

# Characterization of Event-Camera Biases and Operating Conditions for High-Speed Sorting

Paul Bäcker<sup>1,2</sup>; Johannes Baßler<sup>2</sup>; Georg Maier<sup>1</sup>; Thomas Längle<sup>1</sup>; Michael Heizmann<sup>2</sup> 

<sup>1</sup>Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), Karlsruhe

<sup>2</sup>Karlsruhe Institute of Technology, Institute of Industrial Information Technology (IIT), Karlsruhe

## Abstract

Predictive multi-object tracking offers significant potential to improve separation accuracy in sensor-based sorting (SBS) systems. However, conventional frame-based vision pipelines are limited by exposure time, frame rate, and end-to-end latency, which reduces localization precision and actuation performance at high material throughputs. Event cameras offer an attractive alternative due to their asynchronous operation, low latency, and high temporal resolution. Yet, their performance strongly depends on sensor parameters (biases), illumination, and optical setup, and selecting suitable operating points remains a challenging multi-objective problem. In this work, we present a task-driven framework for characterizing event-camera operating points for high-speed sorting. Using a controlled rolling-sphere experiment with known geometry and an LED-based timing reference, we evaluate the influence of bias settings, illumination, and aperture on task-relevant metrics including geometric consistency, event statistics, background activity, hot-pixel behavior, and trigger-aligned latency. We validate the stability of the proposed metrics through repetition experiments and use one-dimensional parameter sweeps to reveal distinct sensitivities and trade-offs. Based on the results, we provide guidelines for selecting suitable operating points. The presented methodology provides a basis for principled manual tuning and future automated operating-point selection in event-based sorting and related low-latency vision applications.

**Keywords:** event cameras, bias tuning, task-driven evaluation, latency, event statistics, geometric consistency

## 1 Introduction

Sensor-based sorting (SBS) is widely used in industrial domains such as recycling and food processing [1]. To improve separation accuracy, predictive multi-object tracking has been proposed to compensate uncertain object motion between detection and actuation [2]. Classical frame-based vision pipelines are however limited by exposure time, frame rate, and end-to-end latency, which can degrade localization and actuation accuracy at high speeds. Event-based vision offers an alternative by reporting per-pixel brightness changes asynchronously with low latency and high temporal resolution [3, 7]. However, the event-based sensing paradigm also introduces specific challenges, that can diminish the benefits of low latency and high temporal resolution if not properly addressed:

**The event rate** is inherently scene dependent, with larger numbers of objects creating more events. Most asynchronous event-based tracking methods perform per-object event assignment resulting in undesirable scaling of computational complexity [8]. Even if the computational load can be handled downstream, the event-based sensor and transmission pipeline is limited by **maximum data throughput**, which can be exceeded under strong global mo-

tion and high event rates [11].

**Latency** is defined as the temporal difference between the time when an intensity change happens in the scene, and when it is detected by the sensor. It is dependent on the sensor operating point and illumination levels [13]. Ideally, this should be as low as possible to extract accurate object positions for precise actuation.

**Shape fidelity** - We want events precisely on the object boundaries in order to capture the true shape. This includes statistical error (jitter) and systematic error (varying latency between edges), which can distort the object shape and make it harder to maintain correspondences across tracking steps.

We address those challenges by investigating sensor settings in combination with different illumination and aperture conditions. We present a controlled benchmark for event cameras using a known shape prior and an LED-based latency measurement. We introduce a task-driven metric set that combines geometric, temporal, and event-statistical quality measures, and we perform bias sweeps to reveal distinct sensitivities, trade-offs, and practical operating regions across hardware settings and sensor parameters (biases). Finally, we show how these metrics support operating-point selection and can serve as a basis for later automated tuning.

This work was supported by the German Federal Ministry of Research, Technology and Space - Grant number 03DPS1146A

## 2 Related Work

Event cameras asynchronously report brightness changes once the change in log-intensity exceeds a polarity-dependent threshold. Foundational work on event cameras describes the pixel-level signal formation process and the asynchronous threshold-crossing principle underlying dynamic vision sensors [3]. Later characterization work measures how observable quantities such as contrast sensitivity, temporal behavior, and latency depend on the sensor and illumination conditions [12, 13].

Multiple methods address the challenges of event-based vision for tracking applications on an application level. For handling the high number of events and global motion, frame-based processing of event-data has been proposed [10, 5]. Further, event-filters can reduce the number of events, based on burst- or background suppression [5], or using a-priori global motion assumptions [9]. Although these methods reduce the computational load of the tracking algorithm, they do not change the amount of events generated at the sensor level, leading to potentially excessive data transfer and processing load, which can result in the loss of events [11].

