This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3456153

Filtering on the Go: Effect of Filters on Gaze Pointing Accuracy During Physical Locomotion in Extended Reality

Pavel Manakhov, Ludwig Sidenmark, Ken Pfeuffer, Hans Gellersen

Abstract—Eye tracking filters have been shown to improve accuracy of gaze estimation and input for stationary settings. However, their effectiveness during physical movement remains underexplored. In this work, we compare common online filters in the context of physical locomotion in extended reality and propose alterations to improve them for on-the-go settings. We conducted a computational experiment where we simulate performance of the online filters using data on participants attending visual targets located in world-, path-, and two head-based reference frames while standing, walking, and jogging. Our results provide insights into the filters' effectiveness and factors that affect it, such as the amount of noise caused by locomotion and differences in compensatory eye movements, and demonstrate that filters with saccade detection prove most useful for on-the-go settings. We discuss the implications of our findings and conclude with guidance on gaze data filtering for interaction in extended reality.

Index Terms—eye tracking, gaze filters, gaze-based pointing, extended reality, spatial reference frames, physical locomotion

1 Introduction

Gaze interaction frequently requires the application of online filters[1] to reduce noise in eye tracking data. Interaction researchers routinely report on filtering to improve pointing accuracy, but concrete guidance on the choice of filters is scarce. Recent work by Feit et al. is notable, as it compared a variety of filters and contributed a procedure to select filter parameters [16]. However, the work was based on gaze data collected in a stationary desktop setting. In contrast, gaze in 3D settings consists of gaze vectors, and users can freely move (i.e. walk) affecting gaze and tracking [34]. As such, it is unclear whether filters suitable for stationary 2D settings are also applicable in mobile 3D settings. In this work, we investigate gaze input filters in extended reality (XR) head-mounted displays (HMDs) while standing, walking, or jogging.

Gaze dynamics "on the go" presents a substantially different challenge from previously studied stationary settings (e.g., [17]). Movement increases noise and makes it harder to acquire and track gaze targets, as is evident, for example, in the decrease in reading and pointing performance during locomotion [4, 32]. However, our visual system has natural compensatory mechanisms that stabilize gaze when we are in motion. These mechanisms include the vestibulo-ocular reflex (VOR) as well as elements of pursuit and vergence [23,24]. The VOR becomes active in response to head rotation, which typically occurs during locomotion [36]. Smooth pursuit and vergence are activated in response to the relative movement of the target in the visual field. Ideally, a filter should be robust to locomotion-induced noise, but should not undermine our natural tracking ability by removing the compensatory eye movements.

In the context of gaze interfaces in HMDs, their spatial reference frames (SRFs) play a crucial role in pointing and target acquisition. In a conventional desktop setting, targets appear in a display space fixed in the world and independent of the user's movement. In an HMD, targets can be presented in more diverse and dynamic ways, for example, anchored in the world, following the user, or affixed to the display [30]. Essentially, SRFs define how the target moves relative to the user, which affects not only gaze pointing but also how to approach filtering. Previous research has shown that VOR naturally supports the acquisition of world-referenced targets during locomotion [4]. Moreover, VOR is effective in stabilizing gaze when the target is affixed in the plane perpendicular to the direction of movement, but moves with the user along their path [34]. On the contrary, VOR makes it harder to focus on the targets affixed in the head SRF [4, 34]. It is not clear how effective filters are in different SRFs and how filtering approaches should consider SRFs.

In this study, we compare five filters on the eye tracking dataset from [34]. The dataset includes data from users visually acquiring targets in virtual reality (VR) while walking or jogging on a straight path, as well as standing. The targets were presented in four different SRFs: *World*, *Head*, *HeadDelay* as head reference with simulated inertia, and *Path* with targets affixed in front of the user but independent of head rotation. The compared filters were weighted average (WA) [25], weighted average with saccade detection (WA-SD) [16], 1€ filter (1Euro) [6], and low-pass filter with window-based saccade detection (LP-WSD) [26] from the literature, and a proposed low-pass filter with n-back saccade detection (LP-SDn). We simulated the filters' performance in a computational experiment in an online manner, i.e. using only past gaze points. As the dataset was based on segmented trials, we used a saccade simulation to assess the delay introduced by the filters. The analysis procedure was inspired by Feit et al. [16] but was extended to work with gaze data in 3D for Pareto optimal selection of filter parameters and accurate estimation of filter performance.

