Event-based eye tracking for smart eyewear*

SIMONE MENTASTI, FRANCESCO LATTARI, RICCARDO SANTAMBROGIO, GIANMARIO CAREDDU, and MATTEO MATTEUCCI, Politecnico di Milano (DEIB), IT

This paper presents an innovative approach to gaze tracking in the context of smart eyewear, utilizing a fully event-based algorithm. Traditional gaze-tracking methods often rely on grayscale or infrared imaging, which can be computationally intensive and raise privacy concerns. Our research addresses these issues by developing an algorithm that exclusively uses data from event-based sensors, optimizing for the limited computational capabilities of smart eyewear. The system uses simple geometrical operations, enabling efficient real-time processing. Experimental results demonstrate the feasibility of this approach, offering a promising solution for gaze tracking in compact, computationally constrained devices. Despite certain limitations in accuracy due to optimization for efficiency, the research underscores the practicality of this approach for practical, privacy-conscious applications in smart eyewear technology.

CCS Concepts: • **Computing methodologies** → **Image processing**.

Additional Key Words and Phrases: Eye tracking, event-based, low-power, smart eyewear

ACM Reference Format:
Simone Mentasti, Francesco Lattari, Riccardo Santambrogio, Gianmario Careddu, and Matteo Matteucci. 2024. Event-based eye tracking for smart eyewear. In *2024 Symposium on Eye Tracking Research and Applications (ETRA '24), June 4–7, 2024, Glasgow, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3649902.3653471

1 INTRODUCTION

Gaze tracking refers to the technology and methodology used to monitor and record the direction in which a person is looking or the specific point in their visual field where their eyes are directed [Liu et al. 2022]. Gaze tracking can be conducted using various types of visual information obtained from the individual, including images of their eyes [Starostenko et al. 2019] or the entire face [Zhang et al. 2017]. Eye tracking specifically entails monitoring and recording eye movements, such as position and motion type. Gaze tracking can be considered one of the scopes within eye tracking. This technology's applications are vast and varied, encompassing areas like Human-Computer Interaction (HCI) [Majaranta and Bulling 2014], where it enhances device-user engagement; usability testing for evaluating digital interface navigation [Wang et al. 2019]; and Virtual Reality (VR) and Augmented Reality (AR) for immersive experiences [Shadiev and Li 2023]. Additionally, gaze tracking is instrumental in neuroscience for understanding brain-eye interactions and in marketing to analyze consumer attention [Bialkova et al. 2020].

The methods of gaze tracking are diverse, each suited to different applications and scenarios. 2D mapping-based methods, often employed in desktop eye trackers, correlate 2D eye features with screen coordinates, requiring personal calibration to understand the user's gaze [Morgante et al. 2012]. 3D model-based methods construct a more spatially accurate representation by considering the depth and distance of the eyes, commonly used in VR/AR settings [Tonsen et al. 2020]. Finally, appearance-based methods, which focus on the overall visual appearance of the user rather than

*This paper was carried out in the EssilorLuxottica Smart Eyewear Lab, a Joint Research Center between EssilorLuxottica and Politecnico di Milano

Manuscript submitted to ACM

specific eye features, are beneficial when detailed eye feature extraction is unfeasible, such as in distance tracking or head-mounted devices.

The intricacies of these methodologies also bring to light critical considerations in their development and deployment. Intrusive methods, while effective, might not be suitable for all contexts. The design of the systems, particularly those that are head-mounted, requires careful attention to ensure they do not disrupt natural behavior or pose safety risks [Cañigueral et al. 2018]. Ethical implications, especially for devices directly engaging with the eyes, are paramount. Thus, the evolution of gaze-tracking technology is a balancing act between technological innovation, user comfort, and ethical responsibility. The use of event sensors has recently been considered a safer approach from an ethical point of view since they do not acquire images, only sparse pixels, making them privacy-preserving and preferable in these scenarios [Bolten et al. 2019].