Other approaches have altered the sensor design, enabling motion specific filtering at the sensor level [4, 6]. However, these methods have, to the best of our knowledge, not been transferred into commercially available sensors, and therefore do not yet offer a practical solution for the SBS setting. Our approach in this work is instead to optimize the sensing conditions, namely illumination, aperture, and sensor parameters. These tunable sensor parameters correspond to bias currents in the sensor pixel circuitry and are therefore referred to as **biases**.

Prior work on bias tuning ranges from empirical characterization and qualitative bias interpretation to benchmark-driven and task-driven optimization. Experimental studies investigate how bias settings affect measurable sensor properties and downstream signal quality [12, 14]. Generic benchmarking work proposes reusable tasks and metrics across different applications [15]. Other approaches adapt biases online through feedback control or black-box optimization for specific downstream tasks such as detection, place recognition, or face tracking [16, 17]. These works show that tuning is important and often task-dependent, but they do not take environmental factors and aperture settings into account. Further, we extend the task-driven evaluation by including geometric consistency and event statistics in addition to latency.

## 3 Background

The event-generation process is influenced by both the optical setup and several sensor biases. The optical setup determines the sharpness of the image projected onto the sensor. A small aperture in-

creases depth of field and yields sharper edges, but reduces signal strength; a large aperture increases signal strength at the cost of reduced depth of field and increased blur. Blurred edges spread intensity changes over time, which leads to uncontrollable jitter in the timing of the first event, as the event-pixel state (e.g. the current reference intensity) is to be assumed unknown. Further, inherent inhomogeneities in terms of contrast threshold, which have been reported to be as high as 30 % [3] lead to timing offsets, if the object edge is not sufficiently sharp. The pixel response can be approximated by cascaded temporal filters whose bandwidth depends on illumination and bias settings [3]. At low illumination, the photoreceptor response is slower and more strongly limited by photon flux, while at high illumination the bandwidth of the second-order low-pass filter of the feedback amplifier circuit becomes more relevant. These filters affect latency, noise sensitivity, and overshoot in the temporal response.

The biases control event generation: increasing `bias_diff_on` or `bias_diff_off` raises the ON or OFF contrast threshold, respectively, requiring stronger intensity changes to trigger an event. Increasing `bias_fo` increases high-frequency bandwidth and can reduce latency at the cost of noise; decreasing `bias_hpf` increases sensitivity to slower intensity changes.

To allow for pixel stabilization after an event, pixels are idle for a fixed time after it was triggered, ignoring any intensity changes. This time can be controlled by adjusting the `bias_refr`, where increasing `bias_refr` shortens the refractory period between successive events.

Together, optics and biases determine the trade-off between event rate, latency, noise, and geometric fidelity, motivating the task-driven characterization proposed in this work.

## 4 Experimental Setup

The experimental setup is shown in Fig. 1. Wooden spheres roll down a chute through the field of view of a fixed event camera, while a diffuse, uniform background illumination provides the dominant intensity contrast at the sphere boundary. From the measured traversal time of around 0.12 s for the 21 cm vertical extent, the object speed is approximately  $1.7 \text{ m s}^{-1}$ , i.e., about  $5853 \text{ px s}^{-1}$  ( $5.8 \text{ px ms}^{-1}$ ) in image space. Spheres are used because their silhouette remains circular for all viewing angles; deviations from circularity in the reconstructed event contour can therefore be attributed to sensing effects rather than object geometry.

We use an IDS uEye XCP-E with a Sony IMX636 event sensor (resolution  $1280 \times 720$  pixels). To study the interaction between optics and sensor operating point, we vary background illumination (high/low) and aperture ( $f/1.4$ ,  $f/6$ ,  $f/16$ ). For each configuration, irradiance at the sensor was recorded using the

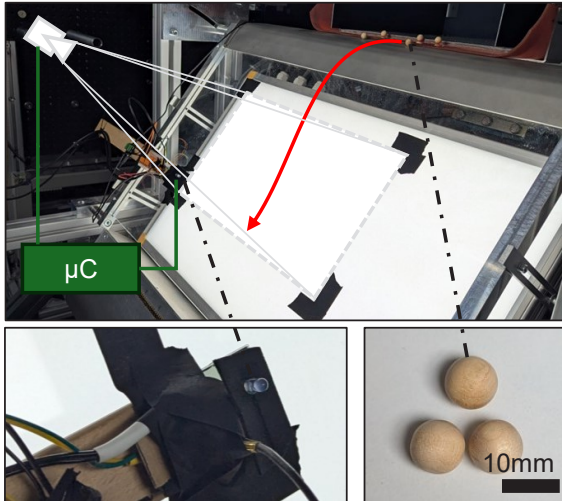


Figure 1: Experimental setup with chute, event camera, illumination, and LED trigger

Table 1: Measured sensor illumination (S. Illum.) for different background illumination and aperture settings.

