



# High-Frequency Near-Eye Ground Truth for Event-Based Eye Tracking

Andrea Simpsi<sup>1</sup> <sup>(✉)</sup>, Andrea Aspesi<sup>1,2</sup> , Simone Mentasti<sup>1</sup> ,  
Luca Merigo<sup>2</sup> , Tommaso Ongarello<sup>2</sup>, and Matteo Matteucci<sup>1</sup> 

<sup>1</sup> Department of Electronics, Information and Bioengineering (DEIB),  
Politecnico di Milano, Via Ponzio 34/5, 20133 Milan, Italy  
{andrea.simpasi, andrea.aspesi, simone.mentasti, matteo.matteucci}@polimi.it  
<sup>2</sup> EssilorLuxottica Italia S.p.A., Piazzale Cadorna 3, 20123 Milan, Italy  
{andrea.aspesi, luca.merigo, tommaso.ongarello}@luxottica.com

**Abstract.** Event-based eye tracking is a promising solution for efficient and low-power eye tracking in smart eyewear technologies. However, the novelty of event-based sensors has resulted in a limited number of available datasets, particularly those with eye-level annotations, crucial for algorithm validation and deep-learning training. This paper addresses this gap by presenting an improved version of a popular event-based eye-tracking dataset. We introduce a semi-automatic annotation pipeline specifically designed for event-based data annotation. Additionally, we provide the scientific community with the computed annotations for pupil detection at 200 Hz.

**Keywords:** Smart Eyewear · Eye Tracking

## 1 Introduction

Eye tracking is a technology that measures and records the eye movements and positions of an individual [1], providing valuable insights into visual attention and cognitive processes [17, 19]. This technology has profound implications for various applications, ranging from user experience research [7] and medical diagnostics [15] to advanced human-computer interaction systems [6, 25]. Moreover, with the advent and rapid diffusion of virtual reality devices [3], augmented reality systems [4], and smart eyewear, eye-tracking technology has become a central component in these systems to enhance the user experience and provide a more immersive and interactive environment.

Thanks to the proliferation of smart wearables and smart eyewear in recent years [20], research on this topic has witnessed a significant boost. In particular, in recent years, we have seen rapid growth in the availability of commercial eye-tracking technologies [13]. These devices, equipped with advanced sensors, enable continuous and unobtrusive monitoring of eye movements in real-world settings. This has opened up new possibilities for applications such as biometric authentication, emotion recognition, and user experience optimization.

Despite the recent advancements, these new technologies still struggle to provide real-time information, or they require heavy computational power and high-end PCs to process the data [21]. Recent advancements in computer vision have significantly enhanced the accuracy of eye-tracking technology. In particular, machine learning and deep learning approaches have enabled more accurate and robust detection and tracking of eye movements, even in challenging conditions such as varying lighting and head movements [27]. Techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been instrumental in improving the precision and speed of eye tracking systems [10, 18]. Nevertheless, the improved accuracy has come at the cost of using powerful GPUs to run the deep-learning models, making the whole eye-tracking system more resource-demanding and distant from a slim eye-wear system, where all the computation is performed directly on the device.

The growing accuracy in the eye tracking process and the increase in computational requirements are related to how traditional eye tracking has been performed, using cameras looking at the user's eye. Traditionally, an eye-tracking pipeline is divided into a series of refinement blocks that gradually find the pupil in the image, then the center, and track it through time. It is then expected that an increase in accuracy can lead to higher computational requirements. For this reason, in recent years, alternative solutions in eye tracking have been proposed, from non-camera based like microelectromechanical systems (MEMSs) to scan the eyes [24] to photodetector-based solutions [8]. Other solutions focused on new and innovative camera systems, particularly event-based cameras [11], which, unlike traditional frame-based sensors, return only the changing pixels from one frame to the next, in an asynchronous way. This feature can be exploited to detect only the pupil in the frame and drastically increase the performance of the pupil detector while maintaining low computational power.

In particular, event-based eye tracking has proved to be a promising solution. However, annotated data with proper ground truth is required to validate eye-tracking algorithms and train deep-learning models. Camera-based eye-tracking datasets are fairly common and have been recorded for many years [23]. In contrast, to the best of our knowledge, the number of datasets for these new sensors is extremely limited due to the novelty of the event-based approach. Indeed, only four datasets are available [2, 5, 22, 26], but each of them presents some limitations that are further discussed in the following sections.

