A DVS-CIS Sensor Data Receiver on FPGA with a 10 Gbps MIPI Controller

Mincheol Cha*, Keehyuk Lee*, Bobaro Chang*, Soosung Kim‡, Taeho Lee*,
Xuan Truong Nguyen*, Tae Sung Kim†, Hyunsurk Ryu*‡
*Dept. of Electrical and Computer Engineering, Seoul National University, South Korea
†Dept. of Electronic Engineering, Sun Moon University, South Korea
‡Neuro Reality Vision Corp., South Korea
{mccha, lkh099, bbrchang, taehov, truongnx}@capp.snu.ac.kr, ts7kim@sunmoon.ac.kr, {kimsoosung, eric.ryu}@nrv.kr

***Abstract*—Fusing a dynamic vision sensor (DVS) and a CMOS image sensor (CIS) is promising in real-time vision applications. However, unlike common CIS, DVS typically come with a custom data format due to their naturally sparse data, which becomes a challenge to fuse DVS and CIS data streams on a general-purpose CPU. To address this problem, this work proposes a DVS-CIS sensor stream receiver on FPGA. The proposed receiver incorporates a cost-effective address decoder and an inline transpose to receive and store a DVS stream on DRAM effectively. At a system level, a host PC can stream the DVS-CIS stream from FPGA via PCIe and display streams on a monitor. Experimental results demonstrate that our architecture can decode up to 13,900fps of DVS frames without incurring any frame drops while concurrently streaming frames at 60fps from a CIS. The design only uses 135 BRAM, 38 DSPs, 69489 LUTs, and 86626 FFs on a Xilinx Zynq+ ZCU106 FPGA board and consumes a power of 6.977 W.**

***Index Terms*—DVS, CIS, High-Speed MIPI Interface, Low Latency, Power Efficiency, Multi-Modal System, G-AER, FPGA**

I. Introduction

In recent years, dynamic vision sensors (DVS) have gained considerable attention due to their complementary capabilities in real-time visual processing. Inspired by the biological visual system, DVS technology has evolved to detect changes in light intensity on a per-pixel basis, operating asynchronously. Each pixel outputs events in response to variations in scene reflectance, providing exceptional temporal precision and low latency, with a dynamic range reaching up to 120dB [2], [8], [14], [15]. This allows DVS to capture fast and dynamic visual information at a submillisecond frame latency [4], [6], [18], enabling DVS to compensate conventional CIS effectively.

DVS-CIS sensor fusion is promising to leverage their unique characteristics, as illustrated in Fig. 1. CIS excels at capturing high-resolution, frame-based images, offering detailed information on the static features of a scene. In addition, the relatively low frame rate of CIS can be compensated for by DVS, which operates at more than 2,000 frames per second (fps). Conversely, the low precision of DVS, e.g., one or two bits per pixel, can be compensated for by CIS. The integration of CIS and DVS technologies enables a richer representation of both static and dynamic environments, benefitting from both the high detail of CIS and the low latency of DVS.

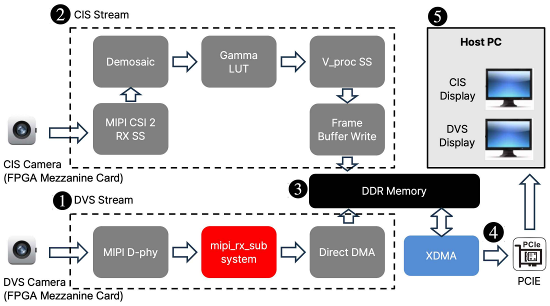

Fig. 1: Overview of DVS-CIS Sensor Fusion Block Diagram

Building a DVS-CIS sensor stream receiver, unfortunately, is a non-trivial task due to the characteristics differences between DVS and CIS. Unlike widely-known CIS, DVS - an emerging technology - typically comes with nonstandard data formats due to its highly sparse data [13], [14]. For example, each pixel in DVS can be represented by address event representation (AER) [13], e.g.,$\langle t, x, y, p\rangle$, where t, x, y, and p represents the timestamp, the row and column addresses, and the polarity flag. Leveraging an advantage of the sparsity of event-based sensory input, a group AER (G-AER) format [14] enables representing an event group, i.e., 1x8 grouped pixels. Additionally, some DVS [13] stream pixel data in a column-wise order instead of the common row-wise order in CIS. These mismatches generally cause a long latency to process a DVS-CIS stream on general-purpose CPU.

