From Corners to Fiducial Tags: Revisiting Checkerboard Calibration for Event Cameras

Taehun Ryu, Changwoo Kang, and Kyungdon Joo*
UNIST
{taehunryu,kangchangwoo,kyungdon}@unist.ac.kr

Abstract

The conventional checkerboard-based calibration for standard cameras faces fundamental limitations when applied to bio-inspired event cameras. Specifically, this stems from two challenges: (i) Events are triggered asynchronously at different timestamps along motion trajectories. If we accumulate them directly on the image plane, it causes temporal misalignment and produces blurred edges. (ii) Checkerboard corners on event cameras show near-zero event occurrence at the corner itself. This hinders reliable corner localization and makes calibration difficult. To address these issues, we present a novel calibration framework that directly detects checkerboard corners from event data without learning-based grayscale image reconstruction. We first mathematically analyze the absence of events at corner points. Based on this fact, we then leverage edge-driven event cues to initialize corner positions. Using the near-zero event occurrence at checkerboard corners, we gradually refine the estimated corner toward low event-density regions, achieving sub-pixel accuracy. Furthermore, we extend the corner detection to fiducial markers such as AprilTag, resulting in reliable detection even under partial visibility or occlusion. Evaluations on self-collected and public data demonstrate reliable checkerboard corner detection and stable camera calibration. Additional information is available at the following link: https://vision3d-lab.github.io/corner2tag/.

1. Introduction

The event camera, a neuromorphic vision sensor, has attracted widespread attention for its robustness to motion blur and low-light conditions compared to conventional cameras [14, 25]. Consequently, event-based vision has shown significant effectiveness in diverse computer vision and robotics domains such as 3D scanning [34], robotic manufacturing [1], high-speed object detection and tracking [18, 32, 35, 54], and autonomous navigation [9, 48].

*Corresponding author.

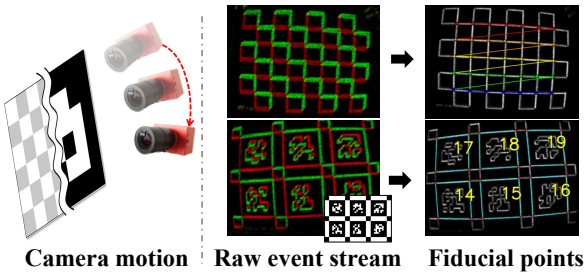


Figure 1. **Illustration of proposed pattern detection.** We utilize checkerboard patterns and AprilTag [50] to directly detect corners from an event stream (left). Our method can extract the corners of the checkerboard and the distinct fiducial (right).

As the demand for such applications increases, calibration of event cameras becomes crucial for reliable vision systems as a process of estimating camera parameters. For frame-based cameras, the camera parameters are typically determined by minimizing the reprojection error between 2D fiducial points detected from images and their corresponding 3D points pre-defined on the calibration target. Various pre-defined patterns (*e.g.,* checkerboard [7], circular grid [23]) have been developed to provide such easily observable fiducial points for accurate calibration. Among them, the checkerboard pattern is the most common and reliable calibration target [3, 4]. Specifically, the fiducial points of the checkerboard are the corners of black-and-white squares, which are defined by the intersections of two edges. These characteristics maintain geometrically well-defined under perspective projection or lens distortion.

However, the checkerboard pattern remains unexplored for event cameras, as it cannot be directly applied to a raw event stream. This is because event cameras do not preserve the image structure. Instead, they record relative brightness changes asynchronously at each pixel as binary values. For this reason, some methods utilize learning-based reconstruction to convert a raw event stream into grayscale images, and then apply frame-based calibration to the reconstructed grayscale images [19, 37]. However, artifacts of reconstructed images inherently constrain calibration precision. Some methods utilize blinking LED boards to directly

process a raw event stream [20, 44]. They generally yield accurate calibration results, but they demand high hardware requirements and rely on dedicated procedures. Other approaches employ circle-grid patterns for event-based calibration [6, 24, 45]. They exhibit motion-invariant characteristics and achieve robust detection performance, yet their design inherits the drawback of circular patterns, which are sensitive to lens distortion [47]. Compared to the above approaches, the checkerboard pattern provides geometrically consistent and precisely localized corners, ensuring high calibration performance. It also holds strong potential for extension to ID-based markers such as ARTag [10], and AprilTag [38, 50], as they share a similar corner-based detection principle.

In this work, we advocate the revisit of the standard checkerboard pattern using event cameras to preserve its proven accuracy and its extensibility to fiducial markers (see Fig. 1). However, this approach faces two key challenges. First, due to the asynchronous characteristic of each event, stacking them into a single image produces blurred edges [12]. Second, events are not triggered at checkerboard corners due to their locally symmetric intensity pattern, which limits reliable corner detection.

To resolve these, we propose the first event camera calibration framework that directly detects checkerboard corners from event representation without grayscale image reconstruction (see Fig. 2). We create an Image of Warped Events (IWE) to temporally align asynchronous events. This process produces a sharp, static representation, which is essential for precise corner detection. Then, we provide a mathematical analysis of Event Rate (ER) at checkerboard corners. This analysis confirms that events are theoretically absent at the exact corner intersections. Grounded in this analysis, our method first initializes corners using strong edge cues, instead of finding corners directly. We then refine the initial corners by moving them toward the point of minimum event density to achieve sub-pixel accuracy. These accurate corner estimates are then utilized to optimize the camera parameters. Furthermore, we demonstrate that our framework extends to fiducial markers, such as AprilTag [50]. This enables reliable tag identification even under partial visibility.

Our main contributions are summarized as follows:

- **Event-based checkerboard calibration**: We present the first calibration framework using detected checkerboard corners without grayscale image reconstruction.
- **Mathematical analysis of corner events**: We analyze the absence of events at corner intersections, which we then exploit for corner initialization and refinement.
- **Analysis-based corner localization**: Based on our analysis of corner events, we initialize the corners and refine them to achieve sub-pixel accuracy.
- **Extension to fiducial markers**: Our framework generalizes to fiducial markers (*e.g.*, AprilTag [50]), ensuring robust detection under partial visibility.

2. Related Work

Camera calibration process can be divided into two steps: Pattern detection and camera parameter optimization [52]. Pattern detection aims to identify accurate fiducial points that serve as 2D-3D correspondences from predefined calibration targets. Camera parameter optimization then estimates the intrinsic and extrinsic parameters of cameras by solving an optimization problem based on these correspondences. In this section, we mainly deal with pattern detection for both frame-based and event cameras. Please refer to other frameworks [3, 4, 11] for details of the optimization process.

Pattern Detection for Frame-Based Cameras. As fiducial points directly provide 2D-3D correspondences, various calibration targets have been proposed [4, 7, 23, 50]. Among them, the checkerboard pattern [3, 4, 7, 30] serves as the standard for high-precision calibration due to its geometrically well-defined and densely distributed corners, which enable robust and sub-pixel localization even under perspective transformation or lens distortion. Building upon this concept, ARTag [10], AprilTag [38, 50], and ArUco marker [15] encode unique marker IDs to ensure reliable detection under partial observations. Accordingly, the checkerboard remains the most widely adopted target in frameworks such as OpenCV [4]. In parallel, other primitives such as circular features or dots are used for calibration targets [23, 27, 28]. They use centroids of circular shapes as fiducial points. They exhibit rotational invariance but suffer from degraded localization accuracy under nonlinear transformations such as lens distortion [29, 47].

In this work, our objective is to apply the well-validated checkerboard pattern to event cameras while leveraging its proven calibration accuracy and structural extensibility to ID-coded fiducial systems [10, 15, 38, 50].

Pattern Detection for Event Cameras. Conventional pattern detection approaches are not directly applicable to event-based data due to its frameless and binary-valued nature [49]. As a straightforward approach, active-target (*e.g.,* LCD or blinking LED) and wand are initially adopted for event cameras [5, 20, 21, 44, 51]. Their design is well-tailored to the triggering mechanism of events, making them intuitive and accurate. However, these approaches require specialized hardware and corresponding methodologies.

In the other directions, several studies attempt to detect circular or circle-grid patterns directly from a raw event stream [6, 24, 45]. These patterns offer robustness, as events are generated along every edge regardless of motion direction. In these approaches, events are typically clustered and

modeled by fitting cylindrical or elliptical shapes to a raw event stream. They share the drawback of high sensitivity to lens distortion with circular patterns, with the cylindrical fitting making precise localization even more difficult.

With the recent advances in learning-based approaches, some methods [19, 37] leverage the event-to-video reconstruction network [43] to convert a raw event stream into grayscale images and apply the Harris corner detector [22] to identify fiducial points. However, event-to-video reconstruction introduces artifacts [53], which may degrade the accuracy of subsequent corner localization. These limitations motivate direct detection of fiducial points from event-domain representations rather than reconstructed images.

In this work, we detect fiducial points directly from event representation, specifically IWE [13], rather than relying on imprecise reconstruction. To this end, we analyze event generation at corner points and propose a calibration pipeline specialized to event characteristics.

3. Method

3.1. Overview

We investigate the event camera calibration directly in the event domain, without relying on grayscale image reconstruction. Calibrating event cameras with a checkerboard is challenging due to asynchronous event generation, which leads to blurred edges when events are accumulated [12]. In addition, the absence of events at the corner intersection makes corner detection difficult.

To address these, we propose an event camera calibration pipeline using a checkerboard pattern (see Fig. 2). We take as input a raw event stream $\mathcal{E} = \{e_k\}_{k=1}^{N}$, where each event $e_k = (\mathbf{x}_k, t_k, p_k)$ consists of the pixel location $\mathbf{x}_k$, timestamp t_k, and polarity p_k (-1 or $+1$). Then, we convert a raw event stream into an IWE [13] via events warping and events accumulation process to obtain sharper edges (Sec. 3.3). Given IWE, we initialize checkerboard corners by leveraging reliable edge cues, and then refine them toward the local minimum of event density to obtain sub-pixel accuracy. Both steps are motivated by our analysis in Sec. 3.2, which shows that events are stably generated along edges while vanishing at the corner. The refined corners are fed into the batch optimization process to estimate the camera parameters $\mathbf{K}, \mathbf{d}$ (Sec. 3.4). Moreover, we show that our framework generalizes to other fiducial markers, such as AprilTag [50], allowing for robust tag identification even under partial occlusion (Sec. 3.5).

