ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Department of Industrial Engineering

Bachelor Degree in
AEROSPACE ENGINEERING

DETECTION AND TRACKING OF RESIDENT SPACE OBJECTS WITH EVENT-BASED CAMERAS

Dissertation in ING-IND/05: Satelliti e Missioni Spaziali

Supervisor
Prof. Dario Modenini

Presented by
Matteo Ponti

Co-Supervisor
Stefano Palmiotto

Graduation session: March 24th 2026

Academic Year 2024/2025

Abstract

This work presents the development and validation of a visual tracking system based on neuromorphic, or event-based, sensors for Space Situational Awareness (SSA) applications. Unlike conventional sensors, event-based cameras generate asynchronous events in response to local changes in brightness, enabling a highly efficient spatiotemporal representation with high temporal resolution and low power consumption. The study explores algorithms from the literature for processing such data, including exponentially decaying time surfaces, a spatial activation filter to reduce noise, an angular template-based feature detector, and an asynchronous event tracker implementing a Leaky Integrate-and-Fire (LIF) neuron-inspired membrane potential mechanism. The system was evaluated on real space observation datasets and in laboratory experiments using a controlled setup with a point-like target moving against a uniform background. Sensor bias values tuning allowed for background noise reduction and improved signal-to-noise ratio, enhancing tracking performance. Results demonstrate that the combination of event-based sensors and neuromorphic algorithms produces a clean, real-time event stream while significantly reducing computational complexity.

Sommario

Il presente lavoro sviluppa e valida un sistema di tracciamento visivo basato su sensori neuromorfici, o event-based, per applicazioni di Space Situational Awareness (SSA). A differenza dei sensori tradizionali, le camere event-based generano eventi asincroni in risposta a variazioni locali di luminosità, permettendo una rappresentazione spaziotemporale altamente efficiente, con elevata risoluzione temporale e basso consumo energetico. La ricerca esplora algoritmi provenienti dalla letteratura per l'elaborazione di questi dati, comprendenti: la costruzione di superfici temporali a decadimento esponenziale, un filtro di attivazione spaziale per ridurre il rumore, un rilevatore di feature basato su template angolari e un tracciatore asincrono di eventi con meccanismo di potenziale membranale ispirato ai neuroni Leaky Integrate-and-Fire (LIF). Il sistema è stato testato su dataset reali di osservazioni spaziali e in laboratorio mediante un setup controllato in cui un target puntiforme si muove su uno sfondo uniforme. La regolazione dei parametri dei sensori ha consentito di ridurre il rumore di fondo e migliorare il rapporto segnale-rumore, aumentando l'efficacia del tracciamento. I risultati mostrano che la combinazione di sensori event-based e algoritmi neuromorfici permette di ottenere uno stream di eventi pulito e gestibile in tempo reale, riducendo significativamente la complessità computazionale.

Nota del Relatore

In qualità di relatore autorizzo la redazione della tesi di laurea in lingua straniera e mi faccio garante della qualità - anche linguistica - dell'elaborato.

Prof. Dario Modenini

Acknowledgements

Acknowledgements, che in italiano significa ringraziamenti. E sì, non preoccupatevi: almeno questi saranno interamente in italiano.
Innanzitutto, desidero ringraziare Stefano e Dario per il tempo, la disponibilità e la pazienza che mi hanno dedicato nella fase conclusiva di questo percorso. Grazie per avermi dato la possibilità di portare a termine questo lavoro, per i consigli, le osservazioni puntuali e gli stimoli continui a fare meglio. Questo progetto non è stato utile solo a raggiungere il traguardo accademico in sè, ma un'occasione di crescita personale, che mi ha permesso di approfondire argomenti che mi appassionano e di mettermi alla prova.
Un grazie immenso va poi a babbo e mamma. Mi avete sostenuto in ogni momento, in ogni fase di questo percorso: quando gli esami andavano bene al primo colpo e quando invece richiedevano più tempo, quando io ero felice e di buon umore per raccontarvi tutto e anche quando ero girato male. Grazie per aver sempre creduto in me, senza di voi, senza i vostri sacrifici e il vostro amore incondizionato, nulla di tutto questo sarebbe stato possibile.
Grazie anche a Bic e a Giulia, per aver fatto il tifo per me in ogni occasione, per la leggerezza, le risate, i dispetti e le bicconate che hanno reso tutto ancor più divertente.
Un ringraziamento speciale anche ai nonni Dedo, Lalla, Miria e Giovanni che mi dà forza da lassù, per il vostro affetto costante e silenzioso, per l'orgoglio con cui avete sempre seguito ogni mio passo e per avermi trasmesso valori che porto con me ogni giorno. E anche per tutte le tagliatelle, tortelli, castagnole e cappelletti che con amore preparate sempre per noi. Ogni volta che non mi ricordavo l'esito di un qualsiasi esame, bastava andare a chiedere al nonno Dedo, tanto se li è segnati tutti.
Grazie anche agli zii Angelo, Katia, Barby, Luca, Gabry e ai cugini Giada, Alice, Sara, Surgo, Giada e Paolo per i giochi, gli scherzi, il sostegno e l'incoraggiamento.
Grazie ai compagni di corso Saragini, Bocco, Mazz, Tibi, Luis, Giulio, Pevoni, Dani, Albi e tutti gli altri: avete reso questi anni più leggeri e indimenticabili. Tra lezioni, esami, pause caffè, momenti di sconforto e di esultanza, la vostra spensieratezza e la vostra allegria hanno fatto la differenza. Condividere questo percorso con voi lo ha reso molto più bello.
Grazie a Marco, Lombo ed Edo per avermi "forgiato" sin dai tempi del liceo: tra piscine di martedì, studio e risate, avete contribuito a costruire una parte importante di ciò che sono oggi.
Grazie ai trapana per tutte le mangiate e avventure vissute insieme fin da piccoli, ai pochi ma buoni per i gelati al Crem Caramel, ai gruppi di Lubiana e Sarajevo per due dei viaggi più belli ed epici di sempre, al gruppo giovanissimi di villanova

per tutte le attività che facciamo insieme, all'Equipe per tutte le cose belle che realizziamo fuori e dentro le persone, al Villanova FC per tutte le vittorie e trasferte a calcetto. Grazie a tutte le amicizie che mi hanno accompagnato lungo tutto il percorso, le passioni e i momenti condivisi: mi avete ricordato sempre che, oltre agli obiettivi e ai traguardi, ciò che conta davvero sono le persone con cui li condividi. A tutti voi, grazie di cuore. Questo traguardo è anche vostro.

Contents

List of Figures

List of Tables

Acronyms

ADC	Analog-to-Digital Converter
AER	Address-Event Representation
ATIS	Asynchronous Time-based Image Sensor
CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide Semiconductor
DAVIS	Dynamic and Active-pixel Vision Sensor
EBC	Event-Based Camera
EBSI	Event-Based Space Imaging
EM	Exposure Measurement
EVS	Event-Based Vision Sensor
FAN	Fast Adapting Network
FAST	Features from Accelerated Segment Test
FEAST	Feature Extraction using Adaptive Selection Thresholds
FIESTA	Fast Iterative Extraction of Salient targets for Tracking Asynchronously
FOV	Field of View
GM-PHD	Gaussian Mixture Probability Hypothesis Density
GNN	Global Nearest Neighbor
GEO	Geosynchronous Earth Orbit
HASTE	multi-Hypothesis Asynchronous Speeded-up Tracking of Events
JPDA	Joint Probabilistic Data Association
LEO	Low Earth Orbit
LIF	Leaky Integrate-and-Fire

LUT	Look Up Table
MEO	Medium Earth Orbit
MHT	Multiple Hypothesis Tracker
MTT	Multiple Target Tracking
NMOS	N-type Metal-Oxide-Semiconductor
PDA	Probabilistic Data Association
PMHT	Probabilistic Multiple Hypothesis Tracker
PMOS	P-type Metal-Oxide-Semiconductor
ROI	Region Of Interest
RSO	Resident Space Object
SAN	Slow Adapting Network
SNR	Signal-to-Noise Ratio
SD	Space Debris
SDA	Spatial Detection Accuracy
SL-8 R/B	SL-8 Rocket Body
SSA	Space Situational Awareness
STT	Single Target Tracking
TD	Threshold Detector

Chapter 1

Introduction

1.1 Space Situational Awareness

Earth's orbital environment has become increasingly crowded, populated by thousands of commercial, scientific, and defense satellites. About 40 000 objects are now tracked by space surveillance networks, of which about 11 000 are active payloads. However, the actual number of space debris (SD) objects larger than 1 cm in size (large enough to be capable of causing catastrophic damage) is estimated to be over 1.2 million, with over 50 000 objects of those larger than 10 cm (European Space Agency, 2025).

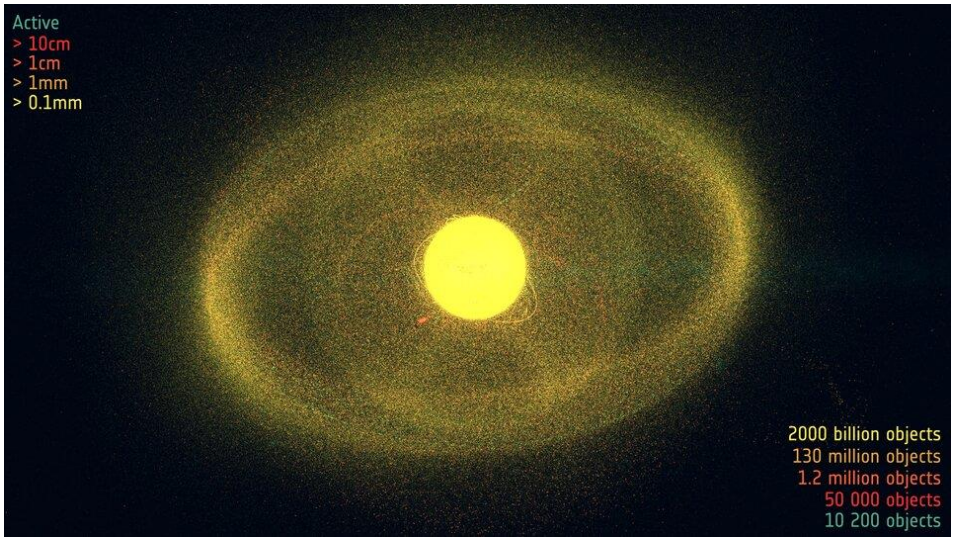


Figure 1.1: Color-coded representation of debris modeled to be circling around Earth in August 2024 with the number of objects of various sizes as well as active satellites (European Space Agency, 2025).

The growing density of resident space objects (RSOs) raises the probability of collisions (Kessler and Cour-Palais, 1978) and emphasizes the urgent need for accurate Space Situational Awareness (SSA). SSA is the set of capabilities, technologies, and activities dedicated to monitoring and understanding what is happening in

the space environment surrounding Earth. Its primary objective is to ensure the safety of space activities, protecting satellites, vehicles, and infrastructure both in orbit and on the ground. This is accomplished by surveying and tracking space debris, monitoring natural space conditions such as solar activity, solar wind, and geomagnetic storms, and observing natural near-Earth objects that could potentially pose a threat to the planet (European Union Agency for the Space Programme, 2025).

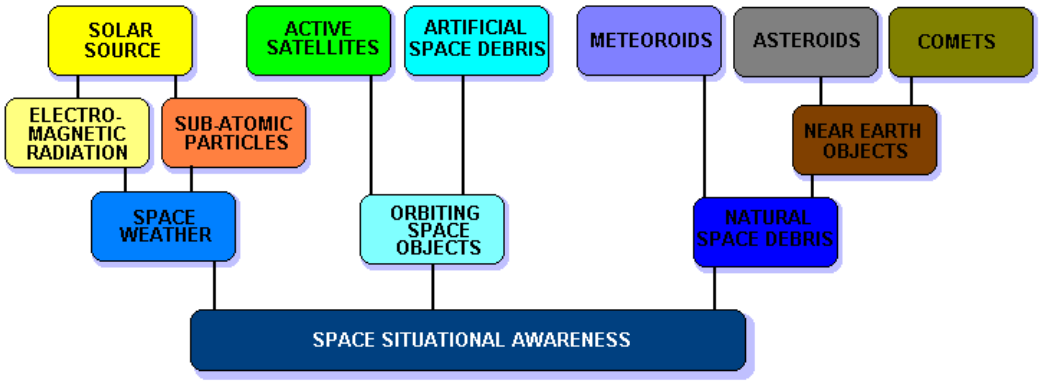


Figure 1.2: The three current components of SSA (Space Academy, 2025).

Some works (The Aerospace Corporation, 2025) focus on the active management of new SD generated by inactive satellites, either by relocating them to graveyard orbits or deorbiting them for complete atmospheric disintegration, while others (Battista et al., 2017) propose the removal of existing SD through dedicated capture mechanisms. In contrast, a separate research direction adopts a passive approach, accepting the presence of space junk and developing collision-avoidance techniques to mitigate its impact to Earth's orbital environment. To this end, SSA relies on a diverse network of sensors, including radars, optical systems, and lasers in order to detect, track, and characterize space objects. Optical sensors are widely utilized for observing objects in Medium Earth Orbit (MEO) and Geosynchronous Earth Orbit (GEO) due to their cost-effectiveness and lower power consumption compared to radars (Argirò et al., 2025) for these specific distances, although their operation is limited by weather conditions and illumination (primarily observing reflected sunlight at night). Traditionally, these systems have employed frame-based imaging sensors, such as charge-coupled devices (CCDs) and complementary metal-oxide semiconductor (CMOS) active pixel sensors. However, recent developments in event-based cameras (EBCs), also known as neuromorphic vision sensors, are reshaping how we capture and process visual information in space.

1.2 Frame-Based versus Event-Based Cameras

In traditional frame-based cameras, the entire pixel array accumulates light intensity during a fixed, predefined exposure time (integration time), after which the signal is digitized in order to produce a single frame. In video recording context, this process is repeated at regular intervals, regardless of whether there are significant changes in

the observed scene. This approach, while intuitive, can be inefficient for monitoring specific space objects, as all pixels acquire information, generating a potential excess of data (spatial redundancy) compared to what is strictly necessary for tracking a point-like target.

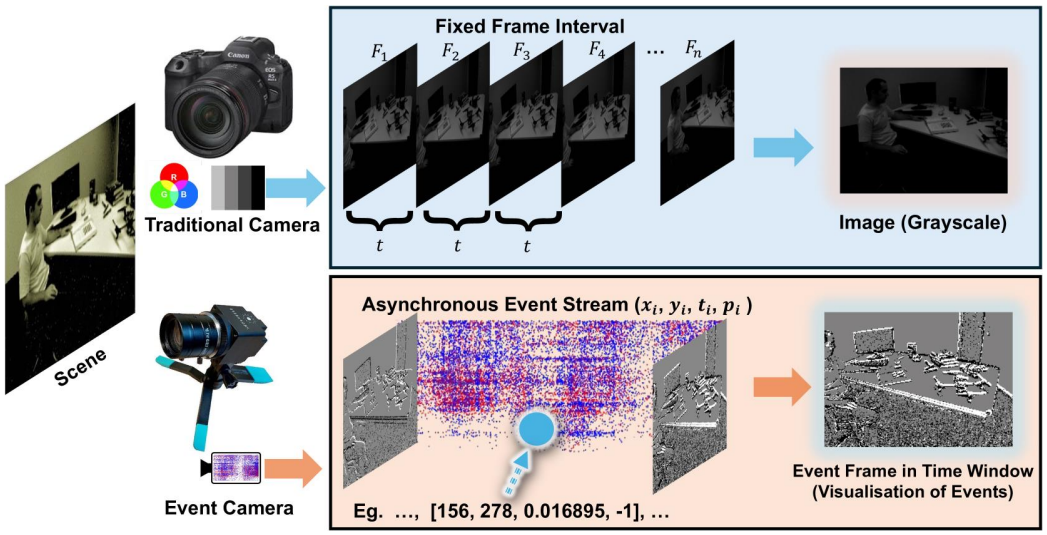


Figure 1.3: Conceptual comparison between conventional frame-based imaging and event-based vision sensors. Event-based cameras asynchronously report brightness changes, enabling sparse and high-temporal-resolution sensing (Xu et al., 2025).

In contrast, event-based sensors produce an event stream rather than conventional images (Cohen et al., 2019). They operate in an event-based paradigm, drawing inspiration from both the functioning of the biological retina and the computation and processing found in biological vision systems (Ralph et al., 2019). Pixels in event-based cameras operate in an independent and asynchronous manner, producing data only in response to changes in the log intensity of the illumination at that pixel. The independent nature of each pixel removes the need for a global exposure time or shutter, allowing these devices to operate in a continuous fashion and diminishing the saturation effects suffered by conventional frame-based devices (Zołnowski et al., 2019). The asynchronous nature of the pixels produces a frame-free output with a high temporal resolution, serving to remove the effects of motion blur and providing the device with a very high dynamic range (Chin et al., 2019).

Furthermore, event-based cameras offer high-speed and low-power operation together with a sparse spatiotemporal representation of visual information (McReynolds et al., 2023). These characteristics lead to intrinsically low data rates and reduced bandwidth requirements, which are critical for spaceborne sensing platforms (Oliver et al., 2025). Because the data output depends on scene dynamics rather than fixed sampling intervals, event-based cameras can achieve effective frame rates of thousands of frames per second while maintaining minimal power and communication overhead (Bagchi and Chin, 2020). The efficiency and temporal precision of this sensing paradigm have enabled significant advances in real-time detection and tracking for space applications, including star tracking (Chin et al., 2019),

multi-object tracking (Cheung et al., 2018), and collision avoidance systems based on neuromorphic vision (Coretti et al., 2025).
Event-based sensors are also emerging as a promising technology for several astronomical and astrometric applications. Their microsecond-level temporal resolution and high dynamic range enable the generation of star maps directly from event streams, supporting high-precision astrometry even under challenging illumination conditions (Cohen et al., 2018), as visualizable in Figure 1.4.

