Asynchronous, Spatiotemporal Filtering using an Analog Cellular Neural Network Processor

Jonah P. Sengupta*, Michael A. Tomlinson*, Daniel R. Mendat*, Martin Villemur†, Andreas G. Andreou*
* Department of Electrical & Computer Engineering, Johns Hopkins University
† Embedded Systems Division, Silicon Austria Labs
Email: {jsengup1, mtomlin5, dmendat4, andreou}@jhu.edu, martin.villemur@silicon-austria.com

2023 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-5109-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCAS46773.2023.10181958

***Abstract*—Neuromorphic processing architectures seek to emulate the functionality of the brain by realizing parallel, efficient, event-based processing which can be directly applied to solve many of the pressing problems within artificial intelligence and big data. However, implementation of these systems leads to slow response times, high power dissipation, or incoherent output. In this paper, an analog cellular neural network processing element is demonstrated to perform asynchronous spatiotemporal filtering operations in an area and power efficient manner. It utilizes a pair of analog memories to encode spike timings and perform event-based bandpass temporal processing. Information from the local clique of temporal filters is leveraged by a parallel, spatial processor which maps CNN arithmetic to the current-domain for compact computation. Preliminary circuit verification demonstrated the ability of the element to perform spatiotemporal filtering operations with latencies less than 1.8μs while only consuming 1.6pJ/spike.**

***Index Terms*—neuromorphic hardware, cellular neural network, asynchronous processing, analog VLSI**

I. Introduction

Neuromorphic sensing and processing pose themselves as a promising solution to many of the prevalent issues pertaining to the scaling of artificial intelligence hardware platforms [1]. These architectures have showcased the ability to perform large-scale optimization, scene recognition, and autonomous control tasks in an energy-efficient, massively parallel fashion [2]–[5]. However, platform implementations leave room for improvement. Digital silicon neuron-based frameworks utilize large memories to store neuron states and synaptic weights [6]. These typically utilize time-multiplexed techniques which places a bound on response speed and require clocks whose distribution networks consume power. In contrast, analog neurons allow for higher implementation density and asynchronous operation but device mismatch makes extensive system calibration requisite [5].

This paper presents an analog, cellular neural network (CNN) processing element which seeks to address both of these issues. CNNs are two-dimensional arrays of cells or processing elements (PEs) that utilize local connections to realize spatial computation. These platform have been shown to perform a large variety tasks in an energy-efficient, parallel, and compact fashion [7], [8]. More recent versions of simplicial-CNN arrays have been used to conduct energy-efficient morphological processing of binary images by leveraging symmetric functions [9]. A prior architecture was proposed that provided the means to spatiotemporally filter event-based data asynchronously [10].

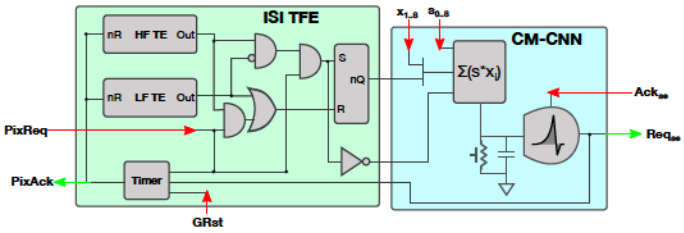


Fig. 1: Block diagram and schematic of the Spike-based CNN Processing Element (SBCNN-PE): it consists of an interspike interval temporal frontend (ISI-TFE) and current-mode CNN (CM-CNN) which are connected to each other, the pixel, and AER with request-acknowledge interfaces.

This paper presents the circuit-level implementation of an enhanced version of this prior work. It utilizes analog memory to represent spike timing information thus providing an area and power efficient alternative to digital memory [11]. Symmetric-CNN arithmetic was then compactly mapped to the current-domain. Low-precision states used for spatial processing allowed for the removal of any external memory and yielded fast response times. Further descriptions of these modules and others will be covered in Section II. Section III demonstrates circuit capability and presents a set of preliminary performance metrics before concluding in Section IV.

