

Polarity-Focused Denoising for Event Cameras

Chenyang Shi^{ID}, Boyi Wei, Xiucheng Wang, Hanxiao Liu, Yibo Zhang^{ID},
Wenzhuo Li^{ID}, Ningfang Song^{ID}, and Jing Jin^{ID}

Abstract—Event cameras, which are highly sensitive to light intensity changes, often generate substantial noise during imaging. Existing denoising methods either lack the speed for real-time processing or struggle with dynamic scenes, mistakenly discarding valid events. To address these issues, we propose a novel dual-stage polarity-focused denoising (PFD) method that leverages the consistency of polarity and its changes within local pixel areas. Whether due to camera motion or dynamic scene changes, the polarity and its changes in triggered events are highly correlated with these movements, allowing for effective noise handling. We introduce two versions: PFD-A, which excels at reducing background activity (BA) noise, and PFD-B, which is designed to address both BA and flicker noise. Both versions are lightweight and computationally efficient. The experimental results show that PFD outperforms benchmark methods in terms of the SNR and ESR metrics, achieving state-of-the-art performance across various datasets. Additionally, we propose an FPGA implementation of PFD processes that handles each event in just 7 clock cycles, ensuring real-time performance. The codes are available at <https://github.com/shicy17/PFD>.

Index Terms—Event camera, denoising method, BA noise, flicker noise.

I. INTRODUCTION

EVENT cameras are bioinspired vision sensors. Unlike traditional cameras that capture absolute light intensity values through global exposure, event cameras encode relative changes in light intensity, producing a sparse, asynchronous stream of events. Each event is represented as a tuple $e = (x, y, p, t)$. This tuple signifies that at the pixel coordinates (x, y) , an event of polarity $p \in \{-1, +1\}$ occurs at the timestamp t . A positive polarity (ON) indicates an increase in light intensity above the threshold, whereas a negative polarity (OFF) indicates a decrease in intensity below the threshold. This unique imaging approach endows event cameras with the remarkable advantages of high temporal resolution, a high dynamic range, and low power consumption. Thus, event cameras outperform conventional cameras in a variety of challenging scenarios, such as object detection [1], [2], [3],

motion analysis [4], [5], [6], [7], [8], simultaneous localization and mapping (SLAM) [9], [10], [11], and navigation [12].

However, the sensitivity of event cameras to light intensity fluctuations poses the challenge of generating significant noise events among the triggered events. These noise events consume valuable data transmission bandwidth, thereby increasing the computational load and potentially compromising the performance of downstream tasks. Given these implications, the effective denoising of events emerges as a crucial and fundamental process, pivotal to harnessing the full potential of this innovative imaging pattern.

Event cameras are susceptible to various types of noise. These can be broadly categorized into noise originating from the pixel circuit and noise induced by external environmental factors. Pixel circuit noise primarily comprises background activity (BA) noise [13], hot noise [14], and $1/f$ noise (also known as shot noise) [14]. Among these, BA noise is the most prevalent, significantly affecting the signal-to-noise ratio of the event stream. Hot noise, manifesting as pixel corruption in event cameras, can be effectively reduced by leveraging the event intensity of impacted pixels. The impact of $1/f$ noise is minimal, posing negligible adverse effects on the event flow. Conversely, flicker noise, caused by external light interference, can be effectively filtered out using the frequency of polarity changes in the events. Our primary focus is on BA noise, which uniquely occurs without any changes in light intensity in the scene. Simultaneously, while addressing BA noise removal, we introduce a derivative approach aimed at mitigating flicker noise.

Prior denoising methods predominantly rely on assessing the spatiotemporal correlations, intensity, and motion consistency of events. These methods exhibit commendable denoising capabilities under simple conditions. However, their effectiveness markedly diminishes in scenarios involving rapid camera movement and complex textures. While some approaches maintain high performance under these conditions, their computational complexity renders them unsuitable for real-time applications.

Leveraging the polarity characteristics of events, we have developed a polarity-focused denoising method (PFD) that is both straightforward and highly effective. The proposed PFD method not only accomplishes efficient noise reduction but also operates at a notably rapid pace, making it well suited for real-time implementation. We also deployed the hardware implementation of PFD on a field-programmable gate array (FPGA) to validate its efficiency. Our main contributions are summarized as follows:

Received 19 August 2024; revised 29 October 2024, 28 November 2024, and 5 December 2024; accepted 14 December 2024. Date of publication 17 December 2024; date of current version 7 May 2025. This article was recommended by Associate Editor D. Grois. (Corresponding author: Jing Jin.)

Chenyang Shi, Boyi Wei, Xiucheng Wang, Yibo Zhang, Wenzhuo Li, Ningfang Song, and Jing Jin are with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China, and also with the Tianmushan Laboratory, Hangzhou 311115, China (e-mail: shicy@buaa.edu.cn; jinjing@buaa.edu.cn).

Hanxiao Liu is with the Department of Precision Instrument, Tsinghua University, Beijing 100084, China.

Digital Object Identifier 10.1109/TCSVT.2024.3519430

- A highly efficient and rapid, dual-stage polarity-focused denoising (PFD) method for event cameras is proposed, which achieves state-of-the-art performance on various benchmark datasets.
- We introduce two versions of PFD: PFD-A, which targets the removal of BA noise, and PFD-B, which is capable of eliminating both BA noise and flicker signals. Additionally, we have developed group-of-event and event-by-event processing variants of PFD to streamline its deployment.
- We design an efficient hardware implementation for the proposed denoising method and verify it on resource-limited FPGA hardware.

Note that a group-of-event process refers to accumulating events over specified time intervals, followed by processing the accumulated events in batches. In contrast, an event-by-event process does not involve accumulation, and each event is processed individually.

The remainder of this article is organized as follows. First, Section II reviews recent works in this field. Next, we present our proposed denoising method in Section III. The hardware implementations of the proposed method are introduced in Section IV. The experimental validation of this method is presented in Section V. Finally, the article is concluded with a discussion in Section VI and Section VII.

II. RELATED WORK

Denoising in event cameras is a critical and foundational computing task that demands not only effective noise reduction but also rapid computational processes. There are four main types of denoising methods for event cameras: those based on the spatiotemporal correlation of events [19], [22], [23], [24], those utilizing event density [17], [25], [26], those relying on motion consistency [27], [28], [29], [30], and learning-based methods [18], [20], [31], [32], [33], [34].

Event-based denoising methods utilize spatiotemporal correlation and event density, operating on the principle that noise is unrelated to changes in scene light intensity. Liu et al. [22] introduced the NNb method, which assesses the validity of a current event by examining the presence of an event in its spatiotemporal vicinity. While this approach is effective, its performance is notably diminished in scenarios containing rapid motion. Wu et al. [17] proposed a probabilistic undirected graph model (PUGM). It transforms the BA denoising problem into an energy optimization problem. However, its calculation is extremely complex and cannot be run in real time. Guo and Delbruck [19] proposed a double window filter (DWF) specifically for BA noise removal. The DWF adeptly establishes the spatiotemporal relationship of events between two queues. Xu et al. [15] proposed a denoising method by enhancing spatiotemporal correlations in events, distinguishing valid signals from noise through the analysis of overlapping events. However, the criterion used in these methods is highly important. A criterion that is too simplistic risks excluding valid events erroneously, whereas an overly complex criterion can severely decelerate computational speed.

Methods based on motion consistency necessitate the estimation of either the motion of the event camera or that

of moving objects within the scene. EV-Gait [28] discriminates between noise and valid events through optical flow estimation. Wang et al. [29] proposed an augmentation of denoising processes with intensity maps. Mohamed et al. [30] developed a noise reduction strategy leveraging event clustering. However, these approaches are characterized by computational complexity, limiting their applicability in real-time scenarios.

Recently, numerous learning-based denoising methods have been introduced, often delivering impressive denoising performance. However, many of these methods necessitate extra information sources or demand significant computing resources, challenging the efficiency and resource constraints fundamental to denoising algorithms. A lightweight multilayer perceptron filter (MLPF) [19], [20] is specifically tailored for denoising on embedded hardware. EDnCNN [31] leverages active pixel sensor (APS) frames and inertial measurement unit (IMU) data to estimate event probabilities, utilizing a classification network to separate events from noise. Guided event filtering (GEF) [32] introduces a frame-based camera to guide the denoising of event cameras. However, the computational processes of these two methods are complex and require additional information sources. EventZoom [18] achieves good denoising performance, but it generates a new event sequence, which compromises data integrity.

While existing methods have shown commendable denoising capabilities, they are often costly because of their computational complexity or supplementary data from extra sensors. Moreover, the effectiveness of these methods significantly decreases in scenes with textured backgrounds and fast motions. This is primarily due to their inability to effectively distinguish between noise and valid signals in moving scenarios, where aliasing occurs. Furthermore, many of these methods overlook the importance of event polarity, which is a crucial factor in the denoising process. To address these gaps, we propose a novel denoising method capitalizing on the polarity of events. Our proposed method is adept at handling both stationary and dynamic conditions of event cameras and is designed to be sufficiently simple for a real-time process.

III. PROPOSED METHODS

In this section, we first introduce the coarse denoising method, which is based on polarity-based spatiotemporal correlation. Next, a fine denoising method that relies on polarity-based motion consistency is proposed. Moreover, we present both group-of-event and event-by-event implementations for the proposed fine denoising method.

A. Overview of the Proposed Method

The proposed method employs a dual-stage processing approach, as shown in Figure 2. Initially, the event stream undergoes processing through the polarity consistency-based coarse filtering method. Subsequently, the events that have been retained are further refined via the motion consistency-based fine filtering method. This sequential processing culminates in the output of a denoised event stream.

