Memory in Motion: Exploring Leaky Integration of Time Surfaces for Event-based Eye-tracking

This is the author's version of an article that has been presented at IEEE BioCAS 2024.
The final version of record is available at http://dx.doi.org/10.1109/BioCAS61083.2024.10798345

Memory in Motion: Exploring Leaky Integration of Time Surfaces for Event-based Eye-tracking

Chiara Boretti*‡, Philippe Bich*, Luciano Prono*, Fabio Pareschi*§, Riccardo Rovatti†§ and Gianluca Setti¶§
*DET, ‡PiC4SeR, Politecnico di Torino, Italy - Email: {philippe.bich, chiara.boretti, luciano.prono, fabio.pareschi}@polito.it
†DEI, §ARCES, University of Bologna, Italy - Email: {riccardo.rovatti}@unibo.it
¶CEMSE, KAUST, Saudi Arabia - Email: {gianluca.setti}@gianluca.setti@kaust.edu.sa

***Abstract*—Augmented and Virtual Reality (AR/VR) technologies are gaining popularity to improve healthcare professionals training, with precise eye tracking playing a crucial role in enhancing performance. However, these systems need to be both low-latency and low-power to operate in real-time scenarios on resource-constrained devices. Event-based cameras can be employed to address these requirements, as they offer energy-efficient, high temporal resolution data with minimal battery drain. However, their sparse data format necessitates specialized processing algorithms. In this work, we propose a data pre-processing technique that improves the performance of non-recurrent Deep Neural Networks (DNNs) for pupil position estimation. With this approach, we integrate over time – with a leakage factor – multiple time surfaces of events, so that the input data is enriched with information from past events. Additionally, in order to better distinguish between recent and old information, we generate multiple memory channels characterized by different leakage/forgetting rates. These memory channels are fed to well-known non-recurrent neural estimators to predict the position of the pupil. As an example, by using time surfaces only and feeding them to a MobileNet-V3L model to track the pupil in DVS recordings, we achieve a P10 accuracy (Euclidean error lower than ten pixels) of 85.40%, whether by using memory channels we achieve a P10 accuracy of 94.37% with a negligible time overhead.**

I. Introduction

Augmented Reality (AR) and Virtual Reality (VR) technologies [1]–[3] are revolutionizing healthcare professionals training, driven by the important role of eye-tracking systems. These systems capture user gaze, enabling intuitive interactions which are essential for immersive learning experiences. Ensuring low-latency, accurate and low power pupil tracking is a crucial challenge for seamless training sessions.

A potential solution to address these requirements is represented by Dynamic Vision Sensors (DVS) or event-based cameras [4]–[6] which are neuromorphic video recording devices that enable low-power and high temporal-resolution acquisition of visual information, gaining increasing popularity in a large variety of fields, including surveillance [7], [8], robotics [9], [10] and biomedical science [11]–[13]. The neuromorphic sensors employed in these cameras generate a sparse and asynchronous stream of events that indicate the pixel-local changes in the brightness of the scene. Each event is of the form (x, y, p, t) where x and y are the pixel coordinates where the brightness alteration occurs, p is the polarity of the event (i.e., the direction in which brightness changes) and t is the detection time.

Nonetheless, DVS advantages come with a trade-off. While traditional RGB cameras provide easily interpretable data, i.e., frames containing full-color information, event-based cameras generate sparse data that encode only changes in the scene.

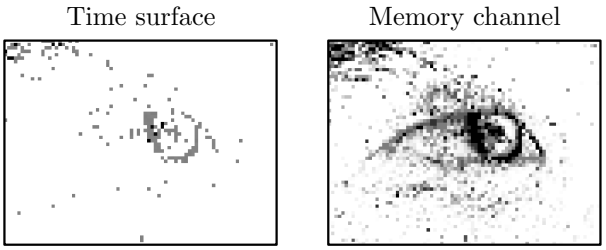

Fig. 1. On the left, a time surface that represents the events collected in a time range Δ_t. On the right, a memory channel that integrates the information of time surfaces over time with a defined leakage/forgetting rate. The method we propose leverages the leaky integration of time surfaces to generate memories channels that improve the capability of non-recurrent estimators to solve eye-tracking tasks.