II. Asynchronous Processing Element

Spatiotemporal processing of event information is performed using the processing element shown in Figure 1. The spike-based, cellular neural network processing element (SBCNN-PE) is composed of a chain of mixed-signal computational blocks. Input spike data can either flow from an AER receiver interface that captures data from a paired transmitter [12] or from a vertically integrated spike generator [13], [14]. First, event data is used as the input to the interspike interval temporal frontend (ISI-TFE) whose programmed band determines whether to send a request to the next stage. Once requested, spatial processing is realized in the current-mode symmetric CNN (CM-CNN) by leveraging temporal data from the 3x3 neighborhood and current-mode arithmetic to represent the symmetric-CNN computation [15]. An analog

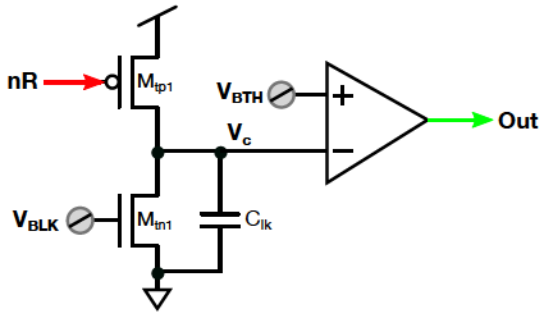


Fig. 2: Schematic of a spike-temporal encoder channel: an analog value, V_c, encodes the time elapsed between spikes, which is then converted into a digital value, Out, using a comparator.

timer circuit is used to perform the requisite handshaking with the spike generator and reset the elements within the ISI-TFE after a programmed amount of time has elapsed.

A. Interspike Interval Temporal Frontend

Data from event-based, silicon retina and other spike generators are anisochronous [16], [17]. Therefore, an interface which provides a decoded, temporal representation of spike timing is needed. Local decoding of incident spikes can be achieved by observing the interspike interval (temporal coding) [18]. Temporal decoders have the benefit of low computation latencies and implementation complexity in the asynchronous domain. These representations can be extended into the spatial domain to render time surfaces which depict spatiotemporal spike activity [19]. Digital realization of time surface activations are area inefficient and require synchronous implementations for local timestamping. In contrast, analog representation of interspike intervals reduces area and power consumption.

To track ISIs, spike timing is encoded in the analog domain by the computational core of the ISI-TFE: the temporal encoder (TE), shown in Figure 2. After a spike activation, the reset (nR) is asserted by the analog timer pulling V_c high. Following its release, pull-up device M_{tp1} is de-activated allowing M_{tn1} to sink current and discharge V_c. When a subsequent spike is received, the ISI will be encoded as the remaining charge on V_c. This voltage is then converted to a digital bit by the comparator. Voltage bias V_{BTH} sets the threshold of the comparator such that Out is raised when $V_c < V_{BTH}$. With current I_{BLK} used to set V_{BLK} on the leak device, each temporal encoder output is logic 0 for time, τ_{te}:

$$\tau_{te} = \frac{C_{lk}(V_{DD} - V_{thr})}{I_{BLK}} \tag{1}$$

where $V_{thr} \approx V_{BTH}$ and assuming M_{tn1} is saturated.

A spike-based, bandpass response can then be constructed by utilizing two temporal encoders with decay times τ_{hf} and τ_{lf} which represent the temporal corners of high-pass and low-pass filters respectively. Assuming identical comparator

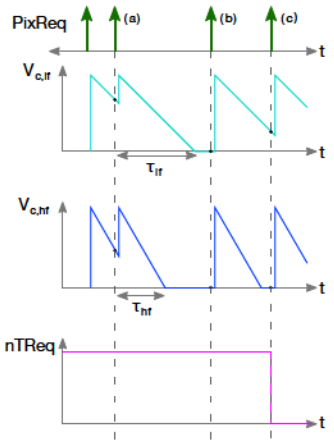


Fig. 3: Timing diagram of the ISI-TFE: three cases are highlighted - (a) high frequency spike (b) low frequency spike (c) spike with requisite interval.

thresholds, maximum bandwidth can be expressed as the difference in leak currents:

$$\tau_{bw} = \tau_{lf} - \tau_{hf} \tag{2}$$

$$= C_{lk}(V_{DD} - V_{thr})\left(\frac{1}{I_{BLK,lf}} - \frac{1}{I_{BLK,hf}}\right) \tag{3}$$

Bandwidth can be further extended by increasing and decreasing the voltage thresholds for the high and low frequency temporal encoders respectively.