Our computational experiment provides new insights into the effect of user locomotion and SRFs on gaze filters. Specifically, filters were most effective when applied in the SRF where the target was located. This is significant, as eye tracking data is commonly filtered at source or in world coordinates, but we show that one can involve more accurate reference frames. Filters improved accuracy under all conditions. Relative improvement increased with user pace and was highest in the Head SRF, where compensatory eye movements interfere with accurate pointing. As expected, the filters with saccade detection performed best, with LP-WSD and LP-SDn demonstrating the highest effectiveness as filters that are the most robust to locomotion-induced noise, while 1Euro performed better than WA for filtering without saccade detection. In sum, this work contributes:

- Insights into the filters' comparative performance and factors that affect it such as the amount of noise originating from physical locomotion and the differences in compensatory eye movements when the eye fixates on the target in the four SRFs.
- Guidance on gaze data filtering in the context of physical movement, in particular, the coordinate system to apply filters, the most

- *Pavel Manakhov and Ken Pfeuffer are with Aarhus University. E-mail: {pmanakhov, ken}@cs.au.dk*
- *Ludwig Sidenmark is with University of Toronto. E-mail: lsidenmark@dgp.toronto.edu*
- *Hans Gellersen is with Lancaster University and Aarhus University. E-mail: h.gellersen@lancaster.ac.uk*

Manuscript received 14 Mar. 2024; accepted 17 Jul. 2024. Date of Publication xx xxx. 201x; date of current version 23 Jul. 2024. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

[1]Here and further we use the term "online filters" to indicate the gaze data filters that are used at runtime, as opposed to in a post-hoc analysis.

effective filters and their parameters.

2 Related Work

2.1 Gaze Interactions for Mobile Head-Worn Extended Reality

XR HMDs are becoming widely adopted by consumers, with current applications predominantly revolve around stationary environments, encompassing homes, workplaces, and exhibitions, where spatial user interfaces (UIs) appear in a fixed position in the world. Gaze pointing has been shown to be a promising input modality [17] and some HMDs, such as Microsoft Hololens 2 and Apple Vision Pro, have already adopted a gaze-based interaction model [44] as their primary means of controlling the operating system. The next generation of personal XR devices is envisioned for ubiquitous integration in everyday and mobile situations [5, 22], implying a variety of spatial UI types – from fixed UI elements anchored to objects in the world, to dynamic UI elements following the user on the go. This can materialize in the form of glanceable UI elements that allow quick access to information, such as checking a weather app or reviewing notifications [8, 31], or information labels overlaid on top of physical objects in the world reference frame [43, 47]. Or, the control of static internet-of-things appliances through smooth pursuit-based eye movements [59], spatial alignment of eye and hand rays [33], or coordinated eye-head interaction in the head reference frame [55]. Mobile applications have been envisioned with gaze control, such as Orbits [14], which leverage eye-based motion correlation to enable hands-free activation of buttons on a smartwatch, Look&Turn [49] as a multi-layer eye-hand menu situated on the forearm, and PalmGazer [45] as a mobile app menu controllable through an eye-hand interface that supports several body-based reference frames. These examples demonstrate a variety of possible SRFs and use cases for the user in motion. Our investigation of the fundamental gaze input accuracy will benefit a range of aforementioned mobile gaze interactions.

2.2 Eye Tracking Errors and Their Impact

Eye trackers are inherently noisy due to sensing and physiological factors and have received significant attention in the research community. The quality of an eye tracking signal is defined by its *trueness* (also known as 'spatial accuracy') and *precision* (in this work, we follow the terminology presented in [19]). Paraphrasing Blignaut & Blignaut [3], trueness is defined as a constant offset between the actual and recorded gaze direction, while precision error is defined as the "jitter" introduced by the eyes' natural level of noise and tracking difficulties. These factors can negatively affect gaze-based interaction and may vary between tracking areas [16], devices [42], or over time due to changing lighting conditions, pupil size, or slippage of a head-mounted display in XR settings [10, 39]. For trueness, significant errors would cause the gaze position to be outside of the target area, making accurate gaze-based interaction and inference difficult, if not impossible [13, 21, 40]. Meanwhile, precision errors can significantly affect gaze-based target selection by reducing user confidence and efficiency [21, 37]. Precision errors also significantly affect algorithms with dispersion or velocity thresholds commonly used to detect gaze movements [50] or interaction based on relative eye movements [52].