| ID         | BG illum. | Aperture | S. Illum. (lux) |
|------------|-----------|----------|-----------------|
| High-f/1.4 | high      | f/1.4    | 11185.27        |
| Low-f/1.4  | low       | f/1.4    | 3026.36         |
| High-f/6   | high      | f/6      | 894.59          |
| Low-f/6    | low       | f/6      | 236.09          |
| High-f/16  | high      | f/16     | 95.00           |
| Low-f/16   | low       | f/16     | 25.00           |

on-sensor photodiode (Tab. 1). We identified five hot pixels at default biases, that we mask out for all recordings to avoid their influence on the metrics. A timing reference is provided by a LED that is visible in the camera image and is simultaneously connected to the camera trigger input (Fig. 1). This yields a shared external reference for the onset of an intensity change and enables direct estimation of event latency, which is later evaluated via the trigger-aligned delay distributions (Fig. 5).

## 5 Metric Extraction Pipeline

Fig. 2 shows the full evaluation pipeline and the per-trajectory metric extraction with two example outputs for different bias settings. The goal is to extract a set of meaningful, independent scalar metrics for each recording that can be used for bias-sweep interpretation and operating-point selection. All metrics are interpreted as error quantities, such that lower values indicate better performance. For each bias setting, we record five spheres rolling down the chute, yielding an event-cube and timestamped trigger signals (actual timing of LED activation). All events in a region of interest (ROI) around the LED are extracted and used exclusively for the **Trigger Analysis** branch, which estimates the mean ON

and OFF latency relative to the trigger signal. We automatically find the center of the LED using a heatmap of the global event activity. We consider a  $5 \times 5$  pixel region around the LED center for the analysis. For each of these pixels, we extract the first positive event after each detected ON-trigger and the first negative event after each detected OFF-trigger. We repeatedly switch the LED on- and off every 10 ms. The figure shows a histogram of the extracted latencies for the two example settings, revealing a clear ON/OFF asymmetry for the setting in the top row, and a more symmetric behavior as well as much lower latencies for the setting in the bottom row. As a numerical per-sample metric  $\text{Latency}_{\text{pos/neg}}$ , we compute the means over all measured latencies.

The remaining events are assigned to individual objects using the Metavision Spatter **Tracker** [18]. For each detected object, the tracker yields a sequence of timestamped bounding boxes together with the corresponding object-center coordinates. These trajectory samples are converted into a structured representation and smoothed by fitting splines to the horizontal and vertical center position over time. Event assignment is then not performed by a single static crop, but by a moving ROI that follows the tracked object. Concretely, for each interval between two tracker timestamps, we retain only the events inside a bounding box enlarged by 30% around the tracked object. This produces trajectory-specific event sets that serve as the basis for the later per-object metrics.

For the **Events per Object** analysis, we sample the tracked centerline along the vertical image coordinate and identify, for each sampled row, the pixel that lies on the spline-predicted object center. At these trajectory pixels, we count the number of negative and positive events separately. This yields a compact per-trajectory measure of how often the sensor fires while the object center passes a given image location and deliberately suppresses edge-related effects away from the centerline. Across trajectories, we then aggregate the mean number of positive and negative event counts per pixel to extract the  $\# \text{Events}_{\text{pos/neg}}$  metric.

In the geometric analysis branch, the per-trajectory events are first filtered to keep only the first negative and last positive event per pixel (**First/Last Filter**). This is based on the strong assumption, that the background is much brighter than the objects, so a pixel will transition from bright to dark to bright again as the object passes, leading to a negative event at the entry edge and a positive event at the exit edge. We want this analysis to be orthogonal to the event statistics, so we deliberately focus on the first and last events, which are more likely to correspond to the physical entry and exit edges of the sphere, rather than texture-related events that can