For this reason, in this work, we present an improved version of the original dataset proposed by Angelopoulos in [2]. In particular, we present a pipeline for semi-automatic annotation of event data for pupil detection and tracking. We also provide the scientific community with the generated pupil position for Angelopoulos's dataset 200 Hz. The computed annotations are made publicly available at<sup>1</sup>.

This work is organized as follows: Sect. 2 presents the current state of the art in eye-tracking, with a specific focus on the event-based scenario and the datasets available. Subsequently, Sect. 3 describes our proposed pipeline for automatic

---

<sup>1</sup> [https://github.com/AIRLab-POLIMI/event\\_based\\_gaze\\_tracking\\_gt](https://github.com/AIRLab-POLIMI/event_based_gaze_tracking_gt).

annotation generation from event data. In Sect. 4, we highlight key statistics regarding the size of the annotations generated. Finally, Sect. 5 offers concluding remarks and summarizes the findings of this paper.

## 2 Related Works

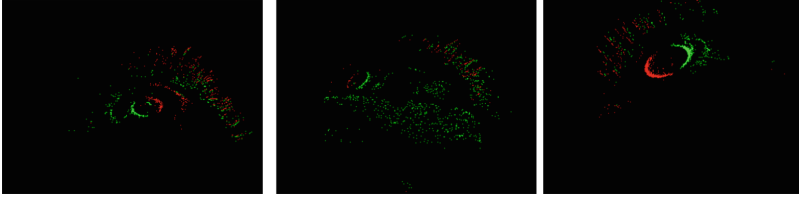
In this section, we discuss the datasets that provide event-camera data specifically for eye-tracking applications. As previously anticipated, due to the novelty of the sensor and application, the number of available datasets is extremely low.

One of the first datasets for the eye-tracking task is the one presented by Angelopoulos in [2]. This dataset has been created using data obtained by the DAVIS sensor, which provides both greyscale and event-based data. The dataset is composed of several eye movements obtained from 27 different users. The ground truth provided is unfortunately limited due to the difficulties in generating a pupil-level annotation from the sparse event data. Indeed, Angelopoulos’s dataset provides as ground truth the looked point on a screen, not the pupil position in the image frame. This type of annotation has been widely used for many years in the eye-tracking field [12] but introduces additional calibration and nonlinearities in the system, which makes a proper evaluation of the pure pupil detection and tracking algorithm less accurate.

A recent addition is the Ini-30 dataset [5], obtained by recording 30 volunteers using a glass frame equipped with two DVXplorer event cameras. To the best of our knowledge, this is the first event-based dataset where labels are provided at high-frequency directly at frame level and where a screen was not employed to guide the users during the acquisition process and with labels dire. The glasses cameras captured natural eye movements; then, data were manually labeled with a variable sampling period ranging from 20.0 ms to 235.77 ms. The 50 Hz labeling frequency, however, is not high enough to capture all eye movements, as presented in the next sections. Additionally, the absence of a standardized protocol during acquisitions, although enhancing variability, might lead to imbalances in the data, thereby affecting its usability.

Another dataset was recently presented as part of the EV-Eye paper [26]. Similarly to [2], two DAVIS346 are used to capture events and also record near-eye grayscale image sequences at a frame rate of 25 fps. In addition, it contains gaze references provided by Tobii Pro Glasses 3. This additional metadata comprises PoGs and pupil diameters of the users at 100 Hz. The dataset is composed by 48 participants (28 male and 20 female) aged between 21 and 35 years. Labels are also provided at sensor level as in [5], but only at slow frequency as estimated from near-eye grayscale images.

Recently a new event-based dataset has also been presented, the 3ET+ dataset [22]. In this scenario the data have been obtained using a DVXplorer Mini event camera. The recording consists of 13 distinct users, each exhibiting various eye movements. Unlike the previous datasets, 3ET+ provides the ground truth annotated 100 Hz. Moreover, they provide two different labels: one binary value to indicate the blink status, and the human-labeled pupil center coordinates. However, the data were collected without an IR-pass filter, so they contain



**Fig. 1.** Examples of frames generated accumulating events 200 Hz

events of object reflections mixed with eye movements. This could impact the generalization capability of deep learning algorithms trained on this dataset.

This work advances the current state of the art by providing additional annotations to the Angelopoulos dataset [2]. Specifically, we retrieve the center of the pupil on the image frame at a rate 200 Hz, a frequency sufficiently high to model all documented eye movements [12]. Additionally, we include annotations describing the current state of the eye, labeling the presence of a blink and the status of the saccade.

