Towards Low-power, High-frequency Gaze Direction Tracking with an Event-camera

Yvonne Vullers[1], Luna Gava[2], Arren Glover[2], and Chiara Bartolozzi[2]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
y.a.g.vullers@student.tue.nl
[2] Istituto Italiano di Tecnologia, Genova, Italy first.last@iit.it

Abstract. Eye tracking is used in smart eye-wear to understand the gaze direction of the users' eyes and must be accurate and with low-latency for pleasant user experience, and low-power for extended use time in a mobile device. However, visual eye tracking using traditional cameras is not ideal given the typical dynamics of eye-motion. The fixed sample rate (fps) is either under-sampling (lower accuracy) during fast eye motion during a saccade or oversampling (wasted computation) during gaze fixations when the eye-ball pose is stationary. An alternative technology, event-cameras, respond only to lighting change and therefore the visual signal adapts to the changing eye dynamics. High-frequency tracking can be achieved during saccades, while redundant processing is avoided during stationary gaze. In addition, due to small data packets, low-latency is also achieved. We present an event-camera-only eye-tracking proof-of-concept using a 5-DoF deformation model and demonstrate strong potential for accurate, low-latency (400 Hz) eye-tracking on a publicly available dataset; demonstrating the first gaze direction (pitch and yaw) estimation with an event-camera. We highlight the challenges still required to be solved to bring event camera technology to commercial viability for eye-tracking.[3]

Keywords: event-camera, eye-tracking, low-latency

1 Introduction

Visual gaze tracking involves estimating the direction at which a person is looking using a camera. Gaze is a cue towards understanding emotional state [23] and attention [26], as well as a requirement for applications such as augmented and virtual reality (AR/VR) [15] and other technology interfaces [16]. Eye and gaze tracking can also be useful in medical applications, such as eye laser-surgery [22] and psychological research [20].

[3] Vullers, Y., Gava, L., Glover, A., and Bartolozzi, C. "Towards Low-power, High-Frequency Gaze Direction Tracking with an Event-camera". The European Conference on Computer Vision workshop on Eyes of the Future: Integrating Computer Vision in Smart Eyewear (ICVSE), Milan Italy, 2024

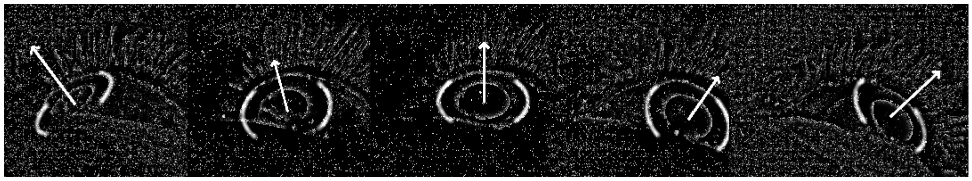


Fig. 1: Visualisation of the gaze-tracking algorithm showing the event-camera processed data, the tracked deformation model, and the extracted gaze direction, as the eye moves from left to right. The final algorithm outputs at over 400 Hz and can track in real-time during saccadic motion.

Gaze-tracking technologies typically need to be worn nearby the eyes, and small, cheap cameras offer a good sensor solution; therefore visual methods for tracking gaze are necessary. However, traditional camera technologies have drawbacks due to limited frame-rate and constant processing requirements. For the task of gaze tracking it is necessary to have a high frame-rate, to track during fast-moving saccadic motion and a high resolution to increase accuracy. The result is a high volume of data that needs constant processing, requiring a constant draw of power. Wearable devices only have limited power available [3].

An alternative option is to use event-cameras for gaze tracking. Event cameras [9] monitor changes in brightness at the sensor level and asynchronously output only those pixels, i.e., *events*, that experience rapid change; therefore detecting moving edges in the scene. Both data transmission bandwidth and subsequent processing are reduced as entire frames are no longer generated by the camera. Low-latency (3.5 ms [11]) is achieved as data is transmitted as events are generated, without waiting for the entire sensor array to be sampled. Tracking can be achieved with lightweight algorithms as the pixels directly indicate the change and can avoid motion blur, a typical problem with fast-moving targets, due to high temporal resolution (sub-millisecond).

Direct adaption of traditional gaze-tracking algorithms is not possible due to the vast difference in data format and only a few event-based methods have been developed for eye-tracking since cameras have only recently become commercially available. Recently, an eye-tracking challenge [31] has been released, triggering many people to use event-based data to detect the iris position. Both convolutional neural networks (CNN) and model-based methods have shown the iris can be detected in event-data [2, 6, 17, 24], however, these methods have not looked at gaze direction. Instead, [3] finds gaze direction using a hybrid frames-and-events approach, but which re-introduces some of the drawbacks of traditional camera technology.