3.2. Event Rate on Checkerboard Corner

Understanding how events are generated around an ideal checkerboard corner helps to explain the underlying characteristics of event data and to design more robust detection methods. Here, we analyze that events are not triggered in the checkerboard corner, using the intensity model of the checkerboard corner in frame-based cameras [30] and the event generation rate formulation [33]. Details of the analysis are described in the supplementary material.

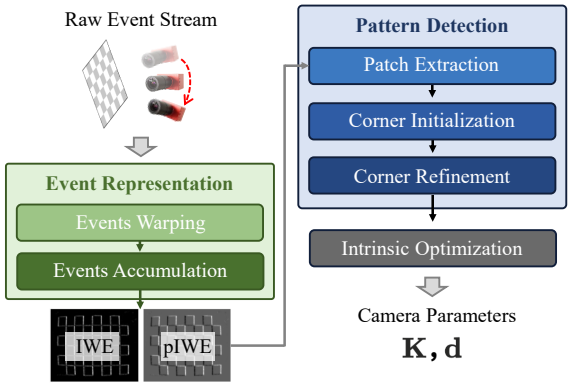


Figure 2. **Overview of the proposed calibration pipeline.** The framework converts a raw event stream into both IWE and pIWE, which are representations well-suited for extracting corner points. These representations are then used by the pattern detection module, which feeds the detected corners into the intrinsic optimization process to estimate the final camera parameters $\mathbf{K}, \mathbf{d}$.

Intensity Model on Checkerboard Corner. In frame-based cameras, the ideal intensity distribution of the checkerboard corner can be defined with a step function $\rho(\cdot)$ as [30]:

$$\hat{I}(\mathbf{x}) = \rho(\mathbf{n}_1^\top \mathbf{x})\rho(\mathbf{n}_2^\top \mathbf{x}),\ \rho(x) = \begin{cases} +1 & \text{if } x > 0, \\ -1 & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathbf{n}_i$ denotes unit normals to checkerboard edges. Using this definition, Krüger *et al.* [30] define the intensity model of the checkerboard corner $\tilde{I}$ by replacing the step function $\rho(\cdot)$ with a centered sigmoid smoothed by a Gaussian filter of width σ:

$$\tilde{I}(\mathbf{x}) \approx S(\mathbf{n}_1^\top \mathbf{x}, \sigma) S(\mathbf{n}_2^\top \mathbf{x}, \sigma), \tag{2}$$

where $S(x, \sigma) = 2/(1+e^{-x/\sigma}) - 1$ is a centered sigmoid. In this model, the ideal checkerboard corner is located $\mathbf{x} = \mathbf{0}$. Note that since construction $S(x, \sigma) \in (-1, 1)$, $\tilde{I}(\mathbf{x}) \in [-1, 1]$ represents the idealized contrast pattern of the checkerboard, rather than the actual brightness observed by the camera.

Event Generation Rate. Based on the event-triggering mechanism, an event is generated whenever the change in logarithmic intensity at a pixel exceeds a fixed contrast threshold C [14]. This implies that, at a given location, faster or larger brightness variations result in a greater number of events accumulated within a fixed time window. Thus, Event Rate (ER) can be defined as the number of events generated per unit time in a given time interval.

Therefore, ER can be interpreted as a statistical measure of the logarithmic intensity change rate, which is related to the image gradient [17, 33].

Following this interpretation, Lin *et al*. [33] establish a relationship between ER and the gradient value in short time interval $[t-\Delta t/2,\ t+\Delta t/2]$ as follows:

$$R_e(\mathbf{x}, t, \Delta t) \approx \frac{1}{C}|\nabla L(\mathbf{x}, t) \cdot \mathbf{v}|, \tag{3}$$

where $L(\mathbf{x}, t) = \log I(\mathbf{x}, t)$ denotes the logarithmic image intensity at pixel location $\mathbf{x}$, and $\mathbf{v}$ is the motion vector. For simplicity, we omit the temporal variable t in the following analysis, using $L(\mathbf{x})$ and $R_e(\mathbf{x})$ instead of $L(\mathbf{x}, t)$ and $R_e(\mathbf{x}, t, \Delta t)$, respectively.

Event Rate on Checkerboard Corner. To analyze ER on checkerboard corner, we combine Eq. (2) and Eq. (3). Before combining them, we first reformulate centered sigmoid $S(x, \sigma)$, noting that $S(x, \sigma)$ can be expressed as a hyperbolic tangent function,

$$S(x, \sigma) = \tanh\left(\frac{x}{2\sigma}\right). \tag{4}$$

Then, we recall that the hyperbolic tangent function has the following Taylor expansion [26]:

$$\tanh(x) = x - \frac{1}{3}x^3 + \frac{2}{15}x^5 - \cdots, \text{ when } |x| < \frac{\pi}{2}. \tag{5}$$

Since we are primarily interested in the neighborhood of the ideal checkerboard corner, where its region is in $|x| \ll 1$, the Taylor expansion of $\tanh(x)$ is valid, and the higher-order terms can be neglected as

$$S(x, \sigma) = \tanh\left(\frac{x}{2\sigma}\right) \approx \frac{x}{2\sigma} + \mathcal{O}(x^3) \approx \frac{x}{2\sigma}, \tag{6}$$

where $\mathcal{O}(x^3)$ denotes the third-order remainder in the Taylor expansion.

Based on this convention, we reformulate Eq. (2) using the approximation in Eq. (6), deriving its local analytical form around the checkerboard corner:

$$\tilde{I}(\mathbf{x}) \approx \frac{1}{4\sigma^2}(\mathbf{n}_1^\top \mathbf{x})(\mathbf{n}_2^\top \mathbf{x}), \quad \nabla \tilde{I}(\mathbf{x}) \approx G\,\mathbf{x}, \tag{7}$$

where $G = \frac{1}{4\sigma^2}(\mathbf{n}_1\mathbf{n}_2^\top + \mathbf{n}_2\mathbf{n}_1^\top)$ is symmetric. Since $\tilde{I}(\mathbf{x})$ represents an idealized contrast pattern rather than an actual brightness value, we obtain the image intensity via a linear camera response applied to $\tilde{I}(\mathbf{x})$ [30]:

$$I(\mathbf{x}) = I_0 + \alpha\,\tilde{I}(\mathbf{x}), \tag{8}$$

where α denotes the camera gain and I_0 offset.

Having established the local intensity model of the checkerboard corner, we now analyze ER on checkerboard corner by substituting Eq. (8) into the ER formulation in Eq. (3). Since $L(\mathbf{x}) = \log I(\mathbf{x})$, we have

$$\nabla L(\mathbf{x}) \approx \frac{\alpha}{I_0 + \alpha\,\tilde{I}(\mathbf{x})}\,G\mathbf{x}. \tag{9}$$

Since $\tilde{I}(\mathbf{x}) \approx 0$ around the corner, the term $\alpha/(I_0 + \alpha\,\tilde{I}(\mathbf{x}))$ can be approximated by α/I_0 and treated as constant. Therefore, Eq. (3) can be rewritten as

$$\begin{aligned} R_e(\mathbf{x}) &\approx \frac{1}{C}\left|\nabla L(\mathbf{x}) \cdot \mathbf{v}\right| \\ &\approx \gamma\left|(G\mathbf{x}) \cdot \mathbf{v}\right| = \gamma\left|(G\mathbf{v})^\top \mathbf{x}\right|, \end{aligned} \tag{10}$$

where $\gamma = \alpha/CI_0$ is a constant factor. Since G cannot be zero unless $\mathbf{n}_1$ and $\mathbf{n}_2$ are parallel, ER becomes zero at the exact corner $\mathbf{x} = \mathbf{0}$ or in the degenerate case $\mathbf{v} = \mathbf{0}$. Thus, from Eq. (10), we find that

$$\mathbf{x} = \mathbf{0} \implies R_e(\mathbf{0}) = 0. \tag{11}$$

This result holds locally near a checkerboard corner, where the image gradients are modeled as a pair of intersecting edges. In contrast, the intensity model of the checkerboard edge can be modeled as

$$\tilde{I}(\mathbf{x}) \approx \frac{1}{2\sigma}(\mathbf{n}^\top \mathbf{x}), \quad \nabla \tilde{I}(\mathbf{x}) \approx \frac{\mathbf{n}}{2\sigma}. \tag{12}$$

Here, the checkerboard edge corresponds to the set of points $\mathbf{x}$ that satisfy $\mathbf{n}^\top \mathbf{x} = 0$. Then, we can formulate ER on the checkerboard edge as follows:

$$R_e(\mathbf{x}) \approx \kappa\left|\mathbf{n} \cdot \mathbf{v}\right|, \quad \text{where } \kappa = \gamma/2\sigma. \tag{13}$$

This expression indicates that events are reliably generated along the edge as long as the motion vector is not parallel to the edge direction.

In conclusion, under the first-order assumption on local models, no events are generated at the checkerboard corner itself, whereas along an edge, events occur whenever $\mathbf{n} \cdot \mathbf{v} \neq 0$. Consequently, ER on edges is strictly higher than at the corner. This contrast between edges and corners forms the basis of our detection pipeline: edge regions provide reliable cues for initialization, while the absence of events at the corner provides a principled criterion for refinement toward the true corner location.