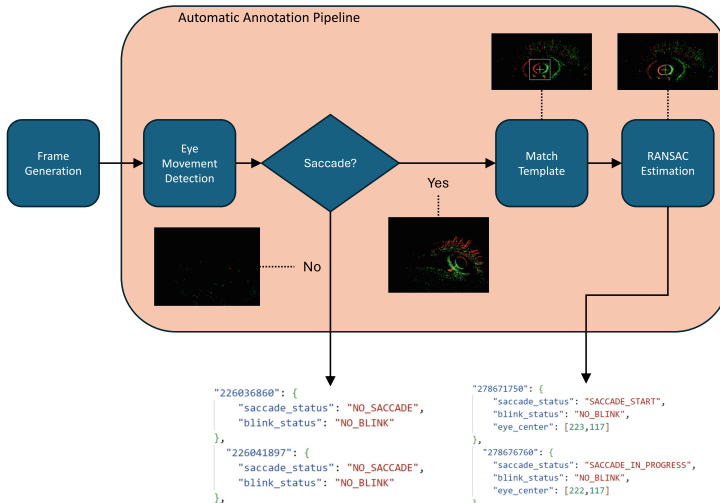
### 3 Ground Truth Semi-automatic Generation

In this section, we describe the methodology used to generate pupil position and blink annotations from Angelopoulos’s dataset [2]. First, the annotations are computed using an automated process. These annotations are subsequently validated and, if necessary, corrected through manual verification.

#### 3.1 Automatic Annotation

A notable drawback of event cameras is the difficulty of directly applying traditional frame-based algorithms to their output data. This issue arises because event cameras generate data as a continuous stream of asynchronous events. Unlike frame-based cameras, event cameras only capture changes in individual pixels, without producing complete images. This limitation impacts the data visualization and the annotation process. Therefore, the first step in the automatic annotation pipeline is to generate RGB frames from the event stream. These frames possess all the characteristics of traditional camera images, allowing them to be visualized and annotated by both the automatic pipeline and the users.

To generate the event frames, we aggregated events at a sampling rate of  $200\text{Hz}$  (i.e., a frame every  $5\text{ms}$ ). This means that all events, which represent changing pixels, in a  $5\text{ms}$  window are summed to reconstruct an RGB image of the eye. This sampling rate was selected to capture all relevant eye movements. Indeed, saccadic movements typically span from  $30\text{ms}$  to  $80\text{ms}$ , while microsaccadic movements occur within a  $10\text{ms}$  to  $30\text{ms}$  range [12]. The need for such a high sampling frequency to accurately capture all types of eye movements



**Fig. 2.** Schema of the automatic annotation pipeline. The system takes the frame generated by accumulating events as input and first predicts if there has been eye movement. If a saccade is detected, the pupil center is determined using a template matching strategy followed by RANSAC estimation.

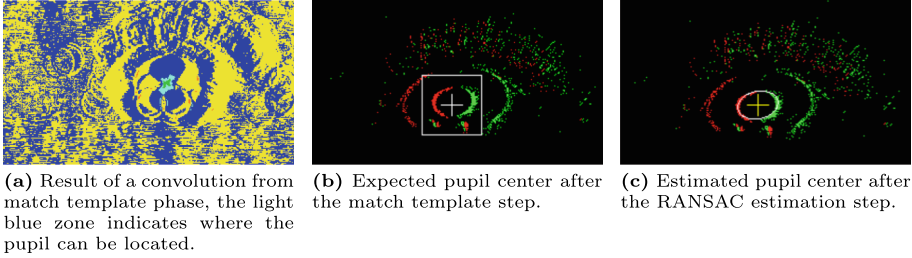
introduces substantial challenges in the annotation process. As the sampling frequency increases, there is a proportional increase in the volume of frames that require annotation, complicating the data processing workflow.

The result of the accumulation process are RGB frames where the only colored points are the events accumulated during the sampling period. The color of the points indicates the polarity of each event: green for positives and red for the negative ones. The output of this first phase of the pipeline is shown in Fig. 1.

The next phase involves automatically generating the raw annotations for each frame. Due to the high frequency at which the data are processed, a significant portion of the frames contain limited information, with most colored pixels attributable to noise. This phenomenon occurs when the eye is nearly stationary; if no movement is detected by the sensor, no events are generated. In these scenarios, where only noise data are available, automatic annotation is problematic due to the lack of a clear pupil. In this scenario, manual inspection would only increase the effort required by the human operator without adding additional benefit to the final result.