In this work, we focus on the specific scenario of smart eyewear, where gaze information has to be collected to provide functionalities and context information to the smart glasses. In this scenario, three crucial aspects must be considered. The first one is the size of the tracking device can have. Since the head-mounted camera for eye tracking has to be integrated into a daily-use product, it cannot drastically change the shape of traditional glasses. Secondly, camera recordings of the user's eye might be considered sensitive information, and their usage could be limited in the future. For this reason, event-based approaches, which, instead of providing a complete image of the eye, return only binary values of the changing pixels, are generally considered a more privacy-preserving and safe option [Gallego et al. 2020]. Finally, traditional eye-tracking algorithms can rely on post-processing and heavy computational power to detect and track the eye effectively. Since our approach is to integrate eye tracking technology into smart eyewear, the available computational power is severely limited, while real-time processing becomes mandatory. For this reason, an optimized algorithm that can run in real-time, leveraging the simplified representation provided by the event cameras, is a promising direction of work.

The central contributions of our research can be distilled into two primary aspects:

- Development of a Fully Event-Based Gaze Tracking Algorithm: We introduce an innovative gaze-tracking algorithm that exclusively relies on data acquired from event-based sensors. This approach differentiates from previous solutions since it omits reconstructed grey-scale images or any infrared data, a common element in traditional gaze-tracking systems.
- Design of an Optimized Gaze Detection and Tracking Pipeline: We have developed a streamlined gaze detection and tracking pipeline. This pipeline is characterized by its reliance solely on simple geometrical operations. The efficiency and simplicity of these operations make the system particularly well-suited for deployment on micro-controllers and within smart eyewear systems. This optimization enhances computational efficiency and ensures compatibility with the limited processing capabilities inherent to these platforms.

2 RELATED WORKS

Eye tracking has been an active research field for many years, employing various setups and technologies, with traditional methods largely using RGB or infrared cameras to monitor the subject's face or eyes. In particular, two different approaches to eye tracking have emerged through the years. The first one is 2D Mapping-based methods. They rely on the correlation of n-dimensional features extracted from a 2D representation of the eye, such as RGB images, with the positional information of the eyes on the 2D coordinates of a screen or image [Morgante et al. 2012]. This approach is commonly employed in desktop eye trackers and integrated laptop/webcam solutions. Gaze-tracking

algorithms based on 2D mapping are generally easier to implement and typically require personal calibration before becoming operational. The personal calibration procedure is essential for learning the relationship between the observed 2D features of the eye and the user's point-of-regard (POR). The second approach is 3D Model-based gaze tracking. They involve the creation of a three-dimensional model of the eye and its movements. These methods account for the depth or distance of the eyes from the tracking system, offering a more spatially accurate representation of gaze direction. Commonly employed in VR and AR, this category of tracking algorithms is more intricate to implement, often requiring stricter setups and controlled lighting conditions. However, it generally excels in handling head movements and can achieve accuracies in 3D gaze line-of-sight (LOS) estimation up to 0.5°, as demonstrated in studies [Bakker et al. 2013; Luo et al. 2020].

Most recent trends in gaze tracking research focus on the problem of gaze tracking in the wild and not in a light-controlled environment. This is indeed one of the most relevant challenges researchers are currently facing. Many of the previous works achieve high accuracy in indoor scenarios, where the lightning does not present significant changes and the eye tracking system can be properly calibrated. Instead, outdoor scenarios are affected by rapid and significant changes in lightning. This affects both the acquisition system (i.e., the camera exposure and the Infrared illuminator) and the eye, which can adapt its shape due to excessive light. In this context, deep learning is widely adopted, and recent models can achieve a sub-0.5° gaze error [Chaudhary et al. 2019], [Yiu et al. 2019]. These algorithms are limited in practical cases by the operational frequency of around 5 Hz, making them unsuitable in real-time scenarios.

With the growing interest in eye-tracking technologies in recent years, many datasets have been published to develop new gaze-tracking algorithms. In particular, nowadays, it is possible to account for a large number of public datasets of real and synthetic eye data, like SynthesEyes [Wood et al. 2015], GazeCapture [Krafka et al. 2016], UT Multi-view [Sugano et al. 2014], UnityEyes [Wood et al. 2016]. While infrared-based eye tracking is a well-established technology, many datasets are available, event-based sensors are relatively new, and the number of datasets is extremely limited. In particular, the only one available is the dataset presented by Angelopoulos et al. [Angelopoulos et al. 2022]. In this work, the authors exploit event cameras as sensors for gaze tracking, complementing greyscale data provided by the DAVIS sensor and specifically focusing on 2D mapping-based methods.