3.3. Event Representation

Event cameras capture visual information asynchronously. This asynchronous stream forms spatio-temporal trajectories based on relative scene motion or ego-motion [12, 13]. Thus, simple accumulation of these events on a static image plane causes motion blur. To mitigate it, various event representations have been proposed [12, 13, 16, 31, 55]. Among them, we adopt IWE [12, 13], which warps events

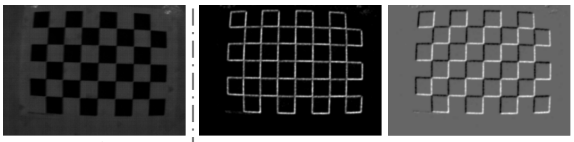


Figure 3. **Event representation.**

to a reference time under a motion model to enforce geometric alignment and suppress motion blur. In contrast to time-surfaces [31, 55] that encode temporal recency at each pixel, IWE aligns spatio-temporal event trajectories onto a common instant, yielding an edge image consistent with the single-instant assumption of camera calibration [52].

Events Warping. To construct IWE, we first divide the input event stream $\mathcal{E}$ into M coherent windows, $\mathcal{E} = \bigcup_{i=1}^{M} \mathcal{E}_i$, and process each window $\mathcal{E}_i$ independently. For each window, we warp each event to reference time t_{ref}, using the warping function [12]:

$$\begin{aligned} e_k = (\mathbf{x}_k, t_k, p_k) \mapsto e'_k = (\mathbf{x}'_k, t_{\text{ref}}, p_k), \\ \mathbf{x}'_k = \mathbf{x}_k + (t_{\text{ref}} - t_k)\,\mathbf{v}, \end{aligned} \tag{14}$$

where $\mathbf{v} \in \mathbb{R}^2$ denotes the constant image-plane motion vector and $\mathbf{x}'_k$ is the pixel location of the warped event e'_k. This model assumes linear camera motion within each window, which is a valid assumption in calibration scenarios where data can be collected under a controlled environment [52].

Then, we build Image of Warped Events (IWE):

$$H(\mathbf{x}; \mathbf{v}) = \sum_{k=1}^{N_e} p_k \delta(\mathbf{x} - \mathbf{x}'_k), \tag{15}$$

where N_e represents the number of events in each window, each pixel $\mathbf{x}$ accumulates the number of warped events whose locations $\mathbf{x}'_k$ spatially coincide with it, and $\delta(\cdot)$ indicates the Dirac delta function. To estimate the optimal motion vector $\mathbf{v}^*$, we maximize the variance of IWE:

$$\mathbf{v}^* = \arg\max_{\mathbf{v} \in \mathbb{R}^2} \operatorname{Var}\big(H(\mathbf{x}; \mathbf{v})\big). \tag{16}$$

This objective encourages sharper and more coherent event alignment [12, 13].

Using the estimated motion vector $\mathbf{v}^*$, we can construct IWE and polarity-aware IWE (pIWE) for each window. IWE $H(\mathbf{x}; \mathbf{v}^*)$ aggregates events regardless of their polarity by setting $p_k = 1$ for all events. It stably captures the landscape of event density without being affected by polarity cancellation. In contrast, pIWE $H_p(\mathbf{x}; \mathbf{v}^*)$ holds the polarity p_k of each event. Using pIWE is effective for identifying black–white alternation of checkerboard patterns, as it preserves polarity contrast. We use these event representations in the subsequent corner detection pipeline (see Fig. 3).

We note that, unlike grayscale images from the reconstruction network [43], which estimates absolute scene brightness, IWE and pIWE are motion-compensated event representations that retain event density. Consequently, ER is directly compatible with these representations.

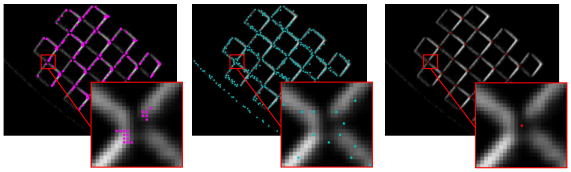


Figure 4. **Comparison with conventional methods.** Harris [22] and Shi-Tomasi [46] fail under event data, whereas the proposed method achieves accurate and stable localization.

3.4. Corner Detection and Calibration

Here, we propose our calibration framework, including a checkerboard corner detection pipeline built upon our analytical insights from Sec. 3.2. Our framework consists of four steps: (i) patch extraction, (ii) corner initialization, (iii) corner refinement, and (iv) camera parameters optimization. First, we extract local patches around checkerboard corners to isolate candidate regions. Second, for each patch, we initialize corner positions by identifying local patterns consistent with the geometry of the checkerboard corner. Third, we refine the initial corners to sub-pixel accuracy by optimization. Finally, we aggregate the refined corners and use them for calibration.

Patch Extraction. To pinpoint the refined corner $\mathbf{c}^*$ precisely, we extract small patches to capture local pattern near each corner. While our overall calibration pipeline primarily utilizes IWE, we leverage polarity contrast of pIWE during the patch extraction to detect initial corners.

To extract each patch in pIWE, we adopt circular boundaries inspired by Bok *et al.* [2]. Since a checkerboard corner lies at the intersection of black and white regions, we select pixels whose circular boundaries contain both positive and negative polarities. Then, the selected pixels close to each other are merged into the same clusters. Each cluster is enclosed by its smallest bounding rectangle within pIWE. Since IWE shares the same pixel coordinates with pIWE, we can extract the corresponding candidate corner regions from IWE. We denote such candidate patches as $\mathcal{P}$.

At first glance, it seems to be possible to apply conventional detection algorithms [22, 46] to IWE. However, even with IWE, corner detection remains highly unreliable on event data (see Fig. 4). As analyzed in Eq. (11), this is because corner regions barely trigger any events.

Corner Initialization. We guess the initial corner $\mathbf{c}_0$ within the single patch $\mathbf{P} \in \mathcal{P}$. Although invalid patches might still exist in the candidate patches $\mathcal{P}$, we can filter out invalid patches during this process.

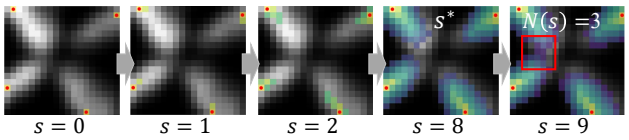


Figure 5. **Pixel propagation from seed pixels.** Each red point means seed pixel $\mathfrak{s} \in \mathcal{S}$.

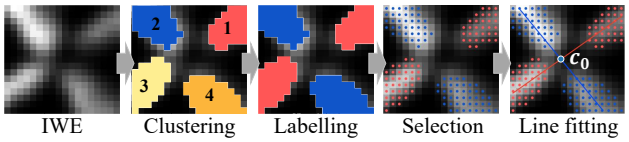


Figure 6. **Corner initialization procedure.**

As we analyzed in Sec. 3.2, events rarely occur at corners, and thus cannot be directly used for corner localization. To handle this issue, we first leverage edge-based cues to estimate the initial corner $\mathbf{c}_0$. As each pixel value in IWE represents the event density, pixels with high values correspond to edges where distinct motion occurs. Leveraging this property, we can define seed pixels $\mathcal{S}$ as pixels that have higher values than their eight neighboring pixels. Seed pixels $\mathcal{S}$ are highly likely to belong to certain edges, so we detect them from the boundary regions within patch $\mathbf{P}$ (see Fig. 5). The seed pixels $\mathcal{S}$ are defined as follows:

$$\mathcal{S} \;=\; \big\{\mathbf{x} \in \mathbf{P} \;\big|\; H(\mathbf{x}) > H(\mathbf{y}),\; \forall \mathbf{y} \in \mathcal{N}_8(\mathbf{x})\big\}, \qquad (17)$$

where $\mathbf{x}$ is a pixel in the patch $\mathbf{P}$ and $\mathcal{N}_8(\mathbf{x})$ represents eight neighbors of $\mathbf{x}$. To initialize precise corner points, we propagate from each seed pixel $\mathfrak{s} \in \mathcal{S}$ toward the second-largest neighbor in $\mathcal{N}_8(\mathfrak{s})$ step by step (see Fig. 5). After each seed pixel $\mathfrak{s}$ propagates one step toward its neighboring pixel, it merges with the propagated pixel, forming a cluster. Let $N(s)$ denote the number of non-empty clusters after merging at step $s \in \mathbb{R}$. Each patch generally maintains $N(s) = 4$, corresponding to the four edges of the checkerboard pattern. When $N(s) \neq 4$, it indicates either a single-edge response or a misdetected seed, and such cases are filtered out. As propagation continues, clusters gradually merge, decreasing $N(s)$. We therefore define s^* as the largest propagation step for which $N(s) = 4$:

$$s^* \;=\; \max\big\{\, s \in \mathbb{R} \;\big|\; N(s) = 4 \,\big\}. \qquad (18)$$

At step s^*, we order the four clusters counterclockwise by label $l \in \{1, 2, 3, 4\}$. We then define two lines based on index parity: $\mathbf{l}_{\text{odd}}$ is the line fitted to all pixels belonging to clusters with an odd label, and $\mathbf{l}_{\text{even}}$ is the line fitted to all pixels belonging to clusters with an even label. Then, we determine the corner $\mathbf{c}_0$ as the intersection $\mathbf{c}_0 = \mathbf{l}_{\text{odd}} \times \mathbf{l}_{\text{even}}$ (see Fig. 6). We can filter out the invalid patch if the two lines are nearly parallel or if obtained $\mathbf{c}_0$ not in $\mathbf{P}$. This approach enables us to initialize a reliable corner $\mathbf{c}_0$ when few or no events occur around the actual corner. The filtered cases are shown in the supplementary material.

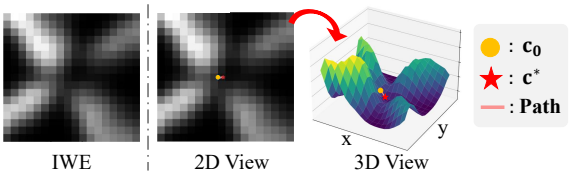


Figure 7. **Visualization of corner refinement on the IWE surface.** Its height represents the count of accumulated events. We show the optimization of an initial corner $\mathbf{c}_0$ to its refined corner $\mathbf{c}^*$ along the gradient path.

