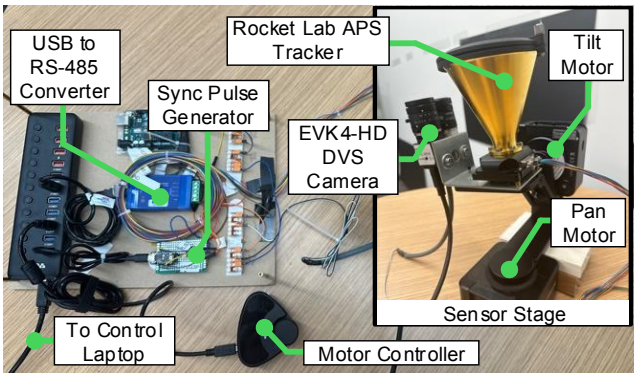


Figure 5. Our night sky data collection system with key components labeled. The sync pulse generator ensures that we can temporally correlate the Rocket Lab APS measurements with the EVK4-HD event camera measurements. The Pan and Tilt Motors are used to sweep both star trackers across the sky based on predefined motion patterns.

than 0.1ms per batch compared to over 100ms for the analytical model. Both of our methods outperform previous approaches in terms of centroiding accuracy and consistency, as they account for intensity-dependent centroiding effects and low-light circuitry effects, which prior methods do not.

4.2. Real World Star Tracking Experiments

We also quantify the performance of ours and existing methods on event data captured from stars in the night sky.

Hardware Configuration: As illustrated in Figure 5, we time-synchronize the event streams from a Prophesee EVK4-HD sensor with the outputs of a space-ready APS star tracker (Rocket Lab ST-16RT2) via a synchronization pulse generator as their fields of view are moved across the night sky. The EVK4-HD sensor is fitted with a 35mm lens, yielding a field of view (FOV) of 10.2°x5.7°. The ST-16RT2 is factory-installed with a 16mm lens, yielding a FOV of 20°x15°. We compute a relative rotation between the rigidly co-mounted cameras for use in our quantitative evaluation. Specifically, we can use the relative rotation to compute a difference between APS and EBS tracks as $||\mathbf{q}_{\text{APS}} \cdot \mathbf{q}_{\text{relative}} - \mathbf{q}_{\text{EBS}}||_2^2$, where $\mathbf{q}$ denotes a quaternion, the dot operator denotes quaternion multiplication, and the minus operator denotes the difference between quaternions. We provide additional details on obtaining their relative rotations in the supplemental material. We also provide details on the cameras' parameters and our method for obtaining the offset correction curve plotted in Fig. 3 (c).

Metrics and Dataset: We report the mean squared difference between the EBS and APS tracks in both across (pointing) and about (roll) accuracies, as is standard in literature [27]. Performance is measured relative to solutions provided from the Rocket Lab tracker, which has its own error margins. Specifically, the Rocket Lab tracker is rated

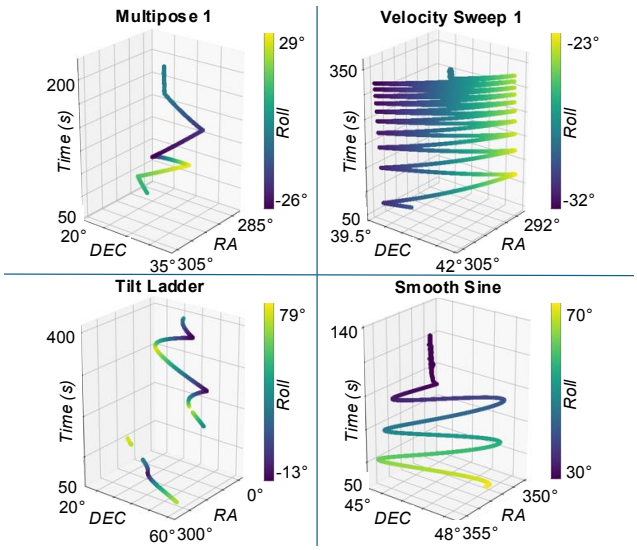


Figure 6. Plots of four real night-sky tracks that were estimated with our EBS-EKF algorithm. Right Ascension and Declination are shown along the bottom, and time flows vertically. Roll is indicated by the color of the line.