Previous research in 2D mapping-based approaches has explored diverse eye features. For instance, some studies have employed the vector formed by the pupil/iris center and eye corner point [Cai et al. 2016], while others have delved into more intricate features such as the state of the eye (open/closed), iris, and eyelid identification [li Tian et al. 2000]. In [Feng et al. 2022], an event camera can drive an auto region of interest (ROI) prediction algorithm for eye tracking using software-emulated events. Due to the nature of event cameras, glints can not be employed in the standard way since a fixed light on the frame will not produce any events. In [Stoffregen et al. 2022], authors introduce a tracking pipeline exploiting blinking glints that the event sensor can, in this way, detect. The work from [Angelopoulos et al. 2022] employed a DAVIS sensor able to capture both events and gray scale frames. Their tracking algorithm uses the frames to localize and estimate the pupil center by fitting an ellipse on the pupil border. Then, the ellipse is propagated in between frames using the incoming events.

Our work extends from state-of-the-art and proposes a purely event-based approach, designed to be extremely lightweight and easily integrated into smart eyewear. This means that differently from the solution proposed in [Angelopoulos et al. 2022], all the components of pupil detection and tracking are performed using only data from the event camera. Since the dataset from [Angelopoulos et al. 2022] is the only one that provides real event recordings, all tests and pipeline validation are performed using this dataset.

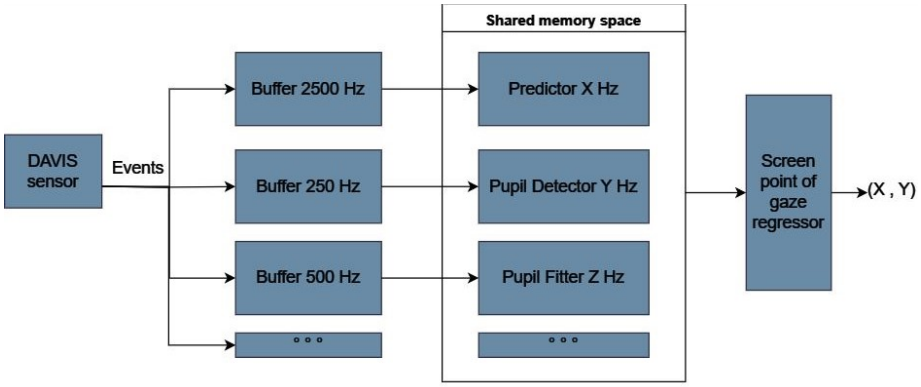


Fig. 1. Architecture of the event-based eye tracking pipeline. Events from the DAVIS sensor are grouped into buffers of different lengths. Each buffer is used for a specific task in the pipeline based on the number of points required. Finally, the gaze position is projected to a screen using a previously computed calibration.

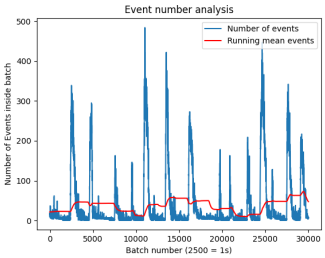
3 EVENT-BASED EYE TRACKING

In this section, we present the complete pipeline for gaze tracking using event-based data. Despite the non-traditional input, this approach maintains the core principle of 2D mapping-based gaze tracking, converting the 2D center of pupil imaging ellipse into corresponding screen coordinates. Our system architecture, depicted in Fig. 1, is composed of three interconnected components: the Predictor, the Detector, and the Fitter, all functioning within a unified memory space to optimize the running frequency of the algorithm.

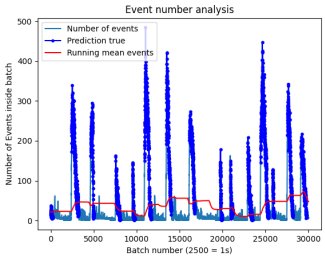
The Predictor is responsible for continuous monitoring of the event stream from the sensor, identifying significant eye movements by tracking event bursts. The Detector, once activated, processes these events at 250 Hz, projecting them onto an RGB-like frame to determine the pupil's center and its region of interest (ROI). The Pupil Fitter is engaged in fitting an ellipse to the pupil within the identified ROI. This component operates at a higher frequency, enabling continuous updates to the ROI and the ellipse center without reconstructing the artificial RGB frame. The interplay between these components, particularly the dynamic activation and deactivation mechanisms, exemplifies an efficient, real-time gaze tracking system optimized for event-based data scenarios.