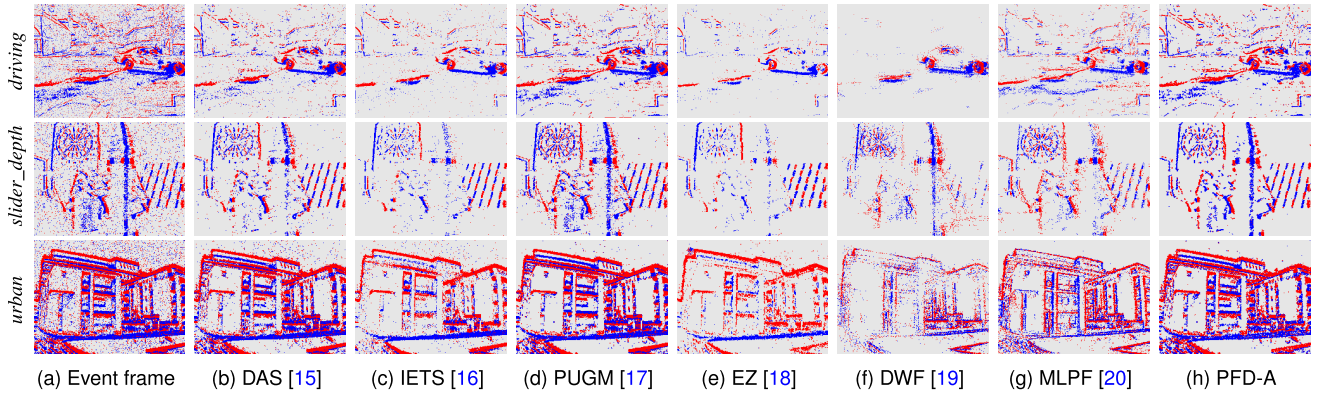


Fig. 1. Qualitative comparison of the DND21 [19] and ECD17 [21] datasets with 30% noise. The *driving* sequence captures urban car journeys, featuring rapid turns and diverse movements. Our proposed PFD-A excels in this environment and adeptly handles significant polarity changes due to motion. Note that EZ represents the EventZoom method.

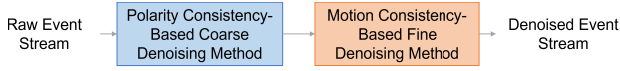


Fig. 2. Computational structure of the proposed method.

B. Polarity Consistency-Based Coarse Denoising Method

The polarity of an event is indicative of the direction of variation in light intensity. Event cameras boast exceptionally high temporal resolution, allowing for a nuanced capture of these changes. Within a sufficiently small spatiotemporal vicinity, it is reasonable to assume uniformity in light intensity alterations. Consequently, this implies that the polarity of events occurring within this spatiotemporal neighborhood remains consistent. We introduce a coarse denoising method based on polarity consistency, which is an adaptation of the classical denoising method NNb [22] that utilizes spatiotemporal correlation.

Initially, we define a polarity timestamp map M_{pt} , sized to match the resolution of the event camera. Each cell in M_p stores the timestamp of the most recent event. Specifically, the map M_{pt}^{ON} records the timestamps of the latest positive events, whereas the map M_{pt}^{OFF} captures those of the latest negative events.

$$M_{pt}^{ON} = \{(x, y) | M_{pt}^{ON}(x, y) = t, p = +1\} \quad (1)$$

$$M_{pt}^{OFF} = \{(x, y) | M_{pt}^{OFF}(x, y) = t, p = -1\} \quad (2)$$

For every event $e = (x, y, p, t)$, we examine its $n \times n$ (n is an odd number) neighborhood $S(x, y)$ on M_{pt}^{ON} or M_{pt}^{OFF} to determine if any event sharing the same polarity occurred within the time window δt_c . Note that each event has its own time window.

$$S = \{(x, y) | x - \frac{n-1}{2} \leq x \leq x + \frac{n-1}{2}, y - \frac{n-1}{2} \leq y \leq y + \frac{n-1}{2}\} \quad (3)$$

In cases where an event is triggered at a boundary, only its 5-neighborhood is considered, whereas for events at a vertex, we restrict our examination to its 3-neighborhood. The number of events with the same polarity in the spatiotemporal

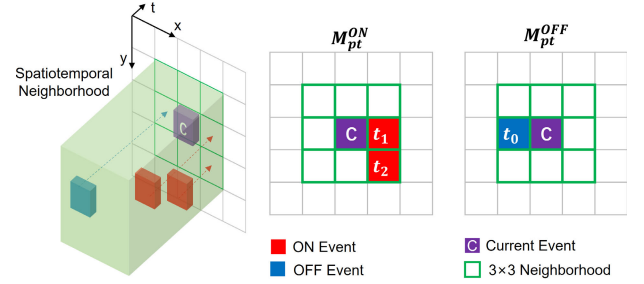


Fig. 3. Polarity consistency-based coarse denoising method. In this example, we check the 3×3 neighborhood of each event. To determine the validity of a current event, we check for the presence of μ nearby events that share its polarity. For example, if $\mu = 2$ and the polarity of the current event is positive (ON), it is deemed valid according to M_{pt}^{ON} . Similarly, if $\mu = 1$ and the event has a negative polarity (OFF), it meets the validation criteria under M_{pt}^{OFF} . This approach ensures that an event is considered significant only when it is part of a larger, consistent pattern within its immediate spatiotemporal neighborhood.

neighborhood is N_p , which is described as follows:

$$N_p = \begin{cases} \sum_{(x,y)} I_{(t-M_{pt}^{ON}(S(x,y)) < \delta t_c)}, p = +1 \\ \sum_{(x,y)} I_{(t-M_{pt}^{OFF}(S(x,y)) < \delta t_c)}, p = -1 \end{cases} \quad (4)$$

where $I(\cdot)$ is an indicator function, which takes a value of 1 when the condition is satisfied and 0 otherwise. An event is deemed a valid signal if it satisfies the following equation:

$$N_p \geq \mu \quad (5)$$

where μ is a parameter. Figure 3 depicts the calculation process of the polarity consistency-based coarse denoising method.

C. Motion Consistency-Based Fine Denoising Method

The event camera captures dynamic information that exhibits inherent motion consistency. For example, the background motion captured during the movement of a camera aligns with its own motion, and the motion within each part of independent dynamic objects in the scene is uniform. We posit that events occurring in a close spatiotemporal neighborhood are the result of consistent motion, whereas the noise distribution is motion independent. Based on this principle,

we introduce a refined denoising method that leverages motion consistency.

1) *Theorem of Consistency in the Polarity Changes of Events*: We utilize the consistency in the polarity changes of events to distinguish between effective signals and noise. Event cameras detect changes in brightness, which can be mathematically linked to the motion of objects in the scene. Below, we present a proof that demonstrates how polarity changes in events are caused by motion.

Let $L(x, y, t)$ represent the logarithm of the brightness at a pixel (x, y) at timestamp t . The brightness is related to the image intensity $I(x, y, t)$ triggered when the change in brightness $\Delta L(x, y, t)$ exceeds a threshold C :

$$\Delta L(x, y, t) = L(x, y, t) - L(x, y, t - \Delta t) \geq C \quad (6)$$

Consider an object moving in the scene with velocity $\mathbf{v} = (v_x, v_y)$. The position of the object changes over time as follows:

$$x(t) = x(0) + v_x t, \quad y(t) = y(0) + v_y t \quad (7)$$

Under the brightness constancy assumption, the brightness remains constant over time:

$$L(x(t), y(t), t) = L(x(0), y(0), 0) \quad (8)$$

A first-order Taylor expansion is used for $L(x, y, t)$ around the initial position:

$$L(x(t), y(t), t) \approx L(x(0), y(0), 0) + \left(\frac{\partial L}{\partial x} v_x + \frac{\partial L}{\partial y} v_y + \frac{\partial L}{\partial t} \right) \Delta t \quad (9)$$

Thus, the change in brightness is given by:

$$\Delta L(x, y, t) = \frac{\partial L}{\partial x} v_x + \frac{\partial L}{\partial y} v_y + \frac{\partial L}{\partial t} \quad (10)$$

The polarity p of an event is defined as:

$$p = \text{sign}(\Delta L(x, y, t)) \quad (11)$$

Substituting the expression for $\Delta L(x, y, t)$ gives:

$$p = \text{sign} \left(\frac{\partial L}{\partial x} v_x + \frac{\partial L}{\partial y} v_y + \frac{\partial L}{\partial t} \right) \quad (12)$$

This derivation shows that the polarity changes detected by an event camera are a direct consequence of object motion in the scene. Consistent motion across a region results in consistent polarity changes, whereas noise, which lacks motion consistency, does not exhibit a coherent polarity pattern. This approach circumvents the need for complex techniques such as optical flow estimation for motion assessment, thereby maintaining the simplicity and efficacy of our denoising method.

2) *Method*: Let us select a short time window δt_f . We then define a grid-shaped event polarity map M_p , an event counting map M_c , and a polarity frequency map M_f [35], whose size is the same as the resolution of the event camera. Each pixel of M_p records the polarity of the latest event triggered at that pixel.

$$M_p = \{(x, y) | M_p(x, y) = p_i\} \quad (13)$$

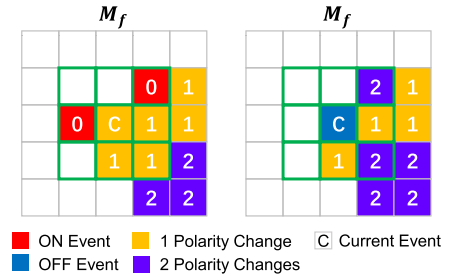


Fig. 4. An example of the motion consistency-based fine denoising method. In this example, we examine the neighborhood of a current event within a 3×3 matrix. In this context, a yellow pixel signifies the pixel where one polarity change of events occurs, whereas a purple-blue pixel indicates the pixel with two polarity changes of events. We assign a value of 1 to τ . On the left side, denoted as M_f , we calculate M_a to be 3 and N_e to be 5. Following Equation 16, we categorize this event as valid. Conversely, on the right side within M_f , the pixel experiencing the current event shows no polarity change within the time window. We then calculate M_a to be 6 and N_e to be 4. According to Equation 16, the event is identified as noise.

On the other hand, each pixel of M_f records the number of polarity changes of the event triggered at that pixel within the selected time window δt_f .

$$M_f(x, y) = \begin{cases} M_f(x, y) + 1, & M_p(x, y) \cdot p_i = -1 \\ M_f(x, y), & M_p(x, y) \cdot p_i = 1 \\ 0, & M_p(x, y) = 0 \end{cases} \quad (14)$$

where $M_f(x, y)$ represents the value of (x, y) on M_f and p_i denotes the polarity of the latest event. An example of M_f is shown in Figure 5b. Polarity changes typically occur at high-contrast edges as the object moves. M_c records the number of events triggered in each pixel.