To address the aforementioned shortcomings, this work presents a DVS-CIS sensor stream receiver leveraging the Field Programmable Gate Arrays (FPGA) technology. Unlike ASIC, FPGA enables quick prototyping and flexibility to reconfigure hardware for specific use case requirements, i.e., custom data formats in DVS. Unfortunately, a MIPI interface for DVS is not available. More importantly, designing a low-latency DVS-CIS receiver on FPGA is non-trivial due to the aforementioned challenges. Addressing the problem, this work has the following main contributions.

- **A DVS-CIS receiver with a MIPI controller.** We propose a low-latency and cost-effective MIPI controller for DVS. Our controller converts the G-AER format data streamed through the MIPI interface and performs an

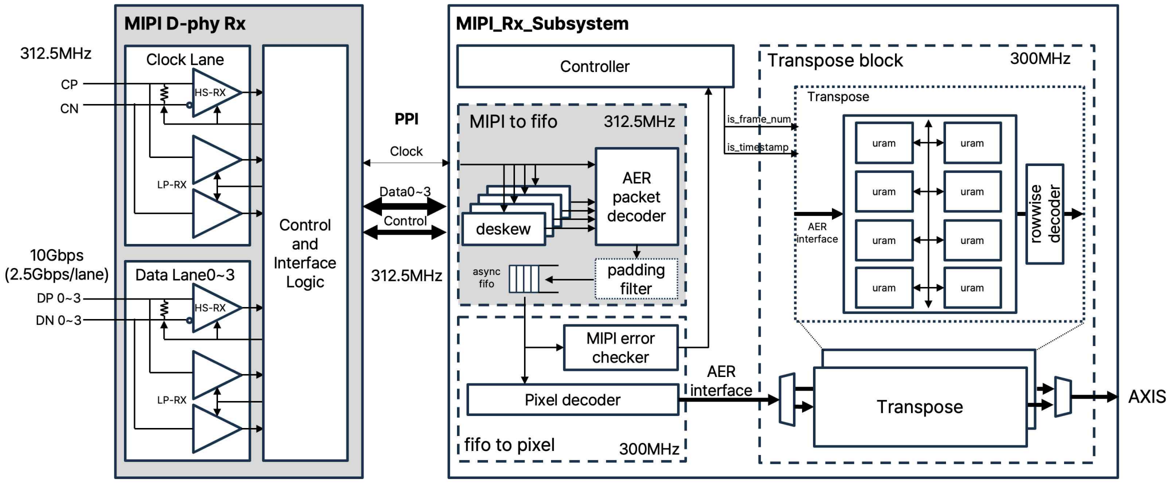

Fig. 2: Block diagram of the MIPI RX subsystem IP, illustrating the key components and their interconnections.

inline transpose to align a column-wise DVS frame to a row-wise CIS one. As DVS and CIS frames are buffered in DRAM, our subsystem is designed to fully utilize the memory bandwidth, avoiding a frame drop. At the system level, a host reads the DVS and CIS streams via PCIe for display and alignment on a monitor.

- **System Prototyping and Evaluation.** We implemented the DVS-CIS receiver prototype with a DVS sensor [13] and LI-IMX274MIPI-FMC CIS camera [5] on an FPGA board [16]. Our architecture can decode up to 13,900fps of DVS frames without incurring any frame drops while concurrently streaming frames at 60fps from a CIS sensor. The host PC can stream the DVS-CIS stream from FPGA via PCIe at the speed of 32Gb/s. The design only uses 135 BRAM, 38 DSPs, 69489 LUTs, and 86626 FFs and consumes a power of 6.977 W.

