LiteDVS: A Low-Data-Redundancy Dynamic Vision Sensor with Hybrid Readout and In-Pixel Denoising

Zichen Kong[1], Zhongyi Wu[1], Xiyuan Tang[2,1,3*], Yuan Wang[1,3,4,5*]
[1]School of Integrated Circuits, Peking University
[2]Institute for Artificial Intelligence, Peking University
[3]Key Laboratory of Microelectronic Devices and Circuits (MoE), MPW Center, Peking University
[4]Beijing Laboratory of Future Integrated Circuit Technology and Science, Peking University
[5]Beijing Advanced Innovation Center for Integrated Circuits

***Abstract*—Dynamic Vision Sensors (DVS) are well suited for latency- and power-sensitive applications such as embodied intelligence and autonomous driving, owing to their event-driven operation and high spatiotemporal efficiency. However, under camera motion or low-light conditions, DVS frequently produces redundant or noisy events, compromising data sparsity and reliability. To address this challenge, we propose LiteDVS, a DVS architecture with region-aware hybrid readout and in-pixel denoising. LiteDVS integrates event streams for regions of interest with event frames for background areas, significantly reducing data redundancy. Furthermore, a lightweight in-pixel filter compatible with both readout modes is designed to suppress noise events with negligible latency overhead. Simulations in a SMIC 55 nm logic CMOS process demonstrate that LiteDVS achieves accurate denoising with energy consumptions of 317 fJ/event in stream mode and 41.8 fJ/event in frame mode.**

***Index Terms*—Dynamic vision sensor, Event denoising, In-pixel processing, Regions of interest**

I. Introduction

Dynamic Vision Sensors (DVS) have emerged as compelling alternatives to conventional frame-based image sensors. Unlike standard cameras that sample scenes synchronously at a fixed frame rate, a DVS operates asynchronously: a pixel emits an event whenever the log-intensity change exceeds a preset threshold [1]. This sensing paradigm offers three salient advantages—microsecond-level latency, ultra-high dynamic range (>120 dB), and substantially lower power consumption than CMOS image sensors. These properties make DVS particularly attractive for latency- and energy-constrained applications such as autonomous driving, agile robotics, and embodied intelligence [2]. Recent hybrid perception systems further underscore this potential: combining a 20-fps RGB camera with a DVS achieves an effective latency comparable to a 5,000-fps imager while requiring bandwidth equivalent to only a 45-fps camera [3].

Nevertheless, the low-power and low-latency benefits of DVS can be undermined by excessive event redundancy. First, during camera ego-motion, even static objects (or semantically unimportant regions) generate large numbers of motion-induced events. Second, in low-light conditions, pixel noise—dominated by shot and thermal noise—triggers spurious events, degrading sparsity and reliability.

To reduce low-importance events, prior studies [4], [5] adopt random event dropping (uniform subsampling) to regulate the event rate, enabling high-throughput, low-redundancy output. However, such stochastic thinning discards events indiscriminately from salient and non-salient regions, thereby compromising data integrity. Other works [5], [6] aggregate events into event frames for readout, which mitigates bandwidth pressure but forfeits the fine temporal structure of event streams and introduces readout latency.

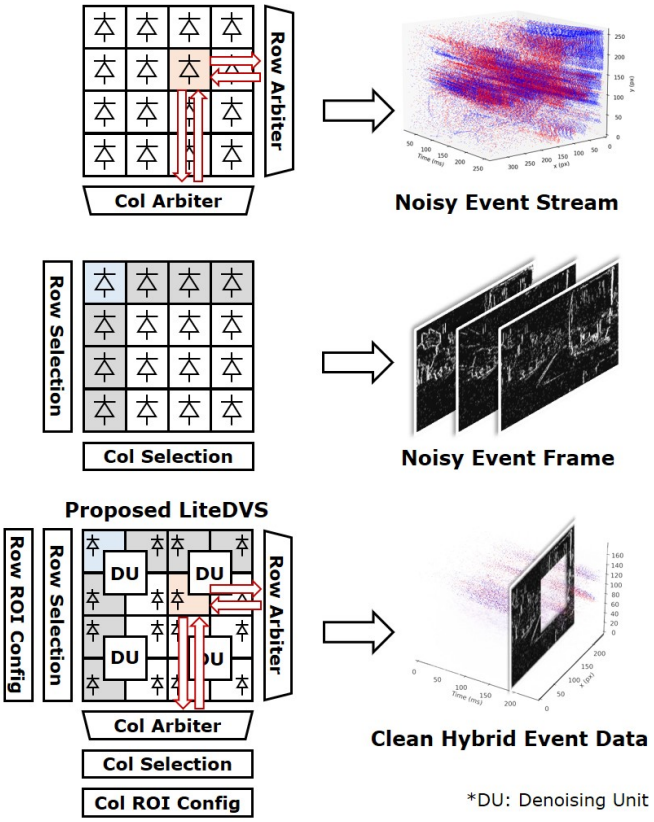

Fig. 1. Proposed LiteDVS with hybrid clean event data.