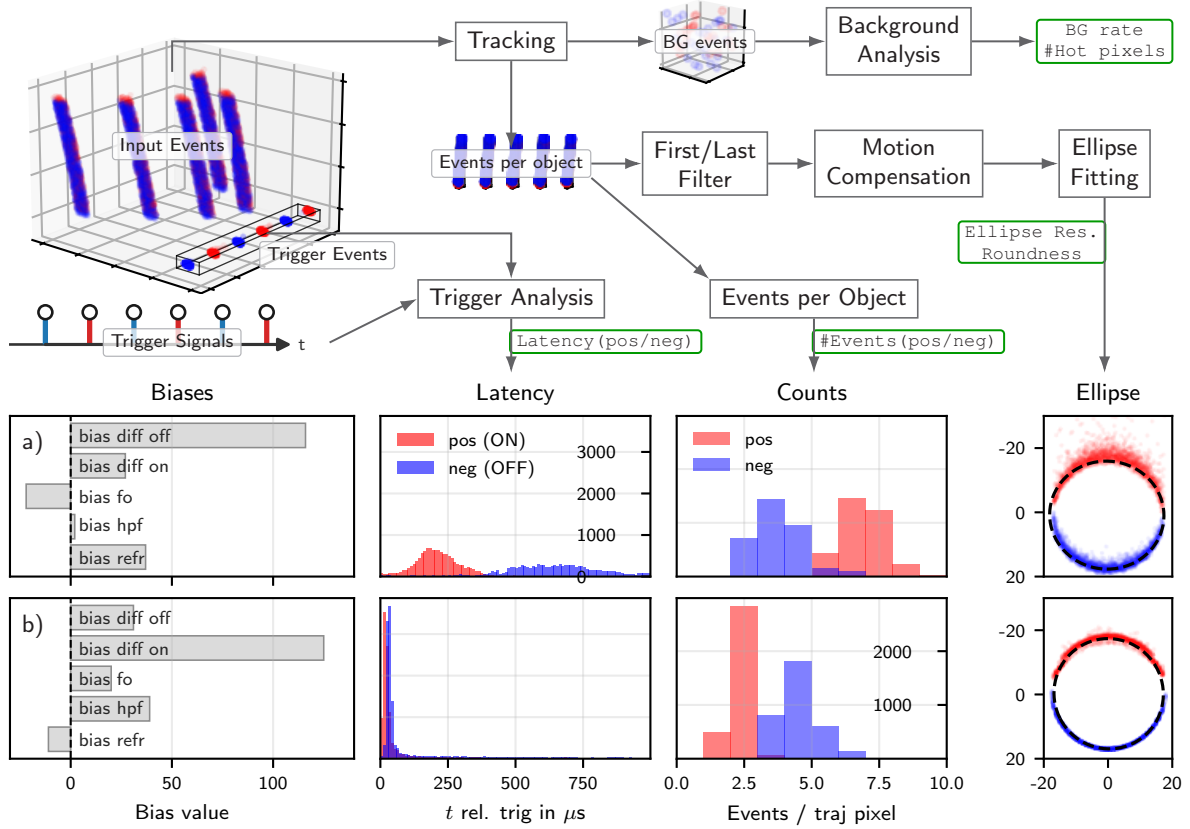


Figure 2: Full evaluation pipeline and per-trajectory metric extraction. The figure illustrates two representative bias settings a) (randomly selected) and b) (empirically tuned). The green boxes show the extracted, single scalar metrics per measurement.

occur inside the object region. For this, the implementation determines the first/last events per pixel, and ignores all other events. Then, for three evenly spaced reference positions at 25%, 50%, and 75% of each trajectory length, a **Motion Compensation** step is applied. For geometric analysis, the temporal window around a certain time reference is crucial, as larger windows introduce motion blur, but too small windows can lead to too few events for a reliable analysis. We therefore use large time windows, but compensate for the motion of the object within these windows to avoid blur. Around each reference time, we extract a temporal event window (10 ms) and fit a second-order polynomial to the event coordinates as a function of time. The resulting offsets are subtracted from the raw event positions so that the correlation between position and timestamp is minimized and the event cloud collapses toward a common spatial frame. This low-order model is consistent with the smooth, constant acceleration motion expected for spheres rolling down the chute. Finally, an **Ellipse Fit** is applied to the aligned event cloud, which yields a fitted ellipse. For a perfect circular object with perfectly aligned events, the fitted

ellipse, major ( $MA$ ) and minor ( $ma$ ) axes should be equal. By taking the mean over the inverse roundness ( $1 - (MA/ma)$ ) for each of the  $3 \cdot 5$  fitted ellipses we extract the `Roundness` metric for each measurement. The ellipse fit also yields a residual, which measures the average distance of the aligned events to the fitted ellipse. This residual is a measure of the spatial precision of the aligned event cloud and can be sensitive to blur or jitter in the event generation process. The mean over the residuals for all three fitted ellipses constitutes the `Ellipse Res.` metric.

Events without any present objects in the scene are used for **Background Analysis**. All events in these windows are accumulated into a per-pixel count map. Pixels with abnormally high activity are classified as hot pixels (`#Hot pixels` metric) and excluded from the final rate estimate. These are pixels whose activity exceeds ten times the median per-pixel rate and that also accumulate more than 50 events within the background window. The remaining background events are used for estimating the background event rate `BG rate` by dividing the total number of remaining events by the number of