II. Background and Motivation

A. DVS and DVS-CIS Sensor Fusion.

DVS [8] achieves a frame rate up to 10,000 fps, offering a highly dynamic range, low latency, and energy efficiency. Samsung's DVS, with a resolution of DVS_(W, H): (640,480) consumes 27mW at 100k events per second and up to 50mW at 300Meps, with a dynamic range exceeding 80 dB [14]. Operating asynchronously, DVS detects pixel-level brightness changes [12], ideal for tasks like hand gesture recognition [3] that require ultra-fast, low-power processing [10], [11]. CIS on the other hand, exhibits higher resolution, e.g., CIS_(W, H) of (1920, 1080), at a lower frame rate, such as 60 fps. The DVS-CIS integrated system can combine high-resolution imaging with low-latency, real-time processing, enhancing performance in dynamic scenes while minimizing power consumption.

B. MIPI Interface and a Custom Data Format in DVS.

The Mobile Industry Processor Interface (MIPI) is widely adopted for DVS and CIS [7], [9], thanks to its high-speed data transmission [1]. Although both sensors stream data via MIPI interface, DVS typically accommodates a custom data format such as address event representation (AER). Specifically, each AER event is represented as $\langle t, x, y, p \rangle$, where t, x, y, and p are the timestamp, the row and column addresses, and the polarity flag. Including position and time information along with pixel data, AER incurs a significant bandwidth overhead. For example, an AER-based pixel requires 53 bits with 2-bit data, a 10-bit row address, an 11-bit column, and a 32-bit time stamp, leading to a 26.5x bandwidth gain over 2-bit pixel data. Meanwhile, group AER (G-AER) groups pixels with unified spatial information, reducing the size of pixel-wise spatial information by 8x at the cost of lower sparsity.

Dedicated to DVS sensors, AER and G-AER with highly sparse data present a challenge for streaming and receiving sensor data on general-purpose CPU or GPU. To address the problem, we present a DVS-CIS data stream receiver on FPGA. Specifically, given a DVS data packet under a custom format, i.e., G-AER, a MIPI interface controller streams and decodes the packet at a low latency. The detailed approach is explained in Section III.

C. Implications of DVS-CIS Multi-Modal Input

As DVS and CIS sensors are capable of streaming high-bandwidth data, an on-board NPU will best benefit by directly utilizing both inputs, minimizing data movement. Furthermore, it will take advantage of both the high resolution of CIS and the low-latency, low-power characteristics of DVS.

III. Proposed System Architecture

A. Methodology and System Overview

This section presents a DVS-CIS receiver system on an FPGA. As illustrated in Fig. 1, the system includes a new DVS MIPI interface (❶) and a CIS MIPI interface (❷) that are connected to DVS and CIS sensor boards, respectively. The DVS captures images of size DVS_(W, H) at a rate of DVS_FPS. The stream is first decoded by the MIPI D-phy interface, which handles the high-speed transmission of event-based data from the DVS. The decoded stream is then processed by the mipi_rx_subsystem, which ensures that the data is properly handled and formatted for subsequent operations. Meanwhile, the CIS captures images of size CIS_(W, H) at a rate of CIS_FPS. Following CITE, the data stream from the CIS is handled by the MIPI CSI-2 Receiver Subsystem, Demosaic, Gamma LUT, and Video Processing Subsystem. Both DVS and CIS streams are stored in DDR memory (❸). At a system level, a host PC (❹) can stream out the DVS and CIS data from memory via XDMA (Xilinx PCIe DMA Engine) (❺) and display them on a monitor.