The automatic annotation pipeline is shown in Fig. 2, and it is divided into three major steps inspired by the method described in: [16]:

- **Eye Movement Detection:** In this step, we need to detect if either a saccade or a blink is ongoing. We can detect the presence of a saccade using a threshold technique. If the events in the frame are more than 150, there is a saccade; otherwise, it is not a saccade. Also, the first time that a frame is detected as a saccade, it is labeled as *SACCADE\_START*. Similarly, the frame is labeled as



**Fig. 3.** Results from different operations performed during the pupil center localization. As we can see, the RANSAC estimation step increases the accuracy of the estimation made by the match template step.

*SACCADE\_END* if it is the last frame before switching to the frames without detected saccades.

- **Match Template:** If a saccade is ongoing, the match template step finds a tentative eye center using a pattern-matching methodology. To do this, we use a 2D convolution similar to [16], but with a slightly improved kernel compared with the one presented in the original work. In particular, to improve system performance, we employ only 8 different templates that represent different movement directions. These kernels are applied to the polarity images (i.e., the frames reconstructed integrating events data). The output of this phase is shown in Fig. 3a. Then, each result is multiplied to create heatmaps, identifying movement direction. The highest-scoring template indicates the motion direction; a bounding box, representing the Region Of Interest (ROI) of the frame, is built centered at the maximum point of the heatmap. The result is visible in 3b.
- **RANSAC Estimation:** After the match template step, we execute the Random Sample Consensus (RANSAC) algorithm [9], considering only the events inside the ROI detected earlier. In particular, we employ RANSAC to fit an ellipse described using the equation of a generic conic with the  $f$  parameter fixed to  $-1$ . Although the RANSAC fit follows the method described in [16], it uses 1000 iterations in this scenario to prioritize reliable results over fast execution. The final result is visible in 3c.

### 3.2 Fine Tuning Annotation Using Human Annotators

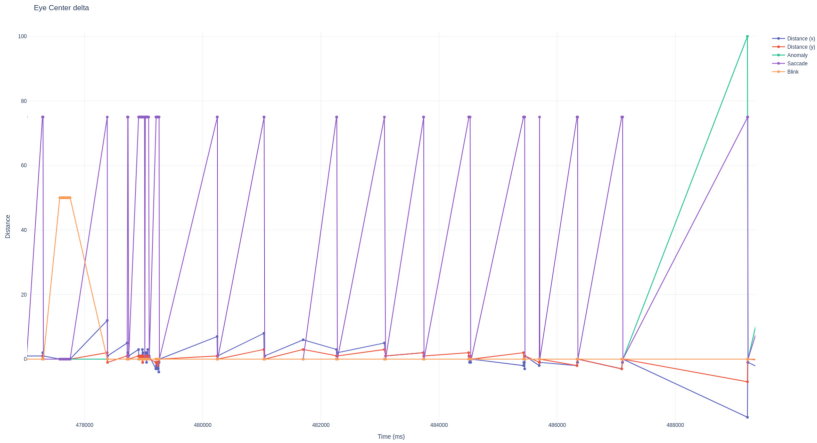
After generating and automatically annotating each frame, as described in the previous section, the labels are inspected by human annotators with expertise in eye-tracking and event cameras to validate and correct the annotations.

The annotation fine-tuning process is divided into two parts. The first part aims to assign each frame a specific label describing the state of the eye, such as a saccade or blink. In this stage, each annotator reviews the automatic annotations generated in the previous step. Annotators are presented with each frame that

contains a number of events exceeding a pre-set threshold, which varies between users. This method excludes all frames with a low number of events, where the eye's state is not clearly defined, and the human annotator would be unable to provide accurate additional information.

Using a simple keyboard shortcut-based interface, each annotator is able to modify the eye center's position, to change the saccade status (*SACCADE\_START*, *SACCADE\_IN\_PROGRESS*, *SACCADE\_END*) and select the current blink status (*BLINK\_STATUS*, *BLINK\_IN\_PROGRESS*, *BLINK\_END*). Having a base annotation greatly speed-up the review process, as for many frames the base annotation itself is correct.

