Event-based optical flow leveraging precise event timing

Supplementary Material

1. Benchmarking and visualization

To benchmark the performance of event-based optical flow models, ground truth data is required. This has been provided through specialized datasets developed within the field. Generally, ground truth data provides optical flow maps at regular time intervals (at a specific frequency) over a period of recorded data. This ground truth represents optical flow as pixel displacements across the visual field. The data is typically sparse (similar to event-based optical flow) due to the collection methodology, which relies on LiDAR, Inertial Measurement Unit (IMU) data, and camera pose information. To evaluate model performance, the predicted flow is compared against ground truth at every available point where both measurements exist. This comparison employs several standard metrics derived from the commonly used definitions provided in [3].

Average End-Point Error (AEE) measures the Euclidean distance between predicted and ground truth flow vectors, providing a direct measure of spatial accuracy. For each pixel location (x, y) at time t, the optical flow predicted by the network is $\mathbf{v}^{\text{pred}}(x, y, t) = [v_x^{\text{pred}}(x, y, t),\, v_y^{\text{pred}}(x, y, t)]^\top$, and the ground-truth vector is $\mathbf{v}^{\text{gt}}(x, y, t) = [v_x^{\text{gt}}(x, y, t),\, v_y^{\text{gt}}(x, y, t)]^\top$, where both represent pixel displacements between frames (in units of px). The AEE is then computed as:

$$\begin{aligned}\text{AEE} &= \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{v}_i^{\text{pred}} - \mathbf{v}_i^{\text{gt}}\right\|_2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\sqrt{(v_{x,i}^{\text{pred}} - v_{x,i}^{\text{gt}})^2 + (v_{y,i}^{\text{pred}} - v_{y,i}^{\text{gt}})^2}. \end{aligned} \quad (1)$$

where N is the number of valid pixels between the ground truth flow and predicted flow, and the index i iterates over all valid spatial locations. Outlier percentage (%Out) is defined as the fraction of flow predictions with an end-point error greater than 3 pixels from the ground truth:

$$\%\text{Out} = \frac{100}{N}\sum_{i=1}^{N} H\left(\|\mathbf{v}_i^{\text{pred}} - \mathbf{v}_i^{\text{gt}}\|_2 - 3\right) \quad (2)$$

where $H(\cdot)$ is the Heaviside step function. Mean Angular Error (MAE) evaluates the directional accuracy of predicted flow by measuring the angular difference between predicted and ground truth vectors. The angular error for each pixel

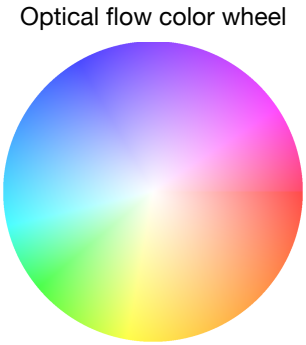

Figure 1. Color encoding for optical flow used in this work, based on [3]. Hue determines the direction of flow and intensity determines the magnitude.

is:

$$\text{AE}_i = \cos^{-1}\left(\frac{\mathbf{v}_i^{\text{pred}} \cdot \mathbf{v}_i^{\text{gt}}}{\|\mathbf{v}_i^{\text{pred}}\|_2\, \|\mathbf{v}_i^{\text{gt}}\|_2}\right), \quad (3)$$

$$\mathbf{v}_i^{\text{pred}} = \begin{bmatrix} v_{x,i}^{\text{pred}} \\ v_{y,i}^{\text{pred}} \\ 1 \end{bmatrix}, \qquad \mathbf{v}_i^{\text{gt}} = \begin{bmatrix} v_{x,i}^{\text{gt}} \\ v_{y,i}^{\text{gt}} \\ 1 \end{bmatrix}. \quad (4)$$

The MAE is then:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\text{AE}_i \quad (5)$$

where the angular error is measured in degrees.
For visualization, optical flow is commonly represented using a color-coding scheme (Figure 1), where pixel hue encodes the flow direction

$$\theta(x, y, t) = \arctan 2\big(v_y(x, y, t),\, v_x(x, y, t)\big), \quad (6)$$

and intensity encodes the magnitude

$$|\mathbf{v}(x, y, t)| = \sqrt{v_x(x, y, t)^2 + v_y(x, y, t)^2}. \quad (7)$$

Alternatively, as also employed in this work, flow vectors can be plotted directly, typically downsampled by skipping vectors to improve visual clarity and interpretability.

2. Additional datasets

2.1. Prophesee

For qualitative assessment, we utilized test datasets provided by Prophesee, including their "Learning to Detect

Figure 2. Optical flow readout from the network when applied to high-resolution Prophesee Gen 4.1 ($1280 \times 720\mathrm{px}$) events capturing pedestrians walking in a street scene. The direction of motion is encoded as color. Optical flow events are accumulated into frames and overlaid at fixed time intervals to visualize the evolution of the scene over time. Top: The network provides clear indication of pedestrian motion, with movement in the distance resulting in lower optical flow magnitude (lighter colors). Bottom: The same event data (cut short for visual clarity) processed with TDE time constants $5\times$ larger ($\tau_{\mathrm{TRG}} = \tau_{\mathrm{FAC}} = 50$ ms, $\tau_{\mathrm{mem}} = 100$ ms), capable of capturing slower motion and therefore more distant pedestrian activity but exhibiting greater residual spiking activity, resulting in a less crisp readout of flow events.