B. Micro Architecture

The proposed MIPI Rx Subsystem IP decodes G-AER format MIPI stream into DVS frames in real-time. Fig. 2 shows its overall architecture, consisting of the MIPI interface with Deskew, Packet and Pixel decoder, and Transpose Module.

*1) **MIPI inteface, Deskew Module:*** Supporting up to four data lanes, each delivering 2.5 Gbps, the MIPI interface operates at a data rate of 10 Gbps. To maintain data integrity at such high speeds, the subsystem integrates a deskew module that synchronizes data streams across lanes, compensating for timing skews that arise from the differential signaling used in high-speed data transmissions.

*2) **G-AER packet decoder, Pixel Decoder:*** Receiving skew-corrected signals from DVS, these modules extract the spatial and temporal information of each event. The G-AER decoder efficiently extracts event coordinates and timestamps, while the Pixel Decoder converts events into pixels, allowing for seamless streaming at 10Gbps without any overhead.

*3) **Transpose Module:*** While the G-AER data format [14] reduces bandwidth and increases throughput over the MIPI interface, it sends packets progressing in the column direction. Consequently, we utilize a transpose module to maintain row-wise pixel order for each frame. The transpose module reorders pixel data in URAM banks [17], optimizing memory access for high-speed processing. Additionally, double buffering is applied for read and write operation concurrency.

Fig. 3 shows how the URAM write location of each G-AER packet is determined using its group and column address. To accomodate input size of 8-pixel groups, each 64-bit URAM word is partitioned into 4 16-bit slots, where each slot can be accessed by partial bit write. Using a very efficient method of bit slicing and concatenation, one of 8 URAM banks are selected, and its write address and slot is determined.

Fig. 4 visualizes this write operation in green boxes. Once all packets of the frame are written to the transpose buffer, all 8 URAMs are read simultaneously to create a 32-pixel group extending in the row direction, shown as a blue box. This is repeated until the entire frame is read out, effectively transposing the input data stream.

C. Host-FPGA Interface and Software Stack

The FPGA's DRAM data layout supports high-speed access between the host and FPGA, assigning for each sensor Frame Buffers and Frame Ready for polling, allowing parallel data reads and writes as show in Table. I.

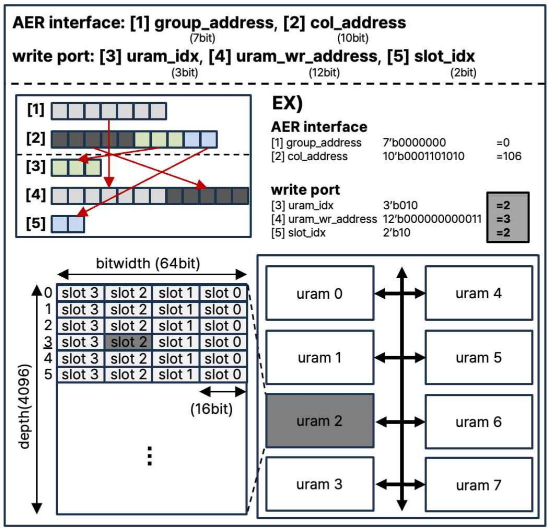
AER interface: [1] group_address, [2] col_address
(7bit)
(10bit)
write port: [3] uram_idx, [4] uram_wr_address, [5] slot_idx
(3bit)
(12bit)
(2bit)
[1]
[2]
[3]
[4]
[5]
EX)
AER interface
[1] group_address 7'b0000000 =0
[2] col_address 10'b0001101010 =106
write port
[3] uram_idx 3'b010 =2
[4] uram_wr_address 12'b000000000011 =3
[5] slot_idx 2'b10 =2
bitwidth (64bit)
0 slot 3 slot 2 slot 1 slot 0
1 slot 3 slot 2 slot 1 slot 0
2 slot 3 slot 2 slot 1 slot 0
3 slot 3 slot 2 slot 1 slot 0
4 slot 3 slot 2 slot 1 slot 0
5 slot 3 slot 2 slot 1 slot 0
(16bit)
depth(4096)
uram 0
uram 1
uram 2
uram 3
uram 4
uram 5
uram 6
uram 7

Fig. 3: Transpose module URAM pixel mapping, illustrating the slot-based memory organization and the DVS pixel data addressing scheme.