This means that the design of algorithms for DVS applications is not as straightforward as it is for standard RGB input data. Because of this, studying to generate meaningful representations from streams of events is of paramount importance [14], [15]. For example, a popular strategy is to create time-surfaces [16] where small volumes containing the events in a short time range are represented in a image-like format.

Unfortunately, not every volume obtained in this way contains enough information to estimate the position of the eye's pupil in the given time range. While enlarging the time range for building more informative inputs seems an intuitive solution, it negates the high-speed benefit of event-based cameras. Possible solutions to this problem are the usage of custom recurrent neural network models [17] which have the drawback of being typically more complex to train compared to non-recurrent ones or to leverage an hybrid RGB-DVS framework to perform the eye-tracking task [18].

In this study, we adopt a middle-ground strategy to enhance the performance of non-recurrent networks without relying on a hybrid framework that would require RGB data. Specifically:

- we propose an event pre-processing technique based on memory channels. Figure 1 represents the difference between a time surface and a memory channel. The usage of memory channels improves the performance of non-recurrent DNNs for estimating the eye's pupil position
- we analyze the performance of different DNN models using memory channels compared to those that use time surfaces only, varying the time range in which events are grouped
- we investigate how the accuracy of the eye tracking system changes as the number of memory channels increases

This is the author's version of an article that has been presented at IEEE BioCAS 2024.
The final version of record is available at http://dx.doi.org/10.1109/BioCAS61083.2024.10798345

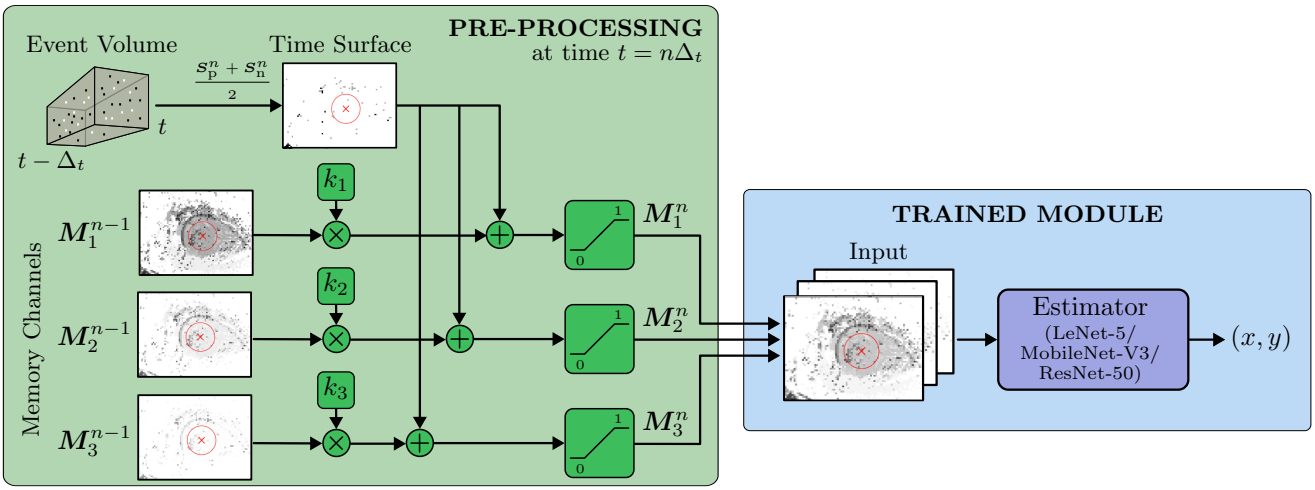

Fig. 2. Schematic of the method used to solve the event-based eye-tracking task. Starting from a volume of events collected in the time range $(t, t + \Delta_t)$, a time surface is generated and then enriched with older information within memory channels built starting from events that were collected in the time range $(0, t)$. From here, the input of the network is structured as the concatenation of three enriched time surfaces while the output is the pupil's position where $x, y \in [0, 1]$. In this work $k_1 = 0.8$, $k_2 = 0.6$ and $k_3 = 0.4$.