A substantial body of research has also explored hardware-friendly denoising for DVS. By output modality, existing methods fall into two categories: *event-stream denoising* and *event-frame denoising*. For event streams, [7] implements a background-activity (BA) filter array using analog timing cells; while compact in logic, it achieves limited denoising accuracy and requires large capacitors for charge storage. In [8], an on-chip multilayer perceptron (MLP) surpasses spatiotemporal correlation filters in accuracy but incurs significantly higher power than non-ML circuits. Algorithms such as [9], [10] deploy sub-quadratic-complexity denoising ($\mathcal{O}(N^2)$) but exhibit accuracy degradation under camera motion or cluttered scenes. For event frames, [11], [12] realize median-like filters using

*Corresponding authors: {xitang, wangyuan}@pku.edu.cn

SRAM/CRAM arrays, and [6] introduces a local-count–based At-Least-At-Most (ALAM) redundancy-removal unit. These designs improve denoising efficiency, but frame-based output sacrifices the fine temporal resolution inherent in event streams.

To address these challenges, we propose **LiteDVS**, illustrated in Fig. 1, a new architecture that removes redundant events efficiently through hybrid event-stream/event-frame readout and pixel-level denoising tailored to this hybrid modality. The contributions of this work are threefold:

- **ROI-based hybrid readout framework.** We introduce a region-of-interest (ROI) policy in which events within salient regions are preserved as asynchronous streams, while events in less important regions are accumulated into frames. This strategy minimizes redundant activity outside the ROI while retaining fine temporal precision where it matters most, thereby improving overall data sparsity without sacrificing critical event information.
- **Hardware-friendly, dual-modality denoising algorithm.** We design a denoising algorithm compatible with both event-stream and event-frame outputs, achieving robust noise suppression across diverse scenes and data formats.
- **Lightweight in-pixel denoising unit.** We develop a compact in-pixel denoising module that suppresses noise before row/column arbitration. This avoids arbitration latency and out-of-pixel buffering. Simulations show denoising energy consumptions of 317 fJ/event in stream mode and 41.8 fJ/event in frame mode under the SMIC 55 nm CMOS process.

II. Preliminaries

A. DVS Output Formats

The conventional output of a DVS is an *event stream*—an asynchronous sequence of events $e = (x, y, t, p)$ consisting of timestamp t, row address y, column address x, and polarity p. Event streams preserve the full temporal resolution of visual signals but are relatively complex for downstream processing. By integrating events along the time axis, one can form *event frames* (binary or ternary images) that are more directly compatible with conventional computer-vision algorithms. An event frame accumulated over a time window $[t, t + \Delta T)$ can be expressed as

$$F(x, y) = \sum_{e_i \in \mathcal{E},\ t \le t_i < t+\Delta T} w(p_i), \tag{1}$$

where $w(p_i)$ maps event polarity to binary $(0/1)$ or ternary $(-1/0/+1)$ values. Eq. 1 effectively integrates asynchronous events into a discrete image representation, making event data more compatible with conventional vision pipelines at the cost of temporal precision.

Event frames trade away the microsecond-level latency of event streams in exchange for bandwidth efficiency under dense activity. For a 128×128 array, a single asynchronous event typically requires 32 bits to encode all fields, whereas a binary event frame needs only 1 bit per pixel but requires synchronous readout of the entire array. At 60 fps, binary event frames generate approximately $128 \times 128 \times 1$ bit/frame $\times 60$ fps ≈ 120 kB/s, which is comparable to a 32-bit event stream at about 3×10^4 events/s.

In static scenes, DVS output is dominated by local transients and noise, with event rates typically ranging from tens to a few thousand events per second [13], favoring the *event-stream* format for its efficiency. In contrast, in highly dynamic scenes, array-level event rates can rise to 10^5–10^6 events/s; in such cases, integrating event streams into event frames provides effective data compression, reducing redundancy and alleviating bandwidth overhead.

B. On-Chip DVS Event Denoising

In DVS pixels, noise primarily arises from junction leakage currents and parasitic photocurrents, which are particularly pronounced under low illumination and generate spurious, task-irrelevant events [14]. Moreover, the photodiode and front-end amplifiers inherently generate thermal and shot noise, which induce current fluctuations and further increase noise events [15]. Consequently, background-activity events are frequently generated in the absence of true log-intensity changes, degrading data utility while inflating bandwidth and power. On-chip filtering of such noise is therefore essential.

Formally, the observed event set $\mathcal{E}$ can be decomposed as

$$\mathcal{E} = \mathcal{E}_{\text{true}} \cup \mathcal{E}_{\text{noise}}, \tag{2}$$

where $\mathcal{E}_{\text{true}}$ corresponds to genuine log-intensity changes and $\mathcal{E}_{\text{noise}}$ denotes spurious background-activity events. Effective denoising seeks to maximize the retention of $\mathcal{E}_{\text{true}}$ while suppressing $\mathcal{E}_{\text{noise}}$ under strict pixel-area and latency constraints.