Corner Refinement. To obtain the sub-pixel accurate refined corner $\mathbf{c}^*$, we refine the initial corner $\mathbf{c}_0$. Based on our analysis in Eq. (11), we assume that the ideal corner corresponds to the minimum value in IWE. Thus, we apply gradient descent to $\mathbf{c}_0$ to find this local minimum (see Fig. 7). During this process, non-corner regions in IWE (*e.g.,* checkerboard squares) may also yield near-zero responses, potentially leading to incorrect convergence. To mitigate this effect, we constrain the optimization to keep corner $\mathbf{c}^*$ within the intersection of two narrow bands around the fitted lines $\mathbf{l}_{\text{odd}}$ and $\mathbf{l}_{\text{even}}$, as follows:

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} H(\mathbf{c}) \quad \text{s.t. } |\mathbf{c}^\top \mathbf{l}_{\text{odd}}| \leq \epsilon, |\mathbf{c}^\top \mathbf{l}_{\text{even}}| \leq \epsilon, \quad (19)$$

where $\mathbf{c}$ denotes a sub-pixel coordinate within the local neighborhood around $\mathbf{c}_0$, and ϵ is a small tolerance.

This refinement process encourages the initial corners to move toward the minimum of the event density within the constrained region, achieving sub-pixel accuracy. Through this process, we obtain geometrically consistent corner positions that align with our theoretical analysis.

Camera Parameter Optimization. Finally, we perform a batch optimization to estimate the camera parameters. Each of the event windows yields a set of refined 2D corners $\mathcal{C}_i^{2D}$, where $i \in \{1, \ldots, M\}$. Each element in $\mathcal{C}_i^{2D}$ corresponds to a refined corner $\mathbf{c}^*$ obtained in the previous process. We also denote the 3D corner coordinates of the calibration board as $\mathcal{C}^{3D}$, which are predefined and fixed during optimization. The refined corner sets from all event windows are jointly used with the predefined 3D corner coordinates to optimize the intrinsic matrix $\mathbf{K}$ and distortion coefficients $\mathbf{d}$ by minimizing the reprojection error using `calibrateCamera`$(\cdot)$ in OpenCV [4]:

$$\mathbf{K}^*, \mathbf{d}^* = \arg\min_{\mathbf{K},\mathbf{d},\mathbf{T}_i} \sum_{i=1}^{M} \left\| \mathcal{C}_i^{2D} - \pi\left(\mathbf{K}, \mathbf{d}, \mathbf{T}_i, \mathcal{C}^{3D}\right) \right\|^2, \qquad (20)$$

where $\mathbf{T}_i$ denotes the extrinsic parameters (camera pose) for the i-th event window, and $\pi(\cdot)$ represents the pinhole projection model including distortion.

(a) Quad fitting (b) Binarization (c) Decoding

Figure 8. **AprilTag detection procedure.**

3.5. AprilTag Detection

By using the polarity contrast of pIWE (Sec. 3.4), our method can be readily extended to AprilTag detection [38, 50]. AprilTag encodes unique IDs within its patterns, enabling reliable identification even under partial visibility or occlusion. Thus, we demonstrate an extension built upon the proposed pattern detection to ID-based fiducial (see Fig. 8). AprilTag detection consists of three steps: (i) quad fitting, (ii) binarization, and (iii) tag decoding. Under the same assumptions in Sec. 3.1, we capture AprilTag patterns using an event camera. By applying our pattern detection method, we can reliably obtain refined corners $\mathcal{C}_i^{2D}$ corresponding to each divided window. For simplicity, we omit the subscript i in the following description.

Quad Fitting. Given the refined corners $\mathcal{C}^{2D}$, we enumerate four-point combinations to obtain quadrilateral candidates $\mathcal{Q}$ (see Fig. 8 (a)). Thus, each candidate $\mathcal{Q}$ is defined as a subset of four corners:

$$\mathcal{Q} \subset \mathcal{C}^{2D}, \quad |\mathcal{Q}| = 4,$$

where $|\cdot|$ denotes the number of elements.

We order each subset $\mathcal{Q}$ clockwise and check whether it satisfies the following geometric constraints to verify its validity as a quadrilateral:

- **Angle range constraint:** All internal angles should remain within a valid range.
- **Minimum area threshold:** The quadrilateral must have sufficient area to ensure geometric stability for detection.
- **Edge length consistency:** The quadrilateral should maintain similar side lengths to prevent distorted shapes.

We then remove overlapping quadrilaterals whose enclosing regions largely overlap using non-maximum suppression (NMS). The remaining quads serve as tag candidates for the subsequent process.

Binarization. In this process, we normalize each valid quadrilateral $\mathbf{q} \in \mathcal{Q}$ into a grid-like binary pattern to easily identify the tag (see Fig. 8 (b)). Based on the four corners of quadrilateral $\mathbf{q}$, we warp the corresponding pIWE region into a normalized square coordinate frame.

From the warped patch, we compute image gradients ∇H_p and project them along motion vector $\mathbf{v}$, which is the same in Sec. 3.3, to obtain a contrast map $g = \nabla H_p \cdot \mathbf{v}$. This operation emphasizes polarity-aligned structural edges of the tag. We then apply ternary thresholding to g to produce $\{-1, 0, +1\}$ values that represent negative edges, neutral regions, and positive edges. Next, we divide the ternarized patch into an $N \times N$ grid and compute the mean value of each cell to construct a coarse binary structure that encodes the tag pattern. Finally, we binarize the cell using Otsu's adaptive thresholding [39].

Tag Decoding. To assign the tag ID to the given binary pattern, we pack the binary pattern into an integer codeword and compare it with the predefined AprilTag codebook. We achieve rotation invariance by evaluating the Hamming distance across four possible orientations and select the best match when the minimum Hamming distance falls below a predefined threshold [38]. By doing so, we can obtain the unique tag ID and orientation corresponding to given quadrilateral $\mathbf{q}$. Examples of various matching results are provided in the supplementary material.

4. Experiments

4.1. Experimental Setup

Our framework is evaluated on real-world data from a DAVIS346 event camera (346×260 resolution), which provides both asynchronous events and grayscale frames. During acquisition, we move the camera so that all checkerboard edges appear in the view, while keeping the motion approximately linear. Our dataset contains three checkerboard patterns: 5×6 (33 mm), 6×8 (35 mm), and 7×10 (28 mm). In addition to our self-collected data, we also incorporate public data from Prophesee Gen 3 and Samsung Gen 3 event cameras [37]. However, since these public data lack checkerboard information, we use them only for qualitative corner detection evaluation.

To the best of our knowledge, existing event camera calibration methods that support checkerboards rely only on image reconstruction [37, 43]. Therefore, we compare our method against two baselines: (1) E2Calib [37], which uses E2VID [43] for grayscale image reconstruction, and (2) calibration using DAVIS grayscale frames. In our real-world evaluation, we exclude circle-grid calibration methods [6, 24, 45] as they require a different calibration target, and using different targets inevitably leads to different camera–pattern relative poses. Since intrinsic estimation is sensitive to the pose distribution, performance differences would be confounded by target-induced pose variations rather than the calibration method itself [41]. Therefore, achieving a fair comparison across different calibration targets is difficult in real-world settings, and we additionally conduct synthetic experiments, including comparisons with circle-grid calibration [6, 45].

4.2. Quantitative Comparison

Our method demonstrates higher repeatability and lower sensitivity than the comparison methods. As shown in Table 1, it consistently yields smaller standard deviations

Table 1. **Calibration results (mean ± std) on checkerboard patterns.** Calibration RMSE denotes the root-mean-square reprojection error (in pixels) evaluated using the mean estimated parameters. Frame-based calibration is used as a pseudo ground-truth reference.

Board	Method	Intrinsic Parameters				Distortion Coefficients				Calibration RMSE
		f_x	f_y	c_x	c_y	k_1 ($\times 10^{-1}$)	k_2 ($\times 10^{-1}$)	p_1 ($\times 10^{-3}$)	p_2 ($\times 10^{-3}$)	
5×6 (B=32)	E2Calib [37]	332.47±3.00	332.48±3.01	187.66±1.36	131.90±1.20	−2.39±0.16	−6.43±1.24	−3.25±0.86	−2.93±0.86	0.747
	Ours	347.96±**1.60**	348.51±**1.62**	184.25±**0.68**	125.43±**0.57**	−3.70±**0.09**	1.39±**0.72**	−1.35±**0.29**	−1.47±**0.48**	**0.487**
	Frame-Based	349.30±1.71	349.99±1.73	186.82±0.50	129.57±0.32	−3.48±0.05	1.21±0.20	−0.42±0.21	−1.90±0.40	0.199
6×8 (B=32)	E2Calib [37]	346.25±2.15	346.95±2.12	185.94±1.48	133.43±1.30	−3.67±0.16	1.82±0.88	−2.22±0.60	−0.10±0.89	0.647
	Ours	347.60±**1.22**	347.99±**1.19**	183.07±**1.32**	131.90±**1.05**	−3.45±**0.05**	0.86±**0.36**	−2.00±**0.50**	0.60±**0.60**	**0.516**
	Frame-Based	349.07±0.87	349.33±0.86	187.63±0.67	130.84±0.49	−3.55±0.04	2.11±0.23	−1.60±0.20	0.40±0.30	0.192
7×10 (B=25)	E2Calib [37]	342.87±2.02	342.89±1.93	187.02±1.07	129.14±**0.43**	−3.39±**0.05**	0.76±0.24	−0.51±0.35	−1.24±0.57	0.559
	Ours	341.60±**0.81**	341.80±**0.78**	184.31±**0.38**	131.79±0.50	−3.42±**0.05**	2.00±**0.01**	−1.16±**0.34**	−1.93±**0.23**	**0.503**
	Frame-Based	342.12±0.74	342.31±0.71	190.18±0.39	130.39±0.28	−3.56±0.02	2.01±0.05	−1.10±0.19	−3.15±0.21	0.153

Table 2. **Corner RMSE (mean ± std).** RMSE is the Euclidean distance (in pixels) between checkerboard corner locations from each method and the frame-based method.