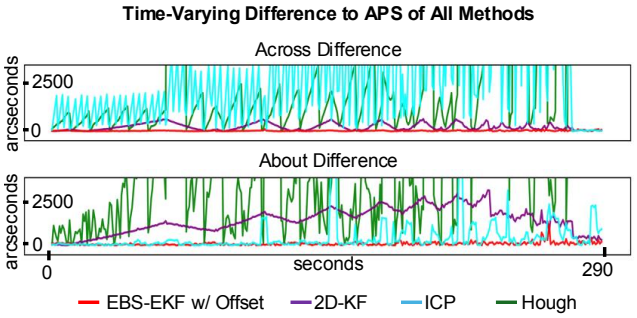


Figure 7. Real World Velocity Sweep 1. Differences of all methods with APS reference measurements. Our method obtains significantly lower difference with the APS reference measurements. Existing methods drift off track by hundreds of arcseconds due to inaccurate centroiding and erroneous state estimation.

for 5 arcseconds across and 50 arcseconds about at slew rates up to 3 deg/sec in yaw and pitch. As such, we speak of methods' performances in terms of their *difference* with respect to the APS tracker. We report differences in arcseconds (denoted using ").

This paper tests all methods on 14 real star data tracks total, which were collected from multiple locations in North America. Track durations are between 1 and 5 minutes, and were acquired using various camera trajectories, velocities, and regions of the night sky. Visualizations of four tracks (estimated with our tracking algorithm) are shown in Figure 6. We summarize key dataset details in the top portion of Table 2 and provide additional details in the supplementary material.

Method Implementations: We implemented all methods in Python and ran experiments on an M2 Silicon MacBook. Open-source code was not available for existing methods (2D-KF [29], ICP [8], and Hough [3]), so we implemented it to the best of our abilities and performed parameter sweeps to optimize the performance on our dataset. Unlike our method, existing approaches require frequent absolute measurements to stay on track. Absolute measurements are obtained by cross-referencing centroids with a star catalog (in our case, using *Astrometry.net* [22]). While they are typically too slow for real-time processing, they are required by existing methods [3, 8], and can be computed offline. Thus, in the spirit of maximizing their performance, we chose to provide frequent (i.e., every 5 seconds) precomputed absolute measurements to ICP and Hough and to 2D KF when it lost track.

Results: We summarize the performance of all methods on our night-sky tracks in Table 2. We observe that our method is typically within 100 arcseconds (across and about) of the APS tracker, whereas existing methods deviate by up to 1-2 degrees (i.e., several thousand arcseconds). 2D-KF is the most competitive to ours, but periodically drifts off track due its 2D state estimation failing to model the 3D dynamics of real data. Notably, ICP is more accurate than Hough, in agreement with reported results [3]. Figure 7 plots the time-varying difference for all methods, showing that existing methods generate saw-tooth shaped difference plots — they drift off track and accumulate difference, but are then snapped on track by the periodic absolute measurements.

The bottom rows of Table 2 summarize our method's performance with and without using the intensity-dependent centroiding offset. We observe that it typically improves performance by 10-20 arcseconds, usually when a bright star is within the FOV. Since there is 1-2 pixel centroid offset between our brightest and dimmer stars (see Fig. 3c), and the FOV of each event pixel is approximately 30 arcseconds, improvements within these ranges agree with intuition. We provide an example scenario from Velocity Sweep 1 of how a bright star can drastically affect EBS tracking performance in Fig. 8, showing that attitude estimation error spikes without using our offset correction.

High Velocity Track: Figure 1c compares our method with 2D-KF on a 7.5 deg/sec velocity track, which is not included in Table 2 since the APS tracker fails to provide consistent solutions past the velocity of 3 deg/sec. Both ours and Ng et al.'s [29] method produce high frequency estimates of the tracks, but only ours reconstructs the correct track, and it maintains an order of magnitude closer agreement to the available APS measurements (total difference of 80.4 arcseconds versus 774.3 arcseconds, respectively). We provide a video of this experiment in the supplemental material.