Objects" driving dataset [15], which employs megapixel event cameras with 1280×720px resolution. While these sequences lack ground-truth optical flow labels, they provide diverse real-world scenarios including driving scenes with multiple moving objects and camera egomotion, as well as static camera recordings of pedestrian movements. These high-resolution datasets not only enable visual assessment of our method's performance on complex motion patterns, but also demonstrate scalability to modern megapixel event sensors such as the Prophesee HD, which offer significantly reduced noise compared to the older, lower-resolution cameras used in previous benchmarks. This results in cleaner event streams with fewer spurious events.

2.2. FPV drone racing

We also evaluate our method on a sample from the "UZH-FPV Drone Racing Dataset" [8], which makes use of an mDAVIS 346 (346×260px) event camera capturing First Person View (FPV) drone racing scenarios. While we do not utilize ground-truth optical flow from this dataset, it provides valuable qualitative assessment of our method under extreme motion conditions. The sequences capture drones navigating racing courses at significantly higher velocities and with more aggressive maneuvers than the indoor flying sequences in MVSEC. This represents an important test case, as optical flow estimation is fundamental to autonomous drone racing pipelines [11], where perception latency directly impacts performance. The dataset thus demonstrates our method's ability to handle high-speed, agile motion typical of competitive FPV drone racing.

2.3. Eye-tracking

We use the 3ET+ dataset from the "AIS 2024 Event-Based Eye-Tracking Challenge" [17], recorded with a DVXplorer Mini Dynamic Vision Sensor (DVS) (640×480px). The dataset contains event streams from 13 subjects performing eye-movement tasks such as saccades, reading, smooth pursuit, and blinking. Eye-tracking is an application where low-level optical flow data proves useful [17] for eye-tracking tasks and could be used to augment state-of-the-art eye tracking methods using event cameras [6].

3. Qualitative analysis

Here we present additional qualitative data to demonstrate how the approach scales to higher resolution event cameras and specific applications, while providing intuition for how model parameters affect performance and optical flow selectivity.

Figure 2 shows data from a fixed event camera observing pedestrians walking, optical flow event frames are overlaid to show the passage of time. We can observe both the fidelity and performance of the model, and compare how changing the time constants of the Time Difference Encoder (TDE) units affects behavior. With larger time constants, the model becomes more sensitive to slower motion, allowing optical flow to be detected at greater distances from the camera. In this scene, pedestrians farther away project slower motion onto the image plane, making them more resolvable with increased time constants. The consequence is temporal blurring for large optical flow, caused by increased spiking activity, where residual responses linger and create 'flow trails' that represent past motion rather than the instantaneous scene.

In Figure 3, the effect of varying the size of σ in the Gaussian convolution is visualized. This shows how the system uses local flow information to output coherent normal flow. As σ increases, the flow estimates become more coarse-grained, eventually converging to global optical flow. Figure 4 shows the optical flow from a megapixel camera on the driving sample over time with $\sigma = 50\mathrm{px}$, sufficient to capture the flow of larger scene elements such as a parked car or trees. Figure 5 shows a more complex driving scene containing both self-motion of the camera and independent motion of other objects. This scene demonstrates a combination of rotations (car turning) and translations (forward driving).

Fast optical flow estimation can be useful for on-board processing in fast-moving drones. Figure 6 shows an example of the network using FPV drone racing event camera data during agile maneuvers, resulting in rapidly changing optical flow patterns. This example demonstrates the network accurately capturing such dynamic optical flow.

Eye tracking using event cameras has attracted significant interest due to their potential for low-latency, low-power embedded systems capable of tracking gaze direction, with direct applications in Augmented Reality (AR) and Virtual Reality (VR) [2]. Various approaches to this problem have emerged, accompanied by specialized datasets for training neural networks. Since some methods utilize optical flow as low-level information for iris tracking [6], we applied our optical flow network to eye motion data to evaluate its performance on this task. We computed optical flow at regular accumulation intervals of 10 ms, with results shown in Figure 7, demonstrating the network's applicability to more microscopic scenarios.

4. System description

This work presents simulations of an event-based system that, while algorithmically faithful to the underlying principles, represents an approximation of what a dedicated hardware implementation would achieve. To fully realize the anticipated benefits in terms of power consumption and processing latency, a specialized system would need to be implemented as an Application Specific Integrated Circuit (ASIC) or programmed on a Field Pro-