To quantify denoising performance, true positive rate (TPR) and false positive rate (FPR) are defined as

$$\text{TPR} = \frac{|\mathcal{E}_{\text{true}} \cap \mathcal{E}_{\text{kept}}|}{|\mathcal{E}_{\text{true}}|}, \quad \text{FPR} = \frac{|\mathcal{E}_{\text{noise}} \cap \mathcal{E}_{\text{kept}}|}{|\mathcal{E}_{\text{noise}}|}, \tag{3}$$

where $\mathcal{E}_{\text{kept}}$ denotes the set of events preserved after filtering. These metrics provide the basis for quantitative evaluation and are later aggregated into the *area under the curve* (AUC) for comprehensive performance comparison.

The core idea of *event-stream denoising* is to exploit the spatiotemporal correlation of real events: genuine object motion produces clusters of events that are continuous in both space and time. In contrast, *event-frame denoising* operates on temporally aggregated representations and leverages spatial correlation, assuming that true events do not appear in isolation but exhibit neighborhood consistency within the frame domain.

III. Proposed LiteDVS

A. Overview

Fig. 2(a) illustrates the proposed LiteDVS architecture. LiteDVS introduces an ROI-based hybrid readout scheme: events inside the ROI are read out as event streams, while events outside the ROI are integrated over time and output as event frames. This hybrid readout significantly reduces data volume, enabling efficient compression. In addition, LiteDVS integrates an in-pixel denoising filter that is compatible with both readout modes. One denoising unit (DU) is shared by every four pixels, which reduces area and power overhead. Moreover, because the denoising logic is embedded within the pixel array, noise suppression is performed at the moment of event generation, introducing no additional latency.

B. Hybrid Event Readout

Fig. 2(b) depicts the ROI-based hybrid readout configuration. The ROI is programmed via row/column one-hot codes. A pixel is configured for event-stream readout if both its row and column ROI bits are set to 1, in which case each event is encoded in 32 bits. Otherwise, the pixel is assigned to a

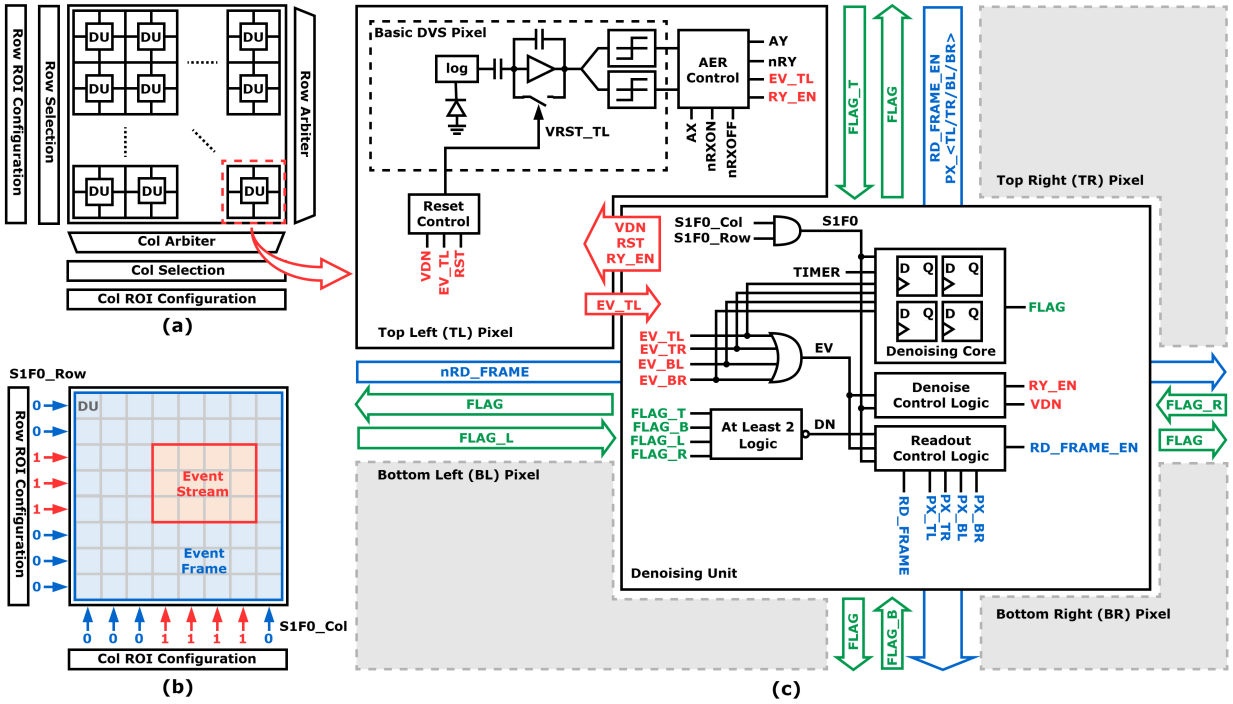

Fig. 2. Overview of proposed LiteDVS: (a) framework, (b) hybrid readout configuration and (c) pixel block of LiteDVS.