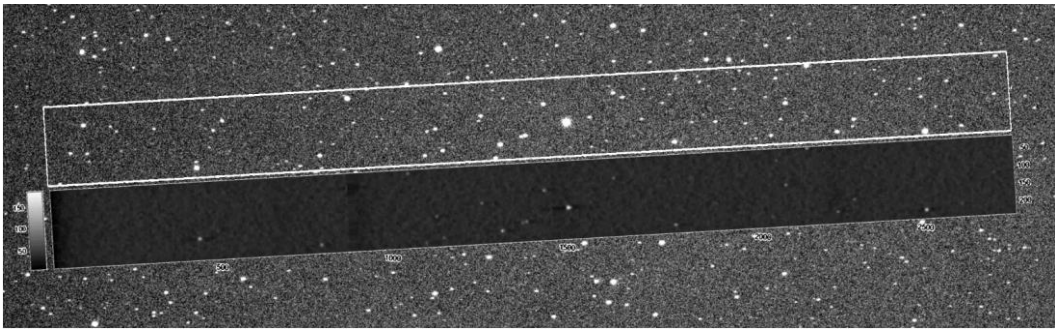


Figure 1.4: A star map generated by the event-based camera and overlaid over the ground truth CCD image. The star map generated by the event-based camera is shown as the inset in the above image. Above the inset, the correct location on the ground-truth image is shown within the white rectangle (Cohen et al., 2018).

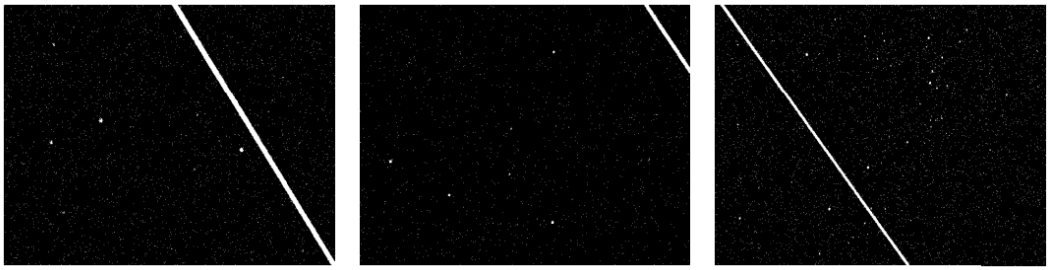


Figure 1.5: Simultaneous capture of a satellite made with three telescopes, each of them mounting an EBC. The full trajectories are shown (Aliste et al., 2023).

Additionally, their ability to detect faint, transient, or fast-moving light sources makes them suitable for astronomical observations involving dim stars, planets, and small bodies (Aliste et al., 2023), as shown in figures 1.5 and 1.6. Event-based sensors are also well suited for all-sky imaging and continuous sky monitoring, where their low data rate, low power consumption and robustness to variation of light allow efficient detection and tracking of events across wide fields of view (FOV) (Ralph et al., 2019). Collectively, these results demonstrate that event-based imaging provides a compelling technological foundation for next-generation space domain awareness systems (Afshar et al., 2020).

Figure 1.6: Simultaneous capture of a satellite made with three telescopes, showing the cropped trajectories of the object (Aliste et al., 2023).

1.3 Thesis objective and outline

The aim of this thesis is to investigate the applicability of event-based vision sensors to the detection and tracking of RSOs for optical SSA applications. In particular, this work aims to (i) analyze the operating principles and constraints of event-based sensors, (ii) design and implement an event-based detection and tracking algorithm tailored to space object observation, and (iii) experimentally validate its performance through laboratory data acquisition and processing, with special attention to the influence of sensor bias settings on tracking performance.
This thesis is structured as follows. Chapter 2 presents the fundamentals of event-based vision, including pixel architecture, bias currents, and a review of detection and tracking techniques for event streams, with a discussion of current challenges in optical SSA. Chapter 3 describes the proposed event-based detection and tracking algorithm, detailing the dataset used for validation, the pre-processing pipeline, the feature detection strategy, and the tracking framework. Chapter 4 focuses on laboratory data acquisition and experimental validation, illustrating the setup configuration, the optical stimulus design, and the results obtained under default and tuned bias settings. Finally, the main findings of this work are summarized and discussed in the conclusions, together with possible future research directions.

Chapter 2

Event-Based Vision

2.1 Event-Based Sensor Pixel Architecture

EVS represent a paradigm shift in imaging, where pixels respond asynchronously to changes in light intensity rather than capturing full frames. Figure 2.1 shows a simplified EVS pixel architecture. Each pixel contains a photodiode that generates a photocurrent I_{ph} in response to incident light, which is converted to an output voltage V_p proportional to the natural logarithm of the photocurrent ($V_p \propto \ln I_{ph}$). This signal is buffered by a source-follower and amplified by a factor of $-C_1/C_2$ (capacities of the two capacitors in the change amplifier stage, as shown in figure 2.1), resulting in a voltage signal V_{diff}.

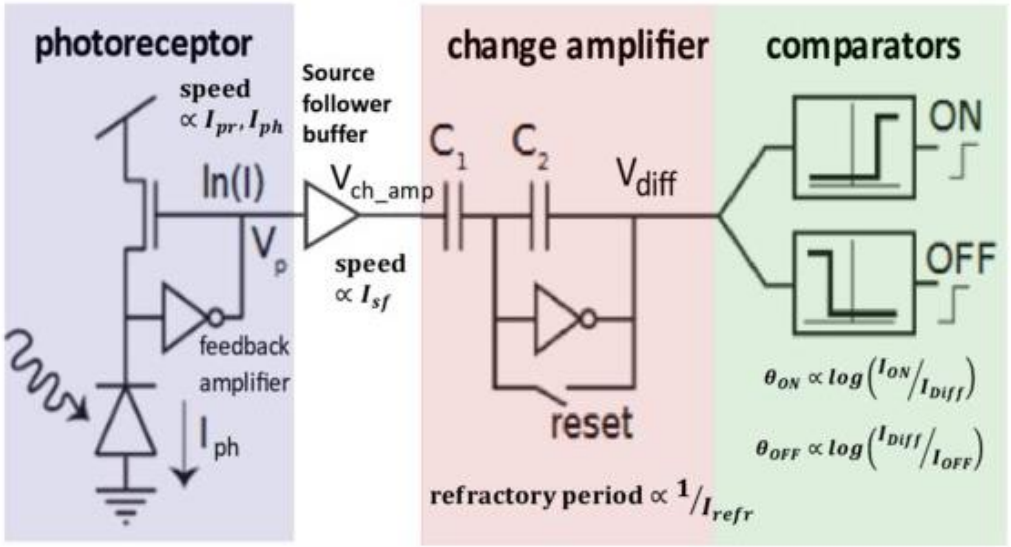


Figure 2.1: Simplified EVS pixel architecture (McReynolds et al., 2023).

In the comparator stage, V_{diff} is continuously monitored and compared to a reference level stored in analog memory, corresponding to the voltage level of the last reported event. An ON event (positive polarity) is generated when the brightness in-

crease exceeds the θ_{ON} threshold, and an OFF event (negative polarity) is generated when the decrease exceeds the θ_{OFF} threshold. Both thresholds are tunable. The output is encoded as asynchronous spikes according to the Address-Event Representation (AER) standard (Cohen et al., 2019). AER is an asynchronous, event-driven communication and coding protocol originally designed to exchange information between neuromorphic chips. As shown in Figure 2.2, a typical AER-based sender implements two functions: encoding and arbitration. The encoder waits for an event and encodes the event identity based on its location. This encoded "address-event" is then sent across the output bus as they occur, preserving the timing information. When two or more events arrive simultaneously, an arbitration mechanism is used to avoid collision and determine which event is communicated first on the output bus. A queuing mechanism is introduced to allow other events to wait for their turn. This queuing can be at the source (per pixel), shared (per AER encoder), or a combination of the two (Liu et al., 2014).

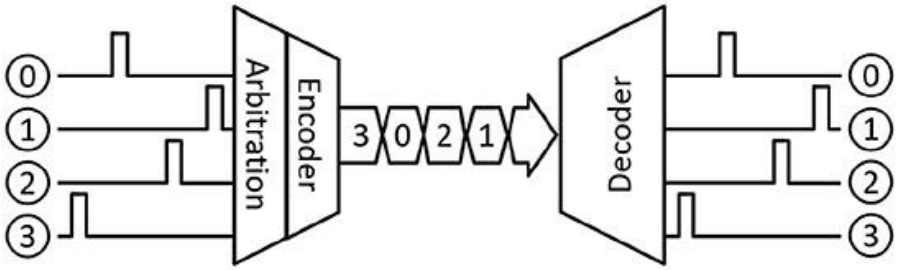


Figure 2.2: AER communication protocol (Purohit and Manohar, 2022).

After reporting an event, the pixel ignores signal changes for a finite period known as the refractory period. After this period, its memorized reference level is reset, and the pixel again monitors the input signal to report subsequent changes relative to this new reference.

2.2 Bias currents

The performance of event-based sensors depends on illumination levels and the configuration of so-called bias currents, which control pixel circuitry behavior. McReynolds et al. (2023) provide a theoretical background. Thresholds $\theta_{ON/OFF}$ are directly controlled by three biases: I_{diff}, I_{ON}, and I_{OFF}:

$$\theta_{ON} = \frac{k_n}{k_p^2}\frac{C_2}{C_1}\ln\frac{I_{ON}}{I_{diff}} \tag{2.1}$$

$$\theta_{OFF} = \frac{k_n}{k_p^2}\frac{C_2}{C_1}\ln\frac{I_{diff}}{I_{OFF}} \tag{2.2}$$

In Equations (2.1) and (2.2), k_n and k_p are the sub-threshold slope coefficients of the N-type Metal-Oxide-Semiconductor (NMOS) and P-type Metal-Oxide-Semiconductor (PMOS) transistors, respectively, determining the effective conversion

gain between the bias currents and the sensor activation thresholds.
To first order, when thresholds are low, the dimensionless term to the left of the logarithm corresponds to the fractional signal change required to generate ON and OFF events. In practice, however, transistor mismatch and noise limit the lowest usable threshold levels, typically between 0.11 and 0.15 fractional change under good lighting conditions, but potentially much higher in the dark, especially when low noise rates are required.
Photoreceptor response speed, often referred to as pixel bandwidth, depends on the induced photocurrent I_{ph} at the photodiode as well as the I_{pr} and I_{sf} bias currents. Under low illumination, bandwidth is primarily determined by photocurrent and increases monotonically with light intensity. At higher illumination levels, either I_{pr} or I_{sf} will determine bandwidth, depending on how these biases are balanced.
The refractory period is uniquely defined by I_{refr}, which controls the rate at which the reset switch node voltage charges after each event. A higher I_{refr} results in a shorter refractory period, while a longer period can reduce recorded information but improve noise performance in low-light scenarios. The effect of bias currents on the EVS performance is summarized in Table 2.1.

Table 2.1: Effects of bias adjustments. The intended influence of tuning each controllable EVS performance parameter is listed above, as well as the unavoidable impact on noise rates. Optimization for spatial detection accuracy (SDA) is a delicate balance of biasing for sensitivity, speed and noise management (McReynolds et al., 2023).

Parameter	**Biases**	**Adjustment**	**Main Effect**	**Noise Rate Effect**
Threshold(s)	$I_{diff}, I_{ON}, I_{OFF}$	Increase	Reduced sensitivity	Decrease
		Decrease	Improved sensitivity	Increase
Pixel Bandwidth	I_{pr}, I_{sf}	Increase	Faster photoreceptor response	Increase
		Decrease	Slower photoreceptor response	Decrease
Refractory Period	I_{refr}	Increase	Shorter interval between consecutive events	Increase
		Decrease	Longer interval between consecutive events	Decrease

2.2.1 ATIS and DAVIS cameras

Several types of event-based cameras have been designed, including temporal contrast sensors, gradient-based sensors, edge-orientation sensors, and optical-flow sensors (Roffe et al., 2021). For example, Cohen et al. (2019) describe two types of EBCs: Asynchronous Time-based Image Sensor (ATIS) by the French company Chronocam and Dynamic and Active-pixel Vision Sensor (DAVIS) by iniLabs, both deriving from the same original prototype. In the ATIS camera, each pixel contains a level-crossing detector, or threshold detector (TD), circuit and an exposure measurement (EM) circuit for absolute illumination measurements. In the TD circuit the events are generated with a polarity to indicate an increase or decrease in the relative illumination. As regards the EM circuitry, the absolute illumination measurement is carried out in two different ways. In global snapshot mode, all the array's pixels are triggered at the same time (for each pixel an event is triggered) and each one starts to integrate the incoming light. When the absolute intensity in a pixel crosses a threshold, a second event in that pixel is triggered. The time duration between the two consecutive events represents the absolute illumination level for that pixel. The brightest pixels respond first. The final product is a complete frame where each pixel records a time-encoded measurement of absolute illumination. This mechanism allows absolute brightness to be encoded without an analog-to-digital converter (ADC), saving space, power, and time. Indeed, it allows the sensor to transmit precise illumination information with just two events, making the output very compact. The global snapshot mode is useful to obtain images in static conditions, such as stars or Geosynchronous Earth Orbit (GEO) satellites, or when calibrating.
The alternative mode is the event-driven snapshot: when the TD circuit of a pixel records an event, it triggers an event in the associated EM circuit which starts integrating the incoming light and measures its absolute luminosity as previously discussed. Thus, each pixel measures its absolute brightness independently of the other pixels only when an event (luminosity change) occurs, reducing the number of active pixels. The event-driven snapshot is thus useful to efficiently capture dynamic scenes, reducing the data to transmit.
The DAVIS imager also offers dual outputs in the form of change detection and absolute pixel illumination measurement. Conversely to the ATIS camera, the DAVIS camera contains a synchronous frame-based pixel sensor in addition to the change detection circuitry, allowing for the output of conventional frames and making this output compatible with the majority of computer vision algorithms and techniques.

2.3 Detection and Tracking Techniques in Event-Based Data

2.3.1 Event-Based Sensors Output

The output of event-based sensors can be described as a sparse, frame-free, asynchronous spatio-temporal stream of events, which are defined as:

$$\mathbf{e} = [x,\, y,\, p,\, t]^T \tag{2.3}$$

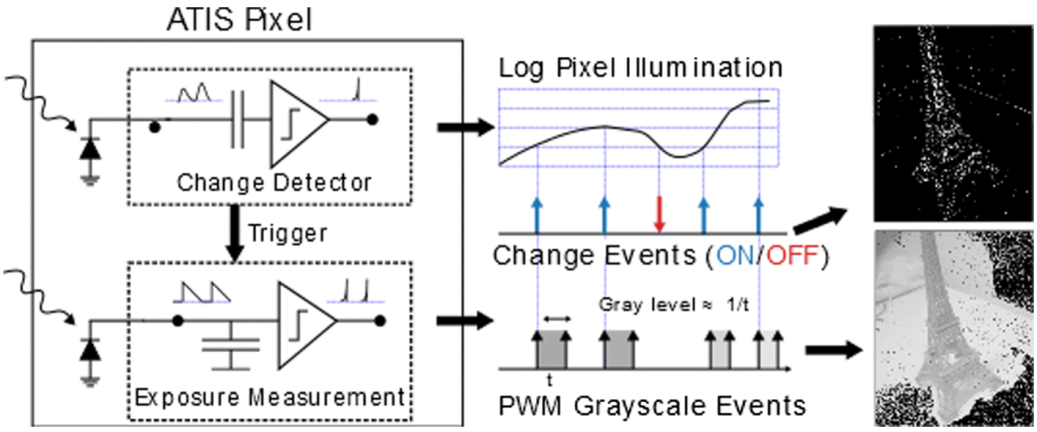


Figure 2.3: Schematic of the two components of each pixel in the ATIS sensor. These are the change detector circuitry and the exposure measurement circuitry (Cohen et al., 2019).

In Eq. (2.3), x and y are the pixel coordinates of the event, while p is the event polarity. An OFF event corresponds to $p = -1$, or $p = 0$ sometimes; an ON event corresponds to $p = +1$. Finally, t is the event timestamp.
As stated in Sec. 1.2, this type of output provides advantages such as low data rates, low power consumption, high temporal resolution, high dynamic range, and reduced motion blur, but requires different interpretation methods compared to conventional frame-based sensors.

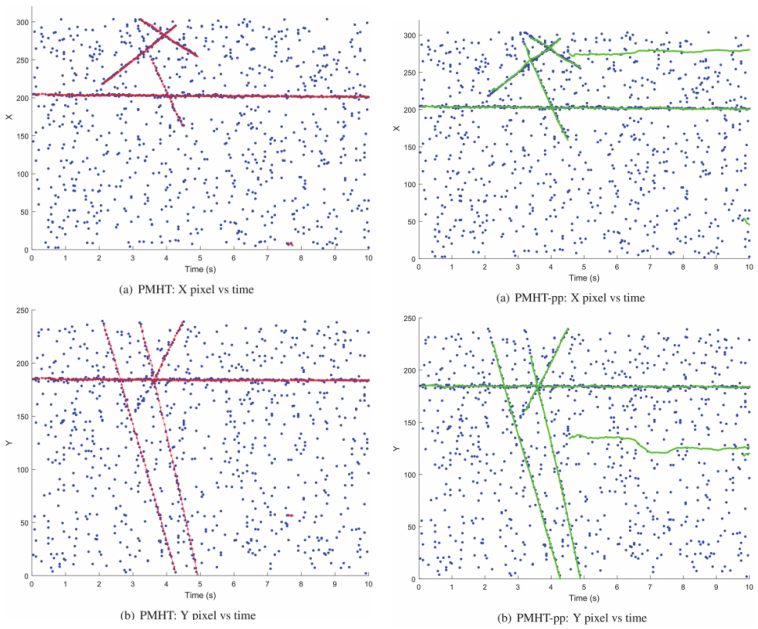
2.3.2 Event-Based Optical SSA and Tracking Challenges

Optical SSA using EBCs is a significantly different process to conventional optical SSA. Instead of exposing a sensor for a predefined period to accumulate light from a scene, an EBC keeps its pixel array continuously active, producing near microsecond asynchronous binary "events" whenever a pixel experiences a temporal change in brightness. Such variations in illumination can be caused by the apparent motion of an RSO or atmospheric scintillation. Because this output consists solely of contrast events, without frames or conventional intensity measurements, it cannot be processed using standard optical detection or tracking techniques.