- we compute the time overhead introduced by our input pipeline and we compare it to various DNNs inference time

The rest of this paper is structured as follows. In Section II, we present the methodology we propose based on memory channels while in Section III the datasets used to validate our methodology are presented. In Section IV, the DNN models together with the description of the training approach and the metrics used to evaluate the results are discussed. In Section V, the results are discussed. Finally, the conclusion is drawn.

II. Proposed Method: The Memory Channels

We present a methodology that employs an input preprocessing pipeline based on memory channels, i.e., we integrate over time the input events to obtain enriched input data.

We use time surfaces [16] as the inputs of this pipeline, generated by means of the *Tonic* framework [19]. These time surfaces are sampled within time intervals of size Δ_t over all the sensor area. They are defined as $\boldsymbol{S}_{\mathrm{p}}^n$ for positive events and $\boldsymbol{S}_{\mathrm{n}}^n$ for negative events, where $n = 1, 2, 3, \ldots$ are the discrete time-step at times $t = \Delta_t, 2\Delta_t, 3\Delta_t, \ldots$ to which they are associated. The way in which they are calculated can be summarized as

$$\mathbf{S}_p^n(x, y) = e^{-\frac{n\Delta_t - \mathcal{T}_p^n(x,y)}{\tau}} \tag{1}$$

indicating the pixel of the time surface at position (x, y) at time $t = n\Delta_t$ with polarity $p \in \{\mathrm{p}, \mathrm{n}\}$, with

$$\mathcal{T}_p^n(x, y) = \max_{\boldsymbol{e} \in \mathcal{E}_p^n(x,y)} t \in \boldsymbol{e} \tag{2}$$

where $\boldsymbol{e} = (x, y, t, p)$ is an event included in the set $\mathcal{E}_p^n(x, y)$ of the events located at pixel (x, y) in the time range $t \in [n\Delta_t - \Delta_t, n\Delta_t]$. In this work, we use a time constant $\tau = 7 \times 10^{-1}$ s.

These time surfaces are integrated over time into multiple memory channels as

$$\boldsymbol{M}_i^n = \left[k_i \boldsymbol{M}_i^{n-1} + \frac{\boldsymbol{S}_{\mathrm{p}}^n + \boldsymbol{S}_{\mathrm{n}}^n}{2} \right]_0^1 \tag{3}$$

where matrix $\boldsymbol{M}_i^n$ indicates the i-th memory channel at time-step n, $k_i \in (0, 1)$ is the leakage/forgetting factor of the i-th channel, $\boldsymbol{S}_{\mathrm{p}}^n$ and $\boldsymbol{S}_{\mathrm{n}}^n$ are the positive and negative time surfaces at time n and operator $[\cdot]_0^1$ saturates the argument between 0 and 1. $\boldsymbol{M}_i^0$ are defined as all-zero matrices. When k_i is close to 1, the information is kept for long periods of time, so the newest data has less relevance, while when k_i is close to 0, data is quickly forgotten and so the newest data has a stronger impact on the estimator. By using multiple memory channels with different values of k_i, we leverage both short and long-term memory information. The memory channels are combined together in a multi-channel tensor which is then sent to the estimator model.

With this input pipeline, it is possible to use any non-recurrent vision model as the estimator that solves the task. This effectively moves the time dependency of the model entirely to the input pipeline and simplifies the optimization process, since the forward and backward passes through the model depend only on the current time step. Furthermore, this approach allows Δ_t, i.e., the distance between two discrete time instants n and $n + 1$, to be taken as small as desired. This would not be possible by simply using time surfaces as inputs, since they would not contain enough information for the vision model to produce meaningful output, limiting the frequency at which the output could be generated. Figure 2 graphically summarizes the proposed approach.

III. Datasets

In this section, we present the two datasets we use in this work to validate the methodology we propose:

- **3ET Dataset [17]**: the Efficient Event-based Eye-tracking (3ET) derives from the transformation of the RGB LPW

This is the author's version of an article that has been presented at IEEE BioCAS 2024.
The final version of record is available at http://dx.doi.org/10.1109/BioCAS61083.2024.10798345

TABLE I
P10 ACCURACY AND MEAN EUCLIDEAN DISTANCE OF DIFFERENT CONFIGURATIONS OF THE EYE-TRACKING SYSTEM, WITH $\Delta t = 50\,\text{ms}$. WE USE AS INPUT EITHER THE POSITIVE AND NEGATIVE TIME SURFACES OR THREE MEMORY CHANNELS.