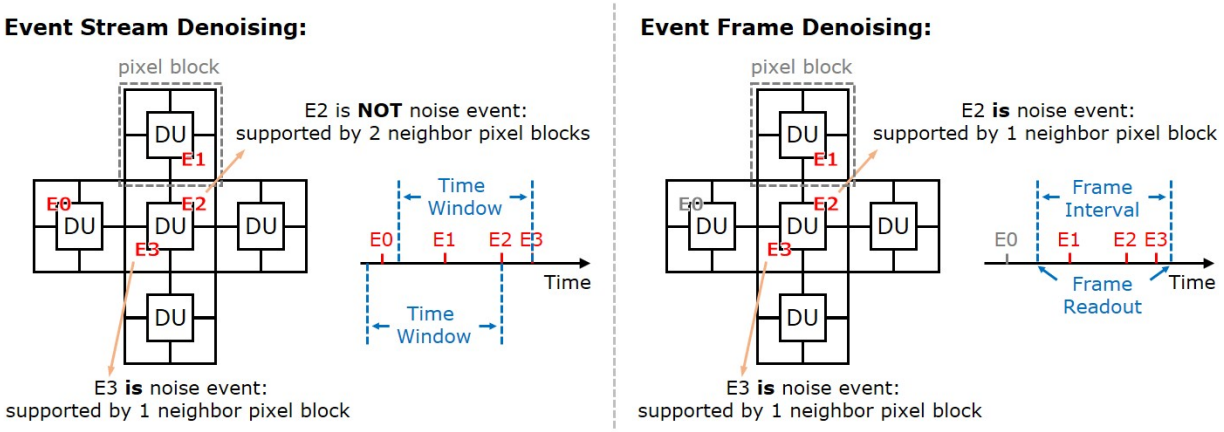

Fig. 3. Denoising algorithm of LiteDVS.

polarity-agnostic event-frame path, where each pixel is represented with 1 bit. This configuration provides fine-grained bandwidth control by reserving asynchronous timing for salient regions while compressing low-importance areas.

C. In-Pixel Denoising Filter

Fig. 2(c) shows the structure of a LiteDVS pixel block, which contains four pixels and one denoising unit. We propose a flip-flop–based in-pixel spatiotemporal correlation filter organized as a 64×64 array of denoising units. Each DU is shared by four pixels and exchanges state with its four immediate neighbors (up, down, left, right), thereby supporting denoising for both event-stream and event-frame outputs.

Fig. 3 summarizes the denoising logic. For event-stream denoising, we define a temporal window of length t. When an event arrives, the corresponding denoising unit performs a logical check; the event is rejected as noise if either of the following conditions is not satisfied (prior events are considered irrespective of whether they were previously accepted or rejected as noise):

- **Condition 1:** The denoising unit has produced at least one event within the past t.
- **Condition 2:** Among the unit's four neighbors, at least two have produced an event within the past t.

Only events satisfying both conditions are forwarded as valid; otherwise, they are suppressed before row/column arbitration.

For the synchronous event-frame modality, a simplified logic is used. The event stream is first accumulated into a binary frame over a frame interval of length t (a pixel is set to 1 if at least one event occurs within the frame interval). The denoising unit then decides whether all events inside the unit are noise.