In this paper, we propose a fully event-based gaze tracking algorithm that uses an iris deformation model, the output of which is visualised in Fig. 1. Our approach is novel in that it combines the following contributions:

- It uses a fully event-based and model-based approach to avoid latency-inducing, power-hungry, CNN-based algorithms.

– It has a high output frequency, giving gaze feedback also during short, fast saccades.
– It estimates the gaze direction, not only the iris position.
– It uses a light-weight approach that takes advantage of the high-temporal resolution of the event-data.

Limited availability of event-camera datasets results in the assumption that the centre of the eye is in a fixed position in the image and does not change and that tracking accuracy can only be measured by pupil centre position. To facilitate comparison, we provide curated event-data, algorithm code and results files[4].

2 Related Work

Research in eye tracking focuses on different parts of the eye including blinks [14, 21, 24], glint [25, 27, 32], and the pupil or iris [1–4, 6, 8, 17, 19, 24, 28, 29]. These features enable either the pupil centre position or the gaze direction, to be tracked. We focus on gaze direction estimation through tracking of the iris, as the iris maintains a consistent size, unlike the pupil.

Using traditional cameras, data-driven methods [1, 8, 19], involve estimating the position and gaze directly from images, typically by using CNN or other deep learning approaches. Model-based methods attempt to fit visual features to a model of the eye and find the iris position and the gaze direction from the state of the model. Current model state can be found by using visual template matching [4, 28] or landmark fitting with optimisation [29]. Both approaches can achieve high-accuracy but are limited to the frame-rate of the sensor.

Event-based eye-tracking is still in its infancy. Models such as YOLO [24], a convolutional Long Short-Term Memory network [6], or a convolutional encoder-decoder structure (U-Net) [17] have been proposed for position estimation. Instead, [2] implements a simpler approach with a convolutional sliding window using a single filter to find the position. Only [18] exists which estimates the gaze direction for extended reality (XR) using only event-based data, by training a CNN system on an available dataset [3]. However, the code is not made publicly available and a quantitative measure of the inference latency suggests the pipeline struggling to work in real-time for the XR headset. [3, 33] are able to estimate gaze direction, but require integration of traditional camera images to do so. [18] focused on estimating gaze direction using solely event-based data, training a CNN system on an available dataset [3]. However, the code is not publicly available, and the quantitative assessment of inference latency and power consumption indicates that the pipeline could operate in real-time on low-power XR headsets for different computational platforms, but lacks a live demonstration.

Other work [5] presented an energy-efficient, low-latency and accurate neuromorphic approach for eye-tracking, leveraging the strengths of both neuromorphic sensors and processors using a spiking neural network. Results provided a

[4] `https://github.com/event-driven-robotics/workbook_yvonne-vullers.git`

quantitative evaluation of latency, power and energy consumption, fundamental for low-power wearable glasses. Code and data are available, but for a fair comparison in terms of latency, we would have required the deployment on a neuromorphic processor, such as Speck. Instead, our algorithm targets simpler computing hardware.

We propose to combine a model-based approach able to estimate gaze direction [4], with inspiration from state-of-the-art event-driven tracking algorithms [10] to enable a fully event-driven, high-frequency tracking algorithm. The tracking algorithm takes advantage of the event-camera data in such a way as to avoid problems with motion-blur and can be made lightweight as event data does not "jump" multiple pixels between image frames.

3 Methods

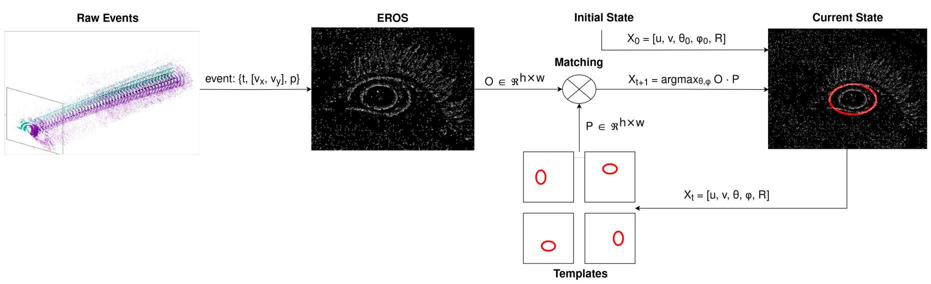


Fig. 2: The full pipeline for event-based gaze direction tracking with an event-camera. Raw events are processed into the speed invariant *EROS* surface. The deformation model projects the current state onto the image plane also at a series of small state variations, from which the best match EROS match is used to update the state. The algorithm is enabled by event-camera technology, which observes the fast saccadic motion without motion-blur or jumps in position.