	3ET				**EET**			
	P10 accuracy		Mean Euclidean distance		P10 accuracy		Mean Euclidean distance	
	TS only	**Ours**	TS only	**Ours**	TS only	**Ours**	TS only	**Ours**
LeNet-5	81.1%	91.9%	7.2	4.9	89.8%	95.0%	5.3	4.3
MobileNet-V3S	84.9%	92.3%	6.4	4.6	93.8%	97.3%	4.3	3.5
MobileNet-V3L	86.5%	94.6%	5.6	4.6	94.9%	99.1%	3.7	3.2
ResNet-50	90.4%	96.9%	5.1	3.9	95.1%	98.9%	3.4	2.4

dataset [20] using the V2E DVS simulator [21]. The resulting dataset comprises recordings from 22 subjects. Each recording lasts approximately 20 s and target labels representing the (x, y) coordinates of the center of the pupil are provided at a frequency of 100 Hz and at a resolution of 640×480 pixels. While the original dataset counts 62 events videos with the corresponding labels, the authors discard videos that do not generate events over a prolonged period of time (following the criterion highlighted in [17]) and they consider only 16 videos for the training set and 2 for the validation. To compensate for the lack of a test set, we test the accuracy on the validation set, while two registrations from the training set are used instead for the validation process.

- **EET Challenge Dataset**: the Event-based Eye-Tracking (EET) dataset has been presented for the CVPR 2024 EET challenge [22]. It comprises 52 videos entirely captured by means of an event-based camera at a resolution of 640×480. The videos are acquired from 13 different subjects performing different activities such as saccades movement, smooth pursuit, and blink. Data of the training and the validation set are labeled at 100 Hz, except for the test set where the frequency is 20 Hz. The annotations include the coordinates (x, y) of the center of the pupil. The dataset is split into 59% of the recordings for training, 18% for validation, and 23% for testing.

IV. EMPLOYED ESTIMATORS, TRAINING AND METRICS

In this section we briefly describe the non-recurrent DNN models we use to estimate the eye pupil's center, their training and the metrics used to validate our approach.

A. Employed architectures

- **LeNet-5 [23]**: we test this model on the event-based eye-tracking task because of its simplicity. The model is composed of two convolutional layers followed by three linear ones with 200, 84, and 2 neurons, respectively. Given an input size of 80×60, the LeNet model we employ has about 670 000 parameters.
- **MobileNet-V3 [24]**: a widely known lightweight DNN already used in production-grade applications and tuned for edge CPU-based devices.We experiment both with the small (≈ 2.5 million parameters) and large (≈ 5 million parameters) versions of this model, both pre-trained on ImageNet-1k [25].
- **ResNet-50 [26]**: this is the largest model we test in this work and it contains around 25 million parameters. In this work, we use the ResNet-50 model, pre-trained on ImageNet-1k [25].

B. Training

The training sets of the two employed datasets are both composed of many recordings/events streams. Each event stream is divided into small chunks containing events in a time range Δ_t. From each chunk, a time surface is generated. In order to train the DNN models with the methodology we propose, we create sub-sequences of time-consecutive time surfaces. Sub-sequences are then fed to the model in random order during training. Conversely, the validation and the test sets are fully fed to the estimator in a single, chronologically-ordered sequence to emulate the behavior of the system.

Each network is trained for 200 epochs, the batch size is set to 32 and the length of each sub-sequence used during training is 30, with stride 15 over the full recorded sequence. The loss function employed is the Mean Squared Error (MSE). We also use the Adam optimizer [27] with an initial learning rate of 2.8×10^{-4}. For both datasets, the input is down-sampled to 60×80. To ensure robustness and avoid over-fitting of the model, we use data augmentation on the training set. Each sub-sequence used to train the model undergoes a random horizontal/vertical flip and a random horizontal/vertical shift. Furthermore, white Gaussian noise is added to 10% of the input instances.