2.3.3 Conventional and Statistically Robust Tracking Methods

The objective of tracking is to recursively estimate the state of a target over time, a task that is nontrivial even in conventional tracking scenarios. Numerous Single Target Tracking (STT) and Multiple Target Tracking (MTT) algorithms have been developed (Ralph et al., 2022). Among the simplest approaches is the Global Nearest Neighbor (GNN) tracker (Blackman and Popoli, 1999), which is computationally efficient and supports MTT, but fails to properly propagate state and measurement uncertainty. This limitation renders GNN trackers susceptible to noise and can lead to irreversible measurement-to-track association errors (Shon, 2019). To address these

shortcomings, statistically robust tracking frameworks such as Probabilistic Data Association (PDA) (Bar-Shalom et al., 2009), Joint Probabilistic Data Association (JPDA) (Bar-Shalom and Tse, 1975), Multiple Hypothesis Tracker (MHT) (Blackman, 2004) and Probabilistic Multiple Hypothesis Tracker (PMHT) (Streit and Luginbuhl, 1994) explicitly model multiple measurement hypotheses and uncertainties through probabilistic association and likelihood weighting.



(a) (b)

Figure 2.4: PMHT tracking results. In (a) the red lines denote tracks of four different objects, while the blue dots denote clustered measurements; in (b) the green lines denote tracks produced by the Poisson Priors variant of PMHT, while the blue dots denote again clustered measurements (Cheung et al., 2018).

2.3.4 Event-Based Tracking Requirements and Existing Approaches

Despite their robustness, conventional multi-hypothesis tracking algorithms are not directly suited to event-based sensing. When applied to event streams, they often incur significantly higher computational costs, as they are not designed to process the asynchronous, high-temporal-resolution, and event-driven outputs of EBCs. Event-based tracking algorithms must therefore exploit these characteristics directly, in particular the high temporal resolution and asynchronous operation. To this end, Alzugaray and Chli (2020) developed HASTE (multi-Hypothesis Asynchronous Speeded-up Tracking of Events), a high-speed and accurate tracking algorithm which has been shown to address many of the challenges above.

The trade-off between rigorous state estimation and individual event processing while operating at real-time and on an event-by-event basis using a novel asynchronous patch-feature tracker that uses pre-learned feature templates have been

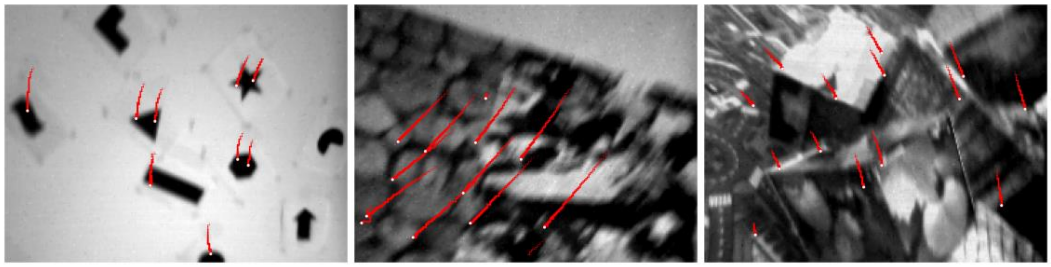


Figure 2.5: Example feature tracks with the Haste-Difference tracker over 40 ms (Alzugaray and Chli, 2020).

explored. Pre-learned features are problematic for systems such as SSA, where it is desirable to operate without priors on the ideal spatio-temporal features that represent trackable targets since target appearance is rarely known and can vary based on tumble rate, illumination angle, material composition, atmospheric seeing conditions, and orbital regime.
A significant part of the literature also focuses on corner detection, employing well-established Harris and FAST (Features from Accelerated Segment Test) key-point trackers (Alzugaray and Chli, 2018). Various optical flow estimation algorithms (Benosman et al., 2013) have also been proposed, which can be used to form the basis of a tracking algorithm. Machine learning approaches to tracking by detection have also been proposed, largely using GNN strategies or learned associations with supervised and unsupervised feature extraction for measurement detection (Lagorce et al., 2015; Afshar et al., 2019). These algorithms operate on various representations of events, such as integrated frames, volumes (Baldwin et al., 2022), graphs (Bi et al., 2020), and time-surfaces (Clady et al., 2015; Afshar et al., 2019).

2.3.5 Probabilistic State Estimation in Neuromorphic Tracking

Many neuromorphic tracking algorithms utilize the Kalman filter or Markov-Bayes recursion for track state estimation (Ralph et al., 2022). However, few use the previously described statistically robust trackers from the conventional tracking literature. Some examples of their use include the development of GM-PHD (Gaussian Mixture Probability Hypothesis Density) tracking (Foster et al., 2019) and MHT (Leclerc et al., 2018). These algorithms operate on event clusters accumulated over time and accumulated frames of event-based data respectively. Additionally, GM-PHD has been designed to operate on normalized event-maps (Chen et al., 2020). A statistically robust tracker in the neuromorphic literature, also termed probabilistic data association, has been proposed (Peng et al., 2024) as an expectation-maximization algorithm for optical flow, not to be confused with the established PDA tracker of the same name.

2.3.6 Real-Time Event-Based SSA Tracking with FIESTA

A notable real-time and robust approach to event-based SSA tracking is the Fast Iterative Extraction of Salient Targets for Tracking Asynchronously (FIESTA) algorithm.

FIESTA operates asynchronously on individual events and selectively processes only those likely to correspond to trackable targets, identified via a feature consolidation mechanism based on spiking neural networks. The system comprises filtering, feature extraction, and tracking modules, with feature consolidation implemented using two feedforward Feature Extraction using Adaptive Selection Thresholds (FEAST) networks: a Fast Adapting Network (FAN) that rapidly learns transient salient features and a Slow Adapting Network (SAN) that consolidates stable features. When a SAN neuron spikes and exceeds an activity threshold, the corresponding event is forwarded to a modified asynchronous PDA tracker as a measurement. Track neurons incrementally learn target appearance from spiking FAN neurons, enabling associations based on both spatio-temporal context and learned features. The tracker output is post-processed to suppress edge artifacts, interpolate track states over time, and remove duplicate detections, mitigating effects such as wake events and hot pixels.

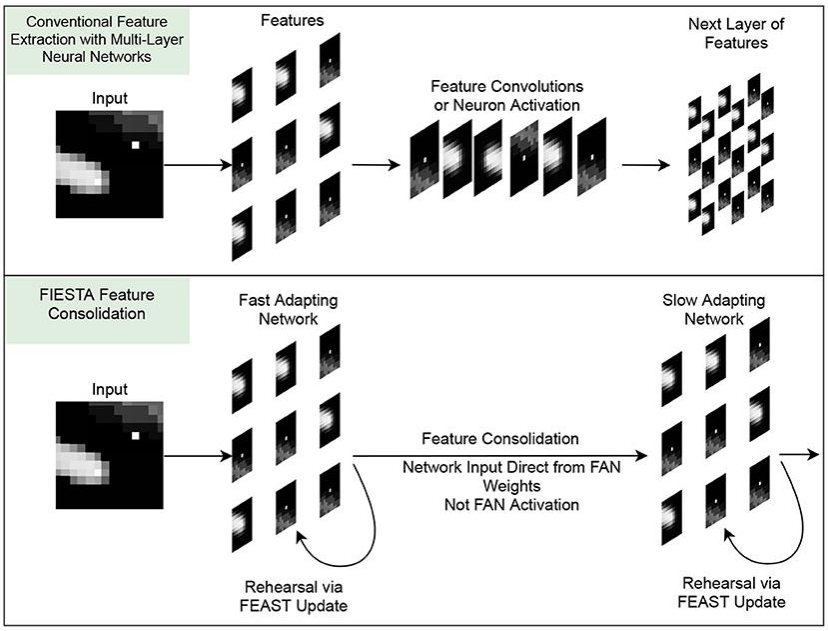


Figure 2.6: Comparison of conventional feature extraction and feature consolidation in FIESTA. In (Ralph et al., 2022) novel feature consolidator, feature weights learned by the FAN are consolidated into the long-term SAN for later processing. This approach is as an alternative to the conventional approach of extracting features over multiple layers using feature convolutions or neuron activation.

2.3.7 Open Challenges and Research Opportunities

Although many neuromorphic tracking algorithms demonstrate high speed and accuracy within their intended applications, relatively few achieve fully real-time operation while incorporating the statistically rigorous state estimation techniques characteristic of conventional tracking frameworks. As a result, despite the variety of tracking approaches, learning-based feature detection and tracking methods also

offer considerable room for research (Gallego et al., 2020).

Chapter 3

Event-Based Detection and Tracking Pipeline

3.1 Dataset for Code Validation

In this work, in order to get started with event-based object detection and tracking, the first publicly available event-based space imaging dataset in the literature was used, with over 8 hours and 377 million events, created by Afshar et al. (2020)[1].

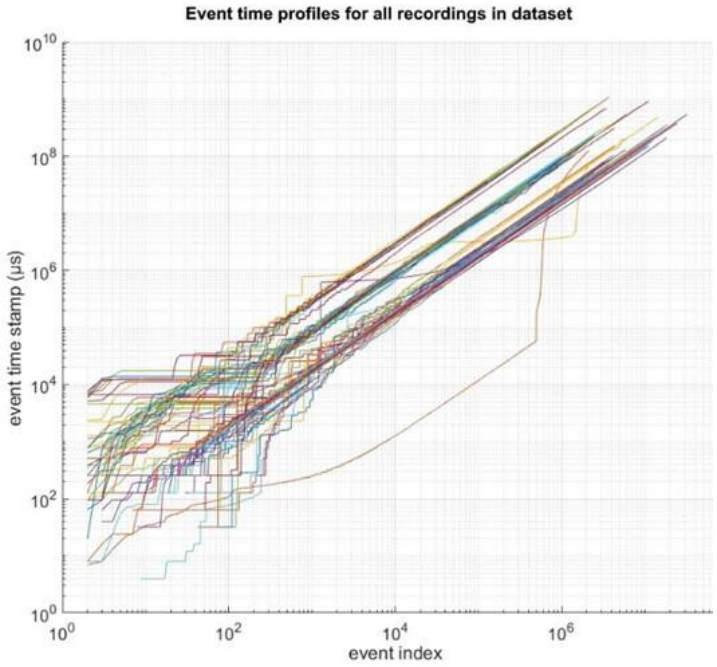


Figure 3.1: Plot figure of the timestamp of all recordings in the dataset as a function of their index. Each line indicates one recording (Afshar et al., 2020).

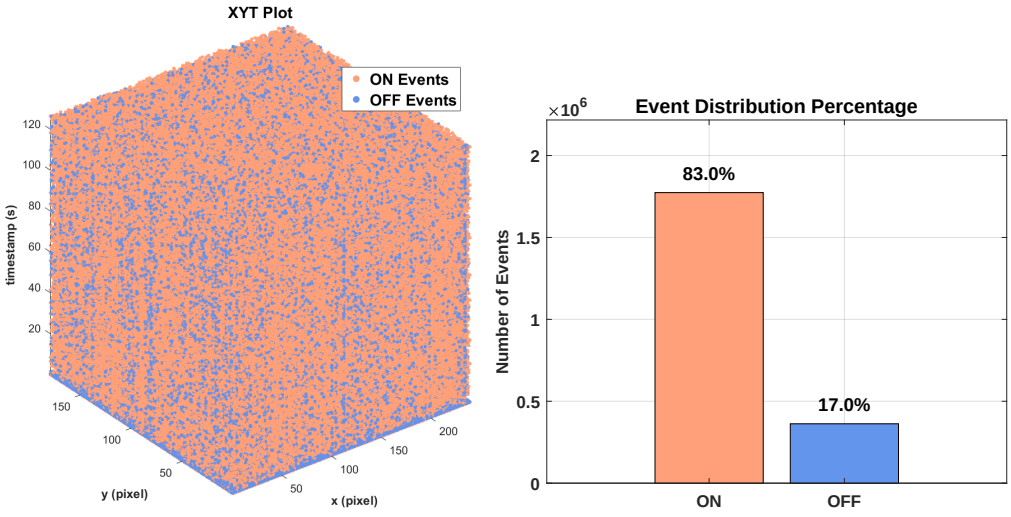
The space imaging dataset was captured using both Asynchronous Time-based Image Sensors (ATIS) and Dynamic and Active-pixel Vision Sensors (DAVIS) and

[1]https://drive.google.com/drive/folders/1TMCkTV4IS4Sxw6rXDFnUFGIE9yDW05CG

was undertaken at the Australian Defence Science and Technology Group's research facility in Edinburgh, South Australia. The experiments made use of their robotic electro-optic telescope facility, which was modified to support the event-based sensors and the existing astronomy equipment simultaneously. The recordings contain portions where the satellites are actively tracked (telescope moves to follow satellites), segments where the stars are sidereally tracked (telescope moves to compensate Earth's rotation), and parts where the telescope remains static relative to the ground. Due to the continuous nature of the event-based recordings, some recordings contain portions of all three types of operation. It is important for event-based systems to reliably track across all modes of movement, as this is what will usually be present when operating these systems.
The time-stamp profiles of all recordings in the dataset show the heterogeneity and non-idealities in the dataset, as shown in Figure 3.1. The discontinuous staircase features in the time-stamp profiles represent event stream timing artifacts. These event stream 'jumps' and 'dumps' occur when multiple events are erroneously assigned simultaneous time-stamps often at periodic intervals. From the authors' perspective, this effect is likely due to USB communication delays in the cameras.

3.2 Pre-processing

A dimetric projection of the raw event stream from a two-minute recording of the SL-8 R/B rocket body is shown in Figure 3.2a.



(a) (b)

Figure 3.2: (a) Event stream of the SL-8 R/B recording by Afshar et al. (2020) in the 3-dimensional $x-y-t$ space. Since neuromorphic sensors do not capture frames, but rather independent pixel changes, each data point is defined by its position in space and its occurrence in time; (b) Percentage distribution of ON and OFF events.

This particular recording of the rocket body is an especially instructive data stream of the dataset because it contains nearly all the sensor non-idealities, scene complexities and processing challenges that can be found in the dataset as a whole.

The inset in Figure 3.3 focuses on a small sub-region of the spatio-temporal event stream centered around the object with the highest Signal-to-Noise Ratio (SNR) which is the rocket body. The plot of this small sub-region shows the high noise rate of a typical Event-Based Space Imaging (EBSI) recording as well data artifacts such as noisy hot pixels (unbroken vertical lines in the inset), event timing jumps (at approximately $t = 93\,\mathrm{s}$), and event dumps (at approximately $t = 0\,\mathrm{s}$). Such non-ideal behavior was observed with both the DAVIS and the ATIS sensors under different conditions. In addition, a cloud of events is clearly visible in the first tenths of a second of the dimetric projection, called initial dump events.

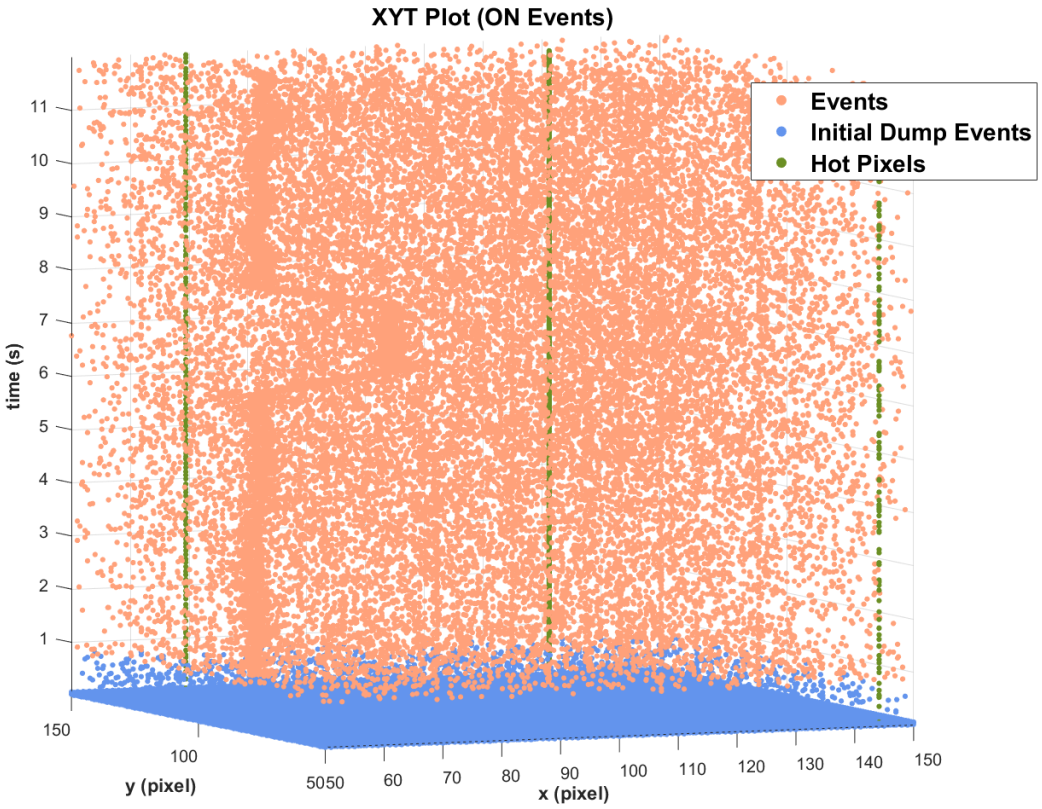


Figure 3.3: The bended curve on the left represents the event stream of the SL-8 R/B, while the green straight lines are some of the hot pixels of the recording. Blue points are the initial dump events.

The entire approach of this work, presented in Afshar et al. (2020), is event-based with all components from the sensors to the detectors and trackers operating totally in the event-based domain. We recall that an event is defined as an array of 4 elements, as shown in Eq. (2.3).

Event-based algorithms typically require some form of memory of recent events to build meaningful spatio-temporal representations. The method used in the work and one which usually outperforms other approaches is the exponentially decaying event time surface (Afshar et al., 2020). A matrix called T, that contains in its entries the time-stamp of the most recent event at each pixel, was built. At each event $\mathbf{e}_i$, the

Table 3.1: Processor 1 Characteristics.