The eye-tracking algorithm aims to determine the gaze direction using an eye state, X, at time t defined as:

$$X_t = [u, v, \theta, \phi, R] \tag{1}$$

where, u and v represent the eyeball centre in pixels, θ and ϕ are yaw and pitch orientations (zeroed when the eye is looking straight), and R is the eyeball radius in pixels.

To compute the state, the algorithm is comprised of three main components (see Fig. 2):

- Processing event data to generate a velocity-invariant, persistent observation O, following the Exponential Reduced Ordinal Surface (EROS) [12] method.

- Defining the eye model to project the iris shape onto the 2D image plane for any possible eye state.
- Tracking the eye to estimate the next state X_{t+1} from the current state X_t and observation O

The algorithm requires the initial state of the eye for each experiment that would ideally be provided by a detection algorithm, which is outside the scope of this paper, and to be developed as future work.

The tracking algorithm is enabled by the event-camera technology in that it searches for only very small incremental changes in the eye state in each update. As the event-camera produces data asynchronously, pixel-by-pixel, as the eye saccades, such an algorithm is possible and can result in high-frequency, on-line tracking.

Data The event-camera generates events as tuples: $event : t, [v_x, v_y], p$, consisting of the timestamp t, pixel position $[v_x, v_y]$, and event polarity p indicating brightness change direction.

The EROS [12] is used to integrate individual events into a speed-invariant spatial representation capable of being used for tracking visual patterns. The EROS generates a 2D array, in a similar fashion to a traditional image, in which each pixel represents the likelihood there is an intensity gradient at the observed point in the world. It is updated asynchronously and allows a high event-throughput with a high-frequency (and can supersede limitations caused by traditional camera frame-rates) back-end tracking algorithm.

In comparison, a simple fixed temporal window can result in motion-blur if the saccade is too fast, or unobservable motion if smooth pursuit is too slow. The inconsistent appearance in the spatial domain of the image plane is detrimental to processing algorithms.

Eye-model Rotating the eyeball in the world space causes the iris to deform from a circle to an ellipse on the image plane. Therefore, accurate representation of the iris shape is crucial when tracking the eyeball state X_t. To achieve this, we defined a representation of a circle (iris) on a rotating sphere (eyeball), following the model in [4], however, we considered yaw and pitch rotations to directly extract gaze direction, instead of the yaw and roll proposed previously.

Assuming roll is negligible, to distort any pixel (x, y) defined on a unit sphere rotated of ϕ and θ to an image plane centered at (u, v), and with scaling of R:

$$\begin{bmatrix} x \\ y \end{bmatrix} = R \cdot \begin{bmatrix} \hat{x} \cdot cos(\phi) + \sqrt{1-\tau^2} \cdot sin(\phi) \\ (\hat{x} \cdot sin(\phi) - \sqrt{1-\tau^2} \cdot cos(\phi)) \cdot sin(\theta) + \hat{y} \cdot cos(\theta) \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix} \tag{2}$$

where, $[u, v, \theta, \phi, R]$ are the eye model state X_t and the values of $\hat{x}$ and $\hat{y}$ must be defined as the iris locations on the unit sphere, and therefore:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \tau \begin{bmatrix} cos(\alpha) \\ sin(\alpha) \end{bmatrix} \tag{3}$$

for $\alpha = [-\pi, \pi]$ and $\tau = 0.5$ based on anthropometric ratio of the iris to the eyeball radius [30]. In detail, Appendix 6 derives the eye-model equation 2.

Tracking The object state is tracked by matching the projections of potential states P to the observation O. Potential states are computed as small variations of the current state, exploiting the event-camera's nature with no significant observation jumps between tracking steps. For the iris and the eyeball, potential states involve slight rotations ($\theta \pm \Delta\theta$ or $\phi \pm \Delta\phi$). The expectations P are obtained by projecting these potential states onto the image plane and then smoothed using a Mexican hat filter to enhance tolerance to noise.

In each iteration, the templates are compared to the current EROS surface, the observation O, selecting the new state by maximising the correlation between the observation and the hypothesis eye model projections, P:

$$X(t+1) = \underset{X_{\Delta\theta,\Delta\phi}}{\operatorname{argmax}}\{P_{\Delta\theta,\Delta\phi} \cdot O_t\}. \tag{4}$$

The template with the highest score in θ and ϕ becomes the new state for the next iteration. If potential state templates do not show improvement, the state remains unchanged.

However, a uniform change in rotation for all possible states may lead to issues, as the pixel shift of the iris centre varies with the eyeball's orientation. To address this, we calculate the change in rotation needed for a desired pixel shift Δ_p:

$$\Delta\phi = \sin^{-1}\left(\frac{x_c + \Delta_p - u}{R \cdot \sqrt{1-\tau^2}}\right) - \phi, \quad \Delta\theta = \sin^{-1}\left(-\frac{y_c + \Delta_p - v}{R \cdot \sqrt{1-\tau^2} \cdot \cos(\phi)}\right) - \theta. \tag{5}$$

Here, $\Delta\phi$ represents the change in rotation for x-axis movement, and $\Delta\theta$ is for y-axis movement.