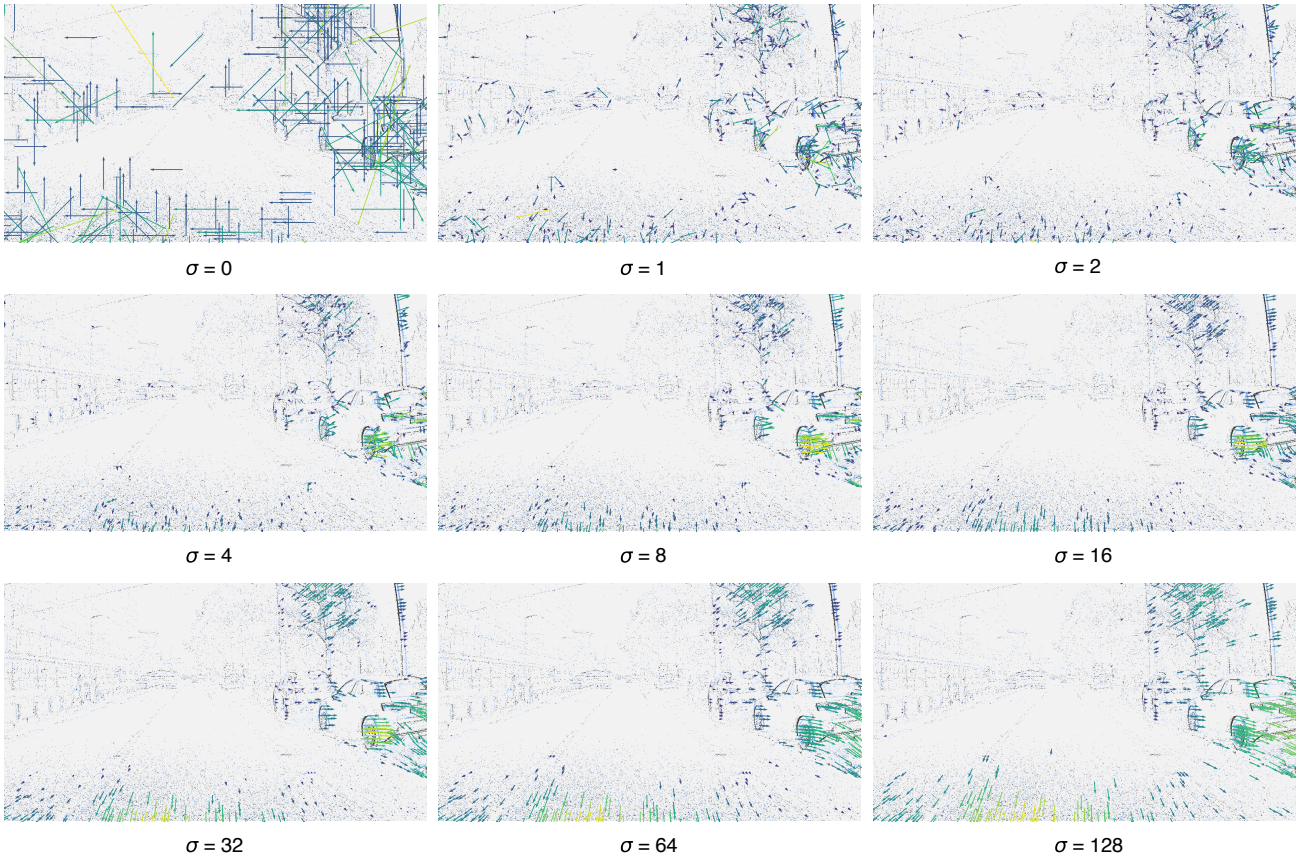

Figure 3. Increasing σ of Gaussian kernels for x and y flow components shows how local flow is smoothed at different scales. The σ value determines the transition from normal flow to flow patterns of larger features. Flow events (arrows) are downsampled by a factor of 10 for visualization and overlaid on top of the raw event-frames. The inset shows a zoomed-in section with flow events downsampled by a factor of 5. This example uses the driving sample from a Prophesee event camera dataset.

grammable Gate Array (FPGA). Such a hardware realization would share architectural similarities with many neuromorphic chips [1, 7, 10, 13, 16, 18] but should incorporate specialized synaptic gating mechanisms and support the high event-throughput of modern event cameras. A key advantage of the approach taken in this work is the dramatic reduction of stored network parameters to a negligible amount, which reduces the memory overhead that is required for storing synaptic weights in many networks with a large number of layers. The simulation-deployment gap is also reduced since there is no need to train the network to be compatible with hardware constraints such as bit precision and asynchronicity [4, 14].

The system architecture would require an event mapper to efficiently route incoming events to dedicated TDE processing cores. These cores would emulate the temporal decay dynamics using either digital logic or analog circuit implementations, with the choice depending on the required scale, power and precision constraints given the potentially large number of processing units needed for high-resolution applications. Kernel operations can be implemented through either dedicated memory arrays storing precomputed values (Look Up Table (LUT)) or real-time on-the-fly computation for parameterizable kernels such as the Gaussian case outlined in Algorithm 1. While our simulations necessarily discretized time for computational tractability, the hardware implementation could operate fully asynchronously or synchronously with low real-time time-step, with kernel calculations also performed in an event-driven manner [1, 16]. Additional traces can maintain real-time flow estimates (analogous to the accumulation period). This representation can then be updated and sampled by input events to provide a graded (non unitary) readout of optical flow in an event packet containing flow phase and magnitude information at a location in the visual field. Although kernel size represents a computational bottleneck, this limitation can be addressed through kernel dilation techniques, as demonstrated in Figure 8. This ap-