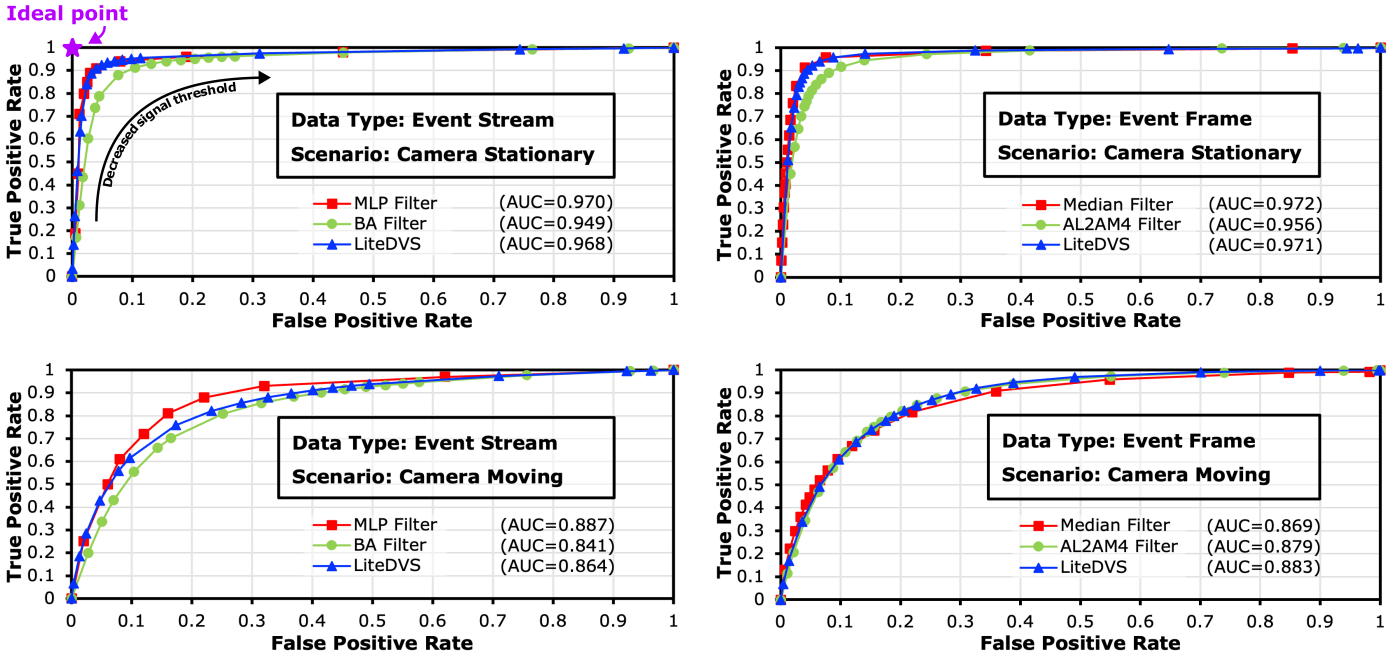

Fig. 4. Comparison of denoising algorithms with different data types and scenarios.

The entire unit is rejected as noise if either of the following conditions is not satisfied:

- **Condition 1:** At least one pixel inside the denoising unit is active (value 1).
- **Condition 2:** Among the unit's four neighbors (up, down, left, right), at least one reports an event within the past t (i.e., has value 1 in the current frame interval).

Fig. 4 compares the TPR and FPR of the proposed denoising module with the MLP Filter [13], BA Filter [7], Median Filter [11], and At-Least-2-At-Most-4 (AL2AM4) Filter [6]. AUC is used to quantify denoising performance. Experiments employ the dataset from [13], covering both event-stream and event-frame outputs under static (no camera motion) and dynamic (with camera motion) scenarios. The results demonstrate that the proposed module achieves a favorable trade-off among computational complexity, scene generality, and denoising effectiveness.

IV. Circuit Implementation

A. Pixel Design

Fig. 5(a) illustrates the LiteDVS pixel design. Relative to a conventional DVS pixel, the main modifications include the addition of a row request enable signal (`RY_EN`) and an auxiliary reset path. These enhancements allow the pixel to participate in ROI-based arbitration and enable more flexible noise suppression. After a pixel fires, it first asserts the `EV_TL` signal (illustrated here with the top-left pixel as an example), which is then processed by the corresponding denoising unit.

In the *event-stream mode*, the denoising unit immediately receives the `EV_TL` signal and determines whether the event corresponds to noise. If the event is classified as noise, the unit actively drives the `VDN` line low, resetting the pixel and suppressing spurious activity before arbitration. In contrast, in the *event-frame mode*, the denoising unit records the occurrence of the event and later contributes to the aggregated frame output during the synchronous readout phase. This dual-mode operation allows LiteDVS pixels to support both high-precision asynchronous readout and efficient bandwidth-reduced frame accumulation within a unified hardware structure.

B. Denoising Unit Design

Fig. 5(b) illustrates the LiteDVS denoising unit. The unit receives event requests from the four pixels within its local pixel block and exchanges denoising flags with its four immediate neighbors (up, down, left, right) to make a joint noise-suppression decision. Functionally, the circuit is partitioned into three main sub-blocks: (i) the denoising core, (ii) the denoising control logic, and (iii) the readout control logic.

- **Denoising core:** Implements spatiotemporal filtering by storing recent activity and combining it with neighboring-unit states, thereby providing the basis for noise suppression.
- **Denoising control logic:** Manages the global reset of the denoising core.
- **Readout control logic:** Interfaces with ROI programming and frame readout, forwarding valid events to the arbitration network while discarding noise before it propagates further.

The global control signals for the denoising array (highlighted in blue in the schematic) include: `nRST_GL`, the global reset; `RST_GL_EN`, the enable for global reset; `TIMER`, which defines the temporal correlation window; `S1F0_Col` and `S1F0_Row`, which configure column and row ROI policies (`S` = stream, `F` = frame); `DN_EN`, which enables or bypasses denoising; and `nRD_FRAME`, the row read signal for event-frame mode. Together, these signals coordinate both local and global behaviors of the denoising array, ensuring consistency across the entire sensor.

The denoising control logic reuses four D flip-flops to support both output modalities. This compact design minimizes area while allowing the same hardware to serve dual purposes.