C. Metrics

In this work, we employ as metrics the mean Euclidean distance and the P10 accuracy. The former is merely the distance between the estimated pupil center and the ground truth, while the latter indicates whether the Euclidean distance of a prediction from the ground truth is below ten pixels.

V. RESULTS

In this section, we report the results obtained using our approach based on memory channels analyzing the improvements in accuracy and the associated time overhead.

A. Accuracy

We first evaluate the effectiveness of our input pipeline based on memory channels against the use of simple time surfaces as inputs. When using time surfaces only, the input of the estimator is composed of three channels, which are $\boldsymbol{S}_{\text{p}}$,

This is the author's version of an article that has been presented at IEEE BioCAS 2024.
The final version of record is available at http://dx.doi.org/10.1109/BioCAS61083.2024.10798345

TABLE II
P10 ACCURACY OF THE EYE-TRACKING SYSTEM EMPLOYING VARIOUS DNNS ON THE EET DATASET, BOTH WITH THE USE OF TIME SURFACES ONLY OR INCORPORATING THE INPUT PIPELINE BASED ON THREE MEMORY CHANNELS WITH DIFFERENT VALUES OF Δ_t.

	$\Delta_t = 20\,\text{ms}$		$\Delta_t = 40\,\text{ms}$	
	TS only	**Ours**	TS only	**Ours**
LeNet-5	73.8%	**85.5%**	78.3%	**86.3%**
MobileNet-V3S	80.9%	**90.6%**	84.1%	**92.4%**
MobileNet-V3L	85.4%	**94.4%**	93.2%	**98.6%**
ResNet-50	84.1%	**93.1%**	94.8%	**98.8%**

TABLE III
PERFORMANCE OF THE EYE-TRACKING SYSTEM EMPLOYING LENET-5 ON THE EET DATASET, USING AS INPUT AN INCREASING NUMBER OF MEMORY CHANNELS WITH $\Delta_t = 50\,\text{ms}$.

# of memory channels	P10 accuracy	Mean Euclidean distance
1	92.7%	4.6
2	93.4%	4.5
3	95.0%	4.3
4	95.1%	4.4

$\boldsymbol{S}_\text{n}$ and their average. Conversely, when using the memory channels, we employ 3 different channels with $k_1 = 0.8$, $k_2 = 0.6$ and $k_3 = 0.4$. Table I compares the performance of the estimators described in Section IV on the 3ET and EET datasets, measured by means of the P10 accuracy and the mean Euclidean distance and with $\Delta_t = 50\,\text{ms}$. The results consistently demonstrate a significant improvement when using memory channels over simple time surfaces.

In Table II, we report the performance of the estimators on the EET dataset with varying values of Δ_t, namely $\Delta_t = 20\,\text{ms}$ and $\Delta_t = 40\,\text{ms}$. Since, at this values of Δ_t, the ground truth labels are not available for the test set, we test the accuracy on the validation set, while two registrations from the train set are used instead for the validation process. The results show that the use of memory channels is required to obtain good performance when lowering Δ_t from $\Delta_t = 40\,\text{ms}$ to $\Delta_t = 20\,\text{ms}$. In fact, the reduction of Δ_t negatively affects the amount of information contained in the time surfaces, but, by integrating them over time, this problem is strongly alleviated.

Finally, we report in Table III the performance of the LeNet-5 model on the original test set of the EET dataset with a varying number of memory channels. The memory channels are defined by the forgetting factors $k_1 = 0.8$, $k_2 = 0.6$, $k_3 = 0.4$ and $k_4 = 0.2$. When one channel is employed, we use only k_1, when two are employed, we use k_1 and k_2, and so on. These results show that in order to get the highest performance, multiple memory channels with different forgetting factors are to be used, so that the estimator can discern the recent information from the old one. Additionally, even by using a single memory channel, we achieve better results compared to the use of simple time surfaces.