Researchers have proposed measures to reduce the impact of eye tracking errors on estimation and interaction. Poor trueness can be addressed by calibration processes, in which points in a tracking area are mapped to eye images through explicit [11, 15, 18, 20, 46] or implicit [15, 53] procedures, while gaze tracking stability during motion can be ensured by using gaze estimation pipelines robust to headset slippage [39] or novel hardware solutions that employ additional motion sensors [58] or even elastic designs of an eye tracker [2]. Trueness error has also received significant attention within gaze-based interaction techniques, where a second more accurate modality refines gaze positions [27, 54, 64].

2.3 Gaze-based Filtering

Precision is most commonly addressed through the application of online filters [6, 7, 16, 26, 57, 61]. As gaze is a noisy, yet ballistic and fast-moving signal, one must balance the amount of filtering without significantly affecting its ballistic nature. A filter that is too strong will stabilize the signal during fixation, but alter the shape of the gaze signal during saccades. A strong filter may also introduce a delay that causes the filtered signal to be significantly behind the actual gaze position. Meanwhile, a weak filter will not affect the noise level during fixation but will retain the ballistic shape of saccades. This trade-off has spurred the design of filters that are strong during fixation but weak during gaze shift [6, 16, 26, 61]. Previous filters have used speed-based parameters to dynamically alter the filter level [6], saccade detection algorithms that remove past gaze points from consideration when a significant gaze shift is detected [16, 26], or weighted filters that prioritize the most recent gaze points [16, 61]. Previous comparisons have shown that considering saccadic movements is vital in reducing jitter and increasing the performance of gaze-based interaction, and that the tuning parameters and window sizes require careful consideration [16, 61]. Feit et al. [16] used a Pareto front of optimal parameters to showcase the trade-off between accuracy and delay. We use a similar methodology in our work to highlight these trade-offs. A limitation of these comparisons is that they were conducted in stationary contexts without user movement. During locomotion, for example, turning the head or walking, the eyes will perform relatively slow low-frequency VOR movements to ensure that the eyes remain fixated on a world-referenced target [48]. Due to the stationary nature of previous filter evaluations, these VOR movements were not present, and the effectiveness of existing gaze-based filters during user movements remains unclear. To further complicate things, these compensatory eye movements may help or interfere with fixation stability depending on the SRF within which the target is attached. For instance, it was shown that during both instrumented [4] and natural walking [34] fixation of head-referenced targets is accompanied by VOR movements that make the gaze wobble around the target. We can assume that, while in this case filtering these low-frequency compensatory movements away can potentially help to increase gaze pointing accuracy, in other cases, it can lead to the opposite effect. As such, our work focuses on comparing filters and their suitability for gaze-based interaction during physical locomotion when targets are affixed in various SRFs.

3 Dataset

In this section, we describe the eye tracking dataset taken from [34] that was used in our computational experiment. The data was collected from 24 participants (mean age 27.7, *SD*=3.36; 16 men and 8 women) who were asked to read and sign informed consent and data processing forms prior to data collection. Participants were asked to follow the linear 5m track wearing an HTC Vive Pro Eye VR headset while simultaneously fixating their gaze on a set of circular virtual targets affixed in the following four SRFs (Figure 1):

- *Head* represents the SRF which origin is always aligned with the head position and which orientation copies the head orientation. The targets are affixed at 1m depth in front of the user, i.e. along the Z axis.
- *HeadDelay* is essentially similar to *Head* except it introduces a delay between the head and the SRF's movements to simulate the inertia effect.
- *Path* is a novel SRF in which the targets float in front of a moving user at a fixed distance and height above the ground. The SRF's origin moves along a predicted path and always stays in the middle of it effectively ignoring lateral oscillations of the user's head induced by locomotion. The path is estimated on the basis of the user's movement trajectory. The SRF's orientation is defined by the estimated forward direction of movement which allows the user to freely rotate their head.
- *World* is the SRF which is stationary relative to the surrounding environment. Its origin is located at the opposite end of the track while the targets are at fixed height above the ground.