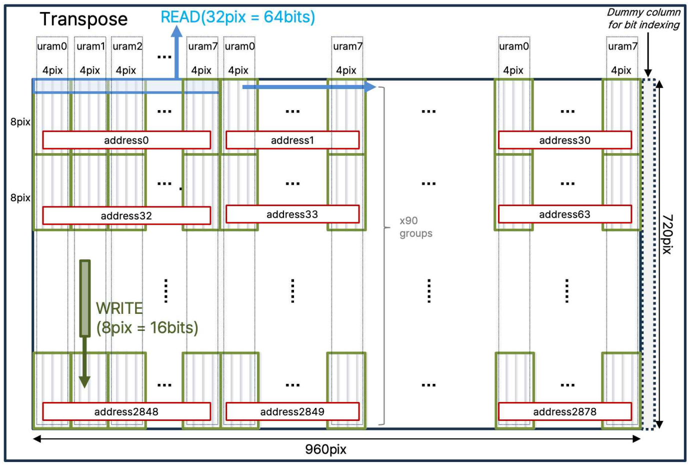

Fig. 4: Detailed architecture of the Transpose module in Fig. 2 in case of (DVS_W, DVS_H) = (960, 720), highlighting data reordering and memory access patterns.

TABLE I: DRAM Data Layout for CIS and DVS

Sensor	Data Type	Size	Address Range
CIS	Frame Ready	5B	0x1010_0000 - 0x1010_0005
DVS	Frame Ready	70B	0x1020_0000 - 0x1020_0046
CIS	Frame Buffer	29MB	0x2000_0000 - 0x21DA_9C00
DVS	Frame Buffer	11MB	0x4000_0000 - 0x40B8_9200

For the Host-FPGA Interface, we used PCIe Gen 3 with four lanes of 32Gbps bandwidth in total to poll data from the Frame Buffer when the frame is ready.

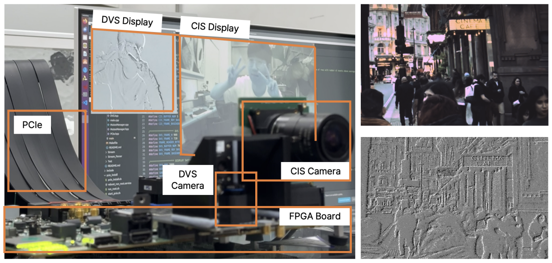

Fig. 5: Sensor fusion environment implemented on the Xilinx ZCU106 board, featuring a sample image of a London street view captured within this setup.

IV. Evaluation

A. Methodology

System Implementation. We implemented the proposed DVS-CIS streaming receiver on a Xilinx ZCU106 FPGA board [16]. The FPGA board is connected to two sensor boards: (1) a custom DVS sensor [13] and (2) an LI-IMX274MIPI-FMC CIS camera [5]. For visualization, a host PC receives and displays DVS and CIS frames streamed from the FPGA.

Settings. Following the specifications [5], [13], the CIS sensor is configured at the FHD resolution at 60 fps, e.g., CIS_(W, H, FPS) of (1920, 1080, 60), and the DVS sensor is configured at the 960x720 resolution at 2,000 fps (e.g., DVS_(W, H, FPS) of (960, 720, 2000)). We empirically determined the CIS and DVS buffer sizes (CIS, DVS)_BUF_NUM as (5, 70). The operating clock frequency of the core is set to 300 MHz while the DRAM clock is configured to 1067MHz. The bus is 64 bits wide, resulting in peak DMA bandwidth of 19.2Gb/s.