3.1 Predictor

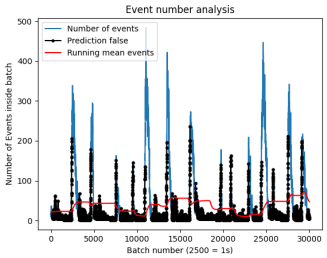
The events from the DAVIS sensor are captured asynchronously with microsecond temporal resolution. Grouped into batches at a fixed 2.5 KHz frequency, each batch represents 0.4 ms of data. Since the eye remains stationary most of the time, processing every batch would be inefficient. To optimize resource use, we employ a movement prediction strategy. This involves tracking the event count in each batch and calculating a running mean over the last 'k' batches. The system processes a batch only if it consistently exceeds the running mean for 'n' consecutive batches. Conversely, it predicts eye movement cessation when the event count falls below the running mean for 'm' consecutive batches. The parameters 'k', 'm', and 'n' are adjustable, affecting the system's sensitivity to movement. For initial settings, considering typical saccade durations in reading and scene perception, we chose 'n' = 25 (10 ms delay for activation) and 'm' = 50 (20 ms delay for deactivation).



(a) Number of events @2500 Hz and running mean



(b) Movement predictions in blue



(c) No movement predictions in black

Fig. 2. Twelve seconds of data extracted from a user recording. The first picture represents the number of events through time and the running mean of the number of events. In the second picture, the point in time where the predictor identifies a movement is presented in blue, while in the third picture, the points in time where the predictor predicts no movement are highlighted in black.

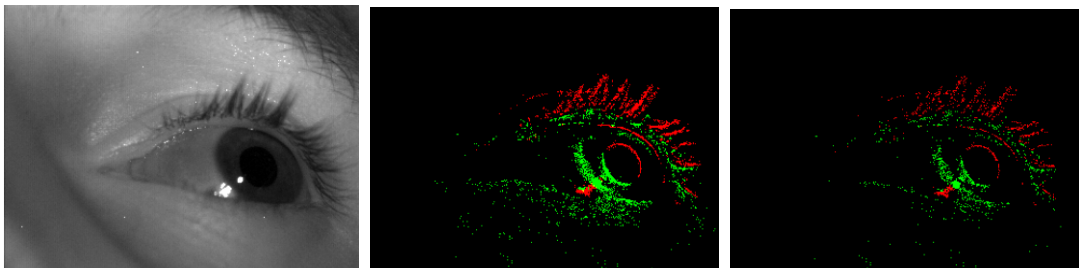


Fig. 3. Event batch rendered as RGB image (second and third images), with the negative polarity pixel in red and the positive in green. The event frames have been acquired with sampling frequencies of 250 Hz and 500 Hz from left to right. The higher the sampling frequency, the fewer events we group in the same batch. The first image is the greyscale returned by the DAVIS sensor.

In Fig. 2, it can be seen how visible the saccadic motion is based on the number of integrated events. The number of events during a saccade is orders of magnitude higher than the running mean over one second. Furthermore, the narrow peaks prevent the running mean from going too high. From 2b and 2c, we can graphically see the sensitivity of the predictor changing from no-movement to movement prediction and vice-versa.

3.2 Pupil detector

The pupil detector, operating at 250 Hz, translates the asynchronous event stream from the DAVIS sensor into event frames. These frames, resembling standard RGB frames, aggregate positive and negative events into separate grids, encoded with red and green colors respectively. This is because events from the camera carry not only the information of a change in brightness of the pixel but also the direction of change, i.e., from dark to bright or the opposite, which is coded into the polarity field. The camera acts as an edge extractor, with the pupil's movement generating boundary pixel changes, creating a distinct pattern on each polarity image. This pattern, mirroring each other across the pupil's center, varies with different eye movements. The sampling frequency crucially affects the edge visibility in these frames, with higher frequencies yielding thinner edges. Slow movements, however, might be less discernible due to event dispersion. To manage this, events are stored in a queue buffer, focusing on arrival order rather than absolute time, ensuring the