A timing diagram of the ISI-TFE is depicted in Figure 3. This illustrates the three possible output cases:

a) High-frequency noise: $t_{isi} < \tau_{hf} < \tau_{lf}$
b) Low-frequency noise: $\tau_{hf} < \tau_{lf} < t_{isi}$
c) Target interval: $\tau_{hf} < t_{isi} < \tau_{lf}$

As shown in Figure 3, output from the ISI-TFE, $nTReq$, is realized on event-based basis using the following logic:

$$nTReq = \neg HF \vee LF \vee \neg PixReq \tag{4}$$

such that HF and LF are the Out comparator signals for the high and low frequency temporal encoders respectively and $PixReq$ is the request from the AER receiver or vertically integrated element. This signal also sets a state register with output $nCMP$ which is then broadcast to the nearest neighbors for spatial processing.

B. Current-mode Symmetric CNN

When successive spikes have an ISI within the passband of the ISI-TFE, a request is then sent to the CM-CNN circuit shown in Figure 4. In this event, the $nTReq$ is asserted and the state bits from the center element and the 8 nearest neighbors, $nCMP[0:8]$, are each used to enable a branch in a pull-up network. Each branch has three devices: the switch tied to the state data, a bias transistor, and a third device representing the structuring element $nSE[0:8]$. Bias V_{BPU} is programmed using a peripheral diode-connected P-device which is sourcing current I_{BPU}. By connecting all the current branches together,

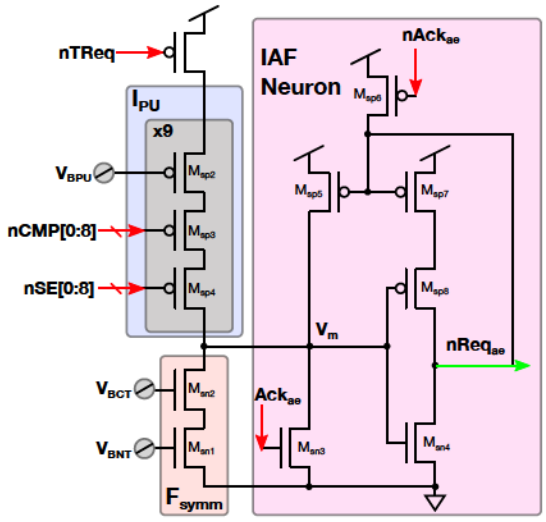


Fig. 4: Schematic of the CM-CNN circuit: Block I_{PU} performs a compact addition using neighborhood activity and structuring element. Block F_{symm} represents a current-mode implementation of the CNN function. Residual current is integrated onto the IAF neuron which handles the output interface with the AER.

TABLE I: Symmetric, Current-mode Functional Mapping

ZRL	F_{symm}	I_{BNT}
1	[0 1 1 ... 1]	0.5*I_{BPU}
2	[0 0 1 ... 1]	1.5*I_{BPU}
3	[0 0 0 ... 1]	2.5*I_{BPU}
N_{ZRL}	[0 0 0 ... 1]	$(N_{ZRL} - 0.5) * I_{BPU}$

a compact addition operation is realized in the current-domain via KCL at the integration node:

$$I_{PU} = \sum_{i=0}^{8} (\neg nCMP[i] \wedge \neg nSE[i]) I_{BPU} \quad (5)$$

Symmetric functions used for spatial processing in CNN arrays consist of vectors of $N_n + 2$ length, such that N_n is the number of connected neighbors. Each bit of the vector represents the vertex of a simplex which is approximating a piecewise linear function, such as the max or min. However, as shown in prior work [10], [15], useful processing operations can be performed by utilizing functions composed of consecutive 0's and 1's. Therefore, to reduce complexity, the symmetric function can be approximated as a threshold value. A current, sunk through devices M_{sn1} and M_{sn2}, implements this functionality in the current domain. An external threshold current I_{BNT} is used to set the gates of two diode connected devices on the periphery which configure cascode bias V_{BCT} and leak bias V_{BNT}. An explicit mapping between the symmetric function, F_{symm}, and programmed leak current, I_{BNT}, is seen in Table I. I_{BNT} is programmed with a 0.5*I_{BPU} margin for each level to accommodate mismatch in the threshold or pull-up devices.