Next, we calculate the total number M_a of polarity changes for all the pixels within the neighborhood $S(x, y)$.

$$M_a = \sum_{i=1}^n M_f((x_i, y_i) | S(x, y)) \quad (15)$$

For each pixel in $S(x, y)$, N_e is defined as the count of pixels that triggered at least one event within the specified time window δt_f . We then check whether the number of polarity changes occurring at the current pixel aligns with the average polarity changes in the nonempty neighborhood. For pixels with $N_e \geq m(m > 0)$, we establish the following criteria to differentiate valid events from noise events in our analysis:

$$|M_f - \frac{M_a}{N_e}| \leq \tau \quad (16)$$

where τ is a parameter. When an event satisfies these conditions, it is classified as a valid event; conversely, if it fails to meet these criteria, it is deemed an invalid event. In summary, this criterion assesses the consistency of polarity changes across nonempty neighborhoods. The proposed dual-stage denoising method with Equation 16 is named PFD-A. We demonstrate an example of whether an event is valid or not in Figure 4.

Furthermore, we propose an additional criterion designed to eliminate flicker noise events. We first analyze the characteristics of flicker noise and how the polarity change density criterion distinguishes it from true motion events.

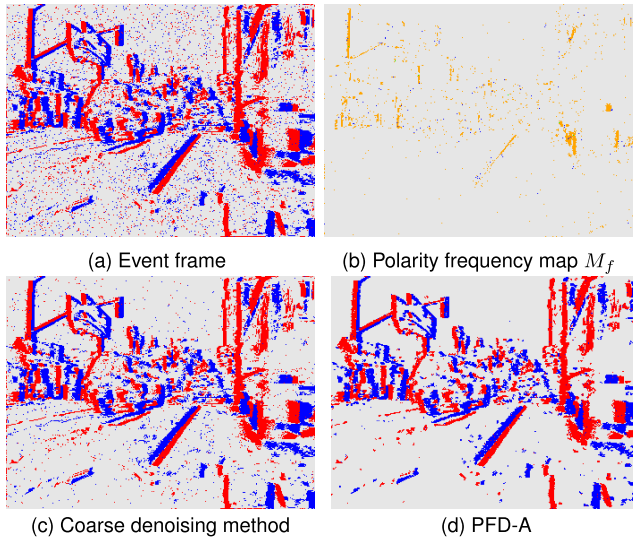


Fig. 5. Demonstration of each step of PFD. (a) displays an event frame in the *driving* sequence of the DND21 dataset [19] with 30% noise, where each pixel captures the latest event. Positive and negative events are differentiated by color, with red indicating positive events and blue indicating negative events. (b) corresponds to this event frame, showcasing a polarity frequency map. Color coding is used to represent the number of polarity changes: orange for a single change and purple-blue for two changes or more within a pixel. Polarity changes in events are strongly associated with motion. (c) depicts the outcome achieved through the coarse denoising method. (d) presents the outcomes following initial coarse denoising and subsequent fine denoising with Equation 16.

Let $D(x, y) = \frac{M_a(x, y)}{N_e(x, y)}$ represent the polarity change density at pixel (x, y) over a time window δt , where $M_a(x, y)$ is the number of polarity changes and $N_e(x, y)$ is the number of events.

For flicker noise, which manifests as frequent, unstructured polarity changes at the same pixel, the polarity change density $D_{FN}(x, y)$ is much greater:

$$D_{FN}(x, y) \gg \sigma \quad (17)$$

where σ is the threshold.

In other words, the polarity change density in areas affected by flickers is significantly greater than that in areas without flickers. This difference in polarity change density is the most distinguishing characteristic between the flickering and nonflickering regions. Thus, we employ the density of polarity changes within the neighborhood to identify and eliminate flicker events, which is described as follows:

$$\frac{M_a}{N_e} \leq \sigma \quad (18)$$

The proposed dual-stage denoising method with Equation.18 is named PFD-B.

We demonstrate each step of the PFD method in Figure 5. Figure 5c illustrates the outcomes following coarse denoising, while Figure 5d displays the further results after fine denoising processing. Initially, coarse denoising effectively eliminates a substantial portion of the noise, yet residual noise persists. The subsequent fine denoising phase further eradicates the remaining noise, demonstrating the efficacy of our proposed dual-stage denoising approach.

3) *Event-by-Event Processing Version*: Notably, while this motion consistency-based fine denoising method primarily

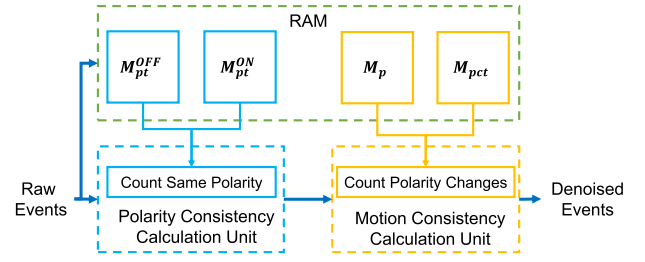


Fig. 6. Data flow diagram of the hardware PFD-A.

addresses groups of events, it is also capable of processing events on an individual basis. With just a few adjustments, we outline the steps to implement this event-by-event processing approach.

First, we establish the polarity change timestamp map M_{pct} , characterized by its dimensions $H \times W \times d$. H and W represent the height and width of the resolution of the event camera, respectively, while d corresponds to the length of the first-in, first-out (FIFO) queue. If a polarity change occurs in comparison to the preceding event at the same pixel, the timestamp of the current event is queued in $M_{pct}(x, y)$.

$$M_{pct}^{n+1}(x, y) = \text{Enqueue}(M_{pct}^n(x, y), t) \quad (19)$$

where $\text{Enqueue}(M_{pct}(x, y), t)$ denotes the process of inserting timestamp t into the end of the queue $M_{pct}(x, y)$. $M_{pct}^n(x, y)$ symbolizes the state of the queue prior to the insertion operation, whereas $M_{pct}^{n+1}(x, y)$ represents the state of the queue subsequent to the insertion of the element.

To process a current event $e(x, y, p, t)$, we examine a specific time window δt_f , retracing from the timestamp of this event. Note that each event has its own corresponding time window. Specifically, we examine the number of timestamps within the time range $(t - \delta t_f, t)$ in $M_{pct}(x, y)$. In other words, this denotes the number of polarity changes within δt_f . We subsequently apply the same evaluation in Equation.16 or 18 as previously conducted within the neighborhood $S(x, y)$ corresponding to this event.

IV. HARDWARE IMPLEMENTATION OF THE PROPOSED METHOD ON AN FPGA

Event streams constitute a type of streaming data. Hardware components such as FPGA and application-specific integrated circuits (ASICs) are capable of delivering high-throughput, low-latency streaming data processing circuits by leveraging techniques such as data pipelines and data parallelism. Therefore, these kinds of hardware are particularly well suited for processing data from event cameras.

In this section, an FPGA-based implementation of the proposed PFD-A method tailored for processing events on an event-by-event basis is presented. The event-by-event processing approach fully leverages the strengths of FPGA data pipelines.

The hardware implementation comprises three key components: a timestamp storage unit, a polarity consistency calculation unit, and a motion consistency calculation unit. The data flow diagram of the hardware PFD-A is depicted in Figure 6.

The timestamp storage unit preserves the timestamps of incoming events, with four random access memories (RAMs) maintained at a size of $W \times H \times B/N$, where W and H denote the pixel dimensions. Bit splicing is employed on adjacent N pixels to increase the data retrieval speed. B denotes the bit width of the data. Two of the RAMs store the timestamps of the most recent positive and negative events of each pixel (which correspond to M_{pt}^{ON} and M_{pt}^{OFF}), facilitating polarity consistency evaluation. The remaining two RAMs serve motion consistency assessment purposes: one retains the polarity of the most recent event to detect polarity changes (which corresponds to M_p), while the other stores timestamps of the last d events triggering such changes (which corresponds to M_{pct}). Note that the timestamp of an event is allocated a 16-bit width, whereas the polarity of an event is recorded in a 1-bit width. We employ bit-stitching to combine timestamps for each pixel and utilize bit assignment for data updates, increasing the calculation speed.

The polarity consistency calculation unit scrutinizes the timestamp of events with the same polarity within the 3×3 pixel neighborhood, employing subtractors and accumulators to calculate the number of events consistent with Equation 4. When neighboring pixels are processed independently, calculations occur concurrently, ensuring swift execution.

The motion consistency calculation unit adopts a circuit structure similar to the polarity consistency calculation unit. Initially, it determines polarity changes for each pixel within the neighborhood in parallel, culminating in judgment based on Equation 16.

V. EXPERIMENTAL METHODOLOGY

In this section, the datasets for evaluation are first introduced. The evaluation metrics and implementation details are then illustrated. Finally, the experimental results are presented.

A. Datasets

We adopt the benchmark datasets ECD17 [21], DND21 [19], and E-MLB [36] for the evaluation of BA noise removal.

The DND21 dataset was recorded using a DAVIS346¹ mono event camera with a resolution of 346×260 , and ECD17 was obtained using a DAVIS240C event camera with a resolution of 240×180 . DND21 and ECD17 are datasets inherently free from noise. We incorporate noise into the dataset at various intensities: 15%, 30%, and 50%. These percentages reflect the proportion of noise relative to the total number of events in the original sequence. The distribution of the introduced noise follows a Poisson pattern, aligning with conventional paradigms [15], [17], [19].

The E-MLB dataset is designed to replicate various lighting conditions by placing neutral density (ND) filters with differing transmittances in front of the lens of the event camera. This setup inherently introduces noise into the dataset, simulating real-world environmental variations. The E-MLB dataset comprises 96 sequences in total, encompassing a wide range of motion speeds and complex features. Note that

all 96 sequences in the E-MLB dataset are used for the experiments.

Furthermore, we employ the light interference event dataset (LIED) [35] to assess the efficacy of our proposed PFD-B method in eliminating flicker noise. This dataset is curated to benchmark the performance of algorithms designed to mitigate light interference phenomena, including strobing.