	Track Name												
	Velocity Sweep							Multipose				Tilt Ladder	Smooth Sine
	1	2	3	4	5	6	7	1	2	3	4		
	Metadata												
Duration (s)	290	290	167	160	160	160	150	151	150	145	265	342	85
RA / Dec / Roll Range (°)	13.9 2.3 8.9	9.8 6.6 14.4	6.8 2.9 0.8	5.7 3.8 0.9	3.6 4.7 1.1	1.3 5.2 1.3	6.4 3.2 5.4	21.7 17.2 55.2	28.01 18.6 36.0	7.1 2.7 44.9	11.3 8.1 90.2	0.01 44.4 92.8	5.1 3.8 40.3
Max Velocity (°/s)	1.8	1.8	1.8	1.8	1.8	1.6	1.7	0.9	0.8	0.8	0.8	0.6	0.9
Star Magnitude (Brightest)	2.2	3.1	3.8	3.6	3.6	3.6	0.0	2.5	3.1	3.6	3.6	2.0	3.8
	Results (across/about) in arcseconds (↓ better)												
ICP [8]	3.3E3 394.6	3.0E3 1.2E3	2.7E3 379.8	2.5E3 404.6	2.6E3 329.2	2.8E3 729.7	2.8E3 680.8	1.6E3 2.2E3	1.6E3 2.1E3	591.3 3.1E3	713.8 3.4E3	2.3E3 6.2E3	1.2E3 2.5E3
Hough [3]	8.6E3 8.7E4	5.0E3 4.8E4	2.6E4 5.2E4	3.5E4 3.4E4	3.1E4 2.5E4	1.8E4 5.9E4	1.1E3 1.6E4	1.1E4 5.7E4	1.6E4 4.8E4	1.5E4 6.3E4	8.2E3 2.9E4	1.1E4 6.5E4	7.6E4 7.3E4
2D-KF [23, 29]	228.9 1.3E3	236.4 1.0E3	92.6 1.2E3	81.9 1.0E3	62.6 813.6	67.9 1.3E3	171.4 2.1E3	337.4 7.3E3	294.6 6.6E3	846.0 1.6E4	607.2 1.2E4	485.1 9.2E3	221.8 7.3E3
EBS-EKF w/o offset	27.0 83.1	31.9 109.3	25.9 84.5	24.2 88.6	26.0 80.8	22.1 86.1	172.1 322.0	57.3 64.8	**71.1** **70.2**	**21.6** **66.2**	17.0 84.7	50.3 79.1	26.7 84.5
EBS-EKF w/ offset	**25.8** **60.3**	**31.7** **87.9**	**26.1** **75.3**	**25.2** **69.1**	**27.0** **77.9**	**22.1** **78.1**	**170.1** **139.0**	**57.4** **52.8**	70.9 74.1	24.7 70.5	**18.6** **82.9**	**49.1** **64.5**	**28.6** **75.9**

Table 2. Metadata and tracking results for our real world dataset. Metrics are reported as (across, about) mean squared difference in arcseconds. Colors rank methods with the first (green), second (yellow), or third (red) lowest across and about means.

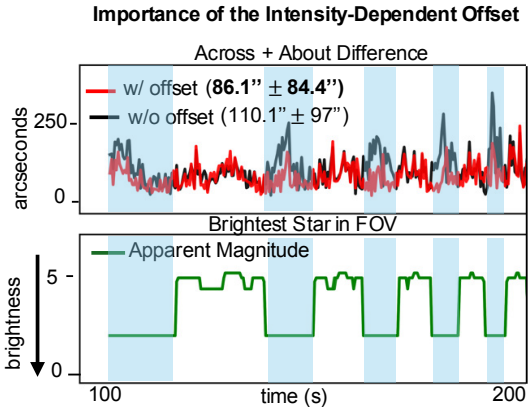


Figure 8. Snippet from Velocity Track 1. Without offset correction (black), error spikes when a bright star (magnitude 2.23) enters the field of view (shaded blue). Our offset correction accounts for the star's brightness, attenuating these spikes and lowering the average difference by over 20 arcseconds.

5. Discussion and Conclusion

Limitations: Obtaining precise ground truth for star tracking presents challenges, as highlighted in prior star tracker studies [17]. Some approaches achieve this by calibrating a motorized pan/tilt mount to reference stars and using the mount's position readout as ground truth [7], while others rely on purely simulated data for benchmarking [8]. To assess performance in real-world conditions, we adopt the solution of benchmarking our system against a finely calibrated, high-accuracy commercial star tracker (the Rocket Lab APS tracker). Although our approach is bounded by the reference star tracker's accuracy, it enables direct, meaningful comparisons between our system and a standard tracker on real data.

While our earth-based data contains some artifacts that wouldn't be present in space—such as atmospheric diffraction variations near the horizon and shifts in background illumination — their impact on our experimental results is expected to be small compared to the order of magnitude larger error observed in our benchmarking methods. Additionally, we verify our centroiding approach separately using synthetic (LCD monitor) stars with idealized ground truth to ensure this component of our algorithm remains unaffected by APS tracker inaccuracy or other external factors.