Name	Base frequency
`Intel(R) Pentium(R) CPU 4415Y`	1.60 GHz (Without Turbo Boost)

entry in T at location $\mathbf{x}_i = (x_i, y_i)$ is updated by the event time stamp t_i with all other entries in T remaining unchanged. Using T, an exponentially decaying time surface matrix S is built.

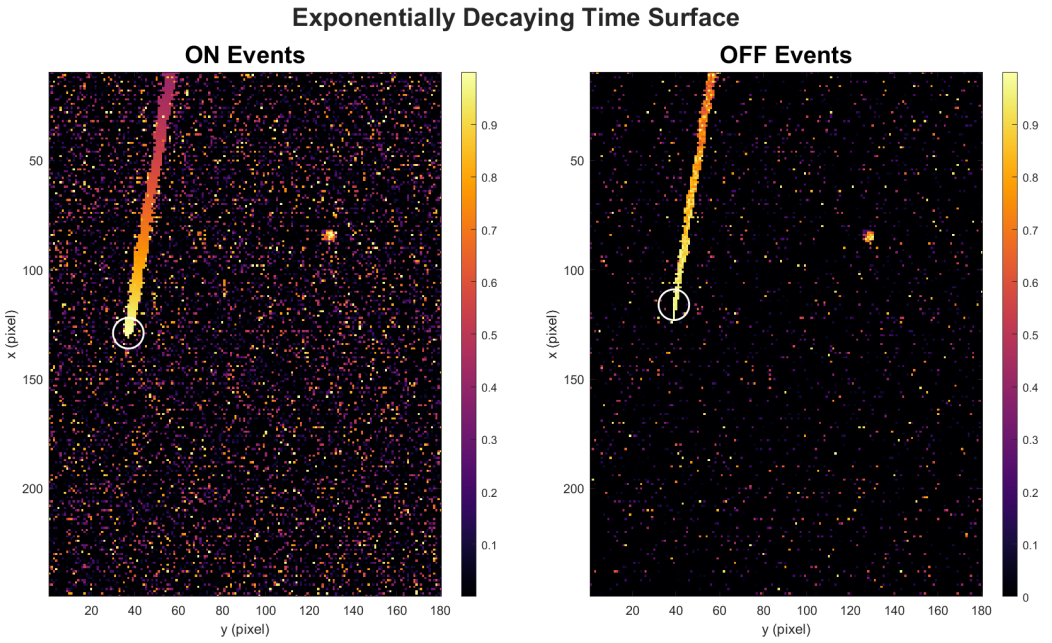


Figure 3.4: Exponentially Decaying Time Surfaces at $t = 27.4\,\mathrm{s}$. The white circles are centered about the last event occurring at the selected time.

The content of its cells can be calculated with:

$$\forall\, \mathbf{x}_k \in X: \quad S(\mathbf{x}_k) = e^{\frac{T(\mathbf{x}_k) - t_i}{\tau}} \tag{3.1}$$

where X is a matrix that contains all pixel addresses on the sensor, k is the index of the sensor pixel, and τ is the decay constant of the time surface. The value of $\tau = 0.4\,\mathrm{s}$ was chosen heuristically and found to maximize accuracy of the algorithm when applied to the dataset (Afshar et al., 2020) At shorter time constants, the memory of recent events fades so quickly on the time surface that faint objects, which generate fewer events over time, generate too short a trace to be distinguishable from noise clusters and thus are rejected, resulting in lower sensitivity (true positive rate). On the contrary, for time constants that are too long, the memory of past events persists for a prolonged period on the time surface. This causes traces from multiple events to accumulate and potentially overlap, including noise events, which can create spurious activations and reduce specificity (increase false positives). Moreover, the temporal resolution of the surface decreases, making

it harder to discriminate fast-moving objects or rapidly changing event patterns. All pixels in T and S are initialized to $-\infty$ and 0 respectively. As an example, Figure 3.4 was obtained using the instance of the ON and OFF time surface for the SL-8 R/B recording used. The particular corresponding instant of time is $t_{\text{now}} = 27.4\,\text{s}$. The white circles are centered around the last event detected at the selected time and represent the Regions Of Interest (ROI), that is, the locations of the image that are worth processing.

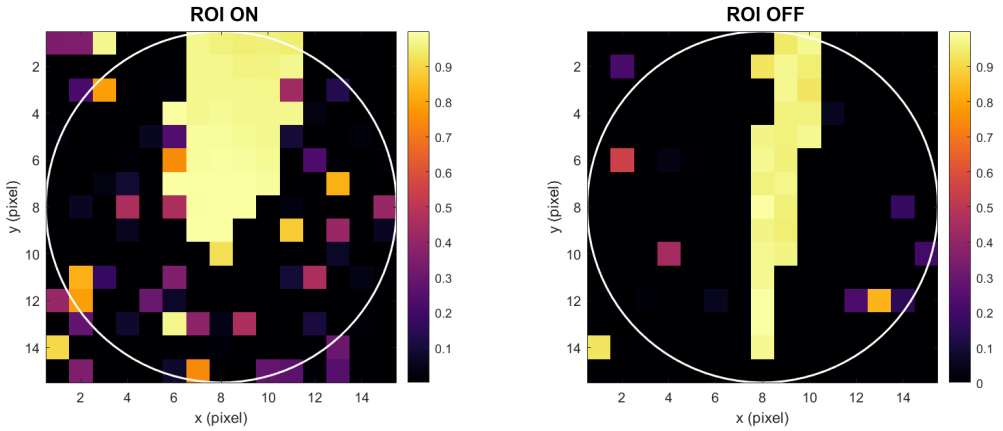


Figure 3.5: Regions of Interest.

3.3 Feature Detection Algorithm

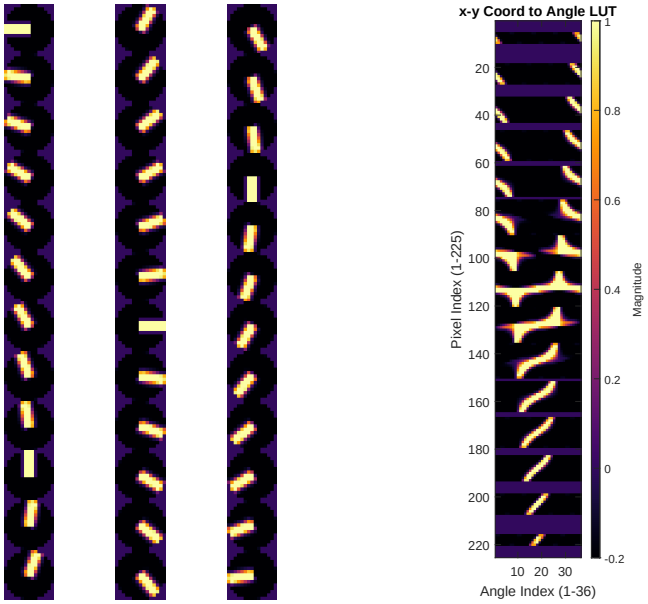
After creating the exponentially decaying time surface S, in fact, a ROI of size $D \times D$ pixels around the last event is cut from the matrix S and is selected for processing. While small ROIs with faster decaying memory suffice in high SNR contexts, in low SNR applications such as EBSI, larger-sized ROIs with slower decaying memory collect more information from a larger spatio-temporal volume, which typically results in better performance. On the other hand, increasing the ROI size reduces computational speed. During the cropping phase of the ROI, it is necessary to control if the event is located too close to the edge of the camera's FOV, to avoid problems in creating the matrix. In this specific case the event is discarded and the ROI is not created, in any other case it is maintained. When created, the ROI is kept if a surface activation test is passed successfully. Through heuristic testing of the space imaging dataset and the algorithms presented in this work, an ROI of 15×15 pixels was found to provide a reasonable trade-off between performance and speed by Afshar et al. (2020). The i-th ROI is then processed by a surface activation test. This test effectively acts as a noise filter and is a generalization of the nearest neighbor filter, presented by Czech and Orchard (2016). If the number of recently activated pixels on the time surface within the ROI is above a predefined value L, then the ROI is accepted. Recency is defined as a pixel that has received an event within Φ seconds and this means that the value of each pixel of the ROI is compared to a threshold η that is between 0 and 1. The relation between Φ and η is viewable in equation (3.2).

$$\eta = e^{-\phi/\tau} \tag{3.2}$$

In this work, the values of $L = 10$ and $\phi = 0.1\,\text{s}$ were chosen.

If this test is passed successfully, the ROI is maintained and the event screening process continues through two additional methods. The first one is called angular activation test and filters the events with the method of half bar templates.

In this stage of processing, a valid $D \times D$ ROI is converted into an angular activation vector $\mathbf{\Lambda}$, generated by matching the ROI with each of N rotated half-bar templates.



(a) (b)

Figure 3.6: (a) 36 Streak templates rotated at 10-degree intervals to calculate the angular activation of the ROI; (b) Look-Up Table (LUT) that converts the ROI values into an angular activation vector $\mathbf{\Lambda}$ through a single vector-matrix multiplication operation.

The template consists of a bar of length $R + 1$ (R being equal to $D/2$) and 3 pixels wide with a magnitude of 1. The bar width parameter at 3 pixels was found heuristically by the authors to produce the best performance across a wide range of object sizes and ROI sizes. This is likely due to the three-pixels bar being close to the size of the smallest resolvable streak in the space imaging dataset. Outside of the bar, the rest of the template has a negative magnitude of -0.2 to penalize activation from noise events. Afshar et al. (2020) selected $N = 36$, meaning that the angular step between adjacent angular templates, i.e., orientations of the half-bar template, is 10°, viewable in Figure 3.6a.

The half bar template also acts as a more general orientation-selective feature detector that can detect the tips of streaks which are much thicker than the 3-pixel half bar template. Furthermore, when the relative activation of the rotated templates is used as a feature, the pattern of template activations also selectively

detects stationary objects of arbitrary size. It was found through experimentation with a range of different bar shapes, lengths, widths and template values, that as long as the template was strongly uni-directional, the precise shape of the template did not significantly impact performance.

Later the N, $D \times D$ templates are re-arranged into an $N \times D^2$ Look Up Table (LUT), as shown in Figure 3.6b, and the $D \times D$ ROI matrix is rearranged into a $D^2 \times 1$ vector, reading its elements column by column.

In this way, matching the ROI with the half-bar templates consists in the matrix multiplication of the LUT with the ROI vector, yielding an $N \times 1$ vector $\mathbf{\Lambda}$, as viewable in Eq. (3.3).

$$\mathbf{\Lambda} = \mathbf{LUT} \cdot \mathrm{vec}(\mathbf{ROI}_i) \tag{3.3}$$

The i-th element of the vector $\mathbf{\Lambda}$ quantifies the activation of the i-th angular template, i.e. how good is the match of the ROI with the i-th orientation of the half-bar template. Each index of $\mathbf{\Lambda}$ is equivalent to an angle, and the step between adjacent indexes is 10°, as stated above. The index m of the maximum value of $\mathbf{\Lambda}$ is associated with the most activated angular template, i.e. the orientation of the half-bar template that best matches the ROI.

Regardless of the implementation environment, the calculation of the angular activation vector $\mathbf{\Lambda}$ is the most computationally expensive step relative to the number of events processed, since it's significantly more computationally expensive than the previous surface activation test, but, unlike subsequent stages which are also computationally intensive, this operation is performed on the majority of the events from the camera.

In conclusion of this process, the angular vector $\mathbf{\Lambda}$ is compared to a static threshold $\Psi = 7$. The use of a static threshold simplifies the algorithm implementation whereas the use of a dynamic threshold can provide greater robustness to noise events. The index of each angle that is above this threshold is taken and put into another vector, called $\mathbf{\Gamma}$. Subsequently, after passing this first step, a $\breve{\mathbf{\Lambda}}$ vector is created, which is $\mathbf{\Lambda}$ circularly shifted by $N/2$, as explained in Algorithm 1. $\breve{\mathbf{\Gamma}}$ is extrapolated from $\breve{\mathbf{\Lambda}}$ in the same way as $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$. At this point, both $\mathbf{\Gamma}$ vectors are compared to another threshold, this time dynamic, calculated as in Eq. (3.4).

$$\Psi_i = \frac{\max(\mathbf{\Lambda}_i) + \min(\mathbf{\Lambda}_i)}{2} \tag{3.4}$$

This test serves as a filter that removes events not associated with a streak on the time surface. However, this filter does not distinguish between events that are on or near a streak and those at the streak's tip (Afshar et al., 2020).

To further extract these later events, a statistical unimodality test must be applied to the angular activation vector $\mathbf{\Lambda}$. To this end, the authors propose a highly simplified hardware amenable circular unimodality test which calculates the algebraic mean q of $\mathbf{\Gamma}$ and checks whether the distance ζ between q and the index m of the maximum element of $\mathbf{\Lambda}$ (and therefore of $\mathbf{\Gamma}$) is below a threshold δ. The authors opted for a value of $\delta = 2$, which is equivalent to a tolerance of 20° in the angular distance. The authors explain that the computation of the mean q of vector $\mathbf{\Gamma}$ as a non-circular algebraic mean, and not as a circular mean, is necessary to decrease the computational cost, but it also allows to deal with pathological cases in

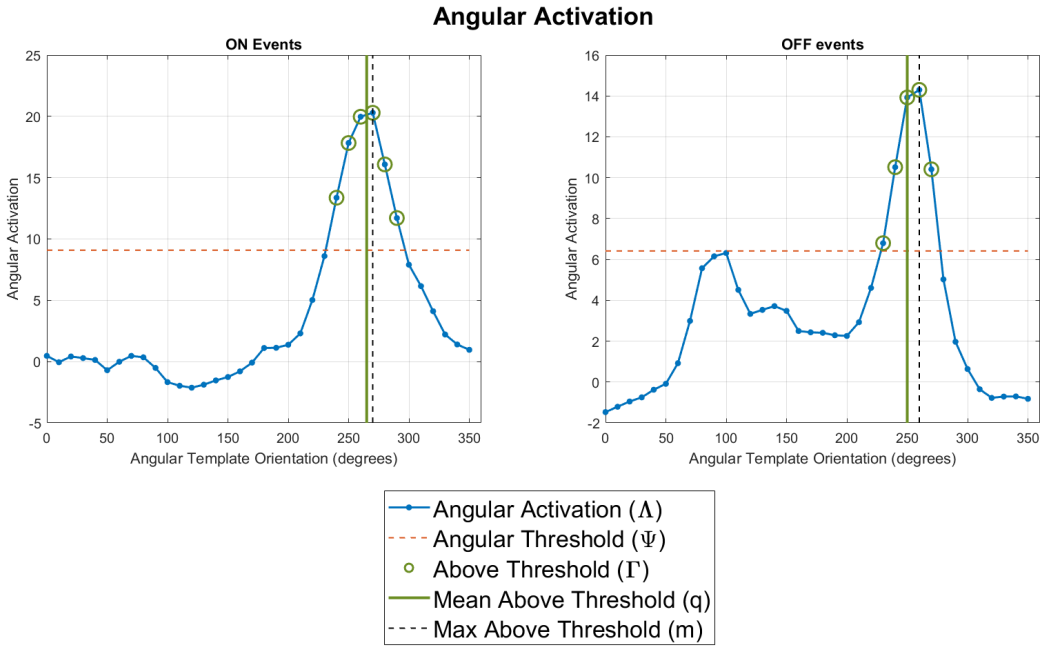


Figure 3.7: Angular activation of the half-bar templates when matching with the ROI centered on the event at $t_{\text{now}} = 27.4 \times 10^6$ s.

the space imaging event-based dataset. These pathological cases make up a small but consistent fraction of the observed ROIs, occurring regularly whenever events are triggered late in the trail of a fast-moving target. These trail events generate bimodal distributions of $\mathbf{\Lambda}$ which regularly have circularly symmetric elements that can cancel each other out. In such cases, the standard circular mean method results in the circular mean index q and circular max index m being close enough to generate false positive detections. The same operation is then performed on $\breve{\mathbf{\Lambda}}$. These two operations result in two non-circular distances ζ and $\breve{\zeta}$, the smaller of which is compared to the threshold δ. The unimodality test is passed if:

$$\min(\zeta_i, \breve{\zeta}_i) = \min(|m_i - q_i|, |\breve{m}_i - \breve{q}_i|) < \delta \tag{3.5}$$

Despite its simplicity, the proposed unimodality test, whose results are viewable in Figure 3.10, is highly selective and performs remarkably well at extracting space targets from the observed event-based space event streams while being robust to noise, object velocity, object size, and orientation (Afshar et al., 2020). If the event e_i passes the angular unimodality test, it is augmented with the mean orientation variable $\theta_i = q_i$, which tells in which direction the object is moving, and results in an output detection event f_j.

The resulting, substantially sparser output stream of streak detection events can then be forwarded to a more computationally demanding event-based tracker. The event-based tracker may in turn be interpreted as an even more restrictive filter, capable of rejecting spurious detection events that are not associated with other nearby detections sharing the same orientation and velocity.