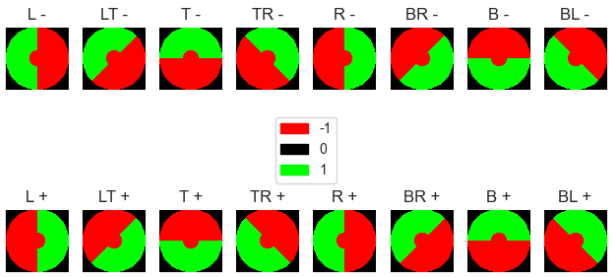


Fig. 4. The 16 directional templates employed for pupil detection. The rows indicate the polarity. The templates are paired columns-wise; every row corresponds to a single direction

detectability of pupil edges during movements. Sample images acquired at different frequencies are shown in Fig. 3. It is important to notice how the eye always has the same pattern, green on one side and red on the other, which indicates the pixel's polarity change due to the movement. Indeed, the event camera functions as an edge extractor: when the pupil moves, the boundary pixels in the direction of its movement switch from a higher intensity of the iris to a darker intensity of the pupil. Conversely, in the pupil region opposite the movement direction, the boundary pixels switch from the pupil's lower intensity to the iris' brighter intensity.

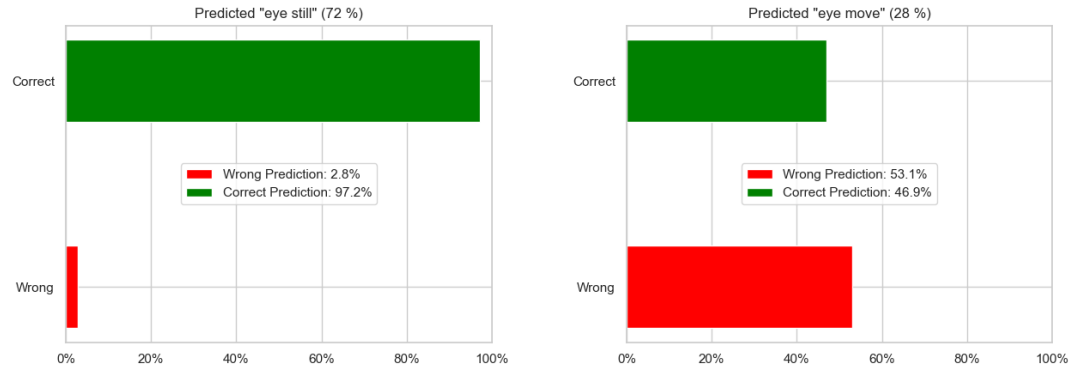
Exploiting this pattern, pupil detection employs a 2D convolution with eight templates, shown in Fig. 4. Each aligns with a potential movement direction. These templates, designed to match two horizontal, two vertical, and four diagonal movements, are applied to the polarity images. Convolution results from both polarities are multiplied to create heatmaps, identifying movement direction. The highest-scoring template pair indicates the motion direction, and a bounding box centered at the maximum point of the heatmap pinpoints the pupil location.

3.3 Pupil Fitter

The initial pupil center estimation obtained from the detection step requires refinement for accuracy. This is achieved by fitting an ellipse to the events within the ROI, identified by the pupil detector. Unlike the RGB case where pupil boundaries are extracted, the event-based approach already has these boundaries defined by the sensor. However, not all events in this region correspond to the pupil boundary. Two distinct fitting strategies are employed to increase the algorithm performance, tailored exclusively for event data. The behavior of the pupil fitter algorithm varies depending on the availability of prior pupil ellipse estimation.

In cases with only the center estimation from the pupil detector, the Random Sample Consensus (RANSAC) algorithm is used [Fischler and Bolles 1981]. RANSAC iteratively selects a random subset of events within the ROI to fit an ellipse, evaluating the fit quality against all points in the ROI. This continues for a set number of iterations, with the best-scoring model being chosen. While RANSAC is robust against outliers, it is computationally intensive, and increasing the number of iterations enhances the probability of finding an optimal model.

The second fitting approach relies on an existing ellipse model, working best when the new model closely aligns with it over time. This method requires events for the new fit near the edge of the pre-identified ellipse. Applied post-RANSAC or independently, it continuously updates the center. In particular, only the events within the ROI are considered from the continuous stream of new events from the sensor. These events are processed in batches of 100, and for each batch, events close to the boundaries of the most recent ellipse are identified. The ellipse model is then



(a) (b)

Fig. 5. Predictor evaluation on the test data. In the first image, the scenario where the predictor predicts a still eye is analyzed. In this case, most of the predictions are correct. The second image presents the scenario where the predictor predicts a movement. In this case predictions are less accurate but, due to the nature of the task false positive are acceptable and preferred to missed movement detections.

updated using these identified events. The update employs a weighted average with the existing ellipse parameters to ensure a smooth and homogeneous transition.