B. Evaluation Metrics

1) *Evaluation Metrics for Removing BA Noise:* We employ multiple sets of established metrics to assess the effectiveness of our proposed denoising method. These include the noise event removal rate (NeRr), valid event removal rate (VeRr), and signal-to-noise ratio (SNR). They can be described as follows.

$$NeRr = \frac{N_{nr}}{N_n} \quad (20)$$

$$VeRr = \frac{N_{vr}}{N_v} \quad (21)$$

$$SNR = 10 \cdot \lg \frac{N_v - N_{vr}}{N_n - N_{nr}} \quad (22)$$

where N_n represents the total noise events, N_{nr} represents the removed noise, N_v represents the total number of valid events, and N_{vr} represents the mistakenly removed valid signals.

Additionally, we evaluate the accuracy (ACC) [37].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (23)$$

where TP is a true positive, FP is a false positive, FN is a false negative, and TN is a true negative. We refrain from utilizing the true positive rate (TPR) and false positive rate (FPR) due to their mutual exclusivity with VeRr and NeRr, respectively. These metrics are selected in alignment with the evaluation paradigms adopted by other baseline methods [16], [18], [19], [31].

Moreover, we adopt the evaluation metric event structural ratio (ESR) [36]. It utilizes the event contrast to evaluate the performance of denoising methods and does not necessitate prelabeling event sequences with valid signals or noise.

$$ESR = \sqrt{\left(\sum_{i=1}^K \frac{n_i(n_i - 1)}{N(N - 1)} \right) \cdot \left(K - \sum_{i=1}^K \left(1 - \frac{M}{N} \right)^{n_i} \right)} \quad (24)$$

where K is the total number of pixels in an image of warped events (IWE) [38], N is the overall number of events, and n_i represents the total number of events at pixel (x_i, y_i) . M refers to the reference number of events used for interpolation. Detailed definitions of the ESR are available in E-MLB [36].

2) *Evaluation Metric for Removing Flicker Noise:* We adopt the filtering rate α [35] to evaluate the effectiveness of our proposed method in removing stroboscopic light (flicker noise).

$$\alpha = \left(1 - \frac{S_0}{S_1} \right) \quad (25)$$

where S_1 represents the number of all events in the pixel area where the strobe light source is located and S_0 represents the number of events in this area after denoising.

¹[Online]. Available at: <https://www.inivation.cn/?list=19>

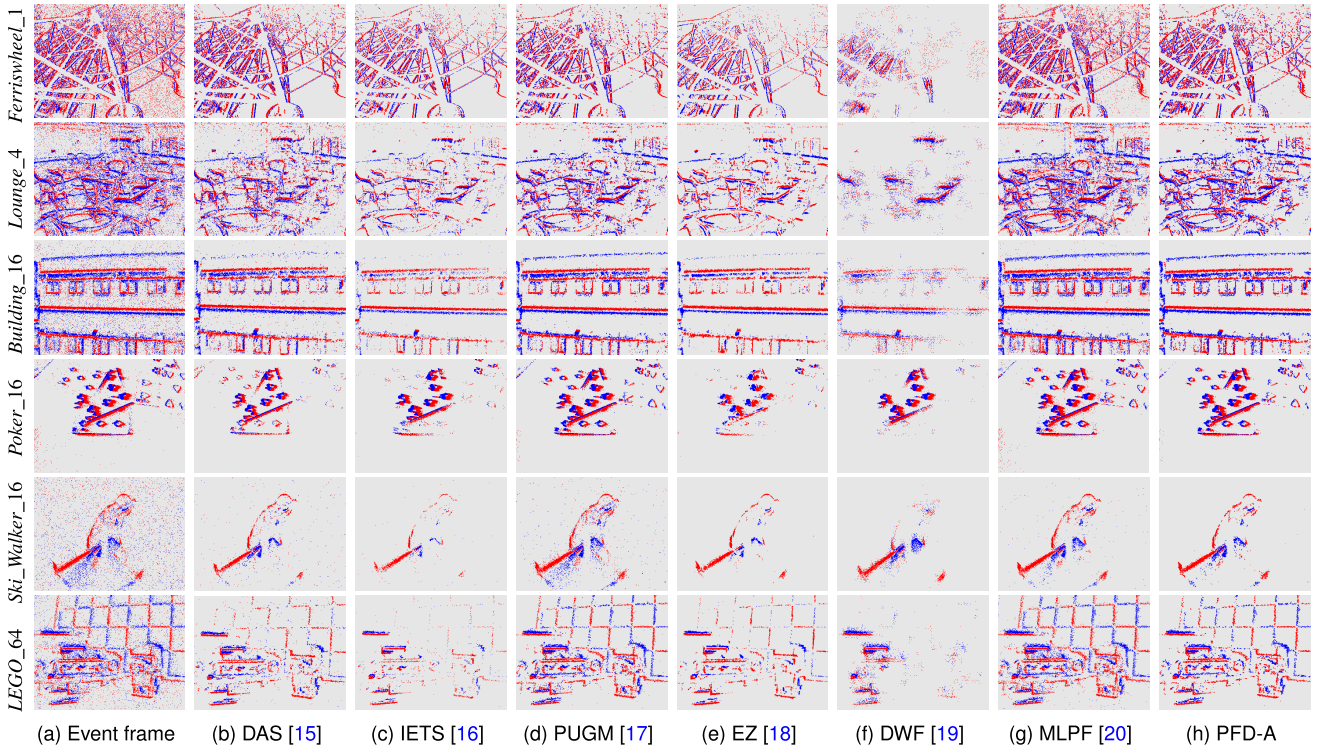


Fig. 7. Qualitative comparison on the E-MLB [36] dataset. 1, 4, 16, and 64 represent ND1, ND4, ND16, and ND64, respectively. The proposed PFD-A method eliminates more noise while retaining more valid signals in sequences containing fast motion. The changing characteristics of event polarity caused by fast motion essentially differ from those caused by noise so that PFD-A can fully play its role.

3) *Computational Performance*: We utilize two metrics for evaluating computing speed: throughput and process time per event.

C. Implementation Details

For highly efficient implementation, the proposed method is written in C++. The computing platform is a laptop with an Intel(R) Core(TM) i7-12700H CPU and 32 GB of memory. In the experiments, the following parameters are selected: time window $\delta t_c = 25$ ms and $\delta t_f = 25$ ms, $\mu = 1$, $m = 3$, $\sigma = 1$, $\tau = 1$, $n = 3$, and $d = 3$.

The hardware PFD is implemented on an AVNET Ultra96-V2² board, featuring the Xilinx Zynq ZU3EG System-on-Chip (SoC). Its logic section comprises 154,000 system logic units, including 70,560 lookup tables (LUTs), 141,120 flip-flops (FFs), and $432 \times 18k$ block random access memories (BRAMs), with 7.6 Mb of memory and 360 digital signal processor (DSP) units. High-level synthesis (HLS) is utilized for the final design completion. The parameter N for bit splicing is set to 4.

D. Experimental Results

1) *Comparisons of BA Noise Removal*: The state-of-the-art (SOTA) denoising methods DAS [15], IETS [16], PUGM [17], EventZoom [18], DWF [19], and MLPF [19], [20] are adopted for comparison. Note that all the event data are processed via the proposed coarse denoising and fine denoising methods.

²[Online]. Available at: <https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/ultra96-v2/>

The quantitative results on the ECD17 and DND21 datasets are shown in Figure 8 and Figure 9, respectively. The *driving* sequence in DND21 and the *slider_depth* and *urban* sequences in ECD17 are selected for comparison and contain the richest features. The qualitative results are shown in Figure 1.

Note that the EventZoom method diverges from conventional denoising approaches by producing denoised frames instead of a denoised event stream. As a result, EventZoom was not quantitatively evaluated on the DND21 and ECD17 datasets. To assess the efficacy of a denoising approach, it is crucial to consider the four metrics of NeRr, VeRr, SNR, and accuracy in a holistic manner. Although achieving a high SNR is possible with elevated NeRr and VeRr, this does not necessarily indicate an effective denoising outcome. The primary concern is that valid events are retained only infrequently, undermining the overall quality of the denoising process. The experimental results indicate that our proposed PFD method achieves unparalleled performance in terms of NeRr and VeRr. By eliminating a greater amount of noise while maximally preserving the effective signal, we also achieve better SNR performance. While methods such as IETS and DWF attain comparable NeRr values, they exhibit significantly higher VeRr values. This indicates that a substantial number of valid events are erroneously eliminated. Moreover, the proposed PFD method outperforms the learning-based MLPF method in terms of these metrics. Furthermore, our proposed method demonstrates exceptional performance in terms of ACC, particularly under high-noise conditions. Although the ACC of our PFD method is slightly lower than that of PUGM in the *urban* and *slider_depth* sequences, it is important to note that the significantly lower NeRr of PUGM indicates