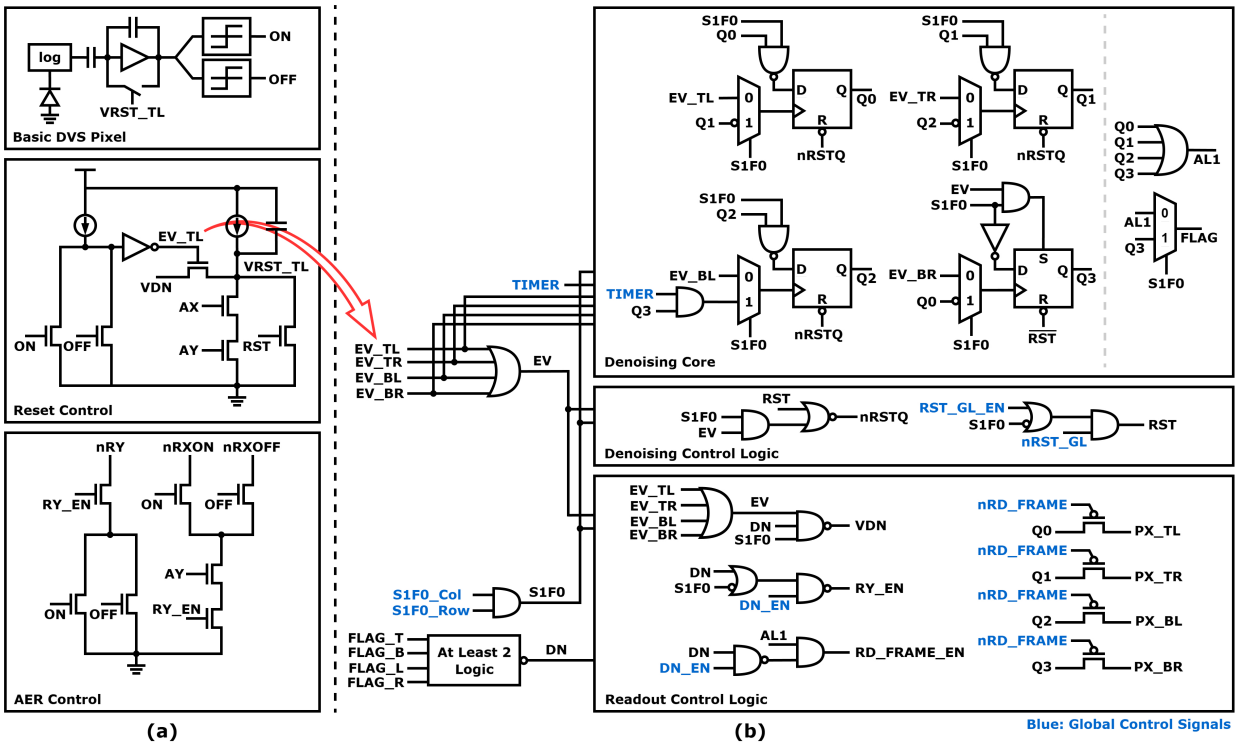

Fig. 5. Circuit implementation of (a) pixel and (b) denoising unit.

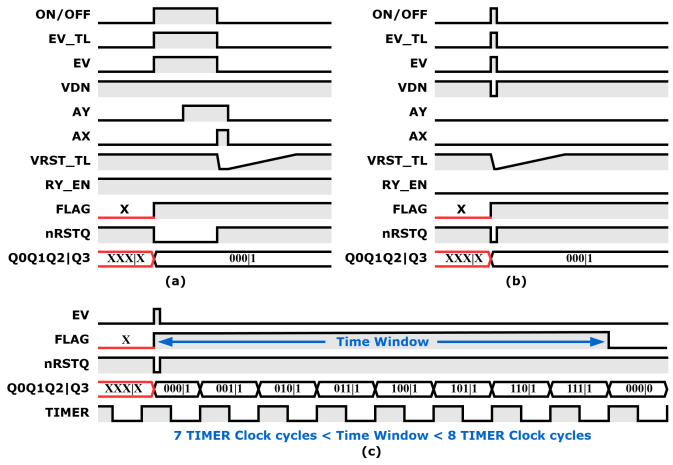

Fig. 6. Timing diagrams of event stream denoising. (a) is when EV_TL is not regarded as noise, (b) is when EV_TL is regarded as noise, and (c) is time window generation.

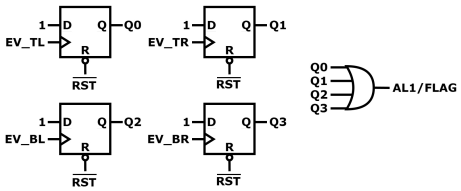

Fig. 7. Equivalent circuits of denoising core in event-frame mode.