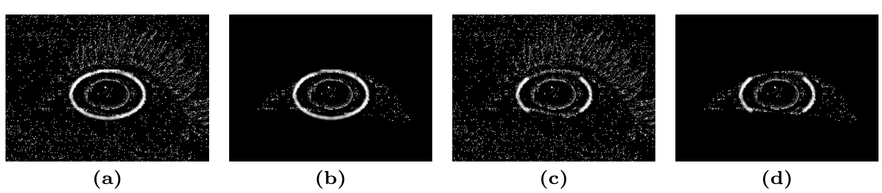


Fig. 3: A comparison of various methods in the ablation study: (a) full-ellipse (no cuts in the data or model), (b) cut-ellipse (the sensor data is only taken in a region defined by the eyelids), (c) full-segments (only side segments of the eye model are used), and (d) cut-segments (data regions and model segments are both in use).

3.1 Robustness to Occlusions

The algorithm can be improved to give robustness to occlusions that occur due to the iris falling under the eyelids. Two possible options exist to model occlusions, 1. modify the observation data to remove occlusions, 2. modify the model to account for occlusions.

Data Modification (Full/Cut) Eyelid exclusion in the data involves considering pixel values only between eyelid positions. Eyelids can be estimated with a polynomial fitting the shape of the curve. Experiments are performed by hand-fitting upper and low eyelid parameters for each dataset to understand if autonomously estimating the eyelid positions (e.g., as in [3]) is beneficial to the tracking algorithm. Once the eyelids parameters are known, the EROS only updates v_x and v_y positions within the boundaries given by the polynomials, yielding the modified EROS as the eye observation, O, as shown in Fig. 3c.

Model Modification (Ellipse/Segments) In the model, ellipses can also be segmented by introducing cuts strategically located in areas corresponding to expected eyelid overlaps ($\alpha \in [-\pi, -3\pi/4] \wedge \alpha \in [-\pi/4, \pi/4] \wedge \alpha \in [3\pi/4, \pi]$), as shown in Fig. 3b. The resulting segmented ellipses form the model expectation (P), minimising the influence of occluded regions on the score in Equation 4.

Either of these methods, or both simultaneously can be used to give robustness to occlusions.

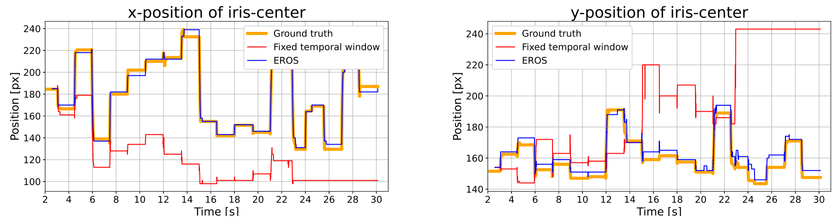
4 Experiments and Results

Experiments assess algorithmic performance for different surrounding features, examining data representation, EROS, and model cuts. The comparative analysis includes accuracy against an existing fully event-based method [2], selected for its dataset compatibility and annotated ground-truth. Gaze direction determination from the eyeball model is demonstrated, followed by a speed evaluation of the algorithm.

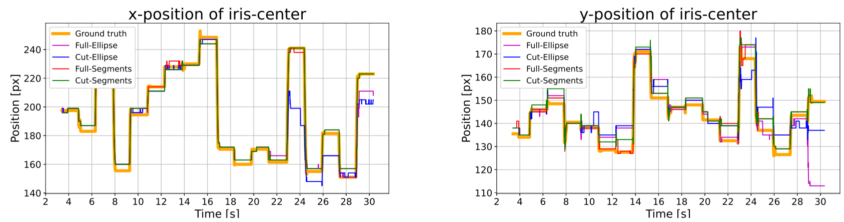
Event-based Eye Dataset The 27-users dataset [3] captured the eye movements of 27 users using the DAVIS346b sensor. The users fixate on a screen, maintaining their head still, resulting in constant eye parameters $[u, v]$, R, θ, and ϕ, forming the initial state X_0 determined manually.

Due to the algorithm's need for recovery post-blink [2], users without blinks in, the initial 30 seconds are selected for tracking evaluation: left and right eyes of users 5, 13, 18, and 19. A uniform algorithm setting is applied across all users.