The number of events that passed the various steps with this set of parameters

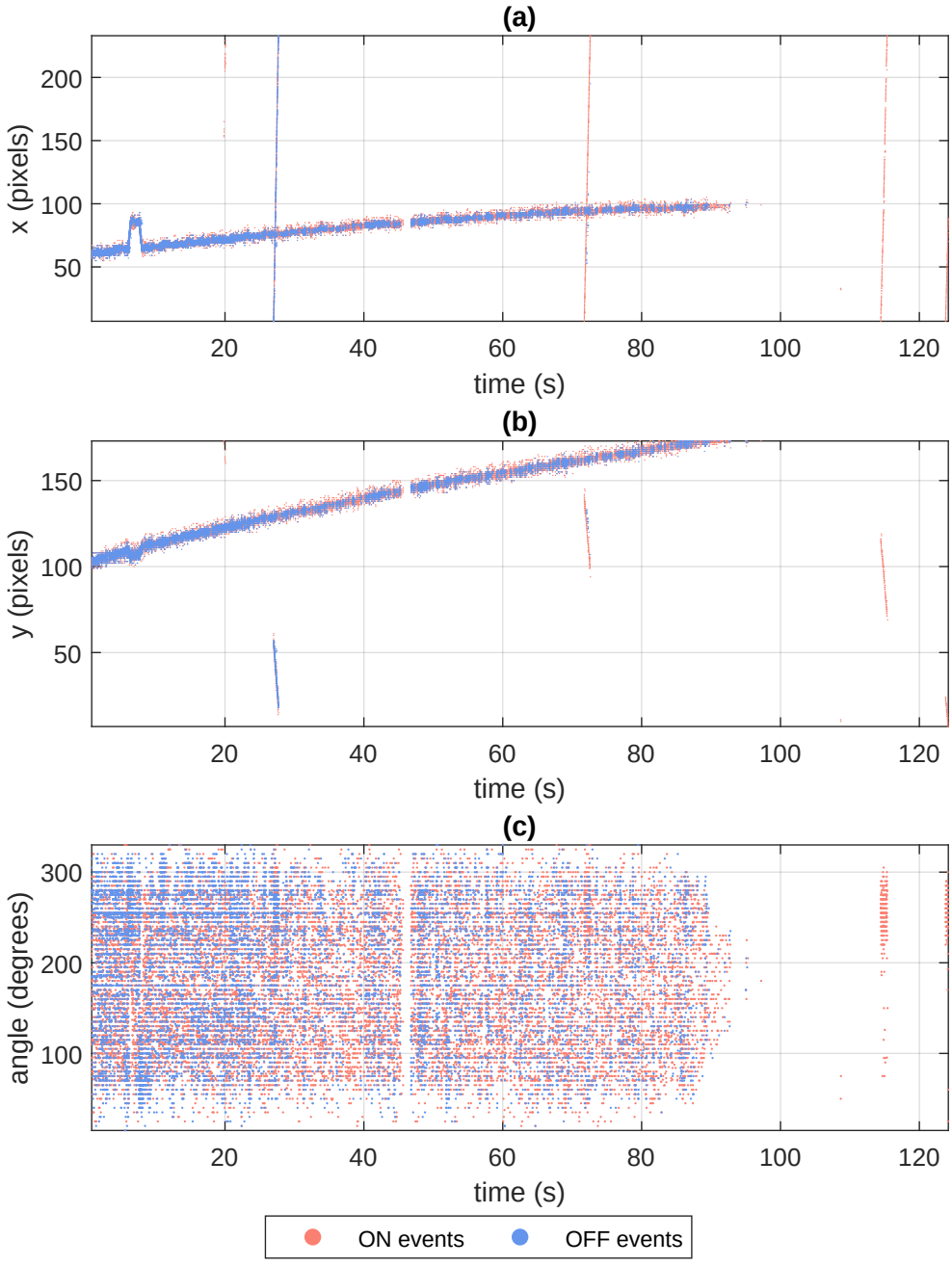


Figure 3.8: The panels in (a), (b) and (c) show the x and y pixel location and orientation of the events which passed the feature detection. The panel in (c) show the orientation of the detection events, based on the mean index of above-threshold templates $\theta_i = q_i$. SL-8 R/B forms the long, slightly curved line.

Table 3.2: Feature Detection Parameters used for processing of Afshar et al. (2020) recording.

Parameters	Values	Name
τ	0.4 s	Decay Constant of the Time Surface
L	10	Number of active ROI pixels
ϕ	0.1 s	Decay Constant of Surface Activation Test
δ	20°	Tolerance of Angular Distance
Ψ_{stat}	7	Static Threshold
Δt_{feat}	102 s	Code Processing Time

is 29 488 ON and 16 222 OFF, starting from the initial 2 136 083, as shown in table 3.4. Actually, events with a timestamp less than 0.8 s were excluded from this analysis because they generated too many false positives, containing 567 331 events of the total recorded. The effectiveness of this feature detection filter can be seen in Figures 3.8 and 3.9.

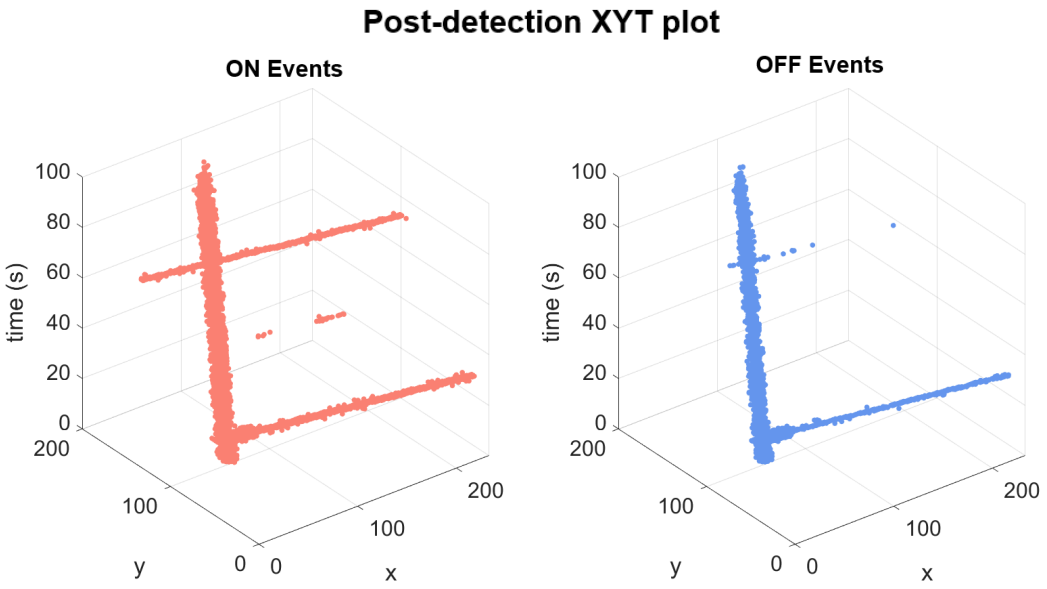


Figure 3.9: Processed event stream derived from the SL-8 R/B recording, following the application of feature detection algorithms and noise filtering of the events.

3.4 Tracking Algorithm

The event-based tracker algorithm removes virtually all false detection events remaining after the feature detection stage while correctly clustering events from each object.

At the core of the proposed event-based tracker lies the concept of hypothesis, or predicted observation, $H_k^{(o)}$ of the o-th object at time step k:

$$H_k^{(o)} = \left[\hat{x}_k,\, \hat{y}_k,\, \hat{\theta}_k,\, \frac{d\hat{x}_k}{dt},\, \frac{d\hat{y}_k}{dt}\, \frac{d\hat{\theta}_k}{dt},\, p_k,\, t_k\right]^T \tag{3.6}$$

When the first detection event f_j of an object o in the scene occurs, the first hypothesis, or a priori hypothesis, for that object is defined as:

$$H_1^{(o)} = [x_j,\, y_j,\, \theta_j,\, 0,\, 0,\, 0,\, p_j,\, t_j]^T \tag{3.7}$$

Each tracked object o is modeled as a Leaky Integrate-and-Fire (LIF) neuron whose membrane potential decays over time and is incremented by spikes whenever a detection event f_j is assigned to it. The membrane potential $M_k^{(o)}$ represents the level of recent observations of object o at time step k and is defined as:

$$M_k^{(o)} = \begin{cases} M_{k-1}^{(o)} - \gamma(t_j - t_{k-1}) + 1, & \text{if } f_j \text{ is assigned to } H_k^{(o)}, \\ 0, & \text{if } M_{k-1}^{(o)} \leq 0, \\ M_{k-1}^{(o)} - \gamma(t_j - t_{k-1}), & \text{otherwise,} \end{cases} \tag{3.8}$$

where $\gamma > 0$ is the decay constant. For the *a priori* hypothesis, $M_{k-1}^{(o)} = 0$, hence $M_1^{(o)} = 1$.

If the membrane potential $M_k^{(o)}$ reaches the activation potential M_A, the object is identified as a true tracked object. Conversely, if $M_k^{(o)}$ reaches zero, the object is deleted.

An activation variable $A_k^{(o)}$ tracks the activation level of the object until it reaches M_A:

$$A_k^{(o)} = \begin{cases} \dfrac{M_k^{(o)}}{M_A}, & \text{if } M_k^{(o)} < M_A, \\ 1, & \text{otherwise.} \end{cases} \tag{3.9}$$

When a new detection event f_j is collected at time $t_j = t_k$, the predicted hypothesis of each active object ($M_k^{(o)} > 0$) is computed assuming linear constant-velocity motion:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{d}{dt_{k-1}} x \\ \frac{d}{dt_{k-1}} y \\ \frac{d}{dt_{k-1}} \theta \end{bmatrix} (t_k - t_{k-1}). \tag{3.10}$$

The weighted Euclidean distance between detection event f_j and the predicted hypothesis $H_k^{(o)}$ is then computed as:

$$d_k^{(o)} = \sqrt{(x_j - x_k^{(o)})^2 + (y_j - y_k^{(o)})^2 + w_k^{(o)}(\theta_j - \theta_k^{(o)})^2}. \tag{3.11}$$

The angular weight $w_k^{(o)}$ is defined as:

$$w_k^{(o)} = V\sqrt{\left(\frac{dx_k^{(o)}}{dt}\right)^2 + \left(\frac{dy_k^{(o)}}{dt}\right)^2}\, A_k^{(o)}, \tag{3.12}$$

where V is a scaling factor, set to $V = 0.1$ pixel/degree.

The detection event f_j is assigned to the object that minimizes $d_k^{(o)}$ subject to the constraint:

$$d_k^{(o)} < d_{\max}. \tag{3.13}$$

Only the winning hypothesis is updated using a sequential least-squares method with online covariance and variance estimation over a rolling window of length K. The membrane potential of the winning object is incremented by one. If no object satisfies the distance constraint, a new object is initialized with hypothesis $H_1^{(o+1)}$.

Finally, the membrane potentials of all active objects decay, and a new tracker iteration begins.

The output of the tracker is an event stream of the form:

$$g_l = [x_l,\, y_l,\, p_l,\, \theta_l,\, o_l,\, t_l]^T \tag{3.14}$$

where o_l denotes the index of the object associated with the l-th tracker event.

Figure 3.10: (a) Number of tracking events per object; panels (b) and (c) show the tracker event position in x and y respectively over time, while (d) shows the angular direction.

As reported by Afshar et al. (2020), the number of objects crossing the scene during the whole is 9. Using the parameters in Table 3.3, it was possible to correctly track 5 objects, unlike the 6 that were tracked in Afshar et al. (2020). Furthermore, at approximately 72 s, a minimal interference occurs between two bodies, resulting in a subtle exchange of events between them. This demonstrates the level of difficulty in automating this tracking process.

It was observed that this method relies on a relatively large set of parameters, shown in Tables 3.2 and 3.3, that must be carefully tuned according to the specific

Table 3.3: Tracking Parameters selected to process the Afshar et al. (2020) recording.

Parameters	**Values**	**Name**
M_A	150	Activation Potential
γ	100 s^{-1}	Decay Constant for the Membrane Potential
d_max	70 pixel	Threshold Acceptable Distance
V	0.1 pixel/deg	Scaling Factor
K	15	Length of Fixed Rolling Window
Δt_{track}	125 s	Code Processing Time

application scenario. The processing time reported in these tables refers to the processor of the table 3.1. While this level of configurability provides flexibility and can lead to strong performance when properly optimized, it also introduces a significant dependency on prior calibration and expert knowledge.
In practice, the need for parameter tuning may limit the method's adaptability in environments where operating conditions are not fully known in advance or may vary over time. For example, changes in target velocity, distance, illumination conditions, or sensor noise characteristics can alter the statistical properties of the event stream, thereby requiring adjustments to multiple parameters to maintain stable performance. This sensitivity can complicate deployment in real-world systems, particularly in autonomous platforms where continuous manual reconfiguration is not possible.
Moreover, parameter inter-dependencies may further increase the complexity of the tuning process. Adjusting one parameter often affects the optimal setting of others, creating a multidimensional optimization problem that can be computationally expensive and time-consuming to solve. In scenarios involving resource-constrained hardware, such as on-board spacecraft systems for SSA or collision avoidance, this additional complexity may represent a non-negligible drawback.
These considerations do not diminish the effectiveness of the approach itself, but they highlight the importance of designing tracking frameworks that either reduce the number of critical parameters or make them less sensitive to environmental variations. In this context, as it will be shown in subsections 4.3.1 and 4.3.2, our work emphasizes a more structured and informative pre-processing stage, with the aim of stabilizing the event representation early in the pipeline. By providing a cleaner and more consistent input to the tracking module, the overall system can potentially operate with fewer application-specific adjustments, thereby improving robustness, scalability, and ease of deployment across different operational scenarios.

Table 3.4: Evolution of the total number of events of Afshar et al. (2020) recording through the event-based processing pipeline.

Raw Events	**Post-Detection**	**Post-Tracking**
2 136 083	45 710	45 081

Algorithm 1 Event-Based Detector + Tracker

Require: Raw event stream $\mathbf{e}_i = [x_i, y_i, p_i, t_i]^T$
Ensure: Tracker event stream $\mathbf{g}_l = [x_l, y_l, p_l, \theta_l, o_l, t_l]^T$

$j \leftarrow 0$
$l \leftarrow 0$
for each incoming raw event $\mathbf{e}_i$ **do**
    Update exp. decaying time surface $\mathbf{S}_i$
    Select $\mathbf{ROI}_i \subset \mathbf{S}_i$
    if $\mathbf{ROI}_i$ passes surface activation test **then**
        $\mathbf{\Lambda}_i \leftarrow \mathbf{LUT} \cdot \mathrm{vec}\,(\mathbf{ROI}_i)$
        if $\max(\mathbf{\Lambda}_i) > \Psi$ **then**
            $\breve{\mathbf{\Lambda}}_i = [\mathbf{\Lambda}_i((N/2+1):N, \mathbf{\Lambda}_i(1:N/2)]$
            $\Psi_i = [\max(\mathbf{\Lambda}_i) + \min(\mathbf{\Lambda}_i)]/2$
            $\breve{\mathbf{\Gamma}}_i \leftarrow \{n\}\ s.t.\ \mathbf{\Lambda}_i(n) > \Psi_i$
            $\mathbf{\Gamma}_i \leftarrow \{n\}\ s.t.\ \mathbf{\Lambda}_i(n) > \Psi_i$
            $q_i = 1/G \sum_{h=1}^{G} \mathbf{\Gamma}_i(h)$
            $\breve{q}_i = 1/G \sum_{h=1}^{G} \breve{\mathbf{\Gamma}}_i(h)$
            $m_i = \mathrm{argmax}_n \mathbf{\Lambda}_i(n)$
            $\breve{m}_i = \mathrm{argmax}_n \breve{\mathbf{\Lambda}}_i(n)$
            if $\min(|m_i - q_i|, |\breve{m}_i - \breve{q}_i|) < \delta$ **then**
                $j \leftarrow j + 1$
                $\theta_j \leftarrow q_i$
                $\mathbf{f}_j \leftarrow [x_i, y_i, p_i, \theta_j, t_i]^T$
                for every object o s.t. $M_k^{(o)} > 0$ **do**
                    Compute predicted hypothesis $H_k^{(o)}$
                end for
                if $\exists\ o$ s.t. $d_k^{(o)} < d_{max}$ **then**
                    $n \leftarrow \mathrm{argmin}_m d_k^{(m)}$ s.t. $d_k^{(m)} < d_{max}$
                    $M_k^{(n)} \leftarrow M_{k-1}^{(n)} + 1$
                    Update hypothesis $H_k^{(n)}$
                    $l \leftarrow l + 1$
                    $\mathbf{g}_l \leftarrow [x_k^{(n)}, y_k^{(n)}, p_k^{(n)}, \theta_k^{(n)}, n, t_k^{(n)}]^T$
                end if
                for every object o s.t. $M_k^{(o)} > 0$ **do**
                    Update $M_k^{(o)}$
                end for
            end if
        end if
    end if
end for

Chapter 4

Laboratory tests

4.1 Testbed Configuration

The experimental data acquisition was carried out using a preliminary laboratory setup specifically designed for recordings with an event-based camera. The hardware and software components are summarized below:

- Vision System:
  - Camera: IDS uEye XCP-E, equipped with a Sony Prophesee IMX636 event-based sensor (1/2.5" format)[1]
  - Lens: Kowa 16-mm lens with a fast aperture of F/1.4 (manual focus fixed at 16 mm)[2]
  - Software: Data acquisition and sensor management were performed using the Prophesee Metavision Studio software suite[3]
- Mechanical Support (Thorlabs Infrastructure):
  - Two MB1590/M optical breadboards arranged in a T-configuration[4]
  - MB1560/M optical breadboard[5]
  - Two posts and two postholders
  - AP90RL/M right-angle bracket and a custom 3D-printed mounting plate for stable alignment between the sensor and the optical axis