Board	E2Calib [37]	Ours
5×6	1.3979 ± 0.8350	**0.8504** ± **0.3203**
6×8	1.0377 ± 0.4879	**0.8356** ± **0.4776**
7×10	0.7808 ± **0.2742**	**0.7791** ± **0.2742**

for all intrinsic and distortion parameters across the three checkerboard patterns, indicating more stable calibration. For this evaluation, we randomly sample the set of B corresponding images in which all modalities (E2Calib [37], ours, and frame-based) detect checkerboard corners successfully and repeat calibration 50 times for each method.

To verify the accuracy of our method, we evaluate corner localization accuracy by measuring the Euclidean distance between the corner locations projected using the camera parameters estimated by each method and the projected locations of the reference parameters obtained from the DAVIS grayscale frames (see Table 2). It shows that our method achieves markedly lower RMSE compared to E2Calib (*i.e.,* E2VID-based calibration). Additional quantitative results, including evaluations on synthetic data and computational time, are provided in the supplementary material.

4.3. Qualitative Results

As shown in Fig. 9, our method yields geometrically consistent corner detections on both self-collected and public data. The extracted corners remain well localized, providing reliable inputs for subsequent calibration. Beyond checkerboards, we showcase that our approach extends to various fiducial markers such as AprilTag [38] (see Fig. 10). Our method successfully detects AprilTag with various IDs and geometric configurations, and handles partial board observations by correctly indexing the visible subset of tags even when others fall outside the field of view. Additional qualitative results in the supplementary material include detections across diverse poses and, for AprilTag, corner indexing under partial observations.

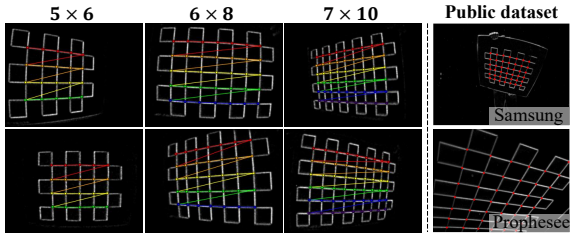


Figure 9. **Detected checkerboard corners.** The left side of the dotted line is self-collected. The right side is public [37].

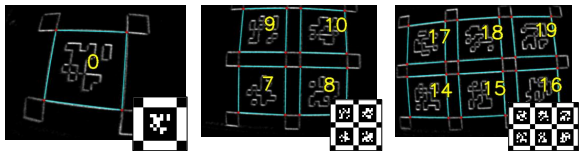


Figure 10. **Qualitative results on various AprilTag.**

5. Conclusion

We presented the first event-based calibration framework that directly detects checkerboard corners without grayscale image reconstruction. Guided by a mathematical analysis revealing the absence of events at corner intersections, our method leverages edge-driven initialization and event density-based refinement to achieve sub-pixel corner accuracy. Extensive experiments demonstrate that our approach achieves accurate calibration precision. By reintroducing checkerboards and extending them to fiducial markers such as AprilTag, this work showcases the feasibility of real-world deployment for event camera calibration.

Limitation. Our method assumes a dominant global motion and estimates a single motion vector in IWE generation stage, which may degrade performance in scenes with spatially varying motion. Incorporating more general motion models (*e.g.,* pixel-wise motion) remains future work.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST); No.RS-2022-II220907, Development of AI Bots Collaboration Platform and Self-organizing; No.RS-2022-II220612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.RS-2024-00457065).

References

[1] Abdulla Ayyad, Mohamad Halwani, Dewald Swart, Rajkumar Muthusamy, Fahad Almaskari, and Yahya Zweiri. Neuromorphic vision based control for the precise positioning of robotic drilling systems. *Robotics and Computer-Integrated Manufacturing*, 79:102419, 2023.

[2] Yunsu Bok, Hyowon Ha, and In So Kweon. Automated checkerboard detection and indexing using circular boundaries. *Pattern Recognition Letters*, 2016.

[3] Jean-Yves Bouguet. Camera calibration toolbox for matlab, 2003.

[4] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008.

[5] Bolin Cai, Ami Zi, Jun Yang, Guoliang Li, Yang Zhang, Qiujie Wu, Chenen Tong, Wenxiang Liu, and Xiangcheng Chen. Accurate event camera calibration with fourier transform. *IEEE TIM*, 73:1–12, 2024.

[6] Shuolong Chen, Xingxing Li, Liu Yuan, and Ziao Liu. ekalibr: Dynamic intrinsic calibration for event cameras from first principles of events. *IEEE RA-L*, 10(7):7094–7101, 2025.

[7] Arturo De la Escalera and Jose María Armingol. Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. *Sensors*, 10(3):2027–2044, 2010.

[8] Burak Ercan, Onur Eker, Canberk Saglam, Aykut Erdem, and Erkut Erdem. HyperE2VID: Improving event-based video reconstruction via hypernetworks. *IEEE TIP*, 33: 1826–1837, 2024.

[9] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020.

[10] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *CVPR*, 2005.

[11] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE IROS*, 2013.

[12] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *CVPR*, 2018.

[13] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. In *CVPR*, 2019.

[14] Guillermo Gallego et al. Event-based vision: A survey. *IEEE TPAMI*, 44(1):154–180, 2020.

[15] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[16] Luna Gava, Marco Monforte, Massimiliano Iacono, Chiara Bartolozzi, and Arren Glover. Puck: Parallel surface and convolution-kernel tracking for event-based cameras. *arXiv preprint arXiv:2205.07657*, 2022.

[17] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. EKLT: Asynchronous, photometric feature tracking using events and frames. *IJCV*, 128(3):601–618, 2020.

[18] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *CVPR*, 2023.

[19] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE RA-L*, 6(3):4947–4954, 2021.

[20] Daniel Gorchard. DVScalibration. https://github.com/gorchard/DVScalibration.git, 2025.

[21] Thomas Gossard, Andreas Ziegler, Levin Kolmar, Jonas Tebbe, and Andreas Zell. ewand: An extrinsic calibration framework for wide baseline frame-based and event-based camera systems. In *IEEE ICRA*, 2024.

[22] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. pages 10–5244, 1988.

[23] Janne Heikkila. Geometric camera calibration using circular control points. *IEEE TPAMI*, 22(10):1066–1077, 2002.

[24] Kun Huang, Yifu Wang, and Laurent Kneip. Dynamic event camera calibration. In *IEEE IROS*, 2021.

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

[26] Javier Ibáñez, José M. Alonso, Jorge Sastre, Emilio Defez, and Pedro Alonso-Jordá. Advances in the approximation of the matrix hyperbolic tangent. *Mathematics*, 9(11), 2021.

[27] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE TPAMI*, 28(8):1335–1340, 2006.

[28] Juho Kannala, Janne Heikkilä, and Sami S Brandt. Geometric camera calibration. *Wiley encyclopedia of computer science and engineering*, 13(6):1–20, 2008.

[29] Jun-Sik Kim, Pierre Gurdjos, and In-So Kweon. Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration. *IEEE TPAMI*, 27(4): 637–642, 2005.

[30] Lars Krüger and Christian Wöhler. Accurate chequerboard corner localisation for camera calibration. *Pattern Recognition Letters*, 32(10):1428–1435, 2011.

[31] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE TPAMI*, 39(7):1346–1359, 2017.

[32] Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE TIP*, 31:2975–2987, 2022.

[33] Shijie Lin, Yinqiang Zhang, Lei Yu, Bin Zhou, Xiaowei Luo, and Jia Pan. Autofocus for event cameras. In *CVPR*, 2022.

[34] Nathan Matsuda, Oliver Cossairt, and Mohit Gupta. MC3D: Motion contrast 3D scanning. In *ICCP*, 2015.

[35] Nico Messikommer, Carter Fang, Mathias Gehrig, and Davide Scaramuzza. Data-driven feature tracking for event cameras. In *CVPR*, 2023.

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

[37] Manasi Muglikar, Mathias Gehrig, Daniel Gehrig, and Davide Scaramuzza. How to calibrate your event camera. In *CVPRW*, 2021.

[38] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *IEEE ICRA*, 2011.

[39] Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27.

[40] Alex Paul Pentland. A new sense for depth of field. *IEEE TPAMI*, PAMI-9(4):523–531, 1987.

[41] Lala Shakti Swarup Ray, Bo Zhou, Lars Krupp, Sungho Suh, and Paul Lukowicz. A synthetic benchmarking pipeline to compare camera calibration algorithms. In *ICPR*, 2024.

[42] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: an open event camera simulator. In *CoRL*, 2018.

[43] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE TPAMI*, 43(6):1964–1980, 2021.

[44] Robotics and Perception Group. rpg_dvs_ros. https://github.com/uzh-rpg/rpg_dvs_ros, 2025. Accessed: Jan. 3, 2025.

[45] Mohammed Salah, Abdulla Ayyad, Muhammad Humais, Daniel Gehrig, Abdelqader Abusafieh, Lakmal Seneviratne, Davide Scaramuzza, and Yahya Zweiri. E-calib: A fast, robust, and accurate calibration toolbox for event cameras. *IEEE TIP*, 33:3977–3990, 2024.

[46] Jianbo Shi and Tomasi. Good features to track. In *CVPRW*, pages 593–600, 1994.

[47] Chaehyeon Song, Dongjae Lee, Jongwoo Lim, and Ayoung Kim. Camera calibration via circular patterns: A comprehensive framework with measurement uncertainty and unbiased projection model. *IEEE TPAMI*, 2025.