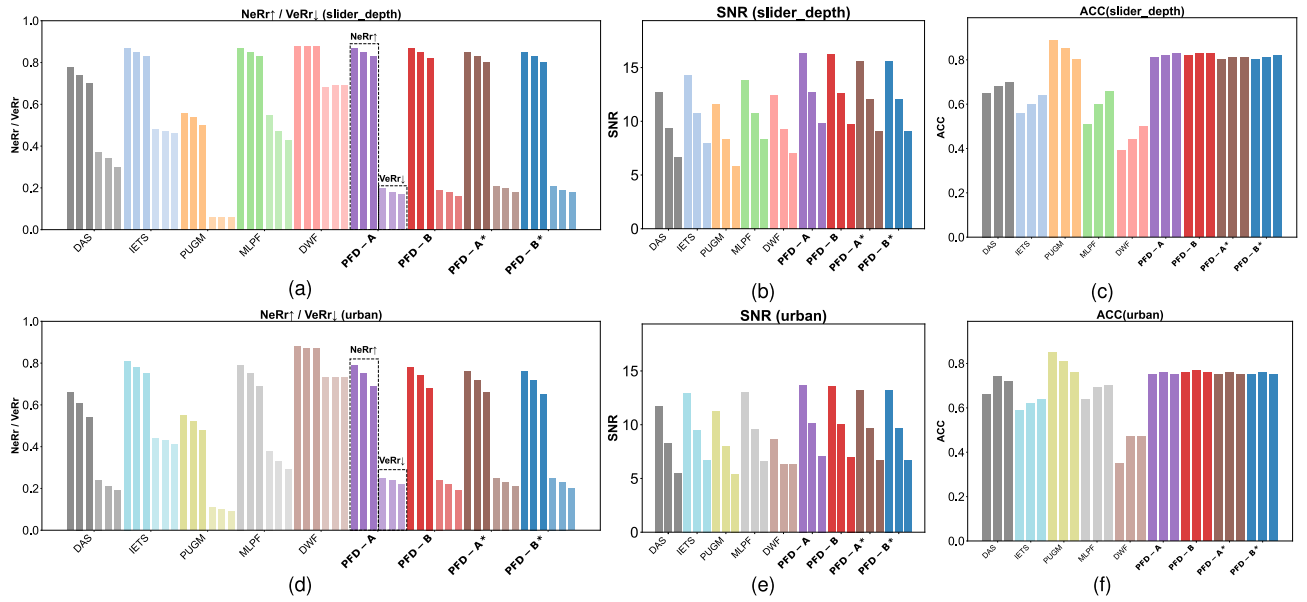


Fig. 8. Experimental results on the ECD17 dataset. The “*” symbol indicates the event-by-event processing version. An upward arrow \uparrow indicates that higher values are better, whereas a downward arrow \downarrow signifies that lower values are preferable. The results for the slider_depth sequence are presented in (a), (b), and (c), while the results for the urban sequence are shown in (d), (e), and (f). Specifically, (a) and (d) illustrate the NeRr and VeRr results, where a higher NeRr and a lower VeRr indicate better performance. For each method, the data are displayed from left to right, corresponding to results under 15%, 30%, and 50% noise levels, respectively. For example, in the IETS results, the three dark blue-gray columns on the left represent NeRr at 15%, 30%, and 50% from left to right, while the three light blue-gray columns on the right correspond to VeRr at 15%, 30%, and 50% from left to right. All other data are organized following this same arrangement. Figures (b) and (e) depict the SNR results, and (c) and (f) show the ACC results, with a higher ACC indicating better denoising performance. These results demonstrate that for the same NeRr, our proposed PFD method achieves a significantly lower VeRr, indicating that we retain more valid events while removing the same amount of noise. This leads to the best SNR performance. Furthermore, our method also has a high ACC. Although the ACC of the PUGM method surpasses that of our method, this method removes very little noise, resulting in a less effective denoising performance.

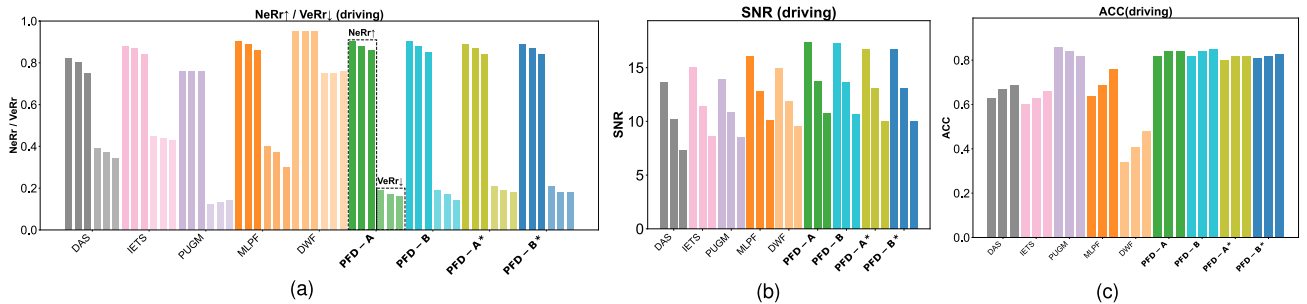


Fig. 9. Experimental results on the DND21 dataset. As in Figure 8a, in Figure 9a, the three dark color columns on the left represent NeRr at 15%, 30%, and 50% from left to right, while the three light color columns on the right correspond to VeRr at 15%, 30%, and 50% from left to right. Our proposed PFD method exhibits superior NeRr and VeRr performance, surpassing the learning-based MLPF approach, which results in the highest observed SNR. Furthermore, it achieves notably high ACC.

limited denoising effectiveness, namely, fewer TNs. PUGM retains more valid events, namely, a higher number of TPs, which increases the ACC, as shown in Equation 23, with the denominator remaining constant for a given sequence. Under low-noise conditions, retaining more valid events can yield a high ACC even if less noise is removed. However, as noise levels increase, the inferior noise removal capability of PUGM causes its ACC to decrease, whereas our PFD maintains a high ACC across all noise levels, increasing even further as noise increases. Given that the ECD17 dataset has a lower resolution (240×180) and scene complexity than the DND21 dataset (whose resolution is 346×260), PFD outperforms PUGM in high-noise sequences, demonstrating superior performance in complex and high-noise scenes. Additionally, the slow computation of PUGM makes it impractical for real-time applications, whereas PFD offers rapid computation with only a slightly lower accuracy in some cases. Therefore, PFD has

clear advantages over PUGM in terms of overall performance and efficiency.

We present the experimental results on the E-MLB dataset in Table I. All 96 sequences in the E-MLB dataset were used for evaluation. Given the specific characteristics of the E-MLB dataset, we exclusively employ the ESR metric for the evaluation. This method initially computes an ESR score for the undenoised sequence, followed by the calculation of the denoised ESR score. The ESR is a positive indicator, with higher values indicating better performance.

Note that since the main structure of PUGM is not open source, we cannot rewrite it into a format that complies with ESR evaluation. We cite the optimal ESR values of the IETS, EventZoom, and MLPF methods as reported in [36]. Our proposed method demonstrates exceptional performance across both Daytime and Night sequences within the E-MLB dataset. It outperforms the SOTA nonlearning-based

TABLE I
QUANTITATIVE ESR RESULTS ON THE E-MLB DATASET

Method	E-MLB(Daytime)				E-MLB(Night)			
	ND1	ND4	ND16	ND64	ND1	ND4	ND16	ND64
Raw	0.821	0.824	0.815	0.786	0.890	0.824	0.786	0.768
DAS [15]	0.838	0.854	0.858	0.861	0.921	0.902	0.952	1.048
IETS [16]	0.772	0.785	0.777	0.753	0.950	0.823	0.804	0.711
EventZoom [18]	0.996	0.988	0.996	0.970	1.055	1.007	1.010	0.988
DWF [19]	0.866	0.870	0.872	0.859	<u>0.972</u>	0.939	0.899	0.890
MLPF [19]	0.851	0.855	0.846	0.840	0.926	0.928	0.910	0.906
PFD-A	0.875	0.873	0.873	0.878	0.946	0.945	<u>0.974</u>	1.063
PFD-B	0.877	0.875	0.874	0.879	0.938	0.934	0.973	1.065
PFD-A*	<u>0.879</u>	<u>0.880</u>	<u>0.880</u>	<u>0.882</u>	0.954	0.954	0.970	1.070
PFD-B*	0.879	<u>0.880</u>	<u>0.880</u>	<u>0.882</u>	0.954	<u>0.954</u>	0.970	1.070

ND1 to ND64 represent a gradual increase in event noise levels. The first row (Raw) shows the ESR scores of the undenoised sequences.

methods and the learning-based MLPF method. While the PFD method may lag slightly behind the SOTA learning-based EventZoom method, it outperforms EventZoom in nighttime ND64 sequences characterized by higher noise levels, showing an improvement of 10%. Note that the ESR evaluation method involves warping events and assigns higher scores to event frames with clearer edges after warping. EventZoom is not a standard bisection network; it also reprojects events, a warping operation on a two-dimensional plane, to enhance edge clarity for subsequent image superresolution tasks. Consequently, the event frames generated by EventZoom have higher spatial gradients than those produced by methods without warping, aligning with the ESR's reward mechanism and resulting in a higher ESR score for EventZoom. However, the primary goal of denoising is to remove noise while retaining valid events to facilitate various subsequent tasks. For tasks such as motion compensation and rotation estimation, preprocessing events with operations such as warping can lead to significant deviations in the calculation results. Moreover, we notice that PFD-A* has a slightly higher ESR score than PFD-A does, whereas the NeRr and VeRr values of PFD-A* are slightly worse. Importantly, a higher VeRr indicates that more effective events are removed, which increases the spatial gradient of the accumulated event frame (i.e., sharper edges), leading to a slightly higher ESR score for PFD-A*. However, the overall performances of PFD-A and PFD-A* remain very close.

The qualitative results on the E-MLB dataset are shown in Figure 7. In the *poker* sequence, the IETS, DWF, and EventZoom methods significantly reduce the number of valid events, notably including elements such as the heart suit in the upper middle part of the figure. However, the proposed PFD method excels at noise removal while preserving more valid events. These results conclusively demonstrate the effectiveness of our proposed method, which relies on polarity consistency, especially in scenarios involving rapid movement.

Moreover, the outcomes of both the event-by-event processing version and the group-of-event processing version are presented in Figure 8, Figure 9, and Table I. The metrics for the event-by-event version are close to those for the group-of-event version. This highlights the efficacy of both event-by-event and group-of-event processing versions of the PFD method.