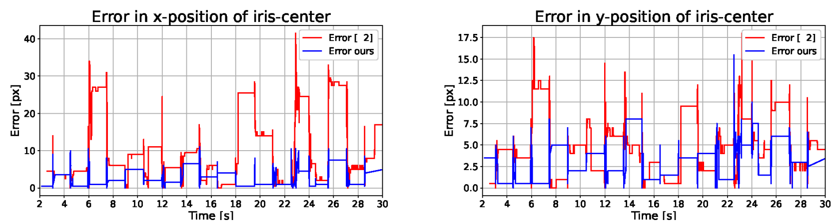
The dataset's ground truth, hand-labelled by [2], provides pupil centre data for the first 30 seconds. Experiments are consequently confined to this period due to available ground-truth.



(a)



(b)



(c)

Fig. 4: Three example sessions comparing the eye position determined by the tracker to the ground truth: (a) EXPERIMENT 1: positions for the left eye of user 5 comparing a fixed temporal window and EROS, (b) EXPERIMENT 2: position for the left eye of user 19 for the four experiments considered in the ablation study, and (c) EXPERIMENT 3: error in position for the left eye of user 5 comparing our proposed algorithm to [2].

EXPERIMENT 1: Data representation study Comparing the algorithm performance on the EROS surface versus a fixed $4ms$ temporal window reveals significantly higher iris centre position error for the fixed window, as shown in Table 1. Fig. 4a displays the x and y positions of the iris centre, illustrating that tracking discontinues after 8 seconds with the fixed temporal window. This underscores the advantageous velocity-independent property of EROS for sustained eye-tracking success.

EXPERIMENT 2: Ablation study To assess the impact of cuts in the EROS and iris model on the tracking performance, an ablation study is conducted. Fig. 3 illustrates the EROS and model configurations for the four experiments. The tracked x and y positions of the iris centre, depicted in Fig. 5, reveal that algorithms using the full EROS and ellipse, as well as cut EROS and full ellipse,

	Mean error [px]
EROS	4.129 ± 3.963
Fixed temporal window	20.619 ± 15.820

Table 1: The mean pixel error and standard deviation over all users in EXPERIMENT 1: Data representation study.

lose track at certain points (e.g., at 13 s, 27 s and 29 s) unlike the other two algorithms that maintain continuous tracking.

Table 2 shows the mean pixel error across all users for these experiments. Cutting the EROS alone does not enhance accuracy, but utilizing ellipse segments on the full surface significantly reduces the error. This remains small relative to the iris diameter (average 105 pixels). Additionally, incorporating cuts in both data and model shows improvements over using the full ellipse, though not surpassing the accuracy achieved with ellipse segments and the full EROS.

	Mean error [px]
Full-ellipse	11.489 ± 10.520
Cut-ellipse	11.613 ± 10.674
Full-segments	4.129 ± 3.963
Cut-segments	8.235 ± 7.958

Table 2: The mean pixel error and standard deviation over all users in EXPERIMENT 2: Ablation Study

EXPERIMENT 3: Comparison study The best performance of the tracking algorithm is compared against the algorithm of [2]. The results of the comparison over the users 5, 13, 18, and 19 for both algorithms are depicted in Table 3. Only the right eye of user 18 is excluded, because this data of [2] was not available. From this, we can conclude that our algorithm has a smaller mean pixel error.

In Fig. 4c the difference in error is visible. In general, the error of [2] is slightly higher, with exceptions at times $6, 18, 23$ and 25 s where the error is much higher. After this high error, the error is quickly reduced to a much smaller value. This is where the detector in the algorithm of [2] needs to correct the algorithm due to mistakes in the tracking. It should be noted that our algorithm only employs tracking and does not use a detector.

Final performance The experiments ran on an Intel Core i7-9750H CPU @ 2.60GHz, achieving a frequency of $424Hz$ without algorithm optimization. With a pixel shift of $\Delta_p = 3$, the algorithm can track shifts up to 1272 pixels per second. Qualitative observations suggest real-time performance for eye-tracking tasks.

	Mean error [px]
Ours	4.191 ± 3.963
[2]	7.349 ± 10.284

Table 3: The mean pixel error and standard deviation over all users in EXPERIMENT 3: Comparison Study.

Table 3 reveals that the best performance utilises the model with ellipse segments. Fig. 5 compares the x and y positions of the iris centre to ground truth for a single user, demonstrating accurate directional responses, even if with minor position errors.

The algorithm does not only identify the iris centre but also estimates the gaze direction, a feature lacking in [2]. Fig. 1 illustrates this with an arrow indicating the gaze direction, directly determined by θ and ϕ, aligning with the eye's fixation point.

5 Discussion and Limitations

The ablation study indicated that the speed invariance of the EROS was crucial for accurate tracking as a simple temporal integration produced motion blur for fast motions, or alternatively, was not able to observe slow eye motions. The most accurate algorithm occurred when using only segments of the iris model, which provided more robustness to occlusion as the iris commonly falls behind the eyelids. The result is significant as segments of the eye can be created more simply than removing the eyelids from the EROS. Removing eyelids would require accurate tracking of the eyelid features, which increases computation requirements and adds additional modes of algorithm failure.