B. Experimental Results

1) DVS-CIS System: The DVS-CIS system resource usage is summarized in Table. II.

TABLE II: CIS DVS System Resource Usage.

Resource	Utilization	Available	Utilization %
LUT	69489	230400	30
FF	86626	70B	18.8
BRAM	134.5	312	43.11
URAM	16	96	16.67
DSP	38	1728	2.2

Notably, only 38 out of 1,728 DSPs are utilized, resulting in a low utilization rate of 2.2%. There is considerable room for integrating additional processing elements, such as dedicated IP cores, which could significantly improve computational efficiency and expand the operational capabilities of the system.

2) Buffer-Overwritten Prevention: Since the transpose buffer read size is four times larger than the write size, the difference between read and write speed is unlikely to cause an overflow. If an overflow does occur, it indicates that the instantaneous frame rate surpasses (300 MHz × 32 pixels) / (960 × 720 pixels) = 13,900 fps which is 550% faster than the base data supply rate of the DVS. After 24 hours of verifying frame consecutivity through DVS metadata, the transpose buffer is deemed safe from potential overflow.

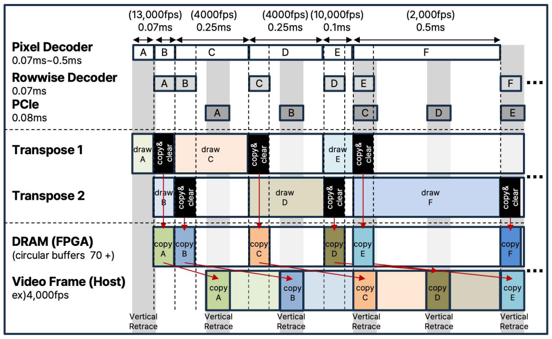

Fig. 6: Timing diagram and buffer management of a high-speed DVS image processing pipeline in response to frame rate fluctuations (2,000fps to 13,000fps) and display at 4,000fps.

3) DVS-CIS Frame Synchronization: To manage fluctuations in event rates generated by the DVS, the timing diagram of how our proposed MIPI Rx Subsystem processes the DVS frames is depicted in Fig. 6. The figure illustrates the detailed flow of video data processing with DVS frames operating at frame rates between 2,000 fps and 13,000 fps and output in desired frame rate. During the vertical retrace, the PCIe interface transfers the decoded data from FPGA DRAM to the Host PC, ensuring synchronization across varying frame rates. As in the figure, this system synchronizes DVS frames at consistent intervals with the desired frequency.

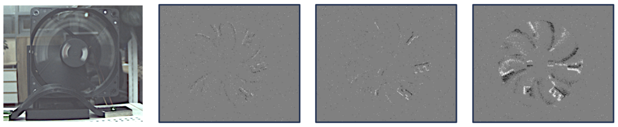

Fig. 7: CIS, DVS frames of a fan running at 30 Hz. The frames are synchronized at 1:33 for CIS frames and DVS frames.

We evaluated the system using a fan running at 30 Hz, as shown in Fig. 7. As our setting of (CIS, DVS)_FPS is (60, 2000) in our experiment, they are synchronized at 1:33 ratio. The letters on the fan blades are visible in the DVS domain.

V. Conclusion

We present a real-time DVS-CIS sensor stream environment capable of decoding up to 13,900 and 60fps respectively. Its high throughput, low latency and low utilization is appropriate for feeding multi-modal sensory input to on-chip NPU, with applications such as enhanced danger detection for surveillance systems, low-latency obstacle detection for Advanced Driver Assistance Systems(ADAS), and providing dynamic, consistent input to robot navigation systems.

Acknowledgment

This work was supported in part by Information Technology Research Center (ITRC) support program (IITP-2024-2020-0-01461) and under the artificial intelligence semiconductor support program to nurture the best talents (IITP-2023-RS-2023-00256081), and in part by the Ministry of SMEs and Startups (No. S3305034).