Zero run length (ZRL) is defined as the number of consecutive 0's in the symmetric function [10]. Practically, this represents the minimum amount of activity needed in the neighborhood to elicit a response. A threshold in the CM-CNN realizes the same behavior when expressed in terms of the unit current, I_{BPU}.

The difference between the summed spatial activity and current-mode symmetric function is then integrated onto a positive-feedback, IAF neuron [20], [21] which handles the interface with the AER network. Residual integration current is

$$I_{int} = I_{PU} - I_{BNT} \quad (6)$$

$$= I_{BPU} \sum_{i=0}^{8} (\neg nCMP[i] \wedge \neg nSE[i]) - (N_{ZRL} - 0.5) \quad (7)$$

Latency between the assertion of $nTReq$ and assertion of $nReq_{ae}$ is proportional to this integration current. Slowest response time, $t_{cnn,max}$, is attained when the spatial activity just exceeds the programmed threshold with residual current $0.5I_{BPU}$

$$t_{cnn,max} = \frac{2C_p V_{thr,iaf}}{I_{BPU}} \quad (8)$$

Therefore, worst-case response time can be improved by scaling up the unit current used in the CNN computation. However, this also increases the power consumed by the CM-CNN.

III. Results and Analysis

The various circuit designs outlined in Section II were implemented in a 65nm CMOS technology and simulated in SPICE to characterize their theoretical performance. Beneficial compactness and low-power operation of analog circuits is balanced by larger sensitivity to noise mechanisms. Therefore, fixed-pattern noise analysis was utilized to explore its effects on the performance of ISI-TFE.

Figure 5 shows the results of such analysis. Nominal ISI-TFE bandpass response is represented by the yellow curve. With the $V_{BTH,HF}$=1.1V, $V_{BTH,LF}$=0.2V, and $I_{BLK} = 100pA$, the high frequency and low frequency cutoffs are 12.5kHz and 800Hz respectively. Varying the device parameters of the different components of the temporal encoders reduces the steepness of the bandpass response. When the comparators are varied, the variance can be represented by a fixed threshold offset which moreso affects the high-frequency corner (left corner of blue curve). In contrast, varying the leak devices (M_{tn1} in Figure 2) manifests the variance as a percentage of programmed current thereby affecting both corners (orange curve).

Figure 6 is a transient simulation which demonstrates the activity-driven current-mode, arithmetic of the CM-CNN circuit. Manual activation of $nCMP[0:8]$ and $nSE[0:8]$ allowed for the exploration of all input combinations. The third trace is the sum of output currents from the enabled current branches. Output current is stepped in increments of $I_{BPU} = 5.5nA$ (bottom blue trace) for $\sum \neg nCMP[0:8] <=$

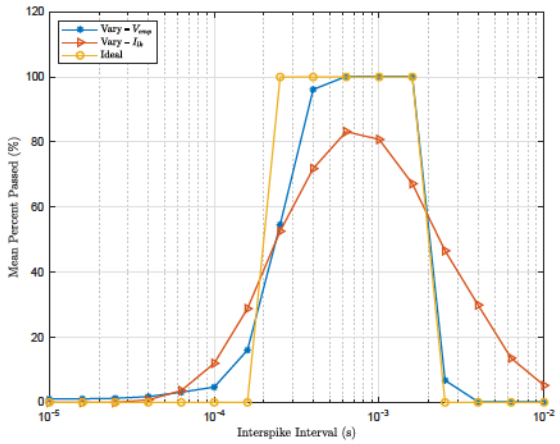


Fig. 5: Bandpass response of the ISI-TFE under three conditions: ideal (yellow), sampling the device parameters within the comparator (blue), and sampling the device parameters within the temporal encoder (orange).