TABLE IV
NUMBER OF FLOPS AND COMPUTATIONAL TIME REQUIRED BY THE MEMORY CHANNELS UPDATE COMPARED TO THE COMPLEXITY AND THE INFERENCE TIME OF THE ESTIMATORS. RESULTS ARE COMPUTED USING THE JETSON ORIN NANO SINGLE BOARD COMPUTER SYSTEM

	FLOPS	**Latency**
Memory update	24.00 k	0.23 ms (CPU)
LeNet-5	1.64 M	1.40 ms (GPU)
MobileNet-V3S	0.12 G	20.04 ms (GPU)
MobileNet-V3L	0.23 G	24.26 ms (GPU)
ResNet-50	4.00 G	23.59 ms (GPU)

B. *Computational efficiency*

The methodology that we propose is able to enrich the event-based input information that is required by conventional non-recurrent DNNs. However, this step involves the update of multiple memory channels through a leaky integration process described in (3) that inevitably adds a computational overhead to the deployed system. To estimate this overhead, we evaluate the amount of Floating Point OperationS (FLOPS) to be performed by the execution of the memory channels update and by the inference through the various estimators tested in this work. Additionally, we measure the time latency introduced by each of the parts of the estimation model on the Jetson Orin Nano single computer board system. We report these values in Table IV. The inference time is unvarying with respect to the number of channels employed, since the Jetson Orin GPU computes them in parallel. The latency introduced by the memory update process is only a fraction compared to the total inference time. As an example, the memory update latency is 16.4% of the inference time of LeNet-5 (i.e., the smallest model under test) and only 0.9% of the inference time of MobileNet-V3L.

VI. CONCLUSION

In this work, we introduced an input pipeline to improve the performance of non-recurrent neural models for event-based eye-tracking tasks. In particular, we employed memory channels that integrate with a leakage multiple time surfaces of events over time. In order to differentiate recent information from the old, multiple memory channels are used, with different leakage/forgetting factors. In this way, it is possible to improve the performance of the neural estimators on this event-based task, resulting in a non-recurrent, straightforward structure compatible with low-power, production-grade models such as MobileNet. We performed multiple tests, both on synthetically generated and recorded datasets and with multiple well-known estimators, namely LeNet, MobileNet, and ResNet. All tests showed improvements when the memory channels pre-processing pipeline is employed.

ACKNOWLEDGMENT

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-Generation EU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

REFERENCES

[1] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, "Mobile Augmented Reality Survey: From Where We Are to Where We Go," *IEEE Access*, vol. 5, pp. 6917–6950, 2017. doi:10.1109/ACCESS.2017.2698164 (Accessed 2024-03-30).

[2] N.-N. Zhou and Y.-L. Deng, "Virtual reality: A state-of-the-art survey," *International Journal of Automation and Computing*, vol. 6, no. 4, pp. 319–325, Nov. 2009. doi:10.1007/s11633-009-0319-9 (Accessed 2024-03-30).

[3] M. Sereno, X. Wang, L. Besançon, M. J. McGuffin, and T. Isenberg, "Collaborative Work in Augmented Reality: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 6, pp. 2530–2549, Jun. 2022. doi:10.1109/TVCG.2020.3032761 (Accessed 2024-03-30).

[4] J. Kramer, "An on/off transient imager with event-driven, asynchronous read-out," in *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*, vol. 2, May 2002, pp. II–II. doi:10.1109/ISCAS.2002.1010950 (Accessed 2024-03-30).

[5] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 X 128 120db 30mw asynchronous vision sensor that responds to relative intensity change," in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, Feb. 2006, pp. 2060–2069. doi:10.1109/ISSCC.2006.1696265 (Accessed 2024-03-30).

[6] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-Based Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, Jan. 2022. doi:10.1109/TPAMI.2020.3008413

[7] J. Rodríguez-Gomez, A. G. Eguíluz, J. Martínez-de Dios, and A. Ollero, "Asynchronous event-based clustering and tracking for intrusion monitoring in UAS," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 8518–8524. doi:10.1109/ICRA40945.2020.9197341 (Accessed 2024-04-02).

[8] J. P. Rodríguez-Gómez, A. G. Eguíluz, J. R. Martínez-De Dios, and A. Ollero, "Auto-Tuned Event-Based Perception Scheme for Intrusion Monitoring With UAS," *IEEE Access*, vol. 9, pp. 44 840–44 854, 2021. doi:10.1109/ACCESS.2021.3066529 (Accessed 2024-04-02).