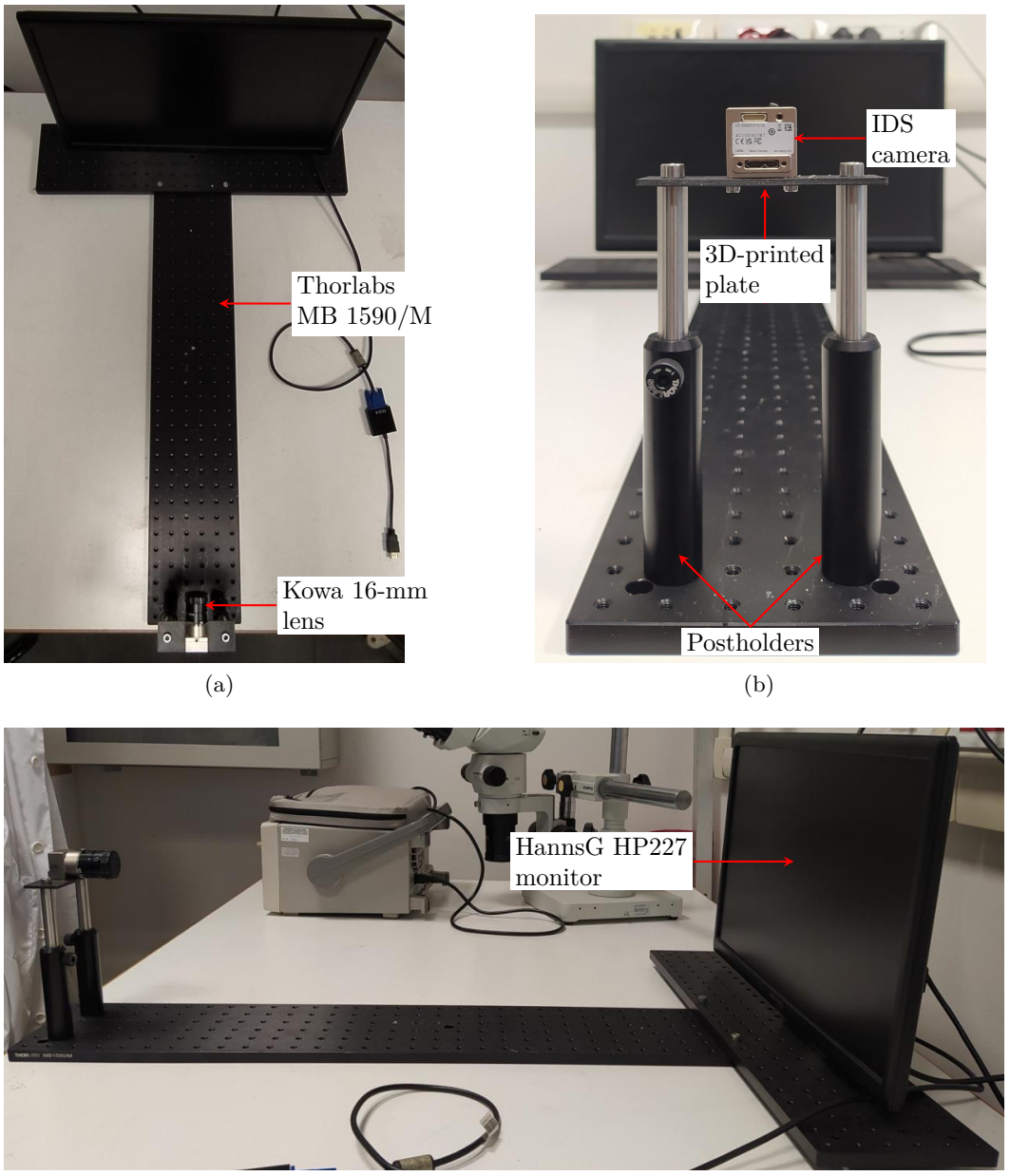
[1] https://en.ids-imaging.com/store/ue-39b0xcp-e.html
[2] https://www.soltecstore.com/visione/254532-LENS-KOWA-04M-23-16MM.html
[3] https://docs.prophesee.ai/stable/metavision-studio/
[4] https://www.thorlabs.com/item/MB1590-M
[5] https://www.thorlabs.com/item/MB1560-M

- Visual Stimulus and Control:
  - Monitor: HannsG HP227 (1920×1080 resolution, 8-bit depth) operating at a fixed refresh rate of 59.94 Hz
  - Connectivity: USB 3.0 interface for camera power and data transfer; HDMI connection for the stimulus display

From the optics–sensor coupling, some key optical parameters were derived. The resulting horizontal FOV is 42.49°, while the vertical FOV is 24.67°. The system's pixel scale is 2 arcmin/pixel, where one arcminute corresponds to 1/60 of a degree. This relatively large pixel scale implies a limited spatial resolution for fine details; however, the wide field of view allows for the observation of a substantial portion of the scene. This trade-off is particularly advantageous in sky observation scenarios, where covering a large area increases the likelihood of capturing the target of interest. The whole testbed is shown in Figure 4.1.

4.2 Optical Stimulus

The experiment was designed to simulate a simplified space observation scenario, in which a point-like object undergoes uniform rectilinear motion against a dark background. For simplicity, the presence of stars, secondary objects crossing the FOV, and other light sources such as the Sun, the Earth, and the Moon were omitted. This represents a further development that will be considered in future work. The synthetic scene was generated and projected onto the monitor described in the previous section, which acted as a fully controllable and repeatable visual stimulus. This experimental configuration enabled precise control over the object's motion, while ensuring a repeatability across multiple acquisition runs.
The projected object moved at constant velocity along a straight trajectory, emulating the apparent motion of a distant object observed in a space environment, such as a satellite or space debris traversing the FOV of an optical sensor. The use of a point-like target allowed the experiment to simulate the appearance of an RSO as seen from the distance and to focus on motion-induced event generation, avoiding additional complexities related to object shape or texture.
The monitor displaying the synthetic scene was continuously observed by the event-based camera, which recorded the resulting asynchronous stream of events generated by temporal changes in pixel intensity. As the object moved across the display, events were primarily triggered at the object's leading and trailing edges, where intensity changes occurred, closely resembling the behavior expected in real space-based observations (Cohen et al., 2019).
This experimental setup made it possible to acquire a controlled event stream representative of an ideal motion scenario, which could then be processed and analyzed using the developed feature detection and tracking algorithms.



(a) (b)

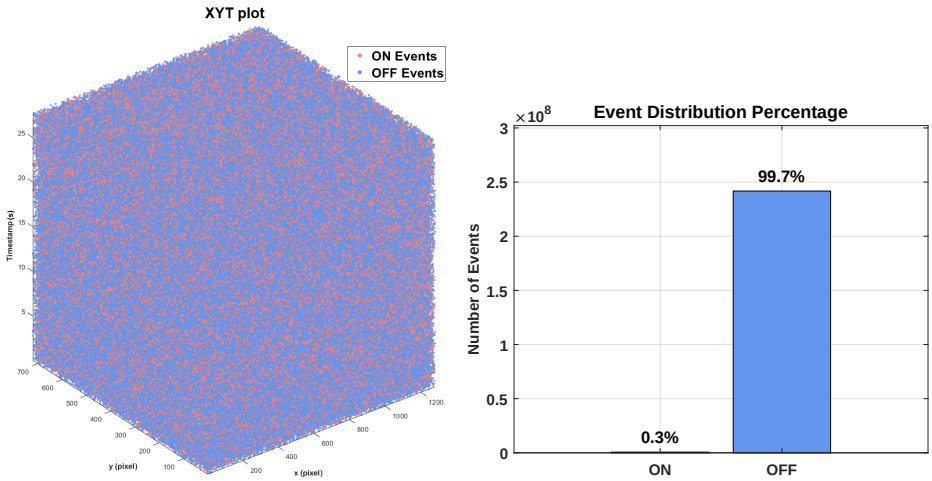
(c)

Figure 4.1: Top (a), Front (b) and Side (c) view of the testbed.

4.3 Results

4.3.1 Default Bias Settings

Initially, the IMX636 sensor was configured using the default bias current settings. As shown in Figure 4.2(a), the 3D XYT representation of the raw event stream clearly reveals a dense background activity, particularly in the OFF polarity channel. The histogram in Figure 4.2(b) quantitatively confirms this imbalance, with a marked predominance of OFF events. This behavior is consistent with the expected effect of overly sensitive bias settings, in particular, too low contrast thresholds, which tend to amplify noise contributions.



(a) (b)

Figure 4.2: (a) 3D XYT event stream (ON and OFF polarities) captured with default bias settings; (b) percentage distribution of ON and OFF events.

The impact of this excessive sensitivity becomes even more evident when analyzing the exponentially decaying time surface in Figure 4.3. While the trajectory of the moving point is still recognizable, a substantial residual background is visible, especially in the OFF stream surface. The presence of widespread spurious activations suggests that, without adequate filtering, the subsequent processing stages would be overwhelmed by irrelevant data.

This huge amount of data did not prevent the feature detection algorithm from effectively filtering the recorded events. Starting from a total of 242 336 544 raw events, the algorithm successfully reduced the data to 8 732 ON events and 6 315 OFF events, as shown in Table 4.4, and the point's trail is correctly visible as shown in figure 4.3.

Table 4.1: Processor 2 Characteristics.

Name	**Base frequency**
`12th Gen Intel(R) Core(TM) i7-12700H`	2.30 GHz

The parameters adopted for this stage are reported in Table 4.2. In particular,

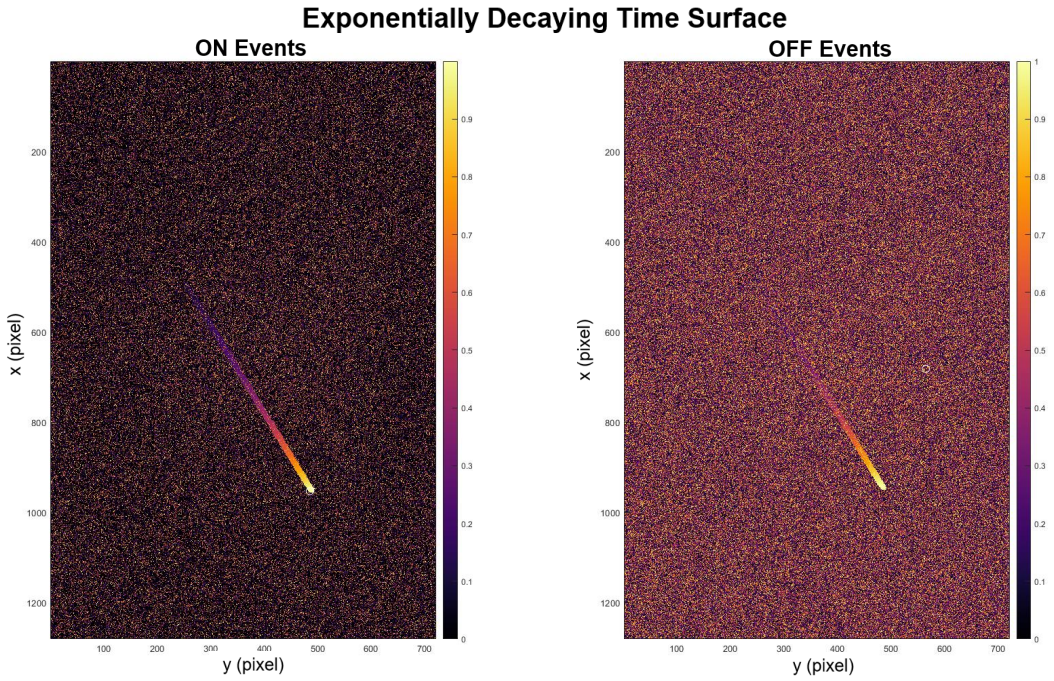


Figure 4.3: Exponentially Decaying Time Surface at $t = 24\,\mathrm{s}$ and default bias currents. The white circles are centered around the last event at the selected time. Notably, a significant background residual noise is evident, specifically affecting the OFF stream surface, where spurious activations appear more pronounced.

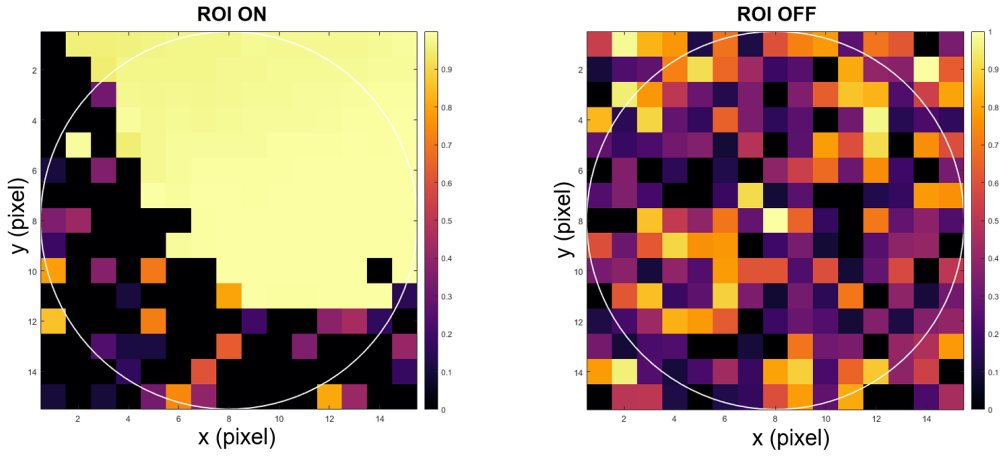


Figure 4.4: Regions of Interest (Default Bias Settings).

the asymmetric choice of L_{ON} and L_{OFF} reflects the need to compensate for the higher noise level in the OFF channel, as previously observed in Figures 4.2 and 4.3.

Table 4.2: Feature Detection Parameters used for the recording with default bias settings.

Parameter	**Value**	**Description**
τ	0.4 s	Decay Constant of the Time Surface
L_{ON}	20	Number of active ROI pixels for ON events
L_{OFF}	100	Number of active ROI pixels for OFF events
ϕ	0.1 s	Decay Constant of Surface Activation Test
δ	20°	Tolerance of Angular Distance
Ψ_{stat}	7	Static Threshold
Δt_{feat}	10 min	Code Processing Time

The temporal evolution of the post-detection events is illustrated in Figure 4.5. The panels clearly show a coherent progression of the x and y coordinates, consistent with the uniform linear motion of the projected object. The orientation estimates also remain stable over time, indicating that the feature detection stage preserves the geometric consistency of the streak despite the strong initial noise.

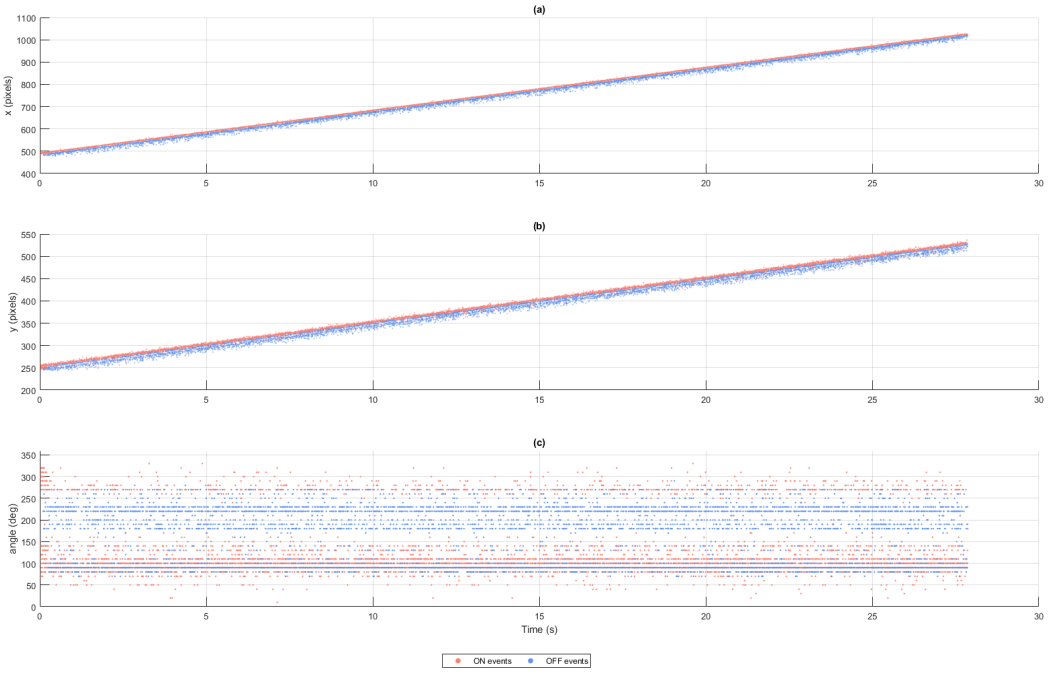


Figure 4.5: The panels show the x and y positions and orientation of the detection events over the time interval corresponding to the single streak produced by the object in uniform linear motion with the default configuration of the bias currents.

A further confirmation of the filtering effectiveness is provided by the post-detection XYT visualization in Figure 4.6. Compared to the raw event map in Figure 4.2(a), the refined stream appears significantly cleaner, with most of the

background noise removed and the object's trajectory clearly isolated.

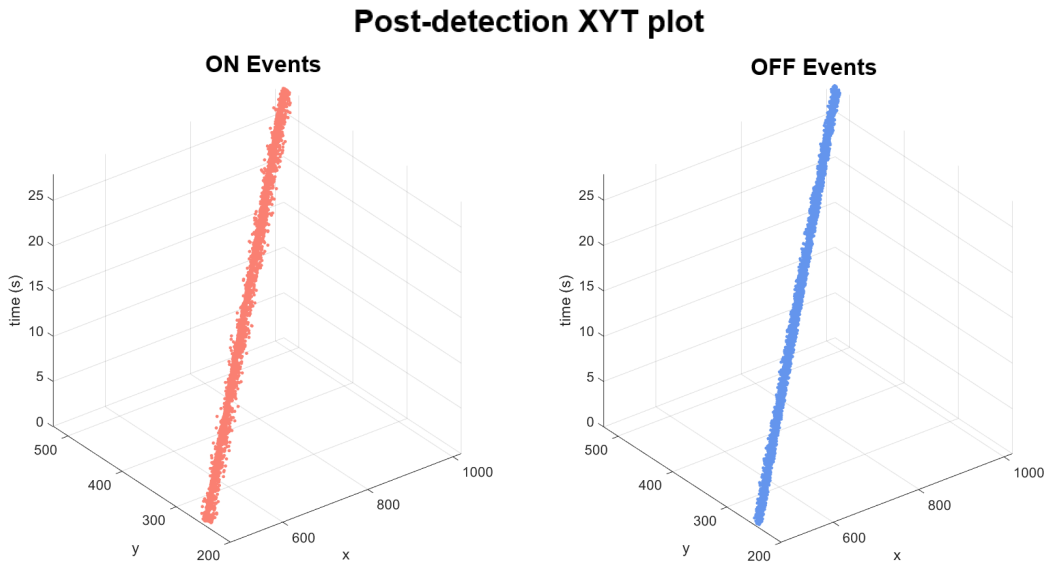


Figure 4.6: XYT visualization of the refined event stream (ON/OFF) post-feature detection, illustrating the reduction in noise events compared to raw data.

Table 4.3: Tracking Parameters of the recording with default bias settings.

Parameter	**Value**	**Description**
M_A	150	Activation Potential
γ	100 s^{-1}	Decay Constant for the Membrane Potential
d_{max}	70 pixel	Threshold Acceptable Distance
V	0.1 pixel/deg	Scaling Factor
K	15	Length of Fixed Rolling Window
Δt_{track}	13 min	Code Processing Time

The tracking results, displayed in Figure 4.7, demonstrate that the system is able to maintain a stable track of the single object present in the scene. The tracking parameters used for this configuration are listed in Table 4.3. These values were selected to ensure stability in the presence of the high residual noise level observed with default bias settings.