The Pupil Detector and RANSAC fitting are used for initial estimation, achieving high tracking accuracy at specific eye movement phases. The second fitting approach is then triggered and tracks the pupil movement processing events up to 500 Hz.

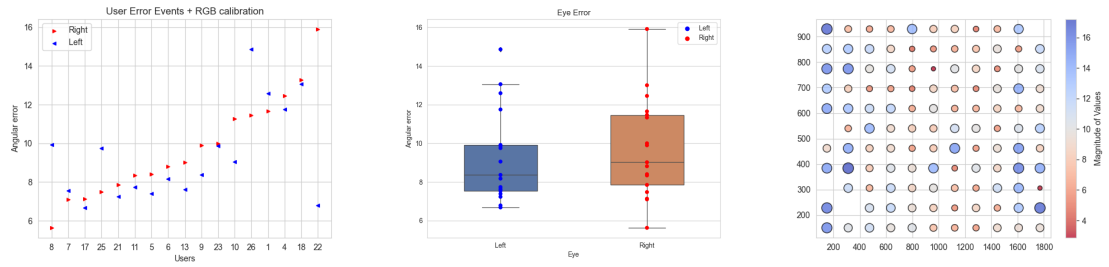
4 EXPERIMENTAL RESULTS

To validate the efficacy of the proposed method, our initial step involves an assessment of the predictor's accuracy, as it is the fundamental element that initiates the entire detection process. Subsequently, we proceed to evaluate the algorithm's performance in its entirety. To perform these tests, we employed the only publicly available event-based gaze dataset from Angelopoulos et al. [Angelopoulos et al. 2022]. A critical aspect to note is that the used dataset does not directly provide the ground truth in terms of the eye's position within the camera frame. Rather, it provides data regarding the focal point of the subject's gaze on a monitor.

It was central to implement a calibration procedure for the system to evaluate the algorithm's precision. In particular, we computed a transformation matrix that maps the observed eye position to its corresponding location on the plane of interest. This transformation is required to validate our algorithm's accuracy, as it enables us to correlate the observed data with the actual point of gaze, thereby providing a more reliable and accurate measure of the algorithm's performance.

4.1 Predictor evaluation

The primary objective of the Predictor is to accurately identify the start point of the ocular movement, which subsequently activates the detection and propagation blocks of the pipeline. The effectiveness of this algorithm is quantitatively evaluated using the F1-score metric, with an average score of 0.6, as depicted in Fig. **??**. However, this F1-score does not



(a) (b) (c)

Fig. 6. Average degree error per user in the left plot users are sorted based on the error in the right eye, the same results are presented on the left but grouping per eye.

fully capture the specific behavioral characteristics of interest. This discrepancy arises primarily due to the imbalance in the class distribution. Moreover, the crucial classification task involves discerning a 'still' state of movement. Conversely, misclassification in the opposite scenario is occasionally deemed tolerable.

Upon examining the distribution of class predictions, it is observed that the algorithm, on average, categorizes 72% of the events as non-movements. Focused analysis of these predictions, as illustrated in Fig. 5a, reveals that a predominant majority of these non-movement classifications are indeed accurate. Nevertheless, a small fraction of potentially significant events is overlooked by the predictor.

Further analysis of the algorithm's performance in movement prediction, as shown in Fig. 5b, indicates less accurate results. Of the average 28% of events predicted as movement, approximately half could potentially trigger the start of the detection pipeline even in the absence of ocular motion. Combining these observations, it can be inferred that when focusing solely on processing relevant events, approximately 70% of the total events are correctly disregarded, since the eye is still, 15% are erroneously processed, 2% are mistakenly ignored, and 13% are accurately processed. This comprehensive analysis highlights how only a minimal part of the events are lost, and the predictor effectively detects the relevant events for the task, with a partial overload on the system in the scenarios where 'still' scenarios are wrongly labeled.