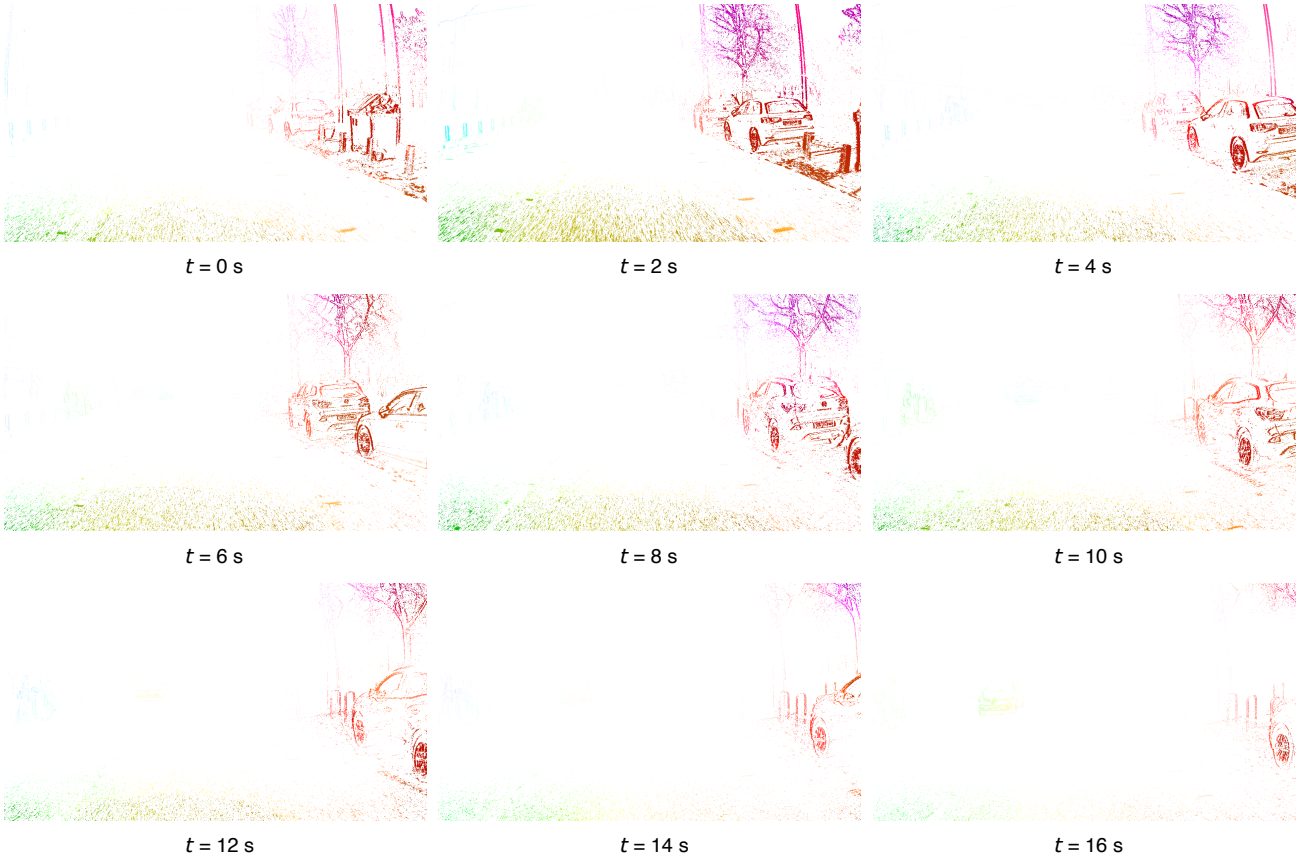

Figure 4. Our optical flow estimation for a megapixel event camera capturing a driving sequence over time.

proach yields comparable results to full-resolution kernels while reducing computational requirements, as evidenced in the results shown in Figure 5 where the kernel was dilated and quantized to the one shown at the bottom right of Figure 8. Existing neuromorphic chips have demonstrated kernel sizes of 5×5 [9], 16×16 [16] and 64×64 [7, 12] with end-to-end latencies reported on the order of microseconds for fully asynchronous and sensor on-chip [5, 16], suggesting that similar end-to-end latencies can be achieved by taking our approach.