Conclusion: This work presents an event-based star tracking method that both quantitatively improves upon the accuracy of previous EBS approaches and extends the operating envelope relative to the motion tolerance and update rate of APS-based star trackers. The accuracy improvements result from our novel centroiding approach based on EBS pixel modeling in low light and our EBS-EKF algorithm's ability to estimate attitude in three dimensions. Both the sparsity of the event data and the sensor's asynchronous output allow us to achieve significantly higher update rates than our APS reference tracker, all while achieving comparable accuracy.

While previous studies have highlighted the potential advantages of event cameras over APS trackers in proof-of-concept scenarios, we demonstrate these benefits using real data. We capture the first dataset of real events of real star fields with a commercially-available event sensor, and provide synchronized attitude estimates from a conventional star tracker as approximate ground truth. This data is released as a public benchmark to allow other researchers to produce comparable results in this growing area of research.

References

[1] rtings.com asus rog swift pro pg248qp review. https://www.rtings.com/monitor/reviews/asus/rog-swift-pro-pg248qp. Accessed: 2024-11-19. 4

[2] Vyas Akondi, M. Roopashree, and Budihal Prasad. Improved Iteratively Weighted Centroiding for accurate spot detection in Laser Guide Star based Shack Hartmann Sensor. *Proceedings of SPIE - The International Society for Optical Engineering*, 7588, 2010. Journal Abbreviation: Proc SPIE Publication Title: Proc SPIE. 2

[3] Samya Bagchi and Tat-Jun Chin. Event-based star tracking via multiresolution progressive Hough transforms. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2143–2152, 2020. 2, 3, 6, 7, 8

[4] Allen L Barker, Donald E Brown, and Worthy N Martin. Bayesian estimation and the kalman filter. *Computers & Mathematics with Applications*, 30(10):55–77, 1995. 5, 1

[5] Conor J Benson and Marcus J Holzinger. Simulation and Analysis of Event Camera Data for Non-Resolved Objects. *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2022. 5

[6] Michael Bloesch, Hannes Sommer, Tristan Laidlow, Michael Burri, Gabriel Nuetzi, Péter Fankhauser, Dario Bellicoso, Christian Gehring, Stefan Leutenegger, Marco Hutter, and Roland Siegwart. A Primer on the Differential Calculus of 3D Orientations, 2016. arXiv:1606.05285 [cs]. 2

[7] T. Brady, S. Buckley, and C. Tillier. Ground validation of the inertial stellar compass. In *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, pages 214–226, Big Sky, MT, USA, 2004. IEEE. 8

[8] 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*, 2019. 2, 3, 6, 7, 8

[9] Tjorven Delabie, Joris De Schutter, and Bart Vandenbussche. An accurate and efficient Gaussian fit centroiding algorithm for star trackers. *Journal of the Astronautical Sciences*, 61 (1):60–84, 2014. 2, 4

[10] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2): 1–139, 2017. 5

[11] Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. 5, 3

[12] James Richard Forbes. Fundamentals of spacecraft attitude determination and control [bookshelf]. *IEEE Control Systems Magazine*, 35(4):56–58, 2015. 3, 1

[13] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Joerg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. arXiv: 1904.08405. 2

[14] Daniel Gehrig and Davide Scaramuzza. Low-latency automotive vision with event cameras. *Nature*, 629 (August 2023), 2024. Publisher: Springer US ISBN: 4158602407409. 2

[15] Rui Graça, Brian McReynolds, and Tobi Delbruck. Shining light on the DVS pixel: A tutorial and discussion about biasing and optimization. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2023-June:4045–4053, 2023. arXiv: 2304.04706 ISBN: 9798350302493. 4

[16] Mitchell Grose, Jason Schmidt, and Keigo Hirakawa. Convolutional neural network for improved event-based Shack-Hartmann wavefront reconstruction. *Applied Optics*, 63(16), 2024. 2

[17] Karl Gudmundson. *Ground Based Attitude Determination Using a SWIR Star Tracker*. PhD thesis, Linkoping University, Linkoping, Sweden, 2019. 8

[18] Christoph Hertzberg, René Wagner, Udo Frese, and Lutz Schröder. Integrating Generic Sensor Fusion Algorithms with Sound State Representations through Encapsulation of Manifolds, 2011. arXiv:1107.1119 [cs]. 1, 6

[19] Yuhuang Hu, Shih Chii Liu, and Tobi Delbruck. V2e: From video frames to realistic dvs events. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, page 1312–1321, 2021. arXiv: 2006.07722. 4