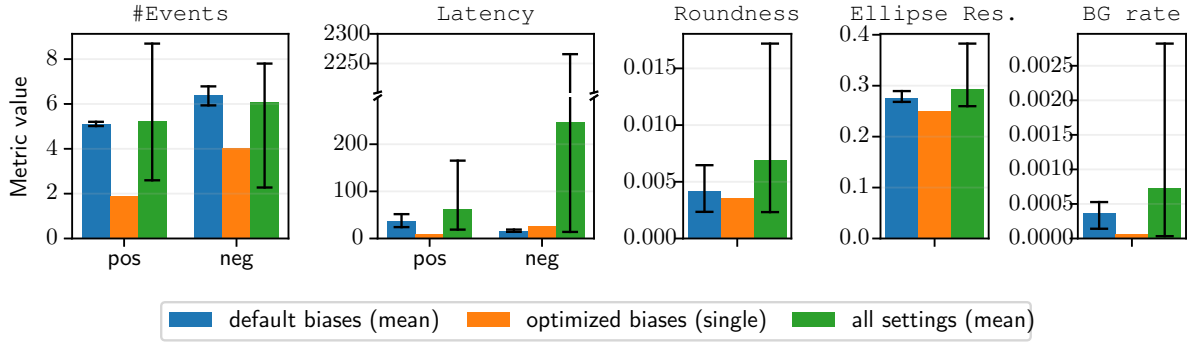


Figure 3: Metric fidelity for the High-f/6 hardware setting. For the default biases, minimum and maximum values of the 5 repeated experiments are shown as whiskers. For the variation across all bias settings, 5- and 95% quantile are shown. As a further reference, metric values for the empirically tuned operating point are also shown.

pixels and the total time across all timespans.

## 6 Bias Sweep Analysis

For each hardware configuration (illumination and aperture), we independently sweep each sensor bias while keeping all remaining biases at their default values. For every measurement, we extract the full metric set described in Sec. 5. We assess metric robustness, compare hardware regimes, and identify dominant sensitivities and trade-offs of individual parameters.

### 6.1 Metric Fidelity

Our measurement setup is subject to both random and systematic disturbances, including variations in rolling-sphere trajectories, illumination fluctuations, and sensor fixed-pattern noise. These disturbances are not a limitation of the proposed method, but reflect the variability expected in an online characterization and tuning scenario. Consequently, the proposed metrics must be robust enough to reveal meaningful and repeatable trends under realistic operating conditions, while remaining sensitive to actual changes in sensor settings.

To assess metric robustness, we repeat the identical experiment—five objects recorded at default bias settings—five times in succession and compare the resulting variation to the range observed across all bias settings. Fig. 3 shows the variation across repeated measurements relative to the variation induced by the bias sweeps.

The `#Events` (`pos/neg`) metric exhibits relatively low noise, with a clear separation between the default setting, the selected operating point, and the overall range across bias settings. The `Latency` (`pos/neg`) metrics show moderate variation, but the default setting remains clearly separated from the selected operating point. Moreover, the absolute magnitude of the noise is small, on the order of tens of  $\mu\text{s}$ , compared to the full range across bias settings.

The `Roundness` score exhibits the highest relative variation. Nevertheless, it remains useful for detecting large degradations in geometric consistency, which occur for a substantial fraction of the tested bias settings. The `Ellipse residual` shows very little variation across repeated measurements. No hot pixels were observed under the default bias settings; consequently, the `#Hot pixels` metric shows no variation across repeated measurements. The `BG rate` exhibits comparatively strong variation, likely due to changes in the static background scene and ambient illumination. Despite this, the observed differences remain significant relative to the selected operating point.

### 6.2 Hardware-Level Comparison

We compare different hardware regimes (background illumination and aperture) at a global level. For each recording in a given hardware configuration (as presented in Tab. 1), we extract the aforementioned metrics using our pipeline. Missing metric values (usually for extreme bias settings) are removed for all hardware settings to remain consistent. We then summarize each error component by taking the median across all available measurements in that hardware configuration, yielding a compact per-hardware “typical error profile”. The median is used to reduce the influence of outliers and to capture the typical behavior of each hardware setting across the various bias sweeps.

To make components comparable across different scales, each metric component is normalized across hardware settings using a per-metric  $z$ -score. Fig. 4 shows these normalized median error components. The plot highlights how the overall operating regime shifts with optics: low light increases noise-related and timing-related penalties, whereas very open apertures can degrade geometry-related components and lead to more events overall. Intermediate settings provide the most balanced trade-off across the task-driven metrics and therefore form the most