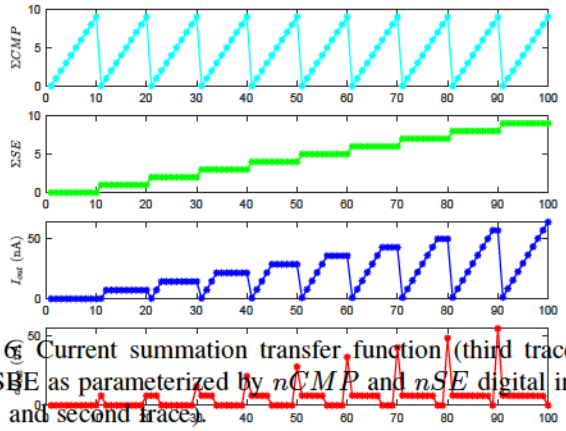
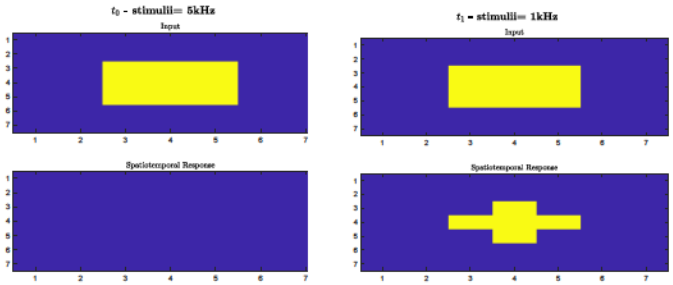


Fig. 6: Current summation transfer function (third trace) of the SBE as parameterized by $nCMP$ and nSE digital inputs (first and second traces).

$\sum \neg nSE[0:8]$ and saturates at a value of $I_{BPU} \sum \neg nSE[0:8]$. It is shown that the $nSE[0:8]$ gates the arithmetic and $nCMP[0:8]$ linearly increments the output current.

Figure 7 showcases the ability of the SBCNN-PE to leverage the temporal filter for spatial processing. In this SPICE simulation, spike streams containing frequencies of 5kHz (a) and 1kHz (b) were injected into a 3x3 array of cells. The ISI-TFE was configured with a passband of 400Hz to 1.3kHz. The neuron threshold was set to $I_{BNT} = 5.5I_{BPU}$ with $I_{BPU} = 1nA$. This corresponds to a F_{symm} with ZRL of 6 which requires a minimum of 5 other active cells in the clique to yield a spike. As shown in the bottom plot of Figure 7a, none of the 5kHz stream is passed to the CM-CNN since the ISI falls outside of the passband. When the 1kHz stream arrives, all of the ISI-TFEs in the array start sending requests to the CM-CNN. However, the bottom plot of Figure 7b shows that only cells in the N, W, E, S and center indices produce output responses thereby realizing the desired filter behavior. This is because those cells have the requisite spatiotemporal activity of 5 active neighbors with valid ISIs.

Table II showcases the metrics captured during verification.



(a) (b)

Fig. 7: Spatiotemporal response of 3x3 SBCNN array during transient simulation. Yellow and purple shaded cells correspond to 10 and 0 spikes per temporal sample respectively ($t_{smp} = 10/(f_{spk,t_0/1})$). Cells outside the 3x3 are inactive.

TABLE II: SBCNN-PE Specifications

Metric	Value
Devices	81T + 2C
Spatial Resolution	3x3 kernel
Max/Min Spike Frequency Corners	4MHz, 72Hz
Worse-case latency	1.8μs
Energy-per-Spike	1.6pJ
Static Power	2nW

A total of 81 devices are needed for the ISI-TFE, CM-CNN, and other modules. Maximum and minimum corner frequencies were extracted using comparator thresholds of $V_{thr} = 1.1, 0.2$V and currents $I_{BLK} = 50nA/10pA$ that ensure subthreshold operation. Worse-case latency and energy-per-spike were extracted using the configuration used to demonstrate the spatiotemporal filtering in Figure 7. The area, energy, and speed preliminary metrics presented are all competitive or strictly better than other asynchronous, neuromorphic architectures [2]–[4].

IV. Conclusion

This paper detailed the design and verification of a spike-based, cellular neural network processing element. This work utilized an analog encoding of spike timing to realize a temporal bandpass filter. If the interspike interval of the incident spike stream resided in the passband of the temporal filter, a request was then sent to the current-mode CNN backend to enable its computation. When simulated in a 65nm technology node, the processing element was shown to be energy (1.6pJ/spike), area (81 devices), and time efficient ($< 1.8\mu$s latency) when performing spatiotemporal filtering.