TABLE II
QUANTITATIVE RESULTS OF FLICKER NOISE REMOVAL ON THE LIED DATASET

Method	$\alpha \uparrow$	
	Sequence 9	Sequence 10
ELIR [35]	90.75%	95.68%
PFD-B	98.39%	99.30%

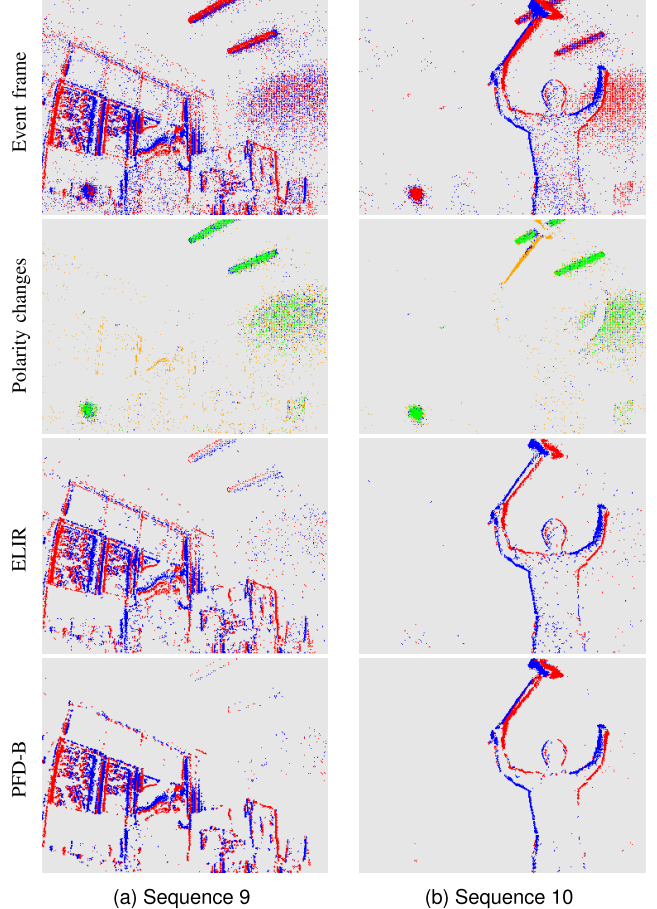


Fig. 10. Qualitative results for Sequence 9 and Sequence 10 within the LIED dataset. The first row displays the accumulated event frames. The strobe light sources, heavily burdened with flicker noise, are located in the upper right (with a frequency of 100 Hz) and lower left corner (with a frequency of 122 Hz) of the scene. The right side of the scene exhibits significant stray light, primarily due to reflections from the wall. The second row provides a schematic diagram of polarity changes, with yellow, purple-blue, and green indicating one, two, and three or more polarity changes in the pixel, respectively. The third and fourth rows showcase the outcomes processed by the ELIR method and by our proposed PFD-B, respectively.

2) *Comparisons of Flicker Noise Removal*: Next, we assess the efficacy of the proposed PFD-B method in eliminating stroboscopic light. For comparative analysis, we choose our previously proposed event-based light interference removal (ELIR) method [35], a foundational baseline method dedicated to stroboscopic light removal. Sequence 9 and Sequence 10 in the LIED dataset are utilized for evaluation. Sequence 9 was captured while the camera was moving, whereas Sequence 10 was recorded with the camera remaining stationary. The quantitative outcomes are detailed in Table II, while the qualitative comparisons are illustrated in Figure 14.

The ELIR method effectively diminishes the majority of stroboscopic light interference and stray light affecting the

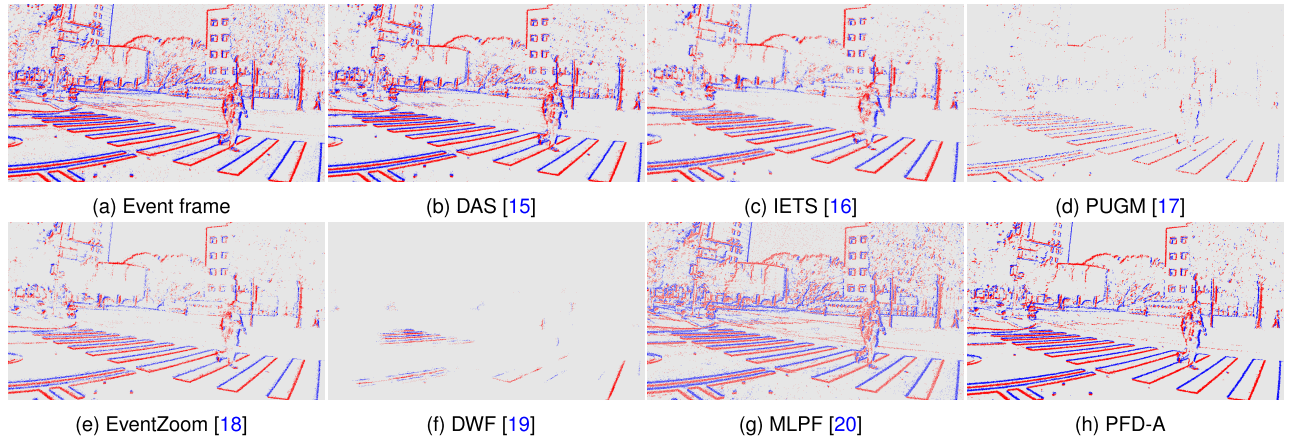


Fig. 11. Visual comparison in a real-world daytime scenario. PUGM and MLPF, which perform well in low-resolution scenarios, struggle in high-resolution conditions. MLPF is trained on event sequences with a resolution of 346×260 , whereas PUGM, which is not a learning-based method, determines whether an event is noise or a valid signal by calculating probability distributions. However, these methods become less effective at higher resolutions. In contrast, our proposed method continues to perform well, retaining most of the valid signals while nearly eliminating BA noise.

scene. Compared with the ELIR method, the newly developed PFD-B method not only effectively eliminates BA noise but also removes flicker noise, notably those with high-density polarity fluctuations. Under dynamic conditions, our proposed PFD-B method outperforms the SOTA alternative, ELIR, by achieving a nearly 8% greater α .

3) *Validation in Real-World Scenarios*: We use the Prophesee EVK3-HD³ event camera to record two event sequences to evaluate the effectiveness of our proposed method in real-world application scenarios. The EVK3-HD camera, with its megapixel resolution (1280×720), is highly sensitive for capturing fine scene details. However, owing to the small pixel size, it is more susceptible to noise. As these sequences inherently contain noise, we refrain from performing quantitative evaluations and instead present the visual results after denoising, as shown in Figure 11 and Figure 12.

The first sequence was recorded on the campus of Beihang University during the day, with the recorder seated on an electric bike. As shown in Figure 11, the scene included dynamic objects such as pedestrians and bicycles and featured a variety of buildings and trees rich in texture. Daytime BA noise was relatively minimal. However, MLPF, which had previously shown strong performance in low-resolution sequences, struggled in high-resolution sequences, failing to remove a significant portion of the BA noise. Additionally, the adaptability of PUGM declines notably under high-resolution conditions, leading to the removal of many valid events. DWF, as a more extreme denoising method, either simultaneously removes a large amount of both noise and valid events or retains a significant portion of both. While IETS and DAS performed well, IETS tended to over-denoise, and DAS left some noise unremoved. The proposed PFD-A method not only effectively removes a large amount of noise but also preserves more valid events, maintaining strong performance in real-world scenarios.

The second sequence was recorded on campus at night. During the night, the amount of BA noise increased significantly, and nearly all the streetlights on the campus were strobe

TABLE III
QUANTITATIVE RESULTS OF COMPUTATIONAL PERFORMANCE

Method	Language	Throughput \uparrow (million events/s)	Time/event \downarrow (μ s)
DAS [15]	C++	0.66	1.514
IETS [16]	Matlab	0.98	1.019
PUGM [17]	Python	0.01	85.740
EventZoom [18]	Python	0.09	11.771
DWF [19]	Python	0.03	34.841
MLPF [19]	Java	0.70	1.431
PFD-A	C++	9.71	0.103
PFD-B	C++	11.63	0.086
PFD-A*	C++	3.51	0.285
PFD-B*	C++	3.69	0.271

lights, resulting in considerable light interference throughout the scene. We continued to use electric bicycles for recording, and the scene retained the same rich features as in the daytime sequence, including dynamic elements and textured environments, as shown in Figure 12. Other methods struggle in real-world scenarios with stroboscopic light sources and fail to even remove BA noise. In contrast, the proposed PFD-B method continues to perform exceptionally well, effectively eliminating stroboscopic light and removing the majority of BA noise.

4) *Computational Performance*: Furthermore, we evaluate the computational performance of the proposed method, as shown in Table III. Since the majority of the baseline methods were not implemented in C++, we merely outline their performance without making direct comparisons to guarantee a fair evaluation. The throughput of the group-of-event version PFD-A achieves an impressive 9.71 million events per second. Although the calculation speed of the event-by-event version is slower than that of the group-of-event version, its throughput still reaches 3.51 million events per second, satisfying the requirements for real-time performance.

E. The Results of Hardware Implementations

We conducted a comparison with the hardware implementation of the MLPF method [20], which was executed on the Xilinx Zynq ZU3CG chip. Notably, this chip shares the

³[Online]. Available at: <https://www.prophesee.ai/event-based-evaluation-kits/>

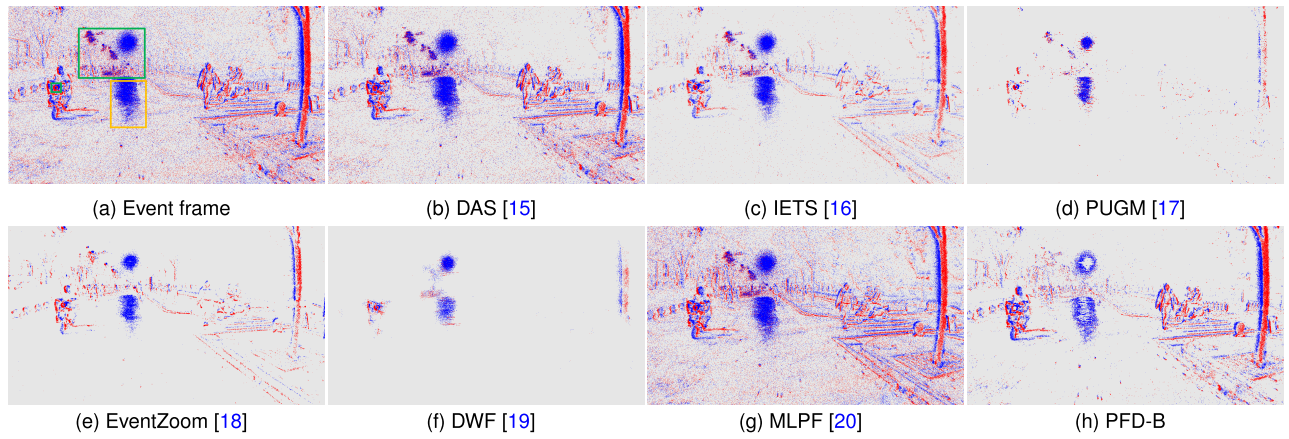


Fig. 12. Visual comparison in a real-world nighttime scenario. This scenario features an extremely bright stroboscopic light source used to capture and record license plates. Additionally, the street lamp itself acts as a stroboscopic light source, contributing to significant BA noise and light interference. In (a), the green box highlights the presence of stroboscopic light interference, while the orange box indicates reflections caused by the stroboscopic light. Our proposed PFD-B not only removes a significant amount of BA noise but also effectively eliminates most of the stroboscopic noise and its reflections. In contrast, other denoising methods lack the ability to address stroboscopic noise and tend to achieve more extreme performance. They either apply overly aggressive denoising, which removes a large portion of useful signals, or offer limited denoising, leaving a substantial amount of noise unaddressed.