4.2 Pipeline evaluation

The pupil centers detected by our pipeline are unsuitable for calibration purposes. This is primarily due to the high sensitivity of the 2D mapping process, where even minor errors in center estimation, like a deviation of a few pixels in determining the pupil center, can significantly impair the mapping quality.

To avoid this limitation, we employed a pre-calibrated Point of Regard (POR) Regressor derived from the frame tracker, i.e., a regressor based on grey-scale images. The assessment methodology employed is akin to the approach described in [Angelopoulos et al. 2022]. This gives us the calibration matrix to remap the pupil position to the 2D plane.

The resultant average angular error, calculated across the training users and considering the entire Field of View (FoV) through frame calibration, stands at 9.31°, as shown in Fig. 6. Notably, this value surpasses the average error expected from conventional image-based eye-tracking techniques. However, several pertinent factors warrant consideration.

Primarily, this angular error comprises not only the intrinsic error of the image-based detector but also the error introduced during the calibration process to the screen. A detailed analysis of the error distribution on the projection

screen reveals that certain points exhibit a markedly higher error, as shown in Fig. 6c, likely stemming from calibration inaccuracies, thereby leading to potential prediction errors.

Furthermore, the algorithm has been designed and optimized for efficient execution on smart eyewear. This necessitated several trade-offs, primarily prioritizing computational efficiency over accuracy. Indeed, the computationally intensive RANSAC algorithm is deployed exclusively during the initialization phase. Consequently, if the tracker loses track of the pupil and awaits reinitialization upon the cessation of eye movement, the accuracy of the measurements significantly diminishes. A potential mitigation strategy could involve increasing the RANSAC algorithm's frequency to promptly identify and rectify such scenarios, thereby reducing the angular error.

Despite these challenges, the results unequivocally demonstrate the viability of a purely event-based approach for embedded gaze-tracking in smart eyewear. This showcases the potential for implementing efficient, effective, gaze-tracking solutions in compact and computationally constrained devices.

5 CONCLUSIONS

This work demonstrates the feasibility and efficacy of a fully event-based gaze-tracking algorithm tailored for smart eyewear. The key contributions of this research include the development of an innovative gaze-tracking algorithm that bypasses the need for grayscale or infrared imaging and the design of an optimized detection and tracking pipeline that is both computationally efficient and well-suited for integration into smart eyewear.

The experimental evaluations show that, while the system prioritizes computational efficiency, it maintains a reasonable level of accuracy, making it a viable solution for real-world applications. This balance is crucial in the context of smart eyewear and embedded devices, where processing power and battery life are limited. Limitations in accuracy, primarily due to the computational constraints, open to future developments that could focus on enhancing this aspect, increasing the frequency of the RANSAC algorithm for more accurate pupil tracking.

Overall, this research highlights the advantages of a fully event-driven approach. It opens up new possibilities for practical, privacy-conscious applications, offering a starting point for future advancements in this rapidly evolving field.

REFERENCES

Anastasios N. Angelopoulos, Julien N. P. Martel, Amit P. S. Kohli, Jorg Conradt, and Gordon Wetzstein. 2022. Event Based, Near Eye Gaze Tracking Beyond 10,000Hz. arXiv:2004.03577 [cs.CV]

Nicole M. Bakker, Boris A. J. Lenseigne, Sander Schutte, Elsbeth B. M. Geukers, Pieter P. Jonker, Frans C. T. Van Der Helm, and Huibert J. Simonsz. 2013. Accurate gaze direction measurements with free head movement for strabismus angle estimation. *IEEE Transactions on Biomedical Engineering* 60, 11 (2013), 3028 – 3035. https://doi.org/10.1109/TBME.2013.2246161 Cited by: 23.

Svetlana Bialkova, Klaus G Grunert, and Hans van Trijp. 2020. From desktop to supermarket shelf: Eye-tracking exploration on consumer attention and choice. *Food Quality and Preference* 81 (2020), 103839.

Tobias Bolten, Regina Pohle-Fröhlich, and Klaus D Tönnies. 2019. Application of hierarchical clustering for object tracking with a dynamic vision sensor. In *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part V 19*. Springer, 164–176.

Haibin Cai, Hui Yu, Xiaolong Zhou, and Honghai Liu. 2016. Robust gaze estimation via normalized iris center-eye corner vector. In *International conference on intelligent robotics and applications*. Springer, 300–309.