The improved performance of the proposed algorithm, when compared to [2], can be attributed to the deformation model of the iris more closely matching the visual observation (i.e., EROS) rather than assuming the iris is always circular. The work in [2] was also supported by an integrated detection method, while our algorithm only performed tracking. Also due to this experiments were limited to a subset of users that did not blink in the datasets. Our work can be extended to improve robustness to blinks by developing a similar detection method, that gives the full 5-DoF state.

The algorithm is designed with the full state space of the eye to track eye position, $[u, v]$, size, $[R]$, and gaze direction, $[\theta, \phi]$, however in the event-driven datasets available the eye position and size are constant over any session. Therefore in these experiments, we only tracked changes in the gaze direction variables. The experimental limitation would need to be further explored to validate how the increase in state dimensionality impacts tracking performance. Often dimensionality increase is a major problem for algorithms, however tracking also $[u, v, R]$ also has the potential to remove human error when setting the initial position. A study of the possible performance degradation (in terms of pixel error) following a perturbation of initialization of such parameters would need

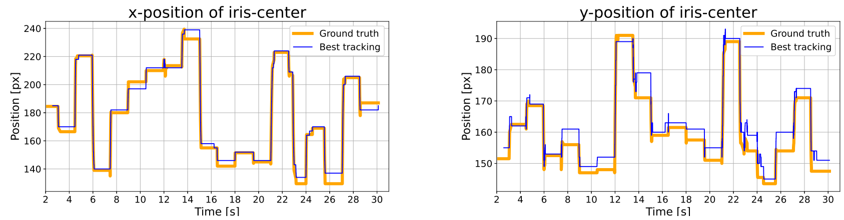


Fig. 5: The eye position for the left eye of user 5 using the best algorithm settings: full EROS and model segments.

to be explored in order to estimate the impact of possible non-perfect initialization. Moreover, experiments would need to be expanded to track the full eye state with the current datasets, and further validate on datasets with large eye position change. Nevertheless, for wearable devices, u, v, R is fixed.

The algorithm is limited to tracking only and requires the correct initial state of the eye model to function. The algorithm is fully compatible with any detection solution that, at a slower frequency, could provide an absolute eye state. A detection algorithm would enable initialisation as well as failure recovery to the presented algorithm, considering that the tracker is not guaranteed to work perfectly.

Evaluation was limited to the availability of ground-truth. The ground-truth provided by [2] (the original work [3] did not provide any ground-truth) was pupil position only. Two issues arise due to this, 1. the performance is only compared in terms of eye position, not in gaze-direction, and 2. there is a systematic offset between ground-truth pupil position and correct iris position due to 3D and transparency effects of the eyeball. Therefore the actual error of the proposed iris tracking algorithm could be substantially smaller than reported. Ground-truth for the dataset would need to be entirely re-labelled and re-processed to evaluate gaze direction and correct for iris position.

A deeper analysis of the power consumption of our method is planned as future work and is scarce in the literature, except [5, 18], testing the pipeline with different types of hardware, towards implementing low-power devices such as smart eyewear.

6 Conclusion

We presented a novel algorithm for gaze direction tracking using only an event-camera sensor capable of real-time tracking during saccades, and improving of previous algorithms, achieving an error of 4.1 pixels. The algorithm novelty arises from using the event-camera, not requiring RGB camera input, and using a deformation model to evaluate the iris position and gaze direction. The limitations of the algorithm have been discussed, and options exist to overcome each of them. The use of an event-camera is a substantial novelty in the wider field of gaze direction estimation as they have potential for low-power, low-latency, and high-frequency gaze tracking technologies. As such the system would be highly suitable for wearable eye-tracking devices, which have limited computation and power resources. We haven't yet performed extensive experiments, and at this point rely on the literature, for claims of low-power [7], and low-latency [13], and acknowledge a full pipeline is required to completely demonstrate the advantages of event-cameras compared to traditional vision sensors.

There are no widely available datasets with ground-truth and results to compare with in the event-camera state-of-the-art. To facilitate comparison, we provide curated event-data, algorithm code and results files.

APPENDIX

Mathematical morphology can be used to describe the projection on the 2D-image plane of a circle on a 3D rotated sphere, using rotation matrices.