[9] H. Blum, A. Dietmüller, M. Milde, J. Conradt, G. Indiveri, and Y. Sandamirskaya, "A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor," *Robotics Science and Systems, RSS 2017*, Jul. 2017. doi:10.15607/RSS.2017.XIII.035 (Accessed 2024-04-02).

[10] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, no. 40, p. eaaz9712, Mar. 2020. doi:10.1126/scirobotics.aaz9712 (Accessed 2024-04-02).

[11] F. Becattini, F. Palai, and A. D. Bimbo, "Understanding Human Reactions Looking at Facial Microexpressions With an Event Camera," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9112–9121, Dec. 2022. doi:10.1109/TII.2022.3195063 (Accessed 2024-05-24).

[12] M. Gouda, A. Lugnan, J. Dambre, G. van den Branden, C. Posch, and P. Bienstman, "Improving the Classification Accuracy in Label-Free Flow Cytometry Using Event-Based Vision and Simple Logistic Regression," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 29, no. 2: Optical Computing, pp. 1–8, Mar. 2023. doi:10.1109/JSTQE.2023.3244040 (Accessed 2024-05-24).

[13] C. Plou, N. Gallego, A. Sabater, E. Montijano, P. Urcola, L. Montesano, R. Martinez-Cantin, and A. C. Murillo, "EventSleep: Sleep Activity Recognition with Event Cameras," Apr. 2024. doi:10.48550/arXiv.2404.01801 (Accessed 2024-05-24).

[14] R. Tapiador-Morales, J.-M. Maro, A. Jimenez-Fernandez, G. Jimenez-Moreno, R. Benosman, and A. Linares-Barranco, "Event-Based Gesture Recognition through a Hierarchy of Time-Surfaces for FPGA," *Sensors*, vol. 20, no. 12, p. 3404, Jan. 2020. doi:10.3390/s20123404 (Accessed 2024-03-29).

[15] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of Averaged Time Surfaces for Robust Event-Based Object Classification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 1731–1740. doi:10.1109/CVPR.2018.00186 (Accessed 2024-03-29).

[16] 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, 2017. doi:10.1109/TPAMI.2016.2574707

[17] Q. Chen, Z. Wang, S.-C. Liu, and C. Gao, "3ET: Efficient Event-based Eye Tracking using a Change-Based ConvLSTM Network," in *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Oct. 2023, pp. 1–5. doi:10.1109/BioCAS58349.2023.10389062 (Accessed 2024-03-27).

[18] A. N. Angelopoulos, J. N. Martel, A. P. Kohli, J. Conradt, and G. Wetzstein, "Event-Based Near-Eye Gaze Tracking Beyond 10,000 Hz," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2577–2586, May 2021. doi:10.1109/TVCG.2021.3067784 (Accessed 2024-03-27).

[19] G. Lenz, K. Chaney, S. B. Shrestha, O. Oubari, S. Picaud, and G. Zarrella, "Tonic: event-based datasets and transformations." Jul. 2021. doi:10.5281/zenodo.5079802

[20] M. Tonsen, X. Zhang, Y. Sugano, and A. Bulling, "Labelled pupils in the wild: A dataset for studying pupil detection in unconstrained environments," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 139–142. doi:10.1145/2857491.2857520

[21] Y. Hu, S.-C. Liu, and T. Delbruck, "V2e: From Video Frames to Realistic DVS Events," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2021, pp. 1312–1321. doi:10.1109/CVPRW53098.2021.00144 (Accessed 2024-03-28).

[22] Z. W. ChrisJudy, Nanashi, "Event-based eye tracking - AIS2024 CVPR workshop," 2024.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. doi:10.1109/5.726791

[24] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 1314–1324. doi:10.1109/ICCV.2019.00140 (Accessed 2024-03-27).

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi:10.1109/CVPR.2016.90 (Accessed 2023-11-21).

[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017. doi:10.48550/arXiv.1412.6980 (Accessed 2022-09-05).