Finally, Table 4.4 summarizes the overall reduction in the number of events across the entire pipeline. From the initial 242 336 544 raw events, only 14 928 events remain after the complete tracking stage. This result highlights the strong selection capability of the proposed approach: even under suboptimal sensor bias conditions, the system is able to drastically reduce the data volume while preserving the essential information required for reliable tracking.

Table 4.4: Evolution of the total number of events thorugh the event-based processing pipeline in the recording with default bias settings.

Raw Events	Post-Detection	Post-Tracking
242 336 544	15 047	14 928

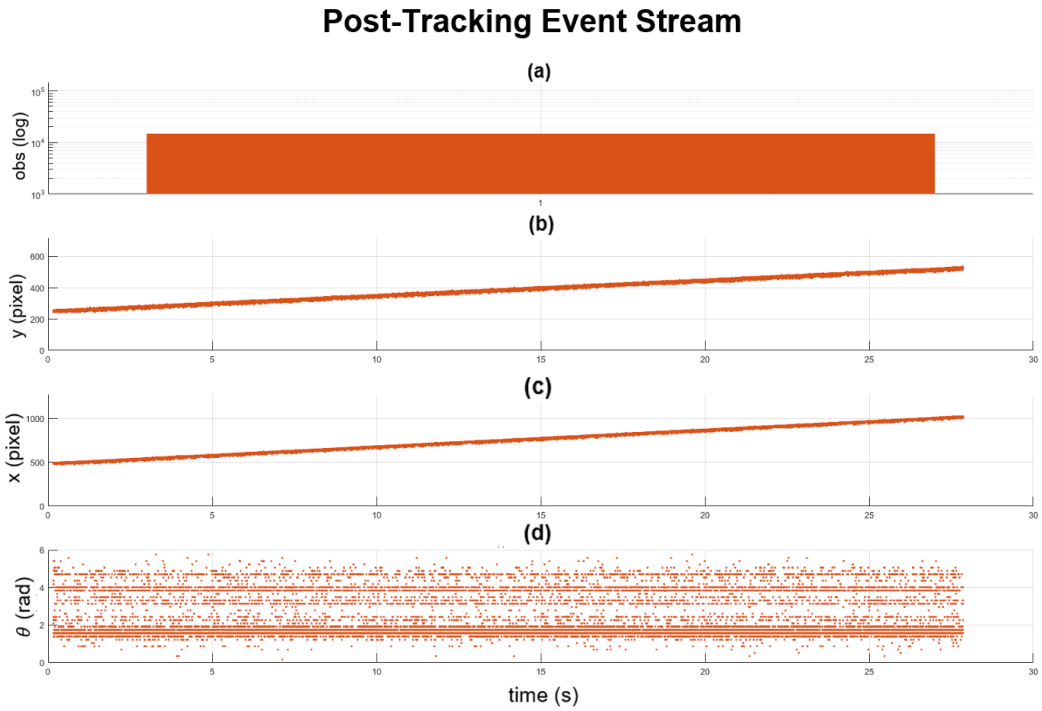


Figure 4.7: In (a) it can be seen the number of tracking events associated with the only object present on the scene. The panels (b), (c) and (d) show the x and y positions and orientation of the detection events over the time interval corresponding to the single streak produced by the object in uniform linear motion.

4.3.2 Bias Tuning

Despite the results in Subsection 4.3.1, in a subsequent phase of the work the IMX636 sensor was configured using a specific set of bias currents, which directly affect the sensor's performance, including sensitivity, temporal response, and noise characteristics, viewable in Table 4.5. The values provided are dimensionless quantities, as they are normalized according to the standard output of the Metavision Studio graphical interface. By adjusting the bias currents by hand, it was possible to significantly reduce background noise and spurious events, yielding a cleaner and more informative event stream prior to the application of feature detection and tracking algorithms. This highlights the role of sensor-level parameter tuning in event-based vision systems, where the quality of the raw asynchronous data has a direct impact on the effectiveness of subsequent processing stages. In addition, in Figure 4.8, 'jumps' and 'dumps' in event time profiles are still present, most notably during the bias tuning test. These event stream 'jumps' and 'dumps' occur when multiple events are erroneously assigned simultaneous time-stamps often at periodic intervals, causing the 'staircase effect'. The time profile takes on a stepped shape because many events share the same time value (the horizontal "step" or dump), followed by a sudden jump to the next timestamp when the buffer empties. The frequency of these fluctuations is significantly reduced compared to the dataset by Afshar et al. (2020).

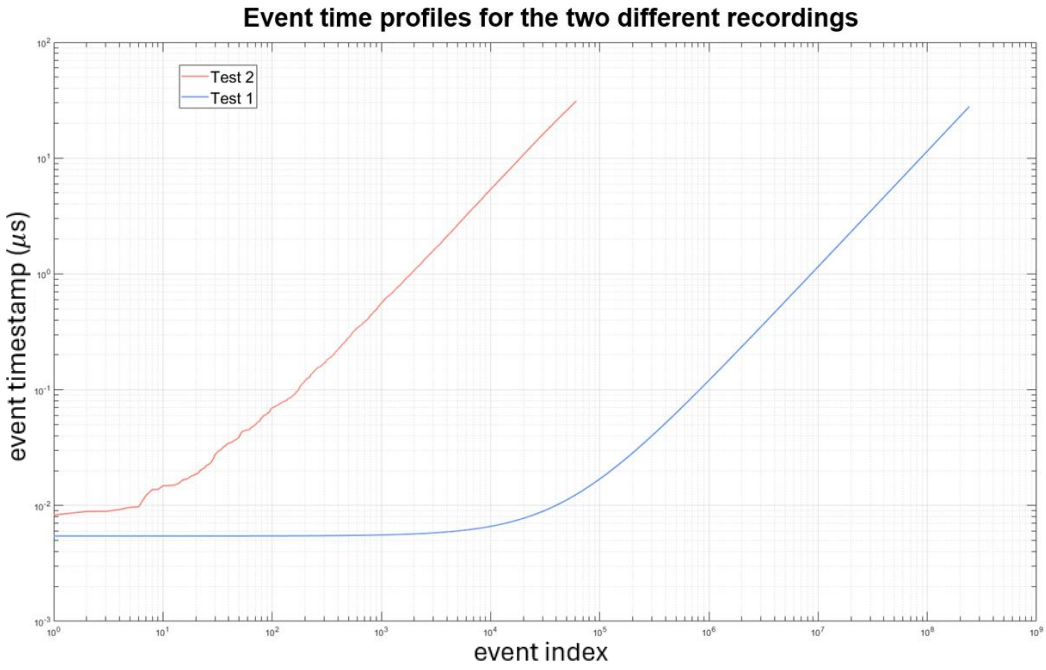


Figure 4.8: Event time profiles for the two different recordings as a function of their index. Test 1 indicates the recording with default bias settings, while Test 2 indicates the recording with bias tuning.

The impact of bias tuning becomes immediately evident when analyzing the 3D XYT event distribution shown in Figure 4.9. Compared to the previous configuration,

Table 4.5: Specific bias set that allowed to significantly reduce the noise.

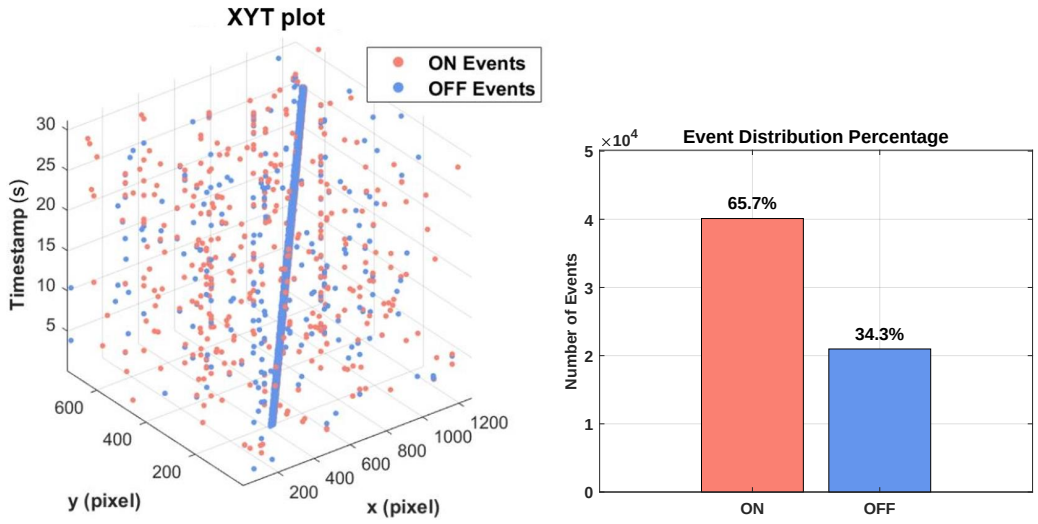
Bias name	**Value**	**Description**
`bias_diff`	0	Common reference level for ON/OFF thresholds.
`bias_diff_off`	60	Adjusts the sensitivity for OFF events (brightness decrease). Higher values reduce sensitivity to shadows.
`bias_diff_on`	0	Adjusts the sensitivity for ON events (brightness increase). Lower values make the camera more sensitive.
`bias_fo`	0	Low-pass filter (First Order). Controls the bandwidth of the pixel; higher values filter out high-frequency noise.
`bias_hpf`	50	High-pass filter. Helps eliminate very slow light changes and continuous background noise.
`bias_refr`	0	Refractory period. Minimum time between two consecutive events for the same pixel.

the event cloud is significantly more compact and concentrated along the object trajectory. This qualitative observation is reinforced by the ON and OFF events percentage distribution in Figure 4.9b, where a more balanced polarity ratio can be observed, indicating improved control over spurious OFF activations.

Further confirmation is provided by the exponentially decaying time surface illustrated in Figure 4.10. In this case, the background appears substantially cleaner, and the streak generated by the moving object is sharply defined. The reduction in residual activity prior to any algorithmic filtering highlights the effectiveness of acting directly at the sensor level. In practical terms, this means that the subsequent processing stages receive a higher SNR input, simplifying their task.

The feature detection parameters adopted for the tuned configuration are summarized in Table 4.7. Owing to the cleaner input stream, it was possible to employ a single threshold L for both polarities, unlike in the default case where asymmetric thresholds were necessary. Additionally, the processing time Δt_{feat} was reduced to 8 seconds, demonstrating the computational advantage of lowering the raw event rate at the source.

The resulting detections are presented in Figure 4.12. The temporal evolution of the x and y coordinates follows a nearly linear trend, consistent with the imposed uniform motion. The orientation estimates remain generally stable, and no significant outliers are visible. While the majority of events are concentrated around the predefined target angle (approximately 90°), which represents the chosen direction of the moving point-like object, a high degree of variability persists. Furthermore, a noticeable distribution of features appears slightly shifted above this expected orientation. This effect is probably due to the fact that the half-bar templates used to generate the LUT matrix, which is essential for calculating the value of θ_i, have remained unchanged from those described in section 3.3. Although this effect does not compromise the overall tracking performance, it deserves further investigation in



(a) (b)

Figure 4.9: (a) 3D XYT event stream (ON and OFF polarities) captured with default bias settings; (b) percentage distribution of ON and OFF events.

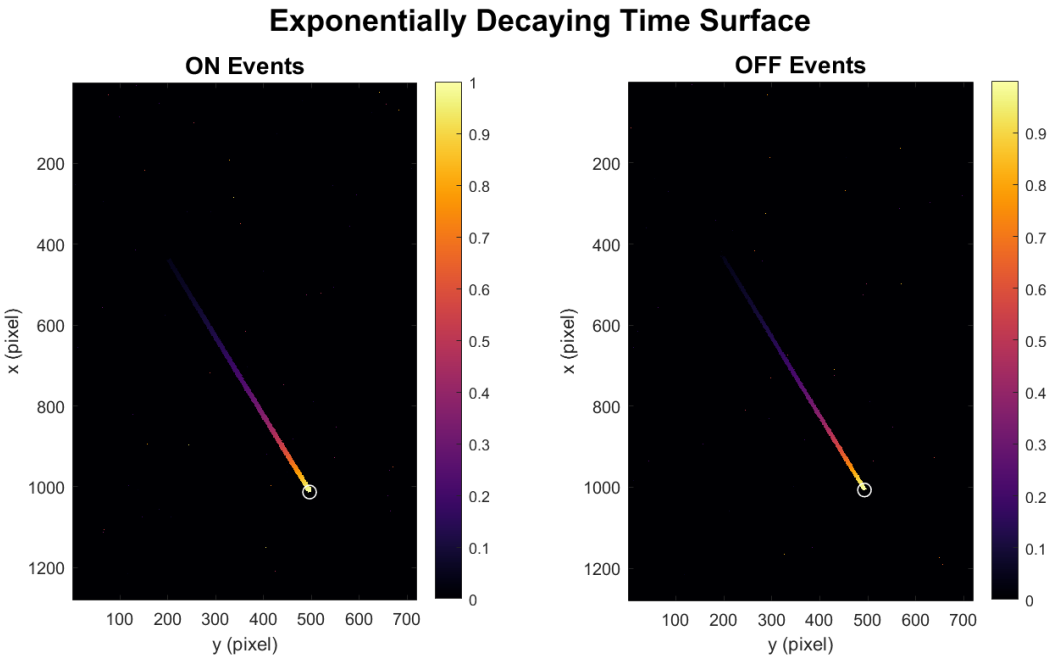


Figure 4.10: Exponentially Decaying Time Surface with the new set of bias currents reducing background activity at $t = 31.2\,\mathrm{s}$.

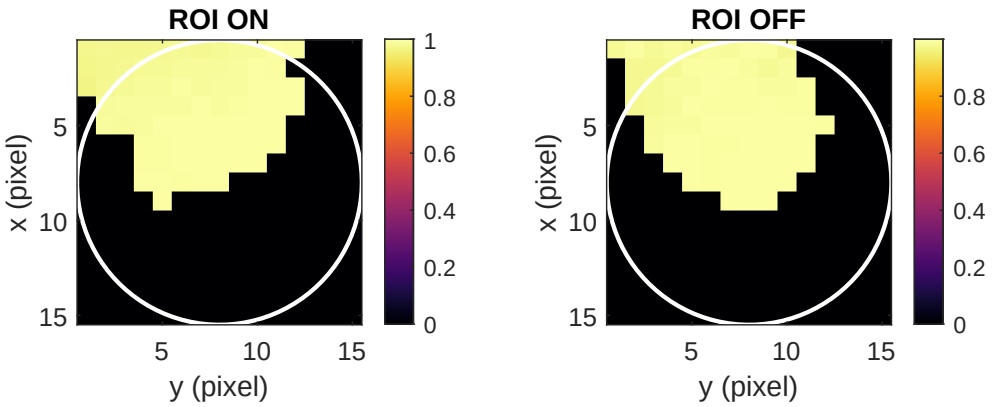


Figure 4.11: Regions of Interest ROI (Bias Tuning).

future work. Overall, the results confirm that the improved pre-processing conditions translate into more regular and reliable feature extraction. Above the whole number of 61 086 events, the strong selection led to remain with 7 138 ON events and 5 327 OFF events.
It is worth noting that the output of the feature detection in the tuned configuration and in the default case are practically identical. This demonstrates the robustness of the algorithm, as it is able to produce consistent feature estimates even when the input event rate and pre-processing conditions vary significantly.

Table 4.6: Feature Detection Parameters of the recording with bias tuning.

Parameter	**Value**	**Description**
τ	0.4 s	Decay Constant of the Time Surface
L	10	Number of active ROI pixels
ϕ	0.1 s	Decay Constant of Surface Activation Test
δ	20°	Tolerance of Angular Distance
Ψ_{stat}	7	Static Threshold
Δt_{feat}	8 s	Code Processing Time

The post-detection XYT visualization in Figure 4.13 further emphasizes the clarity of the refined event stream.

Tracking performance under the tuned configuration is illustrated in Figure 4.14. The detection counter confirms the presence of a single, consistently identified object. The estimated position and orientation evolve smoothly over time, without interruptions or spurious track initiations. The tracking parameters used for this stage are listed in Table 4.7, and notably, the processing time Δt_{track} is reduced to 8 seconds, further demonstrating the computational efficiency gained through bias tuning. Finally, Table 4.8 summarizes the numerical evolution of the dataset across the processing pipeline. Starting from only 61 086 raw events, more than three orders of magnitude fewer than in the default configuration, the system retains 12 346 events after tracking. This indicates that the filter does not significantly reduce the event throughput post-feature detection, functioning primarily as a tool

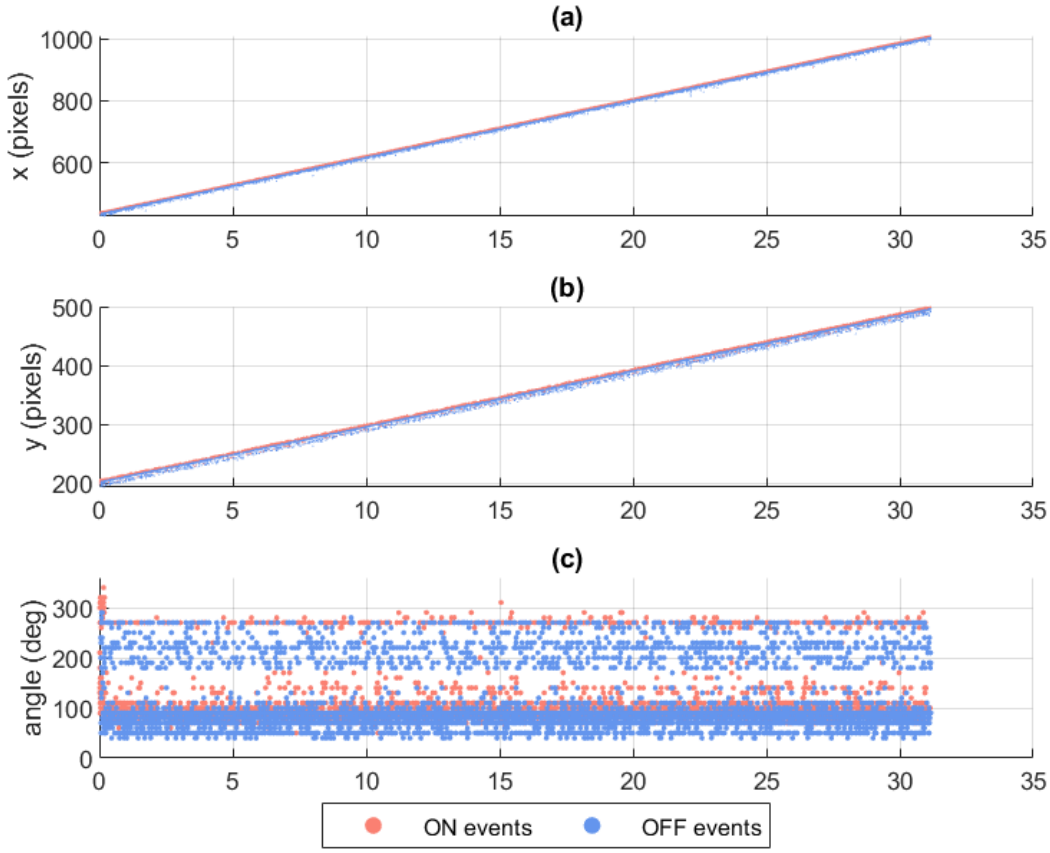


Figure 4.12: The panels show the x and y positions and orientation of the detection events over the time interval corresponding to the single streak produced by the object in uniform linear motion with the new set of bias currents reducing bias activity.