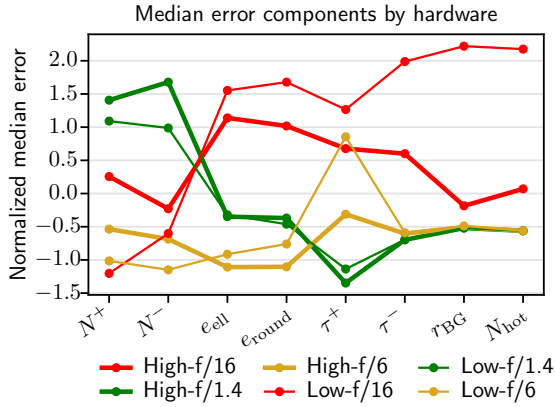


Figure 4: Hardware-level comparison across illumination and aperture settings. Each curve shows the normalized, per-metric  $z$ -score median error components aggregated over all measurements within a hardware configuration. Abbreviations:  $N^+/N^-$  are positive/negative event counts;  $e_{\text{ell}}$  is the ellipse residual;  $e_{\text{round}}$  is the roundness error;  $\tau^+/\tau^-$  are positive/negative trigger delays;  $r_{\text{BG}}$  is the background rate; and  $N_{\text{hot}}$  is the number of hot pixels.

promising region for subsequent bias-sweep interpretation and operating-point selection.

### 6.3 Per-Bias Analysis

The one-dimensional sweeps reveal interpretable sensitivities of individual metrics to specific control parameters and expose practical trade-offs for operating-point selection. We highlight three particularly interesting effects in the following.

#### 6.3.1 Latency and ON/OFF asymmetry

We quantify latency using the LED trigger reference and evaluate ON and OFF responses separately. Fig. 5 compares representative contrast-threshold sweeps for the High-f/1.4 setting. Increasing `bias_diff_on` leads to a comparatively smooth and controllable increase in ON-related delays while leaving the overall behavior stable, which makes it a useful lever for reducing event load without strongly destabilizing timing. In contrast, increasing `bias_diff_off` induces a much stronger latency penalty and a pronounced ON/OFF asymmetry; at higher `bias_diff_off` values, the positive-event delays also increase, indicating that OFF-threshold tuning can affect the timing behavior more broadly than expected from a polarity-isolated view. For comparable event counts, large `bias_diff_off` settings produce substantially higher delays and are accompanied by a clear degradation in the roundness metric, consistent with systematically biased edge timing. Similar sensitivity of OFF-related bias parameters has been reported in benchmark-driven bias tuning studies [15].

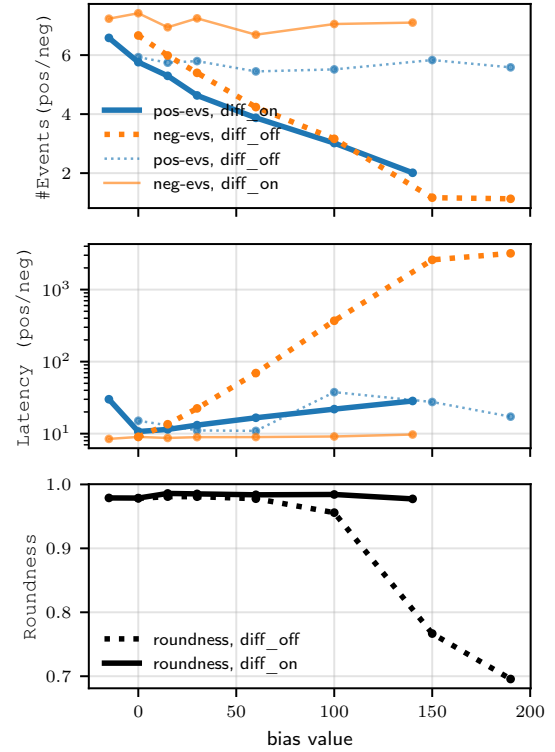


Figure 5: Contrast threshold sweeps for High-f/1.4 setting. The range of `bias_diff_on` is smaller than that of `bias_diff_off`.

#### 6.3.2 Noise and Hot-pixel behavior

Beyond geometry and latency, sensor settings must also be judged by their tendency to generate *background activity* and *hot pixels*. Such pixels consume bandwidth without contributing task-relevant edge information and can degrade downstream tracking by injecting spurious events.

The sweeps in Fig. 6 show that hot-pixel behavior is strongly bias- and optics-dependent. The trends for `bias_diff_on` and `bias_fo` are qualitatively consistent with the expected noise-bandwidth trade-off: Especially at low illumination levels, shot noise has strong influence. This is amplified, if no low-pass filtering is applied (low `bias_fo`). Low contrast thresholds then lead to a large number of events being triggered (each time the noise exceeds the threshold).

Surprisingly, the sweep over `bias_hpf` exhibits a steep increase in the hot-pixel count. The manufacturer documentation suggests that increasing `bias_hpf` should make the sensor less sensitive to slow light changes, therefore decreasing bandwidth on the lower frequency end, but not influence high-frequency changes. Most of this activity is attributed to events with negative polarity.