In the *event-stream mode*, the four flip-flops are configured as a 3-bit counter (Q0, Q1, Q2) and a 1-bit denoising flag (Q3). As illustrated in Fig. 6(c), when EV_TL is asserted, the 3-bit counter is reset, the flag bit is set to 1, and the unit output FLAG becomes active. After EV_TL returns low, the counter increments on each rising edge of TIMER, the temporal window begins. once it reaches a count of 8, all flip-flops (Q0–Q3) and the FLAG are cleared to 0, thus terminating the temporal window. This counter-based mechanism permits arbitrary window lengths by simply tuning the TIMER frequency, providing a flexible trade-off between denoising strength and response time. Representative timing diagrams for event-stream mode, with and without denoising, are shown in Fig. 6(a) and Fig. 6(b), respectively.

In the *event-frame mode*, the same four flip-flops are reconfigured to act as event registers. Each flip-flop stores the activity of one pixel within the block, and a four-input OR gate generates the block-level FLAG. This simple yet effective logic ensures that the unit reports activity only when at least one pixel is active within the accumulation window, while still interacting with neighboring blocks for spatial correlation. By leveraging the same hardware resources across both modes, LiteDVS achieves mode-flexible denoising without duplicating circuitry, reducing silicon area and power consumption while maintaining high functional coverage.

Overall, this unified denoising unit provides scalable and low-overhead noise suppression that seamlessly supports both asynchronous event-stream and synchronous event-frame readout modes, serving as a key enabler of the LiteDVS architecture.

TABLE I
COMPARISON OF DENOISING METHODS

		This Work		ISCAS' 2015	CVPRW' 2023	JSSC' 2022	VLSI' 2019
Data Type		event stream	event frame	event stream	event stream	event frame	event frame
Algorithm		Modified STCF		BA Filter	MLP Filter	Median Filter	AL2AM4 Filter
Implementation		ASIC		ASIC	FPGA/ASIC	ASIC	ASIC
where to denoise		in pixel		out-of pixel	out-of pixel	out-of pixel	in pixel
Technology		55nm		180nm	65nm	65nm	65nm
Pixel Pitch(μm)		9.27		–	–	–	10
Filling Factor (%)		19.5		–	–	–	20
Cell array		128×128		128×128	346×260	320×240	132×104
Denoising Energy (fJ/event)		317/41.8		1.00 × 10^6@1MPS	–/4.00 × 10^6	39	–
Denoising Latency (ns)		0/–		10	43/40	–	–
Denoising AUC	Moving	0.864	0.883	0.841	0.887	0.869	0.879
	Stationary	0.968	0.971	0.949	0.970	0.972	0.956

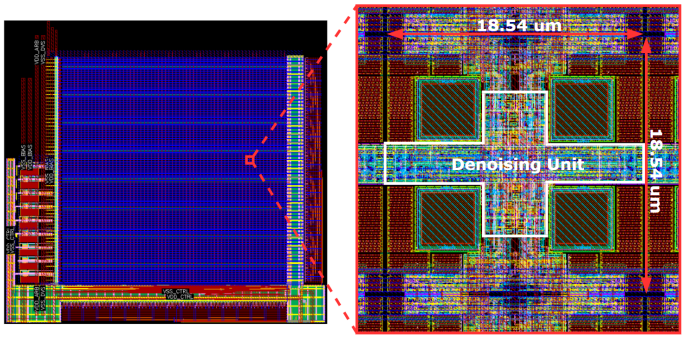

Fig. 8. Layout of LiteDVS.

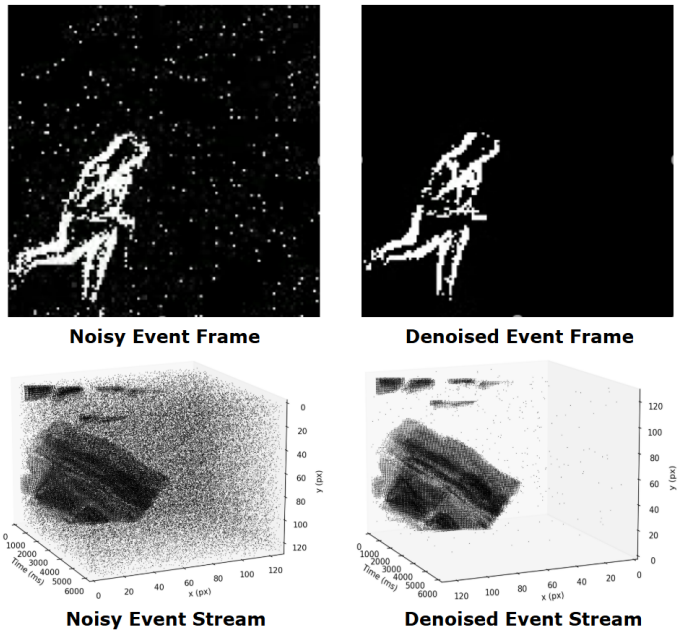

Fig. 9. Simulation results of denoising by LiteDVS.