Self-timed approaches will be the subject of future pursuits. Such schemes would utilize delay-insensitive data representations, asynchronous buffer stages, and metastability filters to coordinate the module handshaking and eliminate the need for the error-prone analog timers used in this work [22]. Finally, implementation and integration need to be undertaken in order to fully realize an asynchronous processor for efficent spatiotemporal filtering.

ACKNOWLEDGMENT

This work was funded by the Defense Advanced Research Projects Agency (DARPA) Microsystems Technology Office (MTO) ReImagine project under Contract HR0011-17-C0071 and in part by the Northrop Grumman Faculty Award to support graduate student research (GT).

REFERENCES

[1] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[2] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *Ieee Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[3] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[4] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in neuroscience*, vol. 9, p. 141, 2015.

[5] A. Neckar, S. Fok, B. V. Benjamin, T. C. Stewart, N. N. Oza, A. R. Voelker, C. Eliasmith, R. Manohar, and K. Boahen, "Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 144–164, 2018.

[6] A. S. Cassidy, J. Georgiou, and A. G. Andreou, "Design of silicon brains in the nano-cmos era: Spiking neurons, learning synapses and neural architecture optimization," *Neural Networks*, vol. 45, pp. 4–26, 2013.

[7] P. S. Mandolesi, P. Julián, and A. G. Andreou, "A scalable and programmable simplicial cnn digital pixel processor architecture," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 5, pp. 988–996, 2004.

[8] M. Di Federico, P. Julián, and P. S. Mandolesi, "Scdvp: A simplicial cnn digital visual processor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 7, pp. 1962–1969, 2014.

[9] M. Villemur, P. Julian, and A. G. Andreou, "Energy aware simplicial processor for embedded morphological visual processing in intelligent internet of things," *IET Electronics Letters*, vol. 54, no. 7, pp. 420–422, Apr. 2018.

[10] J. P. Sengupta, M. Villemur, and A. G. Andreou, "A spike-based cellular-neural network architecture for spatiotemporal filtering," in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2021, pp. 1–6.

[11] R. Sarpeshkar, "Analog versus digital: extrapolating from electronics to neurobiology," *Neural computation*, vol. 10, no. 7, pp. 1601–1638, 1998.

[12] J. Lin and K. Boahen, "A delay-insensitive address-event link," in *2009 15th IEEE Symposium on Asynchronous Circuits and Systems*. IEEE, 2009, pp. 55–62.

[13] M. A. Marwick and A. G. Andreou, "Retinomorphic system design in three dimensional soi-cmos," in *2006 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2006, pp. 4–pp.

[14] T. Finateu, A. Niwa, D. Matolin, K. Tsuchimoto, A. Mascheroni, E. Reynaud, P. Mostafalu, F. Brady, L. Chotard, F. LeGoff, H. Takahashi, H. Wakabayashi, Y. Oike, and C. Posch, "5.10 a 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86µm pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pp. 112–114.

[15] M. Villemur, J. P. Sengupta, P. Julian, and A. G. Andreou, "Morphological, object detection framework for embedded, event-based sensing," in *2022 8th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*. IEEE, 2022, pp. 1–7.

[16] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[17] C. Brandli, R. Berner, M. Yang, S. Liu, and T. Delbruck, "A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[18] D. H. Goldberg and A. G. Andreou, "Spike communication of dynamic stimuli: rate decoding versus temporal decoding," *Neurocomputing*, vol. 58, pp. 101–107, 2004.

[19] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "Hots: a hierarchy of event-based time-surfaces for pattern recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1346–1359, 2016.

[20] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen, "A biomorphic digital image sensor," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 281–294, Feb. 2003.

[21] K. A. Zaghloul, *A silicon implementation of a novel model for retinal processing*. University of Pennsylvania, 2001.

[22] J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs, "A 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver," in *2014 IEEE biomedical circuits and systems conference (BioCAS) proceedings*. IEEE, 2014, pp. 675–678.