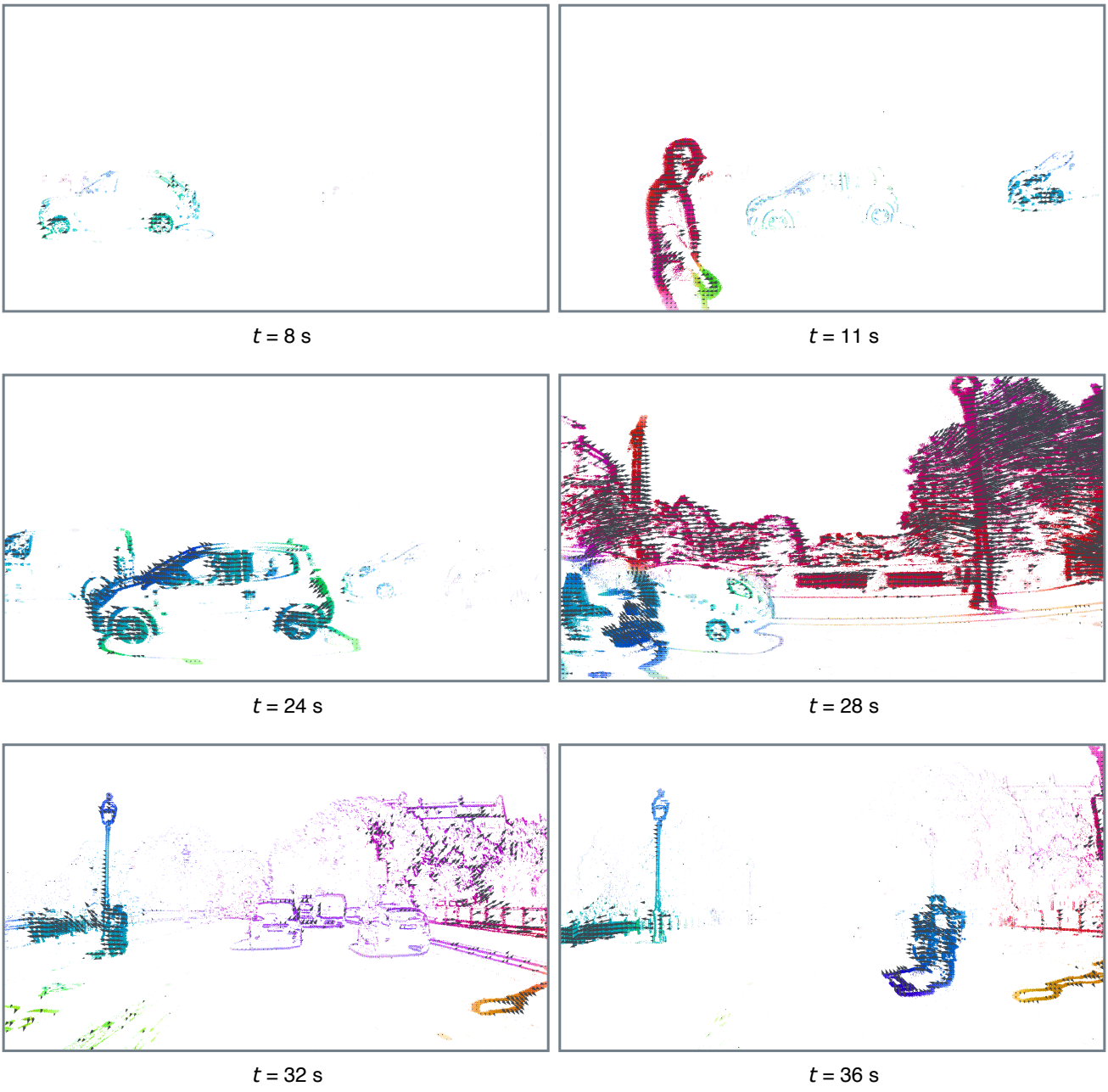

Figure 5. Our optical flow estimation in a dynamic driving scenario. The event-flow frames begin at $8\,\mathrm{s}$ with the vehicle waiting to merge into traffic. At $11\,\mathrm{s}$, a pedestrian crosses in front of the car, note how the swinging motion of their arm produces distinctly different flow patterns from their body. The scene continues at $24\,\mathrm{s}$ as a vehicle ahead pulls away, creating space in traffic. At $28\,\mathrm{s}$, our vehicle executes a maneuver, pulling into the road and turning into the lane, this moment captures both large-scale global motion from the vehicle's movement and localized relative motion from surrounding objects. By $32\,\mathrm{s}$, the vehicle settles into steady forward motion along the road, until $36\,\mathrm{s}$ when a motorcycle overtakes, adding some relative motion to the scene. In this specific example, a Gaussian kernel with $\sigma = 50\mathrm{px}$ and size $100 \times 100\mathrm{px}$ is dilated by a factor of 5, effectively reducing it to a 20×20 kernel. Additionally, the weights are quantized to 2 bits, resulting in a kernel that requires only 100 bytes of storage. This kernel is shown in Figure 8.