TABLE IV
COMPARISON OF HARDWARE IMPLEMENTATIONS

Method	Latency (cycles)	Interval (cycles)	BRAM	DSP	FF	LUT
MLPF [20]	10	-	400	-	3800	24,600
PFD-A*	7	6	418	3	1721	15,160

same FPGA as the one integrated into the ZU3EG, ensuring a fair and direct comparison between the two implementations. Table IV outlines the algorithm resources and delays.

Our proposed PFD implementation requires only 7 clock cycles, and event reading occurs every 6 clock cycles. With the FPGA operating at a frequency of 100 MHz, the single event processing time is only 70 ns, achieving a throughput capacity of up to 14.3 million events per second. Our proposed hardware implementation demonstrates superior computational speed and resource utilization compared with the hardware implementation of MLPF, thereby outperforming it. This comprehensive comparison substantiates the efficiency and effectiveness of our proposed method.

F. Ablation Study

1) *The Impact of Individual Parameters:* We first conducted a comprehensive evaluation to assess the individual effectiveness of the two components within our proposed method.

Initially, we applied solely the polarity consistency-based coarse denoising method (coarse denoising) to the *slider_depth* and *driving* sequences from the ECD17 and DND21 datasets, respectively. The noise level is 30%.

Following this, we exclusively utilized the motion consistency-based fine denoising method (fine denoising) for testing. Equation 16 and Equation 18, which are designed for fine denoising, are both tested. Furthermore, we present the outcomes following the two-step processing with our methods, namely, PFD-A and PFD-B. The experimental results of each step are detailed in Table V.

These results reveal that coarse denoising alone eliminates approximately half of the noise, preserving over 94% of the valid events. In contrast, fine denoising on its own more

TABLE V
THE PERFORMANCE OF EACH COMPONENT OF THE PROPOSED METHOD

Sequence	Coarse Denoising		Fine Denoising				PFD-A		PFD-B	
			Equation.16		Equation.18					
	NeRr	VeRr	NeRr	VeRr	NeRr	VeRr	NeRr	VeRr	NeRr	VeRr
<i>driving</i>	66.54	4.25	81.48	16.25	81.11	16.06	88.22	17.14	88.05	16.95
<i>slider_depth</i>	63.05	5.80	77.98	17.00	77.34	16.58	85.39	18.33	85.12	17.94

To facilitate a more accurate comparison, numerical values are presented as percentages (%).

effectively removes noise but also mistakenly discards some valid events. However, employing both methods sequentially—first coarse denoising followed by fine denoising—significantly improves the outcomes. Both PFD-A and PFD-B demonstrate enhanced performance, effectively balancing noise removal with the preservation of valid events. This synergistic effect of the two-step processing approach outperforms the use of either filter alone, underscoring the efficacy of our proposed method. Furthermore, the two denoising criteria we introduced prove to be highly effective, exhibiting closely matched performance levels.

Next, the effects of varying the time window, as well as the minimum number of neighbors μ and m , on the performance of the proposed coarse denoising method and fine denoising method are studied. Note that μ is applied for the coarse denoising method and that m is applied for the fine denoising method. Using the *driving* sequence from the DND21 dataset with 30% noise, we adjusted the time window from 10 ms to 40 ms in 5 ms steps and set μ and m from 1 to 4 (with $\mu = 0$ or $m = 0$ treating all signals as valid). The experimental results, illustrated in Figure 13, show that smaller time windows and a greater number of neighbors increase NeRr and VeRr, indicating better noise removal but at the risk of discarding valid signals. Therefore, we suggest a balanced setting to effectively reduce noise while preserving valid signals. Moreover, as depicted in Figure 13b and Figure 13c, Equation 16, which is dedicated exclusively to eliminating BA noise, and Equation 18, which is capable of

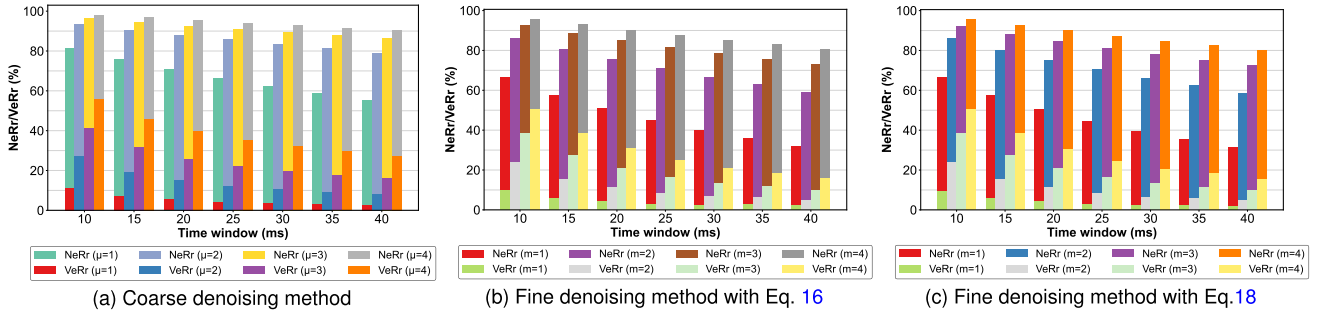


Fig. 13. NeRr and VeRr of the polarity consistency-based coarse denoising method and motion consistency-based fine denoising method. As shown in (a), using a 10 ms time window as an example, the results for $\mu = 1$ to $\mu = 4$ are displayed sequentially from left to right. The colored bars represent the values of NeRr and VeRr, with their heights indicating the respective values. NeRr and VeRr associated with the same parameters are positioned on a common basis for easy comparison. Note that both (b) and (c) depict the outcomes following only fine denoising, omitting the coarse denoising process.

removing both BA noise and flicker noise simultaneously, have both demonstrated commendable effectiveness in mitigating BA noise. To enhance our proposed method, we recommend a lower value of μ during the coarse filtering stage to retain the majority of valid signals while effectively eliminating most of the noise. The experimental results indicate that with $\mu = 1$, only 6% of the valid signals are removed within the time window of 25 ms. For the fine denoising method, we suggest selecting m in the range of 2 to 3 to further reduce noise while keeping VeRr low. Additionally, we advise using a smaller time window of 10 ms to 15 ms in high-mobility applications and a 20 ms to 25 ms window for typical applications (excluding very high-speed scenarios). With these parameters, our proposed PFD method achieves optimal performance.

2) *The Joint Impact of Multiple Parameters:* Moreover, to thoroughly analyze the impact of adjustable parameters on the denoising performance, we conduct a comprehensive experiment of the most important parameters, namely, the time window, μ , and m , on the overall PFD-A and PFD-B to inform the optimal parameter selection strategy. The experimental results are presented in Figure 14. We set μ , m , and the time window as the x , y , and z axes, respectively, and represent the calculated NeRr and VeRr as concentric spheres, with their radii corresponding to the calculated values. For clarity, VeRr is displayed as a solid sphere, while NeRr is shown as a transparent sphere. Better denoising results are indicated by larger NeRr values and smaller VeRr values, represented in the figure by larger transparent spheres and smaller concentric solid spheres.

As shown in Figure 14, the best denoising effect is achieved when m is set between 1 and 3, μ is set between 1 and 2, and the time window is between 20 ms and 30 ms. For high-noise application scenarios, such as nighttime conditions or frequent lighting changes, it is recommended to use larger μ and m values and shorten the time window.

Additionally, we evaluate the influence of the parameters τ and σ on the experimental results. The parameter τ is assessed based on the consistency of polarity changes between a current event and neighboring events. A larger τ value indicates more lenient denoising conditions. For σ , the focus is on the mean value of the polarity change in the neighborhood; a higher σ implies less stringent filtering conditions. We recommend setting σ to 1 and τ to 1. The selection of these two parameters can cover most scenarios, demonstrating excellent denoising

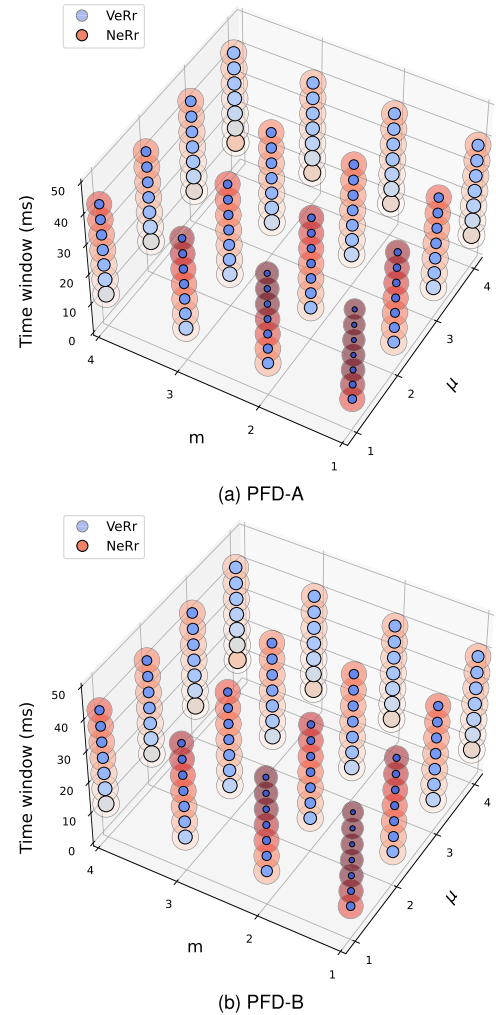


Fig. 14. The NeRr and VeRr of PFD-A and PFD-B are calculated by adjusting three parameters: μ , m , and the time window. The adjustment range for μ and m is from 1 to 4, while the time window ranges from 10 ms to 40 ms, with a step size of 5 ms. We utilize a transparent red sphere to symbolize NeRr and a solid blue sphere for VeRr. The intensity of the blue color indicates the VeRr value, with darker shades representing smaller values, while darker red shades signify lower NeRr values. Additionally, the radius of each sphere is proportional to the respective NeRr and VeRr values; smaller spheres correspond to smaller NeRr or VeRr values. Consistent with previous experimental results, the performances of the PFD-A and PFD-B methods are very similar. Both achieve high NeRr while maintaining very low VeRr with appropriate parameter selection.

effects in complex motion environments and in scenes with high-frequency flicker noise.