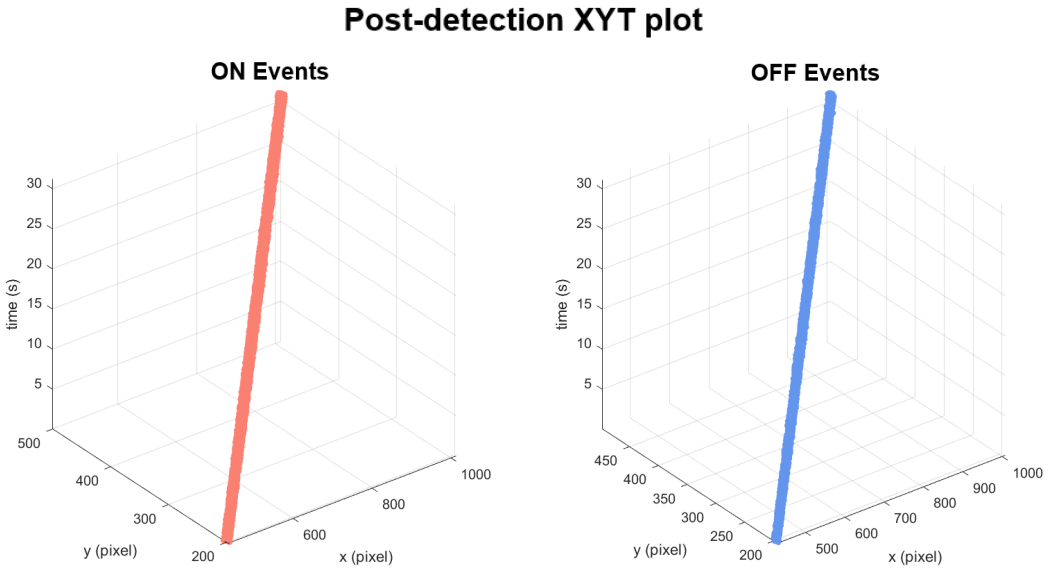


Figure 4.13: XYT visualization of the refined event stream (ON/OFF) post-feature detection, illustrating the reduction in noise events compared to raw data.

to distinguish between different objects rather than a selective event filter.

Table 4.7: Tracking Parameters of the recording with bias tuning.

Parameters	**Values**	**Name**
M_A	150	Activation Potential
γ	$100\,\mathrm{s}^{-1}$	Decay Constant for the Membrane Potential
d_{max}	70 pixel	Threshold Acceptable Distance
V	0.1 pixel/deg	Scaling Factor
K	15	Length of Fixed Rolling Window
Δt_{track}	8 s	Code Processing Time

Although manual bias tuning proved effective in improving pre-processing quality, it represents a time-consuming approach. An important direction for future work consists in developing automatic bias optimization strategies, capable of adapting the sensor configuration to the observed scene dynamics and illumination conditions without human intervention. Such an approach could further enhance the robustness and autonomy of event-based vision systems in real-world and spaceborne applications.

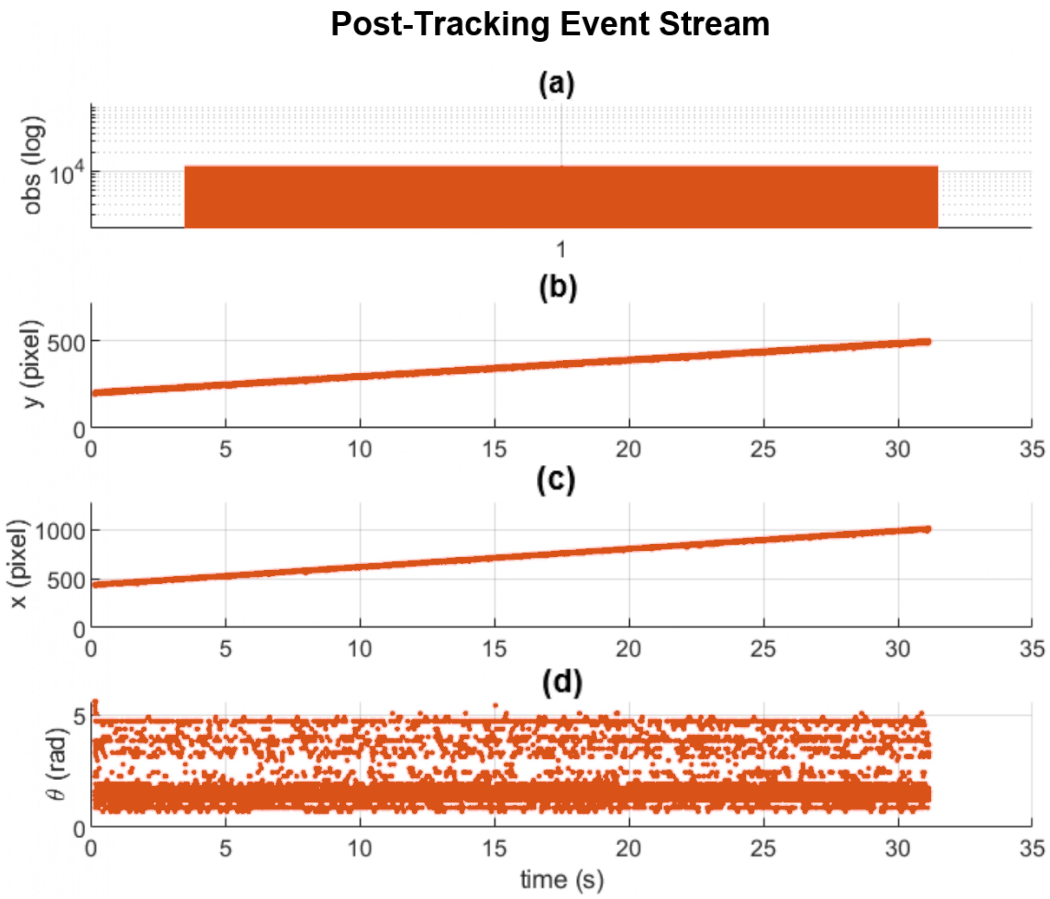


Figure 4.14: In (a) it can be seen the detection number of the only object present on the scene. The panels (b), (c) and (d) show the x and y positions and orientation of the detection events over the time interval corresponding to the single streak produced by the object in uniform linear motion with the new set of bias currents.

Table 4.8: Evolution of the total number of events through the event-based processing pipeline, regarding the recording with bias tuning.

Raw Events	**Post-Detection**	**Post-Tracking**
61 086	12 465	12 346

Conclusions

This work presented the development of an event-based visual tracking system, addressing both the algorithmic design and experimental setup. The results demonstrate that pre-processing stage plays a key role in transforming raw event streams into a more structured and readily usable representation. This significantly improves the robustness and reliability of the overall tracking pipeline, while also reducing the complexity required in the subsequent processing stages.

Having a more refined and informative output already at the pre-processing level opens up interesting directions for future research. In particular, this suggests the possibility of focusing on less computationally demanding tracking algorithms without compromising tracking performance. This approach is especially promising for real-time applications and deployment on resource-constrained systems, such as on-board spacecraft for autonomous collision avoidance, where computational efficiency is a critical requirement and where event-based sensors are increasingly considered as enabling technologies.

An additional consideration emerging from this study concerns the method proposed by Afshar et al. (2020). Although the application of their framework demonstrates competitive performance, it relies on a set of parameters that must be carefully tuned depending on the specific application scenario. Indeed, in our laboratory tests, many parameters of the algorithm have been changed compared to those used for the Afshar et al. (2020) work. This aspect may limit its adaptability and scalability, especially in operational contexts where prior knowledge of the environment is limited or where conditions can change dynamically. Future work could include a systematic sensitivity analysis of this approach, with the goal of identifying parameter configurations that generalize more effectively across different motion regimes and sensing conditions.

From an experimental standpoint, the current validation scenario was intentionally simplified: the projected object moved at constant velocity along a straight trajectory, emulating the apparent motion of a distant object observed in a space environment, such as a satellite or space debris traversing the field of view of an optical sensor. The use of a point-like target allowed the experiment to simulate the appearance of an RSO as seen from large distances and to focus primarily on motion-induced event generation, avoiding additional complexities related to object shape, size variation, rotation, or texture.

Although this choice was appropriate for isolating and analyzing the core behavior of the tracking pipeline, it also highlights an important direction for future research. A natural extension of the present work would be to progressively increase the level of difficulty in the simulations and experimental setups. For example, future scenarios

could include non-uniform motion profiles (with accelerated or maneuvering targets), curved trajectories, varying apparent velocities, or multiple objects simultaneously present in the scene. Furthermore, introducing extended targets with realistic shapes, tumbling dynamics, or partial occlusions would provide a more comprehensive assessment of the system's robustness under conditions that more closely resemble real on-orbit operations.

In conclusion, the presented work demonstrates that investing effort in a well-designed event stream pre-processing stage can substantially simplify the overall tracking architecture while maintaining performance. Future developments aimed at expanding the range of simulated conditions and at analyzing parameter robustness will further consolidate the applicability of this approach in real-world scenarios, particularly in space-based autonomous systems.

Appendix

This appendix provides an example of a visual representation of the raw event stream captured during the experimental phase. Figure 4.15 displays a self-portrait (subject visualization) acquired using the Prophesee Metavision Studio software.

Figure 4.15: Visualization of raw event data (accumulation window) processed via Metavision Studio.

The visualization was generated by accumulating events within a temporal window of 85 ms. This integration time was selected to balance the density of the event cloud with the preservation of structural edges, effectively demonstrating the camera's ability to map high-contrast features in the scene through asynchronous pixel activation. The image was captured after the application of the bias parameters tuning phase discussed in subsection 4.3.2.

Bibliography

Afshar, S., Hamilton, T. J., Tapson, J., Van Schaik, A., and Cohen, G. (2019). Investigation of event-based surfaces for high-speed detection, unsupervised feature extraction, and object recognition. *Frontiers in neuroscience*, 12:1047.

Afshar, S., Ralph, N., Xu, Y., Tapson, J., Schaik, A. v., and Cohen, G. (2020). Event-based feature extraction using adaptive selection thresholds. *Sensors*, 20(6):1600.

Aliste, V. W., Vasquez, S. V., and Rojas, E. V. (2023). Analysis of detection limits in event-based cameras for space situational awareness.

Alzugaray, I. and Chli, M. (2018). Asynchronous corner detection and tracking for event cameras in real time. *IEEE Robotics and Automation Letters*, 3(4):3177–3184.

Alzugaray, I. and Chli, M. (2020). 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.

Argirò, A., Isoletta, G., Opromolla, R., and Fasano, G. (2025). Triangulation of space-based optical measurements for improved space surveillance and tracking. In *Proceedings of the 9th European Conference on Space Debris*, Bonn, Germany. European Space Agency. ESA Space Debris Conference.

Bagchi, S. and Chin, T.-J. (2020). Event-based star tracking via multiresolution progressive hough transforms. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2143–2152.

Baldwin, R. W., Liu, R., Almatrafi, M., Asari, V., and Hirakawa, K. (2022). Time-ordered recent event (tore) volumes for event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2519–2532.

Bar-Shalom, Y., Daum, F., and Huang, J. (2009). The probabilistic data association filter. *IEEE Control Systems Magazine*, 29(6):82–100.

Bar-Shalom, Y. and Tse, E. (1975). Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460.

Battista, U., Landini, A., Gołębiowski, W., Michalczyk, R., Czerwiński, A., Duda, K., and Sochaczewska, A. (2017). Design of net ejector for space debris capturing. In *7th European conference on space debris, Published by the ESA Space Debris office, Germany*.

Benosman, R., Clercq, C., Lagorce, X., Ieng, S.-H., and Bartolozzi, C. (2013). Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25(2):407–417.

Bi, Y., Chadha, A., Abbas, A., Bourtsoulatze, E., and Andreopoulos, Y. (2020). Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Transactions on Image Processing*, 29:9084–9098.

Blackman, S. S. (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18.

Blackman, S. S. and Popoli, R. (1999). Design and analysis of modern tracking systems. *(No Title)*.

Chen, G., Chen, W., Yang, Q., Xu, Z., Yang, L., Conradt, J., and Knoll, A. (2020). A novel visible light positioning system with event-based neuromorphic vision sensor. *IEEE Sensors Journal*, 20(17):10211–10219.

Cheung, B., Rutten, M., Davey, S., and Cohen, G. (2018). Probabilistic multi hypothesis tracker for an event based sensor. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE.

Chin, T.-J., Bagchi, S., Eriksson, A., and Van Schaik, A. (2019). Star tracking using an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0.

Clady, X., Ieng, S.-H., and Benosman, R. (2015). Asynchronous event-based corner detection and matching. *Neural Networks*, 66:91–106.

Cohen, G., Afshar, S., Morreale, B., Bessell, T., Wabnitz, A., Rutten, M., and Van Schaik, A. (2019). Event-based sensing for space situational awareness. *The Journal of the Astronautical Sciences*, 66(2):125–141.

Cohen, G., Afshar, S., and Van Schaik, A. (2018). Approaches for astrometry using event-based sensors. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, volume 1, page 2.

Coretti, A. G., Varile, M., and Bertaina, M. E. (2025). An efficient neuromorphic approach for collision avoidance combining stack-cnn with event cameras. *arXiv preprint arXiv:2506.16436*.

Czech, D. and Orchard, G. (2016). Evaluating noise filtering for event-based asynchronous change detection image sensors. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 19–24. IEEE.

European Space Agency (2025). Esa space environment report 2025.

European Union Agency for the Space Programme (2025). Ssa.

Foster, B. J., Ye, D. H., and Bouman, C. A. (2019). Multi-target tracking with an event-based vision sensor and a partial-update gmphd filter. *Electronic Imaging*, 31:1–7.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., et al. (2020). Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180.

Kessler, D. J. and Cour-Palais, B. G. (1978). Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637–2646.

Lagorce, X., Ieng, S.-H., Clady, X., Pfeiffer, M., and Benosman, R. B. (2015). Spatiotemporal features for asynchronous event-based data. *Frontiers in neuroscience*, 9:46.

Leclerc, M., Tharmarasa, R., Florea, M., Boury-Brisset, A., Kirubarajan, T., and Duclos-Hindie, N. (2018). 2018 21st international conference on information fusion (fusion).

Liu, S.-C., Delbruck, T., Indiveri, G., Whatley, A., and Douglas, R. (2014). *Event-based neuromorphic systems*. John Wiley & Sons.

McReynolds, B., Graca, R., Oliver, R., Nishiguchi, M., and Delbruck, T. (2023). Demystifying event-based sensor biasing to optimize signal to noise for space domain awareness. In *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*. University of Zurich.

Oliver, R., McReynolds, B., and Savransky, D. (2025). Event-based sensor noise modeling for space-based space domain awareness. *The Journal of the Astronautical Sciences*, 72(5):46.

Peng, Y., Chen, C., and Huang, G. (2024). 2024 ieee international conference on robotics and automation (icra).

Purohit, P. and Manohar, R. (2022). Field-programmable encoding for address-event representation. *Frontiers in Neuroscience*, 16:1018166.

Ralph, N., Joubert, D., Jolley, A., Afshar, S., Tothill, N., Van Schaik, A., and Cohen, G. (2022). Real-time event-based unsupervised feature consolidation and tracking for space situational awareness. *Frontiers in neuroscience*, 16:821157.

Ralph, N., Maybour, D., Bethi, Y., and Cohen, G. (2019). Observations and design of a new neuromorphic event-based all-sky and fixed region imaging system. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 71.

Roffe, S., Akolkar, H., George, A. D., Linares-Barranco, B., and Benosman, R. B. (2021). Neutron-induced, single-event effects on neuromorphic event-based vision sensor: A first step and tools to space applications. *IEEE Access*, 9:85748–85763.

Shon, M. (2019). 2019 22th internatio-nal conference on information fusion (fusion). *(No Title)*.

Space Academy (2025). Space situational awareness (ssa).

Streit, R. L. and Luginbuhl, T. E. (1994). Maximum likelihood method for probabilistic multihypothesis tracking. In *Signal and data processing of small targets 1994*, volume 2235, pages 394–405. SPIE.

The Aerospace Corporation (2025). Space debris 101.

Xu, C., Zhou, H., Chen, L., Chen, H., Zhou, Y., Chung, V., Qu, Q., and Cai, W. (2025). A survey of 3d reconstruction with event cameras.

Zołnowski, M., Reszelewski, R., Moeys, D. P., Delbruck, T., and Kamiński, K. (2019). Observational evaluation of event cameras performance in optical space surveillance. In *NEO and Debris Detection Conference, Darmstadt, Germany*.