First, the rotation should be defined. The eye either rotates by an angle θ around the x-axis (pitch) or ϕ around the y-axis (yaw). This rotation is described by

$$R_x(\theta)R_y(\phi) = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ sin(\theta)sin(\phi) & cos(\theta) & -sin(\theta)cos(\phi) \\ -cos(\theta)sin(\phi) & sin(\theta) & cos(\theta)sin(\phi) \end{bmatrix} \tag{6}$$

$R_x(\theta)$ and $R_y(\phi)$ of the 3D rotation group SO(3)

To define the rotation of the eyeball, the spherical shape should be included in the rotation. A sphere is described by the equation $x^2 + y^2 + z^2 = 1$, therefore z can be expressed in terms of x and y as

$$z = \sqrt{1 - x^2 - y^2}, \tag{7}$$

which can be simplified for a point on a circle on the sphere to

$$z = \sqrt{1 - \tau^2} \tag{8}$$

using equation 3. Thus when rotating the sphere with an angle θ and ϕ, the point (x, y, z) on a circle will have the new position

$$\begin{aligned} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} &= R_x(\theta)R_y(\phi) \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ &= \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ sin(\theta)sin(\phi) & cos(\theta) & -sin(\theta)cos(\phi) \\ -cos(\theta)sin(\phi) & sin(\theta) & cos(\theta)sin(\phi) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ \sqrt{1-\tau^2} \end{bmatrix} \\ &= \begin{bmatrix} x \cdot cos(\phi) + \sqrt{1-\tau^2} \cdot sin(\phi) \\ x \cdot sin(\theta)sin(\phi) + y \cdot cos(\theta) - \sqrt{1-\tau^2} \cdot sin(\theta)cos(\phi) \\ -x \cdot cos(\theta)sin(\phi) + y \cdot sin(\theta) + \sqrt{1-\tau^2} \cdot cos(\theta)sin(\phi) \end{bmatrix} \end{aligned} \tag{9}$$

References

1. Alberto Funes Mora, K., Odobez, J.M.: Geometric generative gaze estimation (g3e) for remote rgb-d cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1773–1780 (2014)
2. Alessi, R.: Eye Tracking Algorithm Based on An Event-Driven Camera (11 2023), `https://unire.unige.it/handle/123456789/6818`
3. Angelopoulos, A.N., Martel, J.N., Kohli, A.P., Conradt, J., Wetzstein, G.: Event-based near-eye gaze tracking beyond 10,000 hz. IEEE Transactions on Visualization and Computer Graphics **27**(5), 2577–2586 (2021). `https://doi.org/10.1109/TVCG.2021.3067784`
4. Baek, S.J., Choi, K.A., Ma, C., Kim, Y.H., Ko, S.J.: Eyeball model-based iris center localization for visible image-based eye-gaze tracking systems. IEEE Transactions on Consumer Electronics **59**(2), 415–421 (2013). `https://doi.org/10.1109/TCE.2013.6531125`
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. Chen, Q., Wang, Z., Liu, S.C., Gao, C.: 3et: Efficient event-based eye tracking using a change-based convlstm network (2023)
7. Conradt, J., Cook, M., Berner, R., Lichtsteiner, P., Douglas, R.J., Delbruck, T.: A pencil balancing robot using a pair of aer dynamic vision sensors. In: 2009 IEEE International Symposium on Circuits and Systems. pp. 781–784. IEEE (2009)
8. Demjén, E., Abosi, V., Tomori, Z.: Eye tracking using artificial neural networks for human computer interaction. Physiological research **60**(5), 841 (2011)
9. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., Scaramuzza, D.: Event-based vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(1), 154–180 (2022). `https://doi.org/10.1109/TPAMI.2020.3008413`
10. Gava, L., Bartolozzi, C., Glover, A.: Low-latency visual servoing with event-cameras (2023)
11. Gava, L., Monforte, M., Bartolozzi, C., Glover, A.: How late is too late? a preliminary event-based latency evaluation. In: 2022 8th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP). pp. 1–4 (2022). `https://doi.org/10.1109/EBCCSP56922.2022.9845622`
12. Gava, L., Monforte, M., Iacono, M., Bartolozzi, C., Glover, A.: Puck: Parallel surface and convolution-kernel tracking for event-based cameras (2022)
13. Gehrig, D., Scaramuzza, D.: Low-latency automotive vision with event cameras. Nature **629**(8014), 1034–1040 (2024)
14. Grauman, K., Betke, M., Gips, J., Bradski, G.: Communication via eye blinks - detection and duration analysis in real time. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. vol. 1, pp. I–I (2001). `https://doi.org/10.1109/CVPR.2001.990641`
15. Guenter, B., Finch, M., Drucker, S., Tan, D., Snyder, J.: Foveated 3d graphics. ACM Trans. Graph. **31**(6) (nov 2012). `https://doi.org/10.1145/2366145.2366183`, `https://doi.org/10.1145/2366145.2366183`
16. Kytö, M., Ens, B., Piumsomboon, T., Lee, G.A., Billinghurst, M.: Pinpointing: Precise head- and eye-based target selection for augmented reality. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.