Participants were first asked to look at a target located in the middle of their field of view (FoV), which would then disappear and reappear

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3456153

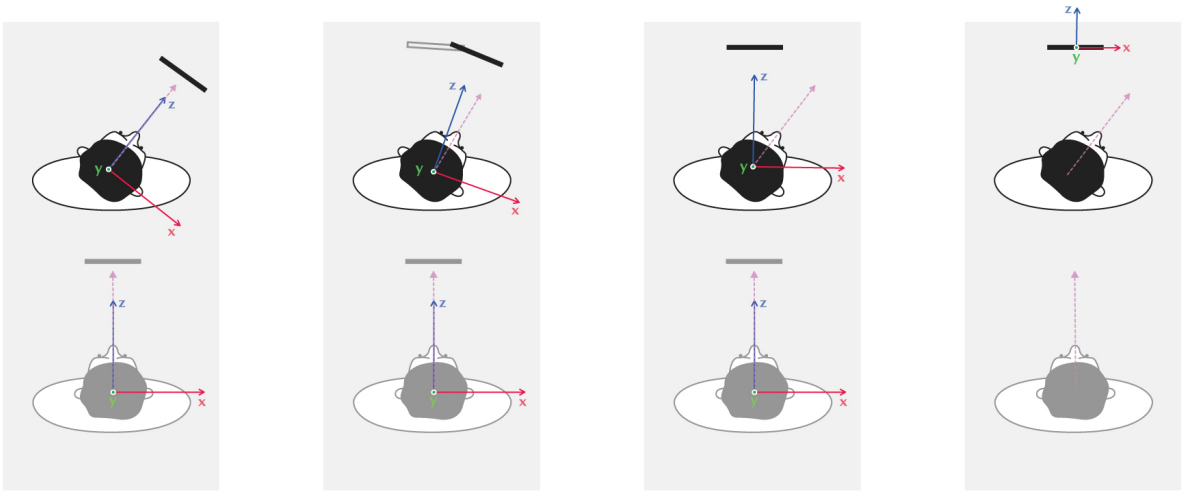

Fig. 1: Movement of a spatial UI, represented as a black rectangle, in response to the movement of a user who is taking a few steps forward along a linear track while simultaneously rotating their head to the right, represented as a purple dashed arrow, depending on a SRF: (a) the UI floats in front of the users's head at a fixed distance (Head); (b) the UI follows the user and their head movement with a delay (HeadDelay); (c) the UI is "pushed" in front of the user while keeping in the middle of the track and on a fixed height (Path); (d) the UI stays at the opposite end of the track regardless of the user's movement (World).

in the periphery. Their task was to follow the target and look at it until they heard a confirmation sound. Each trial was a minimum of 1.2 seconds long and included at least one gaze shift and a tracking fixation (we adopt the term from Lappi [29]). The direction in which the target reappears (in one of the four cardinal directions) and the distance from the FoV center (10 and 20 degrees of visual angle) were also varied. After the trial had ended, the participants were instructed to return their gaze to the FoV center, and the next trial would commence. In most of the conditions, the targets were located at 1 m depth. However, in the World reference frame where they were affixed at the opposite end of the track, the average distance to the target on the go was 3.3 m. The study took place in a virtual replica of a room aligned with its physical counterpart. To investigate the acquisition of targets at different movement paces, the participants were asked to perform the task while standing, walking, and jogging. Participants were asked to move at a predefined pace of 90 and 130 steps per minute for Walking and Jogging respectively, set by a metronome [35, 38]. This ensured consistent movement paces across all participants. Participants kept a pace of 93.17 steps/min (*SD*=5.07) for Walking and 134.26 steps/min (*SD*=4.89) for Jogging during the study. The dataset we used in the current work consisted of 2,296 trials (4 reference frames × 3 movement paces × 4 gaze directions × 2 gaze angles × 24 participants minus one condition, i.e. 8 trials, that was omitted for one of the participants).