In the second stage, the quality of the labels is evaluated using another custom-made tool. Considering the high frequency of annotations, we expect extremely small eye movements to occur between frames. To leverage this, we introduced a metric based on the difference in eye position between consecutive frames. A threshold for this distance is set to identify potential label anomalies, under the assumption that pupil movement should not exceed this distance within a  $5ms$  interval. The output from this phase is displayed as an interactive plot (see Fig. 4). Reviewers can then zoom into specific plot regions where the threshold is breached and directly jump to the corresponding labels for correction.



**Fig. 4.** Interactive plot for annotation correction. Delta in x,y from an eye position to the next is shown in blue and red. Saccades are marked in violet (rising edge when the saccade starts, falling edge when the saccade ends). Blinks are indicated similarly in orange. At the end of the sequence, a possible anomaly is marked in green. (Color figure online)

This second step is performed iteratively until the plot no longer exhibits any significant anomalies. Some peak distances may persist due to eye movements

occurring during blinks. The presence of these peaks is largely attributed to different physiological mechanisms; for example, downward eye movements during blinks are well-documented in the literature [14]. Moreover, saccades and blinks can be simultaneous.

## 4 Dataset Statistics

Using the method proposed in this paper, we annotated the left eye of six different users from the dataset [2].

We reviewed and annotated 114222 frames, each containing a varying number of events. The total number of frames reviewed was 315213; however, many did not contain the minimum number of events necessary to recognize and detect an eye. These empty frames likely resulted from periods when the eyes were still and did not generate events.

**Table 1.** Table reporting core statistics about each annotated user of the Angelopoulos dataset, computed with the proposed method.

Statistic	User4	User5	User10	User15	User18	User19	User20	Total
Frame analyzed	44020	41233	46818	46008	43470	43125	50539	315213
Annotated Frame	13643	16836	26440	12967	11109	9109	24118	114222
Saccade Counts	406	317	288	317	238	248	287	2101
Blink Counts	18	3	32	29	14	10	14	120
Eye Center Position	4447	2911	3400	2981	2510	2917	3139	22305

In the 114222 annotated frames, it was possible to recognize either an open eye or a blink movement. From these frames, we annotated 22305 distinct eye center positions, detected 2101 saccade movements, and identified 120 blink movements. All these elements are summarized in Tab 1, where it is also possible to see this information data for each annotated user.

These annotations can be used during the training and testing of eye-tracking algorithms that utilize event-camera data to also detect various types of eye movements, such as blinks and saccades.

## 5 Conclusions

With this work, we presented an improved version of the original dataset proposed by Angelopoulos in [2], together with the pipeline used to generate improved annotations 200 Hz, providing what we believe could be a valuable resource for the scientific community.

The semi-automatic pipeline presented provides a method that reduces both the time required to annotate an event-camera dataset and the potential human

error that might arise from a completely manual annotation process. This method can also be applied to future datasets, giving more precise annotations that can help develop and test new algorithms for event-based eye-tracking.

Simultaneously, with the increasing interest in event-based technologies, particularly in low-power applications, we aim to expedite research in real-time eye-tracking using purely event-driven approaches.

**Acknowledgments.** This paper was carried out in the EssilorLuxottica Smart Eye-wear Lab, a Joint Research Center between EssilorLuxottica and Politecnico di Milano.

## References

1. Andrychowicz-Trojanowska, A.: Basic terminology of eye-tracking research. *Appli. Linguistics Papers* (**25/2**), 123–132 (2018)
2. Angelopoulos, A.N., Martel, J.N., Kohli, A.P., Conradt, J., Wetzstein, G.: Event based, near eye gaze tracking beyond 10,000 hz. *arXiv preprint [arXiv:2004.03577](https://arxiv.org/abs/2004.03577)* (2020)
3. Angelov, V., Petkov, E., Shipkovenski, G., Kalushkov, T.: Modern virtual reality headsets. In: *2020 International Congress on Human-computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–5. IEEE (2020)
4. Arena, F., Collotta, M., Pau, G., Termine, F.: An overview of augmented reality. *Computers* **11**(2), 28 (2022)
5. Bonazzi, P., Bian, S., Lippolis, G., Li, Y., Sheik, S., Magno, M.: Retina: low-power eye tracking with event camera and spiking hardware. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5684–5692 (2024)
6. Bulling, A., Gellersen, H.W.: Toward mobile eye-based human-computer interaction. *IEEE Pervasive Comput.* **9**, 8–12 (2010). <https://doi.org/10.1109/MPRV.2010.86>
7. Burger, G., Guna, J., Pogacnik, M.: Suitability of inexpensive eye-tracking device for user experience evaluations. *Sensors (Basel, Switzerland)* **18** (2018). <https://doi.org/10.3390/s18061822>
8. Crafa, D.M., Di Giacomo, S., Natali, D., Fiorini, C.E., Carminati, M.: Towards invisible eye tracking with lens-coupled lateral photodetectors. In: *Proceedings of the 2024 Symposium on Eye Tracking Research and Applications*, pp. 1–7 (2024)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
10. Fuhl, W., Santini, T., Kasneci, G., Kasneci, E.: Pupilnet: convolutional neural networks for robust pupil detection. *arXiv preprint [arXiv:1601.04902](https://arxiv.org/abs/1601.04902)* (2016)
11. Gallego, G., et al.: Event-based vision: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(1), 154–180 (2020)
12. Holmqvist, K., Andersson, R.: *Eye-tracking: a comprehensive guide to methods, paradigms and measures* (Nov 2017)
13. Kassner, M., Patera, W., Bulling, A.: Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction (April 2014). <http://arxiv.org/abs/1405.0006>