References

[1] AMD. *MIPI CSI-2 Receiver Subsystem Product Guide: PG232*, March 2023. Accessed: 2024-10-11.

[2] Tobi Delbrück, Bernabe Linares-Barranco, Eugenio Culurciello, and Christoph Posch. Activity-driven, event-based vision sensors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2426–2429, 2010.

[3] Zongpei Fu and Wenbin Ye. A 593nJ/Inference DVS Hand Gesture Recognition Processor Embedded With Reconfigurable Multiple Constant Multiplication Technique. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(6):2749–2759, 2024.

[4] Daniel Gehrig and Davide Scaramuzza. Low-latency automotive vision with event cameras. *Nature*, 629:1034–1040, 05 2024.

[5] Leopard Imaging Inc. Li-imx274mipi-fmc datasheet, 2024. Rev. 1.4.

[6] Hyeongi Lee and Hyoseok Hwang. Ev-ReconNet: Visual Place Recognition Using Event Camera With Spiking Neural Networks. *IEEE Sensors Journal*, 23(17):20390–20399, 2023.

[7] Pil-Ho Lee and Young-Chan Jang. A 20-Gb/s Receiver Bridge Chip With Auto-Skew Calibration for MIPI D-PHY Interface. *IEEE Transactions on Consumer Electronics*, 65(4):484–492, 2019.

[8] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128× 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[9] Fuchun Liu, Lei Wang, and Yang Yang. A UHD MIPI CSI-2 image acquisition system based on FPGA. In *2021 40th Chinese Control Conference (CCC)*, pages 5668–5673, 2021.

[10] Iulia-Alexandra Lungu, Federico Corradi, and Tobi Delbrück. Live demonstration: Convolutional neural network driven by dynamic vision sensor playing RoShamBo. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–1, 2017.

[11] Diederik Paul Moeys, Federico Corradi, Emmett Kerr, Philip Vance, Gautham Das, Daniel Neil, Dermot Kerr, and Tobi Delbrück. Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–8, 2016.

[12] Bodo Rueckauer and Tobi Delbruck. Evaluation of Event-Based Algorithms for Optical Flow with Ground-Truth from Inertial Measurement Sensor. *Frontiers in Neuroscience*, 10, 2016.

[13] Hyunsurk Eric Ryu. Industrial DVS design; key features and applications. In *Conf. on Computer Vision and Pattern Recognition*, volume 3, 2019.

[14] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, Yohan Roh, Hyunku Lee, Yibing Wang, Ilia Ovsiannikov, and Hyunsurk Ryu. 4.1 A 640×480 dynamic vision sensor with a 9µm pixel and 300Meps address-event representation. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 66–67, 2017.

[15] Yunjae Suh, Seungnam Choi, Masamichi Ito, Jeongseok Kim, Youngho Lee, Jongseok Seo, Heejae Jung, Dong-Hee Yeo, Seol Namgung, Jongwoo Bong, Sehoon Yoo, Seung-Hun Shin, Doowon Kwon, Pilkyu Kang, Seokho Kim, Hoonjoo Na, Kihyun Hwang, Changwoo Shin, Jun-Seok Kim, Paul K. J. Park, Joonseok Kim, Hyunsurk Ryu, and Yongin Park. A 1280×960 Dynamic Vision Sensor with a 4.95-m Pixel Pitch and Motion Artifact Minimization. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020.

[16] Xilinx Inc. *ZCU106 Evaluation Board User Guide (UG1244)*, October 2019.

[17] Xilinx Inc. *UltraRAM Readback and Writeback Product Guide (PG356)*, August 2021.

[18] Junwei Zhao, Shiliang Zhang, Zhaofei Yu, and Tiejun Huang. SpiReco: Fast and Efficient Recognition of High-Speed Moving Objects With Spike Camera. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7):5856–5867, 2024.