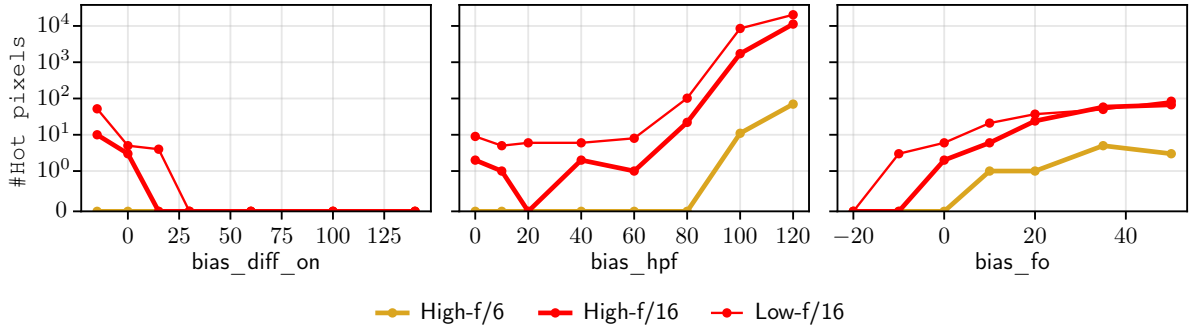


Figure 6: Hot-pixel behavior across bias sweeps and hardware settings. The plot shows the number of hot pixels for each bias setting, revealing strong sensitivities to certain parameters, particularly `bias_hpf`

### 6.3.3 Geometry versus refractory-period trade-off

The refractory period enforces a dead time where the pixel is idle after an event. This can lead to a suppression of event-bursts and is therefore an effective lever in reducing event rate. The upper panel of Fig. 7 shows this effect.

However, an extensive refractory period also means that if a pixel fires early due to texture induced intensity changes on the object, it can be inactive for the actual object boundary, which can lead to gaps in the event cloud and bias in the recovered contour. This is reflected in the roundness metric, which shows a clear deterioration for long refractory times and further can be qualitatively observed in the motion-compensated event clouds and fitted ellipses in the lower panels of Fig. 7.

## 7 Selection of Operating Points

The hardware comparison and bias sweeps provide practical guidance for selecting a suitable operating point without requiring black-box optimization. For the present sorting setup, the High-f/6 configuration offers a useful compromise between noise, and optical blur: compared to very small apertures it provides sufficient signal, while avoiding the stronger blur observed for very open apertures.

Based on the sweep results, we select an operating point with increased contrast thresholds (`bias_diff_on` and `bias_diff_off`) to reduce the event rate, while tuning the negative threshold more conservatively due to its instability at high bias values. The refractory bias `bias_refr` is adjusted to suppress event bursts without introducing visible gaps or degrading the fitted object shape. Finally, a higher `bias_fo` is used to improve temporal response; we found, that under the selected higher contrast thresholds, this does not lead to excessive background activity. The resulting operating point, shown in Fig. 2b), yields lower event multiplicity and latency while preserving the circular event contour. Its effect on the metrics

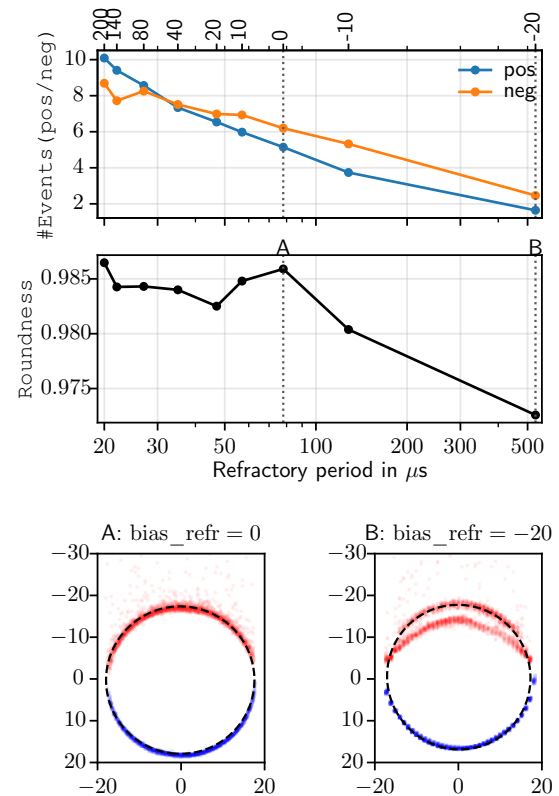


Figure 7: Refractory-period trade-off for the `bias_refr` sweep (at High-f/6). The tick marks on top of the upper axis indicate the numerical bias settings associated with the effective refractory times (lower axis labels). Bottom: two representative motion-compensated event clouds with fitted ellipse overlay, illustrating how overly long refractory times can create gaps and bias the recovered contour while reducing event throughput.

is highlighted in Fig. 3, where it is compared to the default setting.