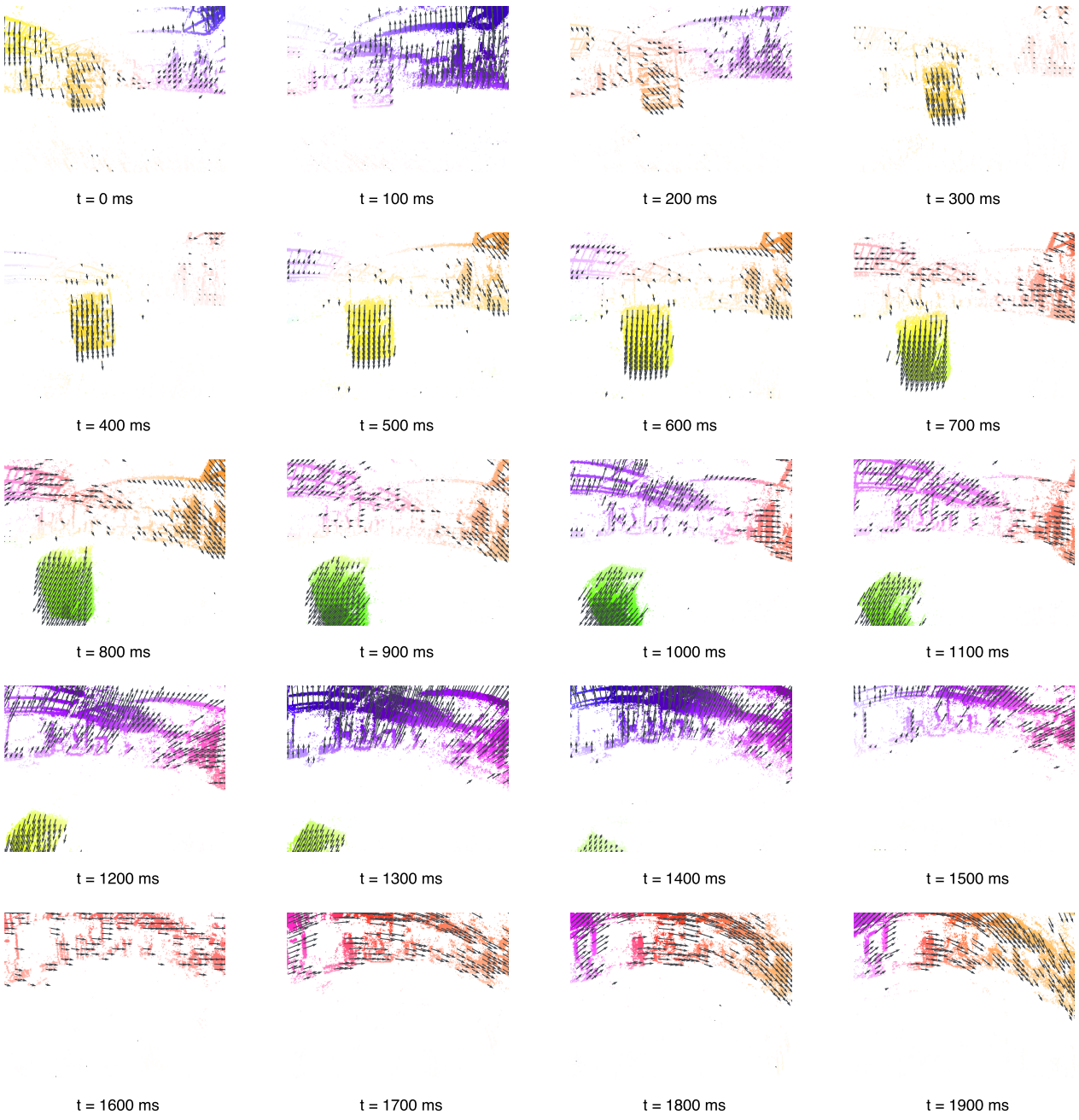

Figure 6. The network applied to FPV drone racing data [8] (indoor_forward_3). The human piloted drone navigates through gates inside a warehouse environment. Color coding represents optical flow magnitude and direction, with downsampled arrows overlaid to visualize flow vectors. This scenario exemplifies a use case where rapid optical flow estimation has high utility [11].

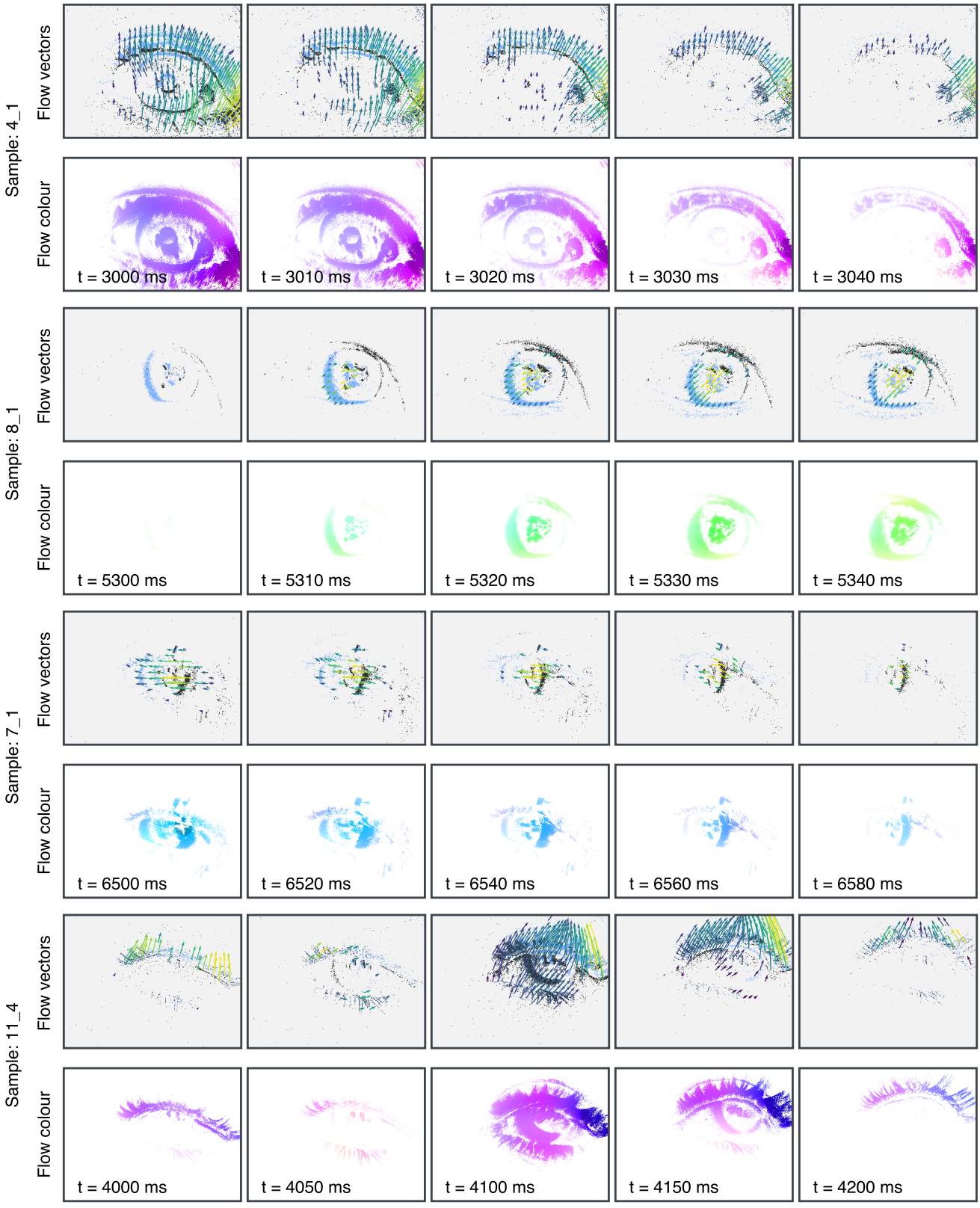

Figure 7. Several examples of the network applied to eye tracking event camera data for different test subjects. Recorded with a DVXplorer Mini DVS (640×480px).