[48] Sihao Sun, Giovanni Cioffi, Coen De Visser, and Davide Scaramuzza. Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs. events. *IEEE RA-L*, 6(2):580–587, 2021.

[49] Varun Sundar, Matthew Dutson, Andrei Ardelean, Claudio Bruschini, Edoardo Charbon, and Mohit Gupta. Generalized event cameras. In *CVPR*, 2024.

[50] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *IEEE IROS*, 2016.

[51] Yongqing Wang, Shiyu He, Yufan Fei, and Xingjian Liu. Motion-error-free calibration of event camera systems using a flashing target. *Optics Express*, 32(15):26833–26845, 2024.

[52] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE TPAMI*, 22(11):1330–1334, 2002.

[53] Zelin Zhang, Anthony J. Yezzi, and Guillermo Gallego. Formulating event-based image reconstruction as a linear inverse problem with deep regularization using optical flow. *IEEE TPAMI*, 45(7):8372–8389, 2023.

[54] Yajing Zheng, Zhaofei Yu, Song Wang, and Tiejun Huang. Spike-based motion estimation for object tracking through bio-inspired unsupervised learning. *IEEE TIP*, 32:335–349, 2022.

[55] Shifan Zhu, Zhipeng Tang, Michael Yang, Erik Learned-Miller, and Donghyun Kim. Event camera-based visual odometry for dynamic motion tracking of a legged robot using adaptive time surface. In *IEEE IROS*, 2023.

From Corners to Fiducial Tags: Revisiting Checkerboard Calibration for Event Cameras

Supplementary Material

Overview

In this supplementary material, we provide additional studies and results as follows:

- **Full description of our analysis:** We provide a full description of our analysis, which is about ER on checkerboard corner and edge described in Sec. 3.2. Description contains the details of previous works and our analysis.
- **Additional studies for checkerboard:** We explain which patches are filtered out during the corner initialization step and how they are removed. Then, we provide additional quantitative and qualitative results.
- **Additional studies for AprilTag:** We present results for binarization of various tag IDs and their corresponding corner indexing. In addition, we provide extra qualitative results for AprilTag detection as well as a calibration result using AprilTag.

A. Full Description of the Analysis in Sec. 3.2

A.1. Intensity Model on Checkerboard Corner

The ideal intensity distribution of checkerboard edge is usually defined with step function $\rho(\cdot)$. Then, checkerboard corner can be represented as the intersection of two edges [30]:

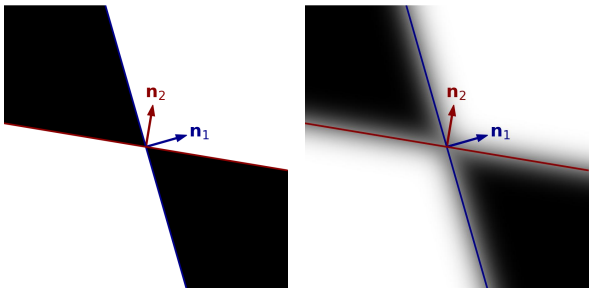
$$\hat{I}(\mathbf{x}) = \rho(\mathbf{n}_1^\top \mathbf{x})\rho(\mathbf{n}_2^\top \mathbf{x}), \quad \text{where } \rho(x) = \begin{cases} +1 & \text{if } x > 0, \\ -1 & \text{otherwise,} \end{cases} \tag{A1}$$

where $\mathbf{n}_i$ denotes unit normals to checkerboard edges (see Fig. A1 (a)) and, in this model, the ideal checkerboard corner is located $\mathbf{x} = \mathbf{0}$.

However, when frame-based camera captures ideal checkerboard $\hat{I}(\mathbf{x})$, there is a blurring effect by the lens (see Fig. A1 (b)). To define this blurring effect, Krüger *et al*. [30] consider the Point Spread Function (PSF) to be an ideal circular Gaussian filter g characterized by the parameter $\tilde{\sigma}$ [40]:

$$\tilde{I}(\mathbf{x}) = g(||\mathbf{x}||_2, \tilde{\sigma}) * \hat{I}(\mathbf{x}), \quad \text{where } g(x, \tilde{\sigma}) = \frac{1}{\sqrt{2\pi}\tilde{\sigma}} \exp\left(\frac{x^2}{2\tilde{\sigma}^2}\right). \tag{A2}$$

Because the Gaussian kernel g is separable, convolving a 1D step function $\rho(\cdot)$ with $g(\cdot, \tilde{\sigma})$ reduces to integrating the Gaussian along the edge normal direction. Since the integral of a Gaussian yields the error function, this step response becomes



(a) Ideal **(b) Blurred**

Figure A1. **Illustration of checkerboard intensity.**

$$g(x, \tilde{\sigma}) * \rho(x) = \text{erf}\left(\frac{x}{\sqrt{2}\tilde{\sigma}}\right), \tag{A3}$$

where $\text{erf}(x) = (2/\sqrt{\pi}) \int_0^x e^{-t^2}\, dt$. Hence, $\tilde{I}$ can be represented as:

$$\tilde{I}(\mathbf{x}) = G(\mathbf{n}_1^\top \mathbf{x}, \tilde{\sigma})\, G(\mathbf{n}_2^\top \mathbf{x}, \tilde{\sigma}), \quad \text{where } G(x, \tilde{\sigma}) = \text{erf}\left(\frac{x}{\sqrt{2}\tilde{\sigma}}\right). \tag{A4}$$

Finally, Krüger *et al*. [30] define the intensity model of the checkerboard corner $\tilde{I}$ by approximating $G(x, \tilde{\sigma})$ with a centered sigmoid $S(\cdot, \sigma)$, corresponding to Eq. (2):

$$\tilde{I}(\mathbf{x}) \approx S(\mathbf{n}_1^\top \mathbf{x}, \sigma) S(\mathbf{n}_2^\top \mathbf{x}, \sigma), \quad \text{where } S(x, \sigma) = \frac{2}{1 + e^{-x/\sigma}} - 1 \tag{A5}$$

where $\sigma = \tilde{\sigma}\sqrt{\pi/8}$ produces an equivalent slope of $G(x, \tilde{\sigma})$ and $S(x, \sigma)$ at $x = 0$. Note that since construction $S(x, \sigma) \in (-1, 1)$, $\tilde{I}(\mathbf{x}) \in [-1, 1]$ represents the idealized contrast pattern of the checkerboard, rather than the actual brightness observed by the camera.

A.2. Event Generation Rate

Event e_k is triggered based on log intensity change with constant threshold [14]:

$$\Delta L(\mathbf{x_k}, t_k) := L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k) = p_k C, \tag{A6}$$

where $L(\mathbf{x}, t) = \log I(\mathbf{x}, t)$ denotes the logarithmic image intensity, $C > 0$ is the constant threshold, and Δt_k represents the duration since the pixel last generated an event.

To analyze Event Rate (ER), it is necessary to first characterize how events accumulate over a given time interval.

Since the rate is defined in terms of the number of events generated within a temporal window, it is necessary to relate the accumulated events to the underlying log-intensity signal that drives event generation. Using the triggering condition in Eq. (A6), an event $e_k = (\mathbf{x}_k, t_k, p_k)$ can be written in distributional form as [36]:

$$e_k(\mathbf{x}, t) = p_k C\, \delta(t - t_k)\, \delta(\mathbf{x} - \mathbf{x}_k), \tag{A7}$$

where $\delta(\cdot)$ indicates the Dirac delta function. Over a time interval of length t, the events accumulated at pixel $\mathbf{x}$ can be written as:

$$\int_0^t \sum_k e_k(\mathbf{x}, \tau)\, d\tau = L(\mathbf{x}, t) - L(\mathbf{x}, 0) + \int_0^t \eta(\mathbf{x}, \tau)\, d\tau, \tag{A8}$$

where $\eta(\mathbf{x}, t)$ denotes sensor noise [14].

From the event models described in Eq. (A7) and Eq. (A8), Lin *et al.* [33] establish a relationship between the Event Rate (ER) and the image gradient. To establish relationship, they first define ER as follows:

$$R_e(\mathbf{x}, t, \Delta t) = \frac{\int_{t-\Delta t/2}^{t+\Delta t/2} \sum_k \delta(t - t_k)\delta(\mathbf{x} - \mathbf{x}_k) d\tau}{\Delta t}, \tag{A9}$$

where $R_e(\mathbf{x}, t, \Delta t)$ is ER within time interval $[t{-}\Delta t/2,\ t{+}\Delta t/2]$, which indicates the average number of events accumulated within a time interval of duration Δt. In addition, they formulate the intensity changing rate within time interval $[t{-}\Delta t/2,\ t{+}\Delta t/2]$, omitting noise tern in Eq. (A8):

$$\begin{aligned} \frac{\Delta L(\mathbf{x}, t)}{\Delta t} &= \frac{L(\mathbf{x}, t + \Delta t/2) - L(\mathbf{x}, t - \Delta t/2)}{\Delta t} \\ &= \frac{\int_{t-\Delta t/2}^{t+\Delta t/2} \sum_k e_k(\mathbf{x}, t) d\tau}{\Delta t}. \end{aligned} \tag{A10}$$

By taking the magnitude of Eq. (A10) and substituting the event model $e_k(\mathbf{x}, t)$ using Eq. (A7), the resulting intensity changing rate magnitude can be approximated by the event rate scaled by the constant C:

$$\begin{aligned} \left| \frac{\Delta L(\mathbf{x}, t)}{\Delta t} \right| &\approx \frac{C \int_{t-\Delta t/2}^{t+\Delta t/2} \sum_k \delta(t - t_k)) \delta(\mathbf{x} - \mathbf{x}_k) d\tau}{\Delta t} \\ &= C \cdot R_e(\mathbf{x}, t, \Delta t), \end{aligned} \tag{A11}$$

where the polarity p_k vanishes since the magnitude operator eliminates its sign.