14. Khazali, M.F., Pomper, J.K., Thier, P.: Blink associated resetting eye movements (barms) are functionally complementary to microsaccades in correcting for fixation errors. *Sci. Rep.* **7**(1), 16823 (2017). <https://doi.org/10.1038/s41598-017-17229-w>
15. Liu, Z., Yang, Z., Gu, Y., Liu, H., Wang, P.: The effectiveness of eye tracking in the diagnosis of cognitive disorders: A systematic review and meta-analysis. *PLoS ONE* **16** (2021). <https://doi.org/10.1371/journal.pone.0254059>
16. Mentasti, S., Lattari, F., Santambrogio, R., Careddu, G., Matteucci, M.: Event-based eye tracking for smart eyewear. In: *Proceedings of the 2024 Symposium on Eye Tracking Research and Applications*, pp. 1–7 (2024)
17. Popa, L., Selejan, O., Scott, A., Mureşanu, D.F., Balea, M., Rafila, A.: Reading beyond the glance: eye tracking in neurosciences. *Neurol. Sci.* **36**(5), 683–688 (2015). <https://doi.org/10.1007/s10072-015-2076-6>
18. Santini, T., Fuhl, W., Kasneci, E.: Pure: Robust pupil detection for real-time pervasive eye tracking. *Comput. Vis. Image Underst.* **170**, 40–50 (2018)
19. Skaramagkas, V., et al.: Review of eye tracking metrics involved in emotional and cognitive processes. *IEEE Rev. Biomed. Eng.* **16**, 260–277 (2021). <https://doi.org/10.1109/RBME.2021.3066072>
20. Spil, T., Kijl, B., Romijnders, V.: The adoption and diffusion of wearables. In: Dwivedi, Y., Ayaburi, E., Boateng, R., Effah, J. (eds.) *TDIT 2019. IAICT*, vol. 558, pp. 31–47. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-20671-0\\_4](https://doi.org/10.1007/978-3-030-20671-0_4)
21. Wan, Z.H., Xiong, C.H., Chen, W.B., Zhang, H.Y.: Robust and accurate pupil detection for head-mounted eye tracking. *Comput. Electr. Eng.* **93**, 107193 (2021)
22. Wang, Z., et al.: Event-based eye tracking. *ais 2024 challenge survey* (2024). <https://arxiv.org/abs/2404.11770>
23. Winkler, S., Subramanian, R.: Overview of eye tracking datasets. In: *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 212–217. IEEE (2013)
24. Zafar, A., Martin Calderon, C., Yeboah, A.M., Dalton, K., Irving, E., Niechwiej-Szwedo, E.: Investigation of camera-free eye-tracking glasses compared to a video-based system. *Sensors* **23**(18), 7753 (2023)
25. Zhang, X., Liu, X., Yuan, S., Lin, S.: Eye tracking based control system for natural human-computer interaction. *Comput. Intel. Neurosci.* **2017** (2017). <https://doi.org/10.1155/2017/5739301>
26. Zhao, G., et al.: Ev-eye: rethinking high-frequency eye tracking through the lenses of event cameras. *Adv. Neural Inform. Process. Syst.* **36** (2024)
27. Zhu, Z., Ji, Q.: Novel eye gaze tracking techniques under natural head movement. *IEEE Trans. Biomed. Eng.* **54**(12), 2246–2260 (2007)