V. EXPERIMENTAL RESULTS

The proposed LiteDVS was implemented in the SMIC 55 nm logic CMOS process, and its physical layout is shown in Fig. 8. The sensor array achieves a resolution of 128 × 128 with a pixel pitch of 9.27 μm, corresponding to a fill factor of 19.5%. This compact integration demonstrates the feasibility of embedding ROI-based hybrid readout and in-pixel denoising within a mainstream CMOS technology node.

To evaluate denoising performance, the *Hotel-bar* dataset [13] was employed, covering both event-stream and event-frame data formats. The denoising results are presented in Fig. 9. For the event-frame format, noise in the input was almost completely removed, yielding clean and spatially consistent outputs. For the event-stream format, most noise events were also successfully suppressed; however, a small number of spurious events remained in originally blank regions. This effect is attributed to the stochastic nature of noise generation, where local clustering occasionally causes a few false events to mimic the spatiotemporal correlation of genuine activity.

Tab. I summarizes the full test results of LiteDVS. All numerical metrics were obtained through pre-layout simulations, while functionality was verified with post-layout simulations. The results show that LiteDVS consumes **317** fJ/event in the event-stream mode, which is substantially lower than comparable state-of-the-art event-stream denoising designs. In the event-frame mode, the denoising energy reaches **41.8** fJ/event, demonstrating competitive efficiency relative to frame-based alternatives. These results confirm that the proposed architecture achieves an effective balance between noise suppression, latency, and energy efficiency across both data formats.

VI. CONCLUSION

This paper presented LiteDVS, a dynamic vision sensor architecture that reduces data redundancy while preserving low-latency event information. By combining asynchronous event-stream readout with synchronous event-frame accumulation, LiteDVS retains microsecond precision in salient regions while compressing redundant activity elsewhere, thus improving bandwidth efficiency. A lightweight in-pixel denoising unit further supports both modes, suppressing noise at the moment of event generation without extra latency.

Implemented in the SMIC 55 nm CMOS process and validated through simulations, LiteDVS achieves denoising energies of 317 fJ/event in stream mode and 41.8 fJ/event in frame mode. These results demonstrate its effectiveness in combining compression, low power, and robust noise suppression, paving the way for scalable, energy-efficient event-based vision sensors for latency- and power-constrained applications.

REFERENCES

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor," JSSC, 2008.

[2] G. Gallego et al., "Event-Based Vision: A Survey," TPAMI, 2022.

[3] D. Gehrig and D. Scaramuzza, "Low-latency automotive vision with event cameras," Nature, 2024.

[4] K. Kodama et al., "1.22um 35.6Mpixel RGB Hybrid Event-Based Vision Sensor with 4.88um-Pitch Event Pixels and up to 10K Event Frame Rate by Adaptive Control on Event Sparsity," ISSCC, 2023.

[5] M. Guo et al., "A 3-Wafer-Stacked Hybrid 15MPixel CIS + 1 MPixel EVS with 4.6GEvent/s Readout, In-Pixel TDC and On-Chip ISP and ESP Function," ISSCC, 2023.

[6] C. Li, L. Longinotti, F. Corradi, and T. Delbruck, "A 132 by 104 10um-Pixel 250uW 1kefps Dynamic Vision Sensor with Pixel-Parallel Noise and Spatial Redundancy Suppression," VLSI, 2019.

[7] H. Liu, C. Brandli, C. Li, S.-C. Liu, and T. Delbruck, "Design of a spatiotemporal correlation filter for event-based sensors," ISCAS, 2015.

[8] A. Rios-Navarro et al., "Within-Camera Multilayer Perceptron DVS Denoising," CVPRW, 2023.

[9] Q. Zhao, J. Wang, Y. Ji, J. Wu, and G. Shi, "An O(m+n)-Space Spatiotemporal Denoising Filter with Cache-Like Memories for Dynamic Vision Sensors," ICCAD, 2024.

[10] S. Guo, Z. Kang, L. Wang, S. Li, and W. Xu, "HashHeat: An O(C) Complexity Hashing-based Filter for Dynamic Vision Sensor," ASP-DAC, 2020.

[11] S. K. Bose, D. Singla, and A. Basu, "A 51.3-TOPS/W, 134.4-GOPS In-Memory Binary Image Filtering in 65-nm CMOS," JSSC, 2022.

[12] X. Zhang and A. Basu, "A 915–1220 TOPS/W, 976–1301 GOPS Hybrid In-Memory Computing Based Always-On Image Processing for Neuromorphic Vision Sensors," JSSC, 2023.

[13] S. Guo and T. Delbruck, "Low Cost and Latency Event Camera Background Activity Denoising," TPAMI, 2023.

[14] Y. Nozaki and T. Delbruck, "Temperature and Parasitic Photocurrent Effects in Dynamic Vision Sensors," TED, 2017.

[15] D. Seo, J.-G. Kim, I. Yeo, H. Lee, and B.-G. Lee, "Analysis of Pixel Noise in Dynamic Vision Sensors," TCAS-I, 2025.