Using the brightness constancy assumption, the intensity changing rate also can be approximated as [17]:

$$\frac{\Delta L(\mathbf{x}, t)}{\Delta t} \approx -\nabla L(\mathbf{x}, t) \cdot \mathbf{v}, \tag{A12}$$

where $\mathbf{v}$ is the constant motion vector on image plane [17]. Finally, combining Eq. (A11) and Eq. (A12), the event rate and gradient value are related as follows, corresponding to Eq. (3):

$$R_e(\mathbf{x}, t, \Delta t) \approx \frac{1}{C} |\nabla L(\mathbf{x}, t) \cdot \mathbf{v}|. \tag{A13}$$

For simplicity, we omit the temporal variable t in the following analysis, using $L(\mathbf{x})$ and $R_e(\mathbf{x})$ instead of $L(\mathbf{x}, t)$ and $R_e(\mathbf{x}, t, \Delta t)$ respectively.

A.3. Event Rate on Checkerboard Corner

To analyze Event Rate on checkerboard corner, we combine Eq. (A5) and Eq. (A13). Before combining them, we first reformulate centered sigmoid $S(x, \sigma)$, noting that $S(x, \sigma)$ can be expressed as a hyperbolic tangent function,

$$\begin{aligned} S(x, \sigma) &= \frac{2}{1 + e^{-x/\sigma}} - 1 \\ &= \frac{1 - e^{-x/\sigma}}{1 + e^{-x/\sigma}} \\ &= \frac{e^{x/2\sigma} - e^{-x/2\sigma}}{e^{x/2\sigma} + e^{-x/2\sigma}} \\ &= \tanh\left(\frac{x}{2\sigma}\right), \end{aligned} \tag{A14}$$

corresponding to Eq. (4). Then, we recall that the hyperbolic tangent function has the following Taylor expansion:

$$\tanh(x) = x - \frac{1}{3}x^3 + \frac{2}{15}x^5 - \cdots, \text{ when } |x| < \frac{\pi}{2}. \tag{A15}$$

Since we are primarily interested in the neighborhood of the ideal checkerboard corner, where its region is in $|x| \ll 1$, the Taylor expansion of $\tanh(x)$ is valid, and the higher-order terms can be neglected as

$$S(x, \sigma) = \tanh\left(\frac{x}{2\sigma}\right) \approx \frac{x}{2\sigma} + \mathcal{O}(x^3) \approx \frac{x}{2\sigma}, \tag{A16}$$

where $\mathcal{O}(x^3)$ denotes the third-order remainder in the Taylor expansion.

Based on this convention, we reformulate Eq. (A5) using the approximation in Eq. (A16), deriving its local analytical form around the checkerboard corner:

$$\tilde{I}(\mathbf{x}) \approx \frac{1}{4\sigma^2}(\mathbf{n}_1^\top \mathbf{x})(\mathbf{n}_2^\top \mathbf{x}), \quad \nabla \tilde{I}(\mathbf{x}) \approx G\,\mathbf{x}, \tag{A17}$$

where $G = \frac{1}{4\sigma^2}(\mathbf{n}_1\mathbf{n}_2^\top + \mathbf{n}_2\mathbf{n}_1^\top)$ is symmetric. Since $\tilde{I}(\mathbf{x})$ represents an idealized contrast pattern rather than an actual brightness value, we obtain the image intensity via a linear camera response applied to $\tilde{I}(\mathbf{x})$ [30]:

$$I(\mathbf{x}) = I_0 + \alpha\, \tilde{I}(\mathbf{x}). \tag{A18}$$

where α and I_0 correspond to the camera gain and offset, respectively. Through this linear response, $\tilde{I}(\mathbf{x}) \in [-1, 1]$ is mapped to the image intensity $I(\mathbf{x})$ within the dynamic range of camera, typically $[0, 255]$.

Having established the local intensity model of the checkerboard corner, we now analyze the event rate on checkerboard corner by substituting Eq. (A18) into the event rate formulation in Eq. (A13). Since $L(\mathbf{x}) = \log I(\mathbf{x})$, we have

$$\begin{aligned} \nabla L(\mathbf{x}) &= \frac{\nabla I(\mathbf{x})}{I(\mathbf{x})} \\ &= \frac{\alpha}{I_0 + \alpha\,\tilde{I}(\mathbf{x})} \nabla \tilde{I}(\mathbf{x}) \\ &\approx \frac{\alpha}{I_0 + \alpha\,\tilde{I}(\mathbf{x})} G\mathbf{x}, \end{aligned} \tag{A19}$$

corresponding to Eq. (9).

Since $\tilde{I}(\mathbf{x}) \approx 0$ around the corner, the term $\alpha/(I_0 + \alpha\,\tilde{I}(\mathbf{x}))$ can be approximated by α/I_0 and treated as constant. Therefore, Eq. (A13) can be rewritten as

$$\begin{aligned} R_e(\mathbf{x}) &\approx \frac{1}{C} \left|\nabla L(\mathbf{x}) \cdot \mathbf{v}\right| \\ &\approx \gamma \left|(G\mathbf{x}) \cdot \mathbf{v}\right| = \gamma \left|(G\mathbf{v})^\top \mathbf{x}\right|, \end{aligned} \tag{A20}$$

where $\gamma = \alpha/CI_0$ is a constant factor and Eq. (A20) corresponds Eq. (10).

To interpret the meaning of Eq. (A20), we first investigate whether G can be zero. Note that, in any valid checkerboard structure, the two edge normals cannot be parallel and are therefore linearly independent. With this geometric constraint, we examine whether G can vanish by multiplying G by $\mathbf{n}_1$ (or $\mathbf{n}_2$):

$$\begin{aligned} G\mathbf{n}_1 &= \frac{1}{4\sigma^2}(\mathbf{n}_1\mathbf{n}_2^\top + \mathbf{n}_2\mathbf{n}_1^\top)\mathbf{n}_1 \\ &= \frac{1}{4\sigma^2}\Big[(\mathbf{n}_1\mathbf{n}_2^\top)\mathbf{n}_1 + (\mathbf{n}_2\mathbf{n}_1^\top)\mathbf{n}_1\Big] \\ &= \frac{1}{4\sigma^2}\Big[(\mathbf{n}_2^\top\mathbf{n}_1)\mathbf{n}_1 + \mathbf{n}_2\Big]. \end{aligned} \tag{A21}$$

If G were zero, then $G\mathbf{n}_1 = \mathbf{0}$ must hold, which means that the following equation holds:

$$(\mathbf{n}_2^\top\mathbf{n}_1)\mathbf{n}_1 + \mathbf{n}_2 = \mathbf{0}. \tag{A22}$$

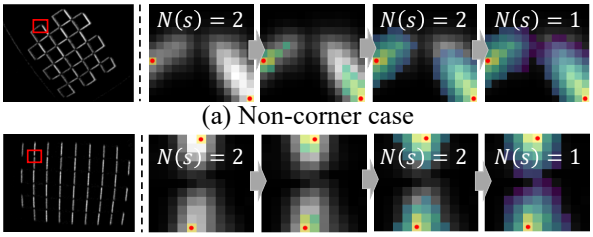
Then we can obtain following relationship between $\mathbf{n}_1$ and $\mathbf{n}_2$:

$$\begin{aligned} \mathbf{n}_2 &= -(\mathbf{n}_2^\top\mathbf{n}_1)\mathbf{n}_1 \\ &= \lambda\mathbf{n}_1, \quad \text{for some scalar } \lambda. \end{aligned} \tag{A23}$$

It is logically equivalent to $\mathbf{n}_2 \in \mathrm{span}\{\mathbf{n}_1\}$, which directly contradicts the valid checkerboard structure that requires the two edge normals to be linearly independent. Hence, G cannot be zero for any valid checkerboard structure.

Since G is nonzero for a valid checkerboard corner, the event rate at the exact corner satisfies $R_e = 0$ for any motion vector $\mathbf{v}$. Thus, from Eq. (A20), we find that

$$\mathbf{x} = \mathbf{0} \implies R_e(\mathbf{0}) = 0. \tag{A24}$$



(a) Non-corner case

(b) Single-edge case

Figure B2. **Filtered cases in pixel propagation.**

This result holds locally near a checkerboard corner, where the image gradients are modeled as a pair of intersecting edges. In contrast, the intensity model of the checkerboard edge can be modeled as

$$\tilde{I}(\mathbf{x}) \approx \frac{1}{2\sigma}(\mathbf{n}^\top\mathbf{x}), \quad \nabla\tilde{I}(\mathbf{x}) \approx \frac{\mathbf{n}}{2\sigma}. \tag{A25}$$

Here, the checkerboard edge corresponds to the set of points $\mathbf{x}$ that satisfy $\mathbf{n}^\top\mathbf{x} = \mathbf{0}$. Then we can calculate $\nabla L(\mathbf{x})$ in edge case as follows:

$$\nabla L(\mathbf{x}) \approx \frac{\alpha}{I_0 + \alpha\,\tilde{I}(\mathbf{x})} \frac{\mathbf{n}}{2\sigma}. \tag{A26}$$

As in the checkerboard corner case, since $\tilde{I}(\mathbf{x}) \approx 0$ near the edge, we can formulate ER on the checkerboard edge as follows:

$$\begin{aligned} R_e(\mathbf{x}) &\approx \frac{1}{C} \left|\nabla L(\mathbf{x}) \cdot \mathbf{v}\right| \\ &\approx \kappa\left|\mathbf{n} \cdot \mathbf{v}\right|, \quad \text{where } \kappa = \gamma/2\sigma, \end{aligned} \tag{A27}$$

and it corresponds to Eq. (13). This expression indicates that events are reliably generated along the edge as long as the motion vector is not parallel to the edge direction. Conversely, when the motion is orthogonal to the edge normal, no events are produced. In other words, events may be triggered only on one side of the checkerboard edge under such motion [45].

B. Additional Studies for Checkerboard

B.1. Filtered Cases in Pixel Propagation

As described in Sec. 3.4, in corner initialization process, we use the quantity $N(s)$ to filter out invalid patches. This section explains which cases are filtered out and why.