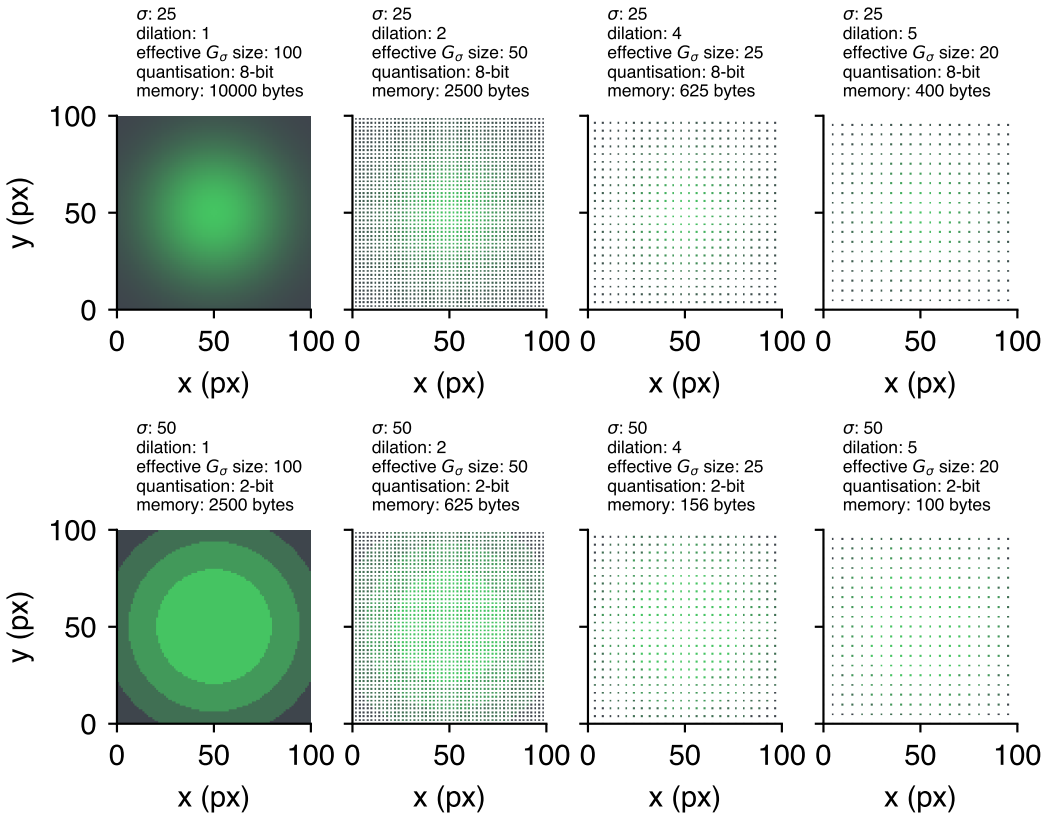

Figure 8. Visualization of kernel dilations and parameter quantization, aimed at reducing the number of synaptic weight parameters and the associated memory usage.

Algorithm 1 Event-based Gaussian convolution

Require: TDE event (x, y, v) at coordinates with value, σ (Gaussian kernel sigma), κ (truncate factor), v is cardinal polarity ($e.g.$ left, right $= -1, 1$)

Ensure: Output events $\mathcal{E}$

function CONV1D($(x, y, v), \sigma, \kappa,$ axis)
  $r \leftarrow \lceil \sigma\kappa \rceil$, $\alpha \leftarrow (2\pi\sigma^2)^{-1/2}$
  for $d = -r, \ldots, r$ **do**
    $(x', y') \leftarrow (x + d, y)$ if horizontal [H], $(x, y + d)$ if vertical [V]
    $w \leftarrow \alpha \exp(-d^2/2\sigma^2)$ ▷ calculated on-the-fly or stored in LUT
    if (x', y') in bounds **then**
      Output (x', y', vw)
    end if
  end for
end function
function PROCESSEVENT($(x, y, v), \sigma, \kappa$)
  $\mathcal{E}_h \leftarrow$ CONV1D($(x, y, v), \sigma, \kappa,$ H) ▷ Horizontal processing unit
  $\mathcal{E} \leftarrow \bigcup_{(x', y', v') \in \mathcal{E}_h}$ CONV1D($(x', y', v'), \sigma, \kappa,$ V) ▷ Vertical processing unit
  return $\mathcal{E}$ ▷ Output event stream
end function

References

[1] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, Brian Taba, Michael Beakes, Bernard Brezzo, Jente B. Kuang, Rajit Manohar, William P. Risk, Bryan Jackson, and Dharmendra S. Modha. TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10):1537–1557, 2015. 4