[20] Louis Jannin and Leonard Felicetti. Calibration and testing strategies to correct atmospheric effects on star tracking algorithms. *Astronomy and Computing*, 46:100794, 2024. 7

[21] Kazutoshi Kodama, Yusuke Sato, Yuhi Yorikado, Raphael Berner, Kyoji Mizoguchi, Takahiro Miyazaki, Masahiro Tsukamoto, Yoshihisa Matoba, Hirotaka Shinozaki, Atsumi Niwa, Tetsuji Yamaguchi, Christian Brandli, Hayato Wakabayashi, and Yusuke Oike. 1.22um 35.6Mpixel RGB Hybrid Event-Based Vision Sensor with 4.88um-Pitch Event Pixels and up to 10K Event Frame Rate by Adaptive Control on Event Sparsity. In *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 92–94, 2023. ISSN: 2376-8606. 2

[22] Dustin Lang, David W. Hogg, Keir Mierle, Michael Blanton, and Sam Roweis. Astrometry. net: Blind astrometric calibration of arbitrary astronomical images. *The astronomical journal*, 139(5):1782, 2010. Publisher: IOP Publishing. 2, 3, 5, 7

[23] Yasir Latif, Peter Anastasiou, Yonhon Ng, Zebb Prime, Tien-Fu Lu, Matthew Tetlow, Robert Mahony, and Tat-Jun Chin. High Frequency, High Accuracy Pointing onboard Nanosats using Neuromorphic Event Sensing and Piezoelectric Actuation. 2023. arXiv: 2309.01361. 2, 3, 6, 8

[24] Scikit learn developers. sklearn.cluster.dbscan, 2024. Accessed: 2024-11-18. 3

[25] Jian Li, Xinguo Wei, and Guangjun Zhang. An Extended Kalman Filter-Based Attitude Tracking Algorithm for Star Sensors. *Sensors*, 17(8):1921, 2017. 3

[26] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 x 128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 2, 3, 4

[27] C.C. Liebe. Accuracy performance of star trackers - a tutorial. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):587–599, 2002. 2, 6, 7

[28] Haoyue Liu, Shihan Peng, Lin Zhu, Yi Chang, Hanyu Zhou, and Luxin Yan. Seeing Motion at Nighttime with an Event Camera. *Computer Vision and Pattern Recognition Conference (CVPR)*, 1:25648–25658, 2024. arXiv: 2404.11884. 4

[29] Yonhon Ng, Yasir Latif, Tat-Jun Chin, and Robert Mahony. Asynchronous Kalman Filter for Event-Based Star Tracking. *European Conference on Computer Vision Workshop*, pages 66–79, 2022. 1, 2, 3, 4, 6, 7, 8

[30] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1): 259–275, 2011. 4

[31] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-Time Intensity Estimation Using Event Cameras. *Asian Conference on Computer Vision*, pages 308–324, 2018. arXiv: 1811.00386 ISBN: 9783030208721. 2

[32] Nathan Stacey and Simone D'Amico. Analytical process noise covariance modeling for absolute and relative orbits. *Acta Astronautica*, 194:34–47, 2022. 2

[33] Ronald C. Stone. A comparison of digital centering algorithms. *The Astronomical Journal*, 97:1227, 1989. 2

[34] Ting Sun, Fei Xing, Xiaochu Wang, Jin Li, Minsong Wei, and Zheng You. Effective star tracking method based on optical flow analysis for star trackers. *Appl. Opt.*, 55(36): 10335–10340, 2016. 3

[35] Eric A. Wan and Rudolph Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. IEEE, 2000. 5

[36] Xiaowei Wan, Gangyi Wang, Xinguo Wei, Jian Li, and Guangjun Zhang. Star centroiding based on fast gaussian fitting for star sensors. *Sensors (Switzerland)*, 18(9), 2018. 2

[37] Ziyun Wang, Fernando Cladera Ojeda, Anthony Bisulco, Daewon Lee, Camillo J. Taylor, Kostas Daniilidis, M. Ani Hsieh, Daniel D. Lee, and Volkan Isler. EV-Catcher: High-Speed Object Catching Using Low-Latency Event-Based Neural Networks. *IEEE Robotics and Automation Letters*, 7(4):8737–8744, 2022. arXiv: 2304.07200. 2

[38] Bohan Yu, Jieji Ren, Jin Han, Feishi Wang, Jinxiu Liang, and Boxin Shi. EventPS: Real-Time Photometric Stereo Using an Event Camera. *Computer Vision and Pattern Recognition Conference (CVPR)*, pages 9602–9611, 2024. 2