VI. LIMITATIONS

Our proposed denoising method demonstrates significant efficacy in environments with rapid motion by effectively leveraging the inherent distinction of events with pronounced polarity changes from BA noise. However, the absence or minimal presence of polarity changes can reduce the effectiveness of the proposed PFD method. Nevertheless, as our proposed method consists of two stages, the first stage removes most of the noise on the basis of polarity consistency rather than the consistency of polarity changes. Therefore, in scenarios with moderate motion, where there are no significant polarity changes, our method still performs quite well. Despite this, the high temporal resolution of event cameras makes them particularly well suited for capturing scenes with rapid movement. Consequently, our proposed method retains broad applicability across various dynamic scenarios.

VII. CONCLUSION

In this paper, we introduced the polarity-focused denoising (PFD) method for event cameras. PFD employs a dual-stage denoising process, incorporating a polarity consistency-based coarse filter and a motion consistency-based fine filter. Moreover, we establish two distinct criteria: PFD-A for effectively eliminating BA noise and PFD-B for concurrently removing both BA noise and flicker noise. Our proposed method is characterized by its straightforward computational process; thus, it has rapid computational speed. Furthermore, a comprehensive evaluation across five metrics, NeRr, VeRr, SNR, ACC, and ESR, reveals that PFD-A outperforms existing SOTA event-based denoising methods in BA noise removal across various benchmark datasets. Moreover, the experiments on the LIED dataset show that PFD-B effectively eliminates flicker and BA noise. Significantly, our method achieves these outcomes without the need for external data sources, such as APS frames and IMU data. Finally, we present an FPGA implementation of PFD-A, highlighting its efficiency by processing each event in 7 clock cycles.

ACKNOWLEDGMENT

The authors express their profound gratitude to Dr. Peiqi Duan for sharing the codes of the EventZoom method.

REFERENCES

- [1] Z. Chen, J. Wu, J. Hou, L. Li, W. Dong, and G. Shi, "ECSNet: Spatiotemporal feature learning for event camera," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 2, pp. 701–712, Feb. 2023.
- [2] D. Li, Y. Tian, and J. Li, "SODFormer: Streaming object detection with transformer using events and frames," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 14020–14037, Nov. 2023.
- [3] H. Yu, H. Li, W. Yang, L. Yu, and G.-S. Xia, "Detecting line segments in motion-blurred images with events," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 2866–2881, May 2024.
- [4] U. M. Nunes and Y. Demiris, "Robust event-based vision model estimation by dispersion minimisation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9561–9573, Dec. 2022.
- [5] Z. Liu, J. Wu, G. Shi, W. Yang, W. Dong, and Q. Zhao, "Motion-oriented hybrid spiking neural networks for event-based motion deblurring," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3742–3754, May 2024.
- [6] Y. Gao et al., "Action recognition and benchmark using event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14081–14097, Dec. 2023.
- [7] J. Zhao et al., "The 3rd anti-uav workshop & challenge: Methods and results," 2023, *arXiv:2305.07290*.
- [8] Z. Wang et al., "Learning to detect head movement in unconstrained remote gaze estimation in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 3432–3441.
- [9] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, Oct. 2021.
- [10] X. Peng, L. Gao, Y. Wang, and L. Kneip, "Globally-optimal contrast maximisation for event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3479–3495, Jul. 2022.
- [11] P. Chen, W. Guan, and P. Lu, "ESVIO: Event-based stereo visual inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3661–3668, Jun. 2023.
- [12] P. Chen et al., "Point-to-box network for accurate object detection via single point supervision," in *Proc. Eur. Conf. Comput. Vis.*, vol. 13669, 2022, pp. 51–67.
- [13] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.
- [14] R. Graca and T. Delbruck, "Unraveling the paradox of intensity-dependent DVS pixel noise," 2021, *arXiv:2109.08640*.
- [15] N. Xu, L. Wang, J. Zhao, and Z. Yao, "Denoising for dynamic vision sensor based on augmented spatiotemporal correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 4812–4824, Sep. 2023.
- [16] R. W. Baldwin, M. Almatrafi, J. R. Kaufman, V. K. Asari, and K. Hirakawa, "Inceptive event time-surfaces for object classification using neuromorphic cameras," in *Proc. 16th Int. Conf. Image Anal. Recognit.* Cham, Switzerland: Springer, Aug. 2019, pp. 395–403.
- [17] J. Wu, C. Ma, L. Li, W. Dong, and G. Shi, "Probabilistic undirected graph based denoising method for dynamic vision sensor," *IEEE Trans. Multimedia*, vol. 23, pp. 1148–1159, 2021.
- [18] P. Duan, Z. W. Wang, X. Zhou, Y. Ma, and B. Shi, "EventZoom: Learning to denoise and super resolve neuromorphic events," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 12824–12833.
- [19] S. Guo and T. Delbruck, "Low cost and latency event camera background activity denoising," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 785–795, Jan. 2023.
- [20] A. Rios-Navarro et al., "Within-camera multilayer perceptron DVS denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 3933–3942.
- [21] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, Feb. 2017.
- [22] H. Liu, C. Brandli, C. Li, S.-C. Liu, and T. Delbruck, "Design of a spatiotemporal correlation filter for event-based sensors," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 722–725.
- [23] A. Khodamoradi and R. Kastner, " $O(N)O(N)$ -space spatiotemporal filter for reducing noise in neuromorphic vision sensors," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 1, pp. 15–23, Jan. 2021.
- [24] F. Li, Y. Huang, Y. Chen, X. Zeng, W. Li, and M. Wang, "Queue-based spatiotemporal filter and clustering for dynamic vision sensor," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2023, pp. 1–4.
- [25] Y. Feng, H. Lv, H. Liu, Y. Zhang, Y. Xiao, and C. Han, "Event density based denoising method for dynamic vision sensor," *Appl. Sci.*, vol. 10, no. 6, p. 2024, Mar. 2020.
- [26] Y. Chen, Y. Huang, F. Li, X. Zeng, W. Li, and M. Wang, "Denoising method for dynamic vision sensor based on two-dimensional event density," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2023, pp. 1–4.
- [27] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, "Fast event-based corner detection," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–12.
- [28] Y. Wang et al., "EV-gait: Event-based robust gait recognition using dynamic vision sensors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6351–6360.
- [29] Z. W. Wang, P. Duan, O. Cossairt, A. Katsaggelos, T. Huang, and B. Shi, "Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1606–1616.

- [30] S. A. Mohamed, J. N. Yasin, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "DBA-filter: A dynamic background activity noise filtering algorithm for event cameras," in *Proc. Comput. Conf.*, vol. 1. Springer, 2022, pp. 685–696.
- [31] R. Baldwin, M. Almatrafi, V. Asari, and K. Hirakawa, "Event probability mask (EPM) and event denoising convolutional neural network (EDnCNN) for neuromorphic cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1701–1710.
- [32] P. Duan, Z. W. Wang, B. Shi, O. Cossairt, T. Huang, and A. K. Katsaggelos, "Guided event filtering: Synergy between intensity images and neuromorphic events for high performance imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8261–8275, Nov. 2022.
- [33] P. Duan et al., "NeuroZoom: Denoising and super resolving neuromorphic events and spikes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 15219–15232, Dec. 2023.
- [34] P. Zhang, G. Zhou, L. Song, and E. Y. Lam, "Neuromorphic imaging with density-based spatiotemporal denoising," *IEEE Trans. Comput. Imag.*, vol. 9, pp. 530–541, 2023.
- [35] C. Shi et al., "Identifying light interference in event-based vision," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 6, pp. 4800–4816, Jun. 2024.
- [36] S. Ding et al., "E-MLB: Multilevel benchmark for event-based camera denoising," *IEEE Trans. Multimedia*, vol. 26, pp. 65–76, 2023.
- [37] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.
- [38] G. Gallego, M. Gehrig, and D. Scaramuzza, "Focus is all you need: Loss functions for event-based vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12272–12281.



Chenyang Shi was born in Guangxi, China, in 1997. He is currently pursuing the Ph.D. degree in optical engineering with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing, China. His research interests focus mainly on event-based vision and vision navigation.



Boyi Wei was born in Shanxi, China, in 1998. He received the master's degree in optical engineering from Beihang University, Beijing, China, in 2023, where he is currently pursuing the Ph.D. degree with the School of Instrumentation and Optoelectronic Engineering. His main research interests include event cameras and heterogeneous computing systems.



Xiucheng Wang was born in Shandong, China, in 1999. He is currently pursuing the master's degree in electronic information with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing, China. His research interests focus mainly on event-based vision and vision navigation.



Hanxiao Liu was born in Chongqing, China, in 2000. He is currently pursuing the master's degree with the Department of Precision Instrumentation, Tsinghua University, Beijing, China. His research interests focus mainly on event-based vision and electrostatic accelerometers.



Yibo Zhang was born in Shaanxi, China, in 1996. She received the master's degree from Beihang University, Beijing, China, in 2021, where she is currently pursuing the Ph.D. degree with the School of Instrumentation and Optoelectronic Engineering. Her main research interests include event cameras.



Wenzhuo Li was born in Beijing, China, in 2000. She is currently pursuing the master's degree in optical engineering with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing. Her research interests focus mainly on event-based vision and visual navigation.



Ningfang Song was born in Tianjin, China, in 1968. She received the Ph.D. degree in precision instrumentation and mechanism from Beihang University, Beijing, China, in 2004, where she is currently a Professor with the School of Instrumentation and Optoelectronic Engineering. Her main research interests include fiber optic gyroscopes and inertial navigation systems.



Jing Jin was born in Inner Mongolia, China, in 1975. He received the Ph.D. degree in precision instrumentation and mechanism from Beihang University, Beijing, China, in 2008, where he is currently a Professor with the School of Instrumentation and Optoelectronic Engineering. His research focuses on optical sensors, event-based vision, and navigation.