## 8 Discussion and Conclusion

We introduce a task-driven framework for characterizing event-camera operating points using a controlled rolling-sphere experiment with known geometry and an LED-based timing reference. Using one-dimensional bias sweeps, we analyze how individual parameters affect the proposed output metrics across different illumination and aperture settings. The sweeps reveal distinct sensitivities of the metrics to contrast thresholds, refractory time, and filter-related biases, informing bias tuning. Based on these trends, we derive a valid operating point by selecting a balanced illumination-aperture regime and combining higher contrast thresholds, moderate refractory suppression, and faster temporal response. Future work will focus on extending the method to unknown objects and enabling online closed-loop parameter tuning. The presented methodology provides a basis for automated operating-point selection in event-based sorting and other event-camera applications where latency, data rate, noise, and geometric fidelity must be balanced explicitly.

## References

- [1] G. Maier *et al.*, “A Survey of the State of the Art in Sensor-Based Sorting Technology and Research,” *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3350987.
- [2] G. Maier *et al.*, “Experimental Evaluation of a Novel Sensor-Based Sorting Approach Featuring Predictive Real-Time Multiobject Tracking,” *IEEE Transactions on Industrial Electronics*, 2021, doi: 10.1109/TIE.2020.2970643.
- [3] P. Lichtsteiner *et al.*, “A  $128 \times 128$  120 dB 15  $\mu$ s Latency Asynchronous Temporal Contrast Vision Sensor,” *IEEE Journal of Solid-State Circuits*, 2008, doi: 10.1109/JSSC.2007.914337.
- [4] T. Delbrück, “Silicon retina with correlation-based, velocity-tuned pixels,” *IEEE Transactions on Neural Networks*, 1993, doi: 10.1109/72.217194.
- [5] G. Gallego *et al.*, “Event-Based Vision: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, doi: 10.1109/TPAMI.2020.3008413.
- [6] Z. Yin *et al.*, “IRIS: Integrated Retinal Functionality in Image Sensors,” *bioRxiv*, 2022, doi: 10.1101/2022.08.14.503909.
- [7] P. Bäcker *et al.*, “Fusion between Event-Based and Line-Scan Camera for Sensor Based Sorting,” in *Conference “Sensor-Based Sorting & Control”* 2024, 2024.
- [8] F. Barranco *et al.*, “Real-Time Clustering and Multi-Target Tracking Using Event-Based Sensors,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, doi: 10.1109/IROS.2018.8593380.
- [9] P. Bäcker and G. Maier, “Motion Compensation of Event-Streams for Multitarget Tracking in Sensor-Based Sorting,” *Sensor and Measurement Science International*, pp. 115–116, 2025, doi: 10.5162/SMSI2025/B6.3.
- [10] J. Baßler and M. Heizmann, “Beyond the sparsity paradigm: efficient binary frames for event-based vision,” in *Automated Visual Inspection and Machine Vision VI*, 2025, doi: 10.1117/12.3062298.
- [11] J. Baßler *et al.*, “Ev2Gray: event-only intensity imaging – occlusion-referenced log-intensity reconstruction,” *tm - Technisches Messen*, 2026, doi: 10.1515/teme-2025-0124.
- [12] B. J. McReynolds *et al.*, “Experimental methods to predict dynamic vision sensor event camera performance,” *Optical Engineering*, 2022, doi: 10.1117/1.OE.61.7.074103.
- [13] A. Manakov *et al.*, “Characterisation metrics for performance comparison of area scan and event-based sensors,” *tm - Technisches Messen*, 2025, doi: 10.1515/teme-2025-0018.
- [14] B. McReynolds *et al.*, “Demystifying Event-based Vision Sensor Biasing to Optimize Signal to Noise for Space Domain Awareness,” 2023.
- [15] A. Ziegler *et al.*, “BiasBench: A reproducible benchmark for tuning the biases of event cameras.”
- [16] T. Delbruck *et al.*, “Feedback control of event cameras,” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, doi: 10.1109/CVPRW53098.2021.00146.
- [17] G. B. Nair *et al.*, “Enhancing Visual Place Recognition via Fast and Slow Adaptive Biasing in Event Cameras,” *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, doi: 10.1109/IROS58592.2024.10802384.
- [18] “Spatter Tracking using Python — Metavision SDK Docs 4.6.2 documentation,” online. Available: [https://docs.prophesee.ai/4.6.2/samples/modules/analytics/tracking\\_spatter\\_py.html?highlight=spatter%20tracking](https://docs.prophesee.ai/4.6.2/samples/modules/analytics/tracking_spatter_py.html?highlight=spatter%20tracking).