[2] Anastasios N. Angelopoulos, Julien N.P. Martel, Amit P. Kohli, Jörg Conradt, and Gordon Wetzstein. Event-Based Near-Eye Gaze Tracking Beyond 10,000 Hz. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2577–2586, 2021. 3

[3] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. 1

[4] Ugurcan Cakal, Maryada, Chenxi Wu, Ilkay Ulusoy, and Dylan Richard Muir. Gradient-descent hardware-aware training and deployment for mixed-signal neuromorphic processors. *Neuromorphic Computing and Engineering*, 4(1):014011, 2024. 4

[5] Luis Camunas-Mesa, Carlos Zamarreno-Ramos, Alejandro Linares-Barranco, Antonio J. Acosta-Jimenez, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. An Event-Driven Multi-Kernel Convolution Processor Module for Event-Driven Vision Sensors. *IEEE Journal of Solid-State Circuits*, 47(2):504–517, 2012. 5

[6] Qinyu Chen, Chang Gao, Min Liu, Daniele Perrone, Yan Ru Pei, Zuowen Wang, Zhuo Zou, Shihang Tan, Tao Han, Guorui Lu, Zhen Xu, Junyuan Ding, Ziteng Wang, Zongwei Wu, Han Han, Yuliang Wu, Jinze Chen, Wei Zhai, Yang Cao, Zheng-jun Zha, Nuwan Bandara, Thivya Kandappu, Archan Misra, Xiaopeng Lin, Hongxiang Huang, Hongwei Ren, Bojun Cheng, Hoang M. Truong, Vinh-Thuan Ly, Huy G. Tran, Thuan-Phat Nguyen, and Tram T. Doan. Event-Based Eye Tracking. 2025 Event-based Vision Workshop, 2025. arXiv:2504.18249. 3

[7] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(01):82–99, 2018. 4, 5

[8] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are We Ready for Autonomous Drone Racing? The UZH-FPV Drone Racing Dataset. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, pages 6713–6719, 2019. 3, 7

[9] Charlotte Frenkel, Jean-Didier Legat, and David Bol. A 28-nm Convolutional Neuromorphic Processor Enabling Online Learning with Spike-Based Retinas. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020. 5

[10] Hector A. Gonzalez, Jiaxin Huang, Florian Kelber, Khaleelulla Khan Nazeer, Tim Langer, Chen Liu, Matthias Lohrmann, Amirhossein Rostami, Mark Schöne, Bernhard Vogginger, Timo C. Wunderlich, Yexin Yan, Mahmoud Akl, and Christian Mayr. SpiNNaker2: A Large-Scale Neuromorphic System for Event-Based and Asynchronous Machine Learning, 2024. 4

[11] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023. 3, 7

[12] Andrew Lines, Prasad Joshi, Ruokun Liu, Steve McCoy, Jonathan Tse, Yi-Hsin Weng, and Mike Davies. Loihi Asynchronous Neuromorphic Research Chip. In *Proceedings of the International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 32–33, 2018. 5

[13] Garrick Orchard, E. Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T. Sommer, and Mike Davies. Efficient Neuromorphic Signal Processing with Loihi 2. In *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259, 2021. 4

[14] Alberto Patino-Saucedo, Roy Meijer, Amirreza Yousefzadeh, Manil-Dev Gomony, Federico Corradi, Paul Detteter, Laura Garrido-Regife, Bernabe Linares-Barranco, and Manolis Sifalakis. Hardware-aware training of models with synaptic delays for digital event-driven neuromorphic processors, 2024. arXiv:2404.10597. 4

[15] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to Detect Objects with a 1 Megapixel Event Camera. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 16639–16652. Curran Associates, Inc., 2020. 3

[16] Ole Richter, Yannan Xing, Michele De Marchi, Carsten Nielsen, Merkourios Katsimpris, Roberto Cattaneo, Yudi Ren, Yalun Hu, Qian Liu, Sadique Sheik, Tugba Demirci, and Ning Qiao. Speck: A Smart event-based Vision Sensor with a low latency 327K Neuron Convolutional Neuronal Network Processing Pipeline, 2024. arXiv:2304.06793. 4, 5

[17] Zuowen Wang, Chang Gao, Zongwei Wu, Marcos V. Conde, Radu Timofte, Shih-Chii Liu, Qinyu Chen, Zheng-Jun Zha, Wei Zhai, Han Han, Bohao Liao, Yuliang Wu, Zengyu Wan, Zhong Wang, Yang Cao, Ganchao Tan, Jinze Chen, Yan Ru Pei, Sasskia Bruers, Sebastien Crouzet, Douglas Mclelland, Oliver Coenen, Baoheng Zhang, Yizhao Gao, Jingyuan Li, Hayden Kwok-Hay So, Philippe Bich, Chiara Boretti, Luciano Prono, Mircea Lica, David Dinucu-Jianu, Catalin Griu, Xiaopeng Lin, Hongwei Ren, Bojun Cheng, Xinan Zhang, Valentin Vial, Anthony Yezzi, and James Tsai. Event-Based Eye Tracking. AIS 2024 Challenge Survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5810–5825, 2024. 3

[18] Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, Lei Deng, Tianyi Yan, Carsten

Nielsen, Sadique Sheik, Chenxi Wu, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1):4464, 2024. 4