p. 1–14. CHI '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3173574.3173655, https://doi.org/10.1145/3173574.3173655
17. Li, N., Bhat, A., Raychowdhury, A.: E-track: Eye tracking with event camera for extended reality (xr) applications. In: 2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS). pp. 1–5 (2023). https://doi.org/10.1109/AICAS57966.2023.10168551
18. Li, N., Bhat, A., Raychowdhury, A.: E-track: Eye tracking with event camera for extended reality (xr) applications. In: 2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS). pp. 1–5. IEEE (2023)
19. Lu, F., Sugano, Y., Okabe, T., Sato, Y.: Inferring human gaze from appearance via adaptive linear regression. In: 2011 International Conference on Computer Vision. pp. 153–160. IEEE (2011)
20. Mele, M.L., Federici, S.: Gaze and eye-tracking solutions for psychological research. Cognitive Processing **13**(S1), 261–265 (7 2012). https://doi.org/10.1007/s10339-012-0499-z, https://doi.org/10.1007/s10339-012-0499-z
21. Morris, T., Blenkhorn, P., Zaidi, F.: Blink detection for real-time eye tracking. Journal of Network and Computer Applications **25**(2), 129–143 (2002). https://doi.org/https://doi.org/10.1006/jnca.2002.0130, https://www.sciencedirect.com/science/article/pii/S108480450290130X
22. Neuhann, I.M., Lege, B.A., Bauer, M., Hassel, J.M., Hilger, A., Neuhann, T.F.: Static and dynamic rotational eye tracking during lasik treatment of myopic astigmatism with the zyoptix laser platform and advanced control eye tracker. Journal of Refractive Surgery **26**(1), 17–27 (2010). https://doi.org/10.3928/1081597X-20101215-03, https://journals.healio.com/doi/abs/10.3928/1081597X-20101215-03
23. Reginald B. Adams, J., Kleck, R.E.: Perceived gaze direction and the processing of facial displays of emotion. Psychological Science **14**(6), 644–647 (2003). https://doi.org/10.1046/j.0956-7976.2003.psci_1479.x, https://doi.org/10.1046/j.0956-7976.2003.psci_1479.x, pMID: 14629700
24. Ryan, C., Sullivan, B.O., Elrasad, A., Lemley, J., Kielty, P., Posch, C., Perot, E.: Real-time face & eye tracking and blink detection using event cameras (2020)
25. Sigut, J., Sidha, S.A.: Iris center corneal reflection method for gaze tracking using visible light. IEEE Transactions on Biomedical Engineering **58**(2), 411–419 (2011). https://doi.org/10.1109/TBME.2010.2087330
26. Stiefelhagen, R., Yang, J., Waibel, A.: Estimating focus of attention based on gaze and sound. In: Proceedings of the 2001 Workshop on Perceptive User Interfaces. p. 1–9. PUI '01, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/971478.971505, https://doi.org/10.1145/971478.971505
27. Stoffregen, T., Daraei, H., Robinson, C., Fix, A.: Event-based kilohertz eye tracking using coded differential lighting. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 3937–3945 (2022). https://doi.org/10.1109/WACV51458.2022.00399
28. Tamura, K., Hashimoto, K., Aoki, Y.: Head pose-invariant eyelid and iris tracking method. Electronics and Communications in Japan **99**(2), 19–27 (2016)
29. Vester-Christensen, M., Leimberg, D., Ersbøll, B.K., Hansen, L.K.: Deformable models for eye tracking. In: Den 14. Danske Konference i Mønstergenkendelse og Billedanalyse (aug 2005), http://www2.compute.dtu.dk/pubdb/pubs/3900-full.html

30. Wang, J.G., Sung, E., Venkateswarlu, R.: Estimating the eye gaze from one eye. Computer Vision and Image Understanding **98**(1), 83–103 (2005). https://doi.org/https://doi.org/10.1016/j.cviu.2004.07.008, https://www.sciencedirect.com/science/article/pii/S1077314204001134, special Issue on Eye Detection and Tracking
31. Wang, Z., Gao, C., Wu, Z., Conde, M.V., Timofte, R., Liu, S.C., Chen, Q., Zha, Z.j., Zhai, W., Han, H., et al.: Event-based eye tracking. ais 2024 challenge survey. arXiv preprint arXiv:2404.11770 (2024)
32. Zhao, F., Wang, H., Yan, S.: Eye glint detection and location algorithm in eye tracking. In: 2016 IEEE/CIC International Conference on Communications in China (ICCC). pp. 1–5 (2016). https://doi.org/10.1109/ICCChina.2016.7636791
33. Zhao, G., Yang, Y., Liu, J., Chen, N., Shen, Y., Wen, H., Lan, G.: Ev-eye: Rethinking high-frequency eye tracking through the lenses of event cameras. Advances in Neural Information Processing Systems **36** (2024)