Roser Cañigueral, Antonia Hamilton, and Jamie A Ward. 2018. Don't Look at Me, I'm Wearing an Eyetracker!. In *Proceedings of the 2018 ACM international joint conference and 2018 international symposium on pervasive and ubiquitous computing and wearable computers*. 994–998.

Aayush K. Chaudhary, Rakshit Sunil Kothari, Manoj Acharya, Shusil Dangi, Nitinraj Nair, Reynold Bailey, Christopher Kanan, Gabriel J. Diaz, and Jeff B. Pelz. 2019. RITnet: Real-time Semantic Segmentation of the Eye for Gaze Tracking. *CoRR* abs/1910.00694 (2019). arXiv:1910.00694 http://arxiv.org/abs/1910.00694

Yu Feng, Nathan Goulding-Hotta, Asif Khan, Hans Reyserhove, and Yuhao Zhu. 2022. Real-Time Gaze Tracking with Event-Driven Eye Segmentation. *CoRR* abs/2201.07367 (2022). arXiv:2201.07367 https://arxiv.org/abs/2201.07367

Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. 2020. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 1 (2020), 154–180.

Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra M. Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye Tracking for Everyone. *CoRR* abs/1606.05814 (2016). arXiv:1606.05814 http://arxiv.org/abs/1606.05814

Ying li Tian, T. Kanade, and J.F. Cohn. 2000. Dual-state parametric eye tracking. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. 110–115. https://doi.org/10.1109/AFGR.2000.840620

Jiahui Liu, Jiannan Chi, Huijie Yang, and Xucheng Yin. 2022. In the eye of the beholder: A survey of gaze tracking techniques. *Pattern Recognition* (2022), 108944.

Kaiqing Luo, Xuan Jia, Hua Xiao, Dongmei Liu, Li Peng, Jian Qiu, and Peng Han. 2020. A New Gaze Estimation Method Based on Homography Transformation Derived from Geometric Relationship. *Applied Sciences* 10, 24 (2020). https://doi.org/10.3390/app10249079

Päivi Majaranta and Andreas Bulling. 2014. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*. Springer, 39–65.

James D Morgante, Rahman Zolfaghari, and Scott P Johnson. 2012. A critical test of temporal and spatial accuracy of the Tobii T60XL eye tracker. *Infancy* 17, 1 (2012), 9–32.

Rustam Shadiev and Dandan Li. 2023. A review study on eye-tracking technology usage in immersive virtual reality learning environments. *Computers & Education* 196 (2023), 104681.

Aleksey Starostenko, Filipp Kozin, and Roman Gorbachev. 2019. Real-Time Algorithms for Head Mounted Gaze Tracker. 86–863. https://doi.org/10.1109/ICAIAI48757.2019.00025

Timo Stoffregen, Hossein Daraei, Clare Robinson, and Alexander Fix. 2022. Event-Based Kilohertz Eye Tracking using Coded Differential Lighting. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 3937–3945. https://doi.org/10.1109/WACV51458.2022.00399

Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. 2014. Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation. *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), 1821–1828. https://api.semanticscholar.org/CorpusID:15500329

Marc Tonsen, Chris Kay Baumann, and Kai Dierkes. 2020. A high-level description and performance evaluation of pupil invisible. *arXiv preprint arXiv:2009.00508* (2020).

Jiahui Wang, Pavlo Antonenko, Mehmet Celepkolu, Yerika Jimenez, Ethan Fieldman, and Ashley Fieldman. 2019. Exploring relationships between eye tracking and traditional usability testing data. *International Journal of Human–Computer Interaction* 35, 6 (2019), 483–494.

Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. 2016. Learning an appearance-based gaze estimator from one million synthesised images. In *Proc. ACM International Symposium on Eye Tracking Research and Applications (ETRA)*. 131–138. https://doi.org/10.1145/2857491.2857492

Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. 2015. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. *CoRR* abs/1505.05916 (2015). arXiv:1505.05916 http://arxiv.org/abs/1505.05916

Y.H. Yiu, M. Aboulatta, T. Raiser, L. Ophey, V.L. Flanagin, P. Zu Eulenburg, and S.A. Ahmadi. 2019. DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of Neuroscience Methods* 324 (1 Aug 2019), 108307. https://doi.org/10.1016/j.jneumeth.2019.05.016 arXiv:2019-06-06

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence* 41, 1 (2017), 162–175.