We broadly categorize discarded patches into two representative types: non-corner case and single-edge case (see Fig. B2).

Patches with $N(s) \neq 4$ are filtered out as invalid patches. (a) non-corner case and (b) single-edge case are common cases. During seed propagation, $N(s)$ does not increase and instead only decreases, so these patches are not selected as valid ones. (a) non-corner case corresponds to regions that

Table B1. **Intrinsics results ($\mu \pm \sigma$) on synthetic data with GT.**

Method	f_x	f_y	c_x	c_y
GT	250	250	150	100
E-Calib*	**250.16**±7.64	249.04±8.38	147.62±1.64	98.56±1.45
eKalibr*	251.20±8.06	248.81±4.69	151.01±0.91	97.95±2.75
E2VID	249.51±3.02	249.59±3.28	149.36±0.80	99.40±1.38
HyperE2VID	248.27±8.44	248.15±8.63	149.18±2.40	100.10±1.66
Ours	250.38±**2.60**	**250.00**±**2.36**	**149.39**±**0.44**	**99.93**±**0.69**

*: circular-based method; E-Calib shows the f_x mean closest to GT but high σ.

do not contain a true corner, yielding $N(s) = 2$, whereas (b) single-edge case corresponds to camera motion parallel to the checkerboard such that no corner is observed, also resulting in $N(s) = 2$.

B.2. Quantitative results

In this section, we present additional quantitative results on synthetic and real-world data, including comparisons with circular patterns, detection ratio analysis, and runtime measurements. Since E2Calib [37] adopts a reconstruction-based pipeline in which event-to-image conversion is followed by standard calibration (e.g., Kalibr [11] or OpenCV [4]), replacing E2VID [43] with a more recent reconstruction network does not change the framework. Hence, we additionally evaluate the pipeline using HyperE2VID [8] to ensure a fair comparison with recent reconstruction models.

To compare our method with circle-grid calibration methods [6, 45], we synthesize data in Blender using fixed calibration targets and randomly generated camera trajectories, varying only the calibration pattern. Event streams are simulated with ESIM [42]. As shown in Tab. B1, Our method estimates parameters close to GT with lower standard deviations averaged over 10 sequences.

For real-world experiments, we extend evaluation in Sec. 4.2 by incorporating HyperE2VID and report only the standard deviation. Using the same 7×10 checkerboard dataset in the paper, we set the sampling number B to 15 due to fewer corresponding images, which increases the standard deviations compared to the paper; nevertheless, our method still achieves the lowest values (see Tab. B2).

We further compare the detection ratio and runtime with E2VID. Although E2VID detects more images, calibration accuracy is governed by corner quality and pose diversity, and our fewer but higher-quality detections yield better results (see Tab. B3). Treating grayscale reconstruction in E2VID and IWE generation as the same event-processing step, we measure runtime of each step. Our method remains substantially more efficient (see Tab. B4).

B.3. Qualitative results

We provide additional qualitative results for our corner detection method in this section (see Fig. B3). These examples illustrate its robustness across diverse motion trajectories and checkerboard poses.

As a qualitative complement to Tab. 2, we further provide qualitative results on checkerboard corner location in Fig. B4. Our method produces corner locations that are nearly coincident with the pseudo ground truth (*i.e.,* frame-

Table B2. **Intrinsics results ($\pm\sigma$) on real data.**

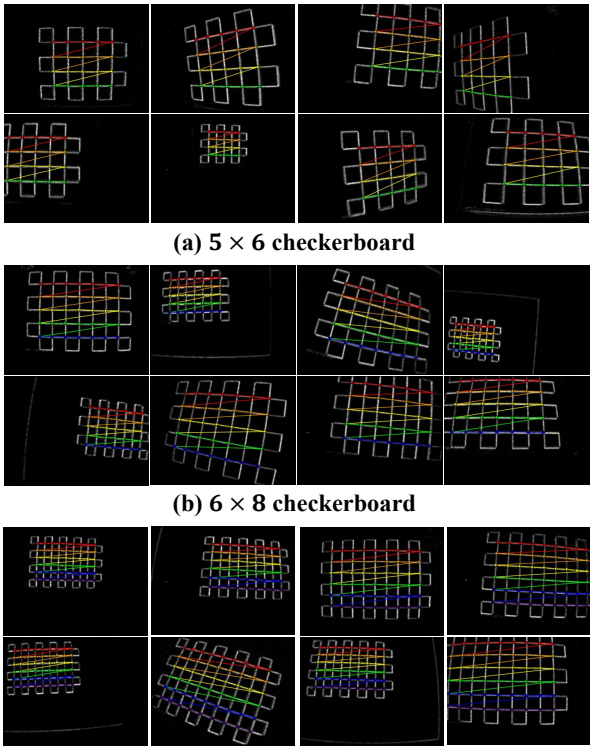
	σf_x	σf_y	σc_x	σc_y	σk_1
E2VID	8.51	8.47	1.59	2.79	0.018
HyperE2VID	9.45	9.26	1.46	4.38	0.020
Ours	**3.79**	**3.80**	**0.73**	**1.31**	**0.009**

Table B3. **Detection ratio and intrinsic results.**

Method	#img	f_x	f_y	c_x	c_y
GT	300	250	250	150	100
E2VID	**187**	244.12	243.37	148.35	103.21
Ours	99	**247.38**	**248.08**	**150.31**	**99.39**

Table B4. **Comparison of runtime (sec).**

Method	Event processing	Corner detection	Total
E2VID	182.96	**59.86**	242.82
Ours	**35.15**	70.58	**105.73**



(a) 5×6 checkerboard

(b) 6×8 checkerboard

(c) 7×10 checkerboard

Figure B3. **Qualitative results for corner detection**

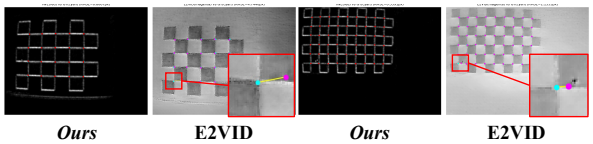


Figure B4. **Comparison with E2VID** Cyan denotes pseudo ground-truth corners obtained from frame-based detection; pink corresponds to E2VID, and red to our method.

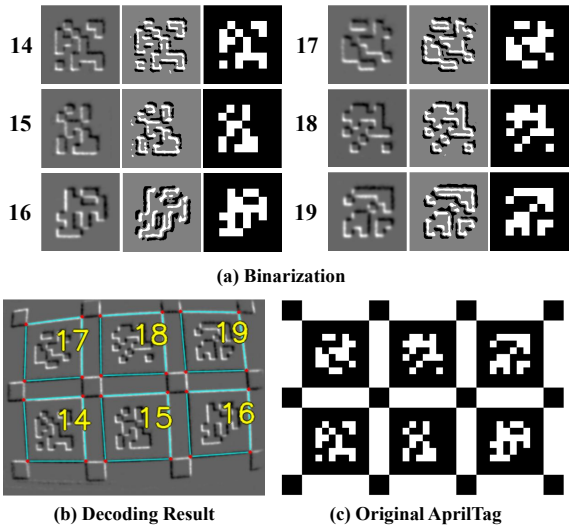


Figure C5. **Qualitative results of binarization.**

based), whereas E2VID exhibits systematic offsets and less consistent alignment, particularly around blurred or low-contrast regions.

B.4. AprilTag Detection

We present various matching results for AprilTag detection in various tag IDs. As shown in Fig. C5, our binarization of the quad regions extracted from the pIWE remains reliable across different tag IDs despite variations in their visual patterns. Although a few grid cells may be misclassified during binarization, the resulting patterns remain within the error tolerance margin inherently built into the AprilTag design, which is intended to accommodate such small perturbations. Additional qualitative results are illustrated in Fig. C6, where all visible AprilTags are successfully detected even when the board is rotated.

C. Additional Studies for AprilTag

C.1. Corner Indexing from AprilTag

The advantage of AprilTag in camera calibration lies in its ability to create diverse observation sets, even for partial observations, by assigning a unique number to each corner. This correspondence eliminates the ambiguity that typically arises in calibration using checkerboard or circle-grid,

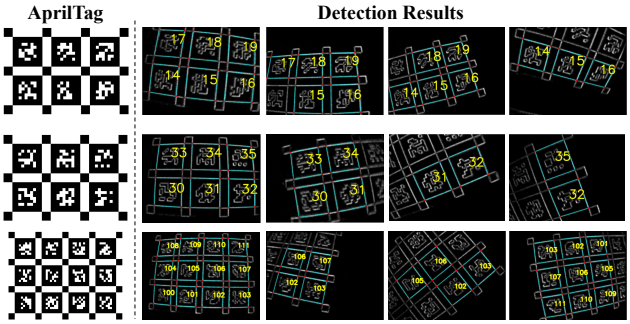


Figure C6. **Qualitative results for AprilTag detection.** The rightmost example in the last row illustrates a 180° rotation.

Figure C7. **AprilTag detection and corner indexing results for various rotations.** Each quad and detected tag ID is colored cyan and yellow, respectively. The unique number for the corner is colored the same as its corner color.

Table C5. **Calibration results using AprilTag.**

Parameters	Values
Intrinsics (f_x, f_y, c_x, c_y)	332.25, 332.31, 179.10, 137.02
Distortion (k_1, k_2, p_1, p_2)	-0.3265, 0.1596, -0.0042, 0.00138
Reprojection error	0.516

where corner ordering must be inferred from geometry.

As shown in Fig. C7, our detection algorithm robustly identifies each corner even under rotation, occlusion, and other extreme conditions. This property enables stable calibration input without requiring an additional corner ordering process.

C.2. Calibration using AprilTag

Using a robust corner indexing process, we perform event camera calibration with AprilTag. As shown in Tab. C5, our pipeline yields reliable parameter estimates when applied to AprilTag.