The built-in filtering of the Tobii eye tracker embedded in the HTC Vive Pro Eye was turned off during data collection. Thus, all trials contained raw unfiltered gaze data sampled at 120 Hz. For the purposes of further analysis, we require finding the longest interval within each trial, which is devoid of natural gaze shifts and uninterrupted by blinks. To find those intervals, we employed the following data processing pipeline:

1. Segments marked by the eye tracker as invalid at least for one eye were excluded along with 5 data points on both sides of such segments. At this stage we did not interpolate the resulting gaps.
2. Local (eyes-in-head) 'cyclopean' gaze direction and origin were reconstructed from the gaze directions and origins for both eyes. The data in the local coordinate system was converted to a global one (gaze-in-world) using the known head pose.
3. The outliers left in the data at this stage were removed using a slightly modified version of the algorithm proposed by Stuart et al. [56, Section 3E]. In our version, we merged outliers when a gap between the two consecutive ones would be less than or equal to 3 frames (25 ms at 120 Hz). Angular velocity and acceleration were calculated using gaze-in-world data [1, 9]. The velocity threshold was set to 750 deg/sec [11], while the acceleration threshold was calculated from the velocity threshold by multiplying it by 1,000 similarly to [56], thus making it 750,000 deg/s². The resulting gaps would be linearly interpolated. Additionally, we filtered resulting data using 9-point wide median filter (75 ms at 120Hz) [41].
4. Gaze shifts were found using the Stuart et al.'s algorithm validated on data from walking participants [56]. It implies applying the angular gaze velocity (> 240 deg/s) and acceleration (> 3000 deg/s²) thresholds simultaneously. We used the frames before and after angular acceleration local maxima and minima to identify exact moments when a gaze shift begins and ends, respectively [9, p. 12].
5. Within each trial a tracking fixation was defined as a data segment following the last gaze shift. Within such the segment we would find the longest interval uninterrupted by the gaps left by step #1.

To ensure that we have enough data to work with, we calculated the number of trials that have an uninterrupted tracking fixation at least 40 data points long (333.3(3)ms at 120Hz). It resulted in 2,166 trials which constituted 94.34% of the original data. For weighted average-based filters, when we varied the size of their window, we also removed the trials with a fixation length lower than the current window size, which resulted in no less than 88.2% of original data for the widest window. It is important to mention that the data processing pipeline described here was used solely to identify the uninterrupted tracking fixations. In all the following calculations, we used raw data that belongs to those segments uninterrupted by gaps, including outliers if they are present in them, making the data as close as possible to what an eye tracker might output in run-time.

4 Baseline Gaze Pointing Accuracy

To understand the advantages (or disadvantages) of the online filters considered in this work, we should first establish the baseline, that is, how accurate gaze-pointing is on the go without any filters applied. A common approach to assess tracking accuracy is to calculate trueness

and precision. While these two measures are ideal for getting insight into the reasons why accuracy is the way it is, we require an aggregate measure that would help us to compare various online filters in terms of their effect on gaze-pointing accuracy. For this purpose, we have chosen the measure of target size proposed by Feit et al. [16] that is calculated as follows:

$$TargetSize = 2 \cdot (Trueness + 2 \cdot Precision) \quad (1)$$

Target size, along with trueness and precision below, is calculated in degrees of visual angle. Similarly to Feit et al., we calculate precision as the standard deviation of the gaze vector. Thus, since two standard deviations include approximately 95% of the values given that they are distributed normally[2], the 2 before precision makes it so that 95% of gaze points fall within the calculated target size. Trueness and precision, in turn, are calculated using the following equations:

$$Trueness = \overline{g} \angle t' \quad (2)$$

where $\overline{g}$ is the mean gaze direction, i.e. $\overline{g} = \frac{1}{N}\sum_{i=1}^{N} g_i$, t' is the vector from the gaze origin to the target, $\angle$ denotes an angle between two vectors.

$$Precision = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(g_i \angle \overline{g})^2} \quad (3)$$

where $g_i = (x_i, y_i, z_i)$ is the i-th gaze direction, $\overline{g}$ is the mean gaze direction.

The calculations are conducted in the constructed coordinate system whose origin coincides with the combined ('cyclopean') gaze origin, and the Z axis is always oriented towards the target to account for the target movement relative to the observer and the movement of the observer relative to the environment [1, 28].

Results. We calculated the target sizes separately for four SRFs and three movement paces aggregating gaze angles and gaze directions. One missing condition was filled using winsorization. According to the Shapiro-Wilk test not all levels of the dependent variable were normally distributed, therefore we applied the aligned rank transform to correct for it [12, 63]. We used two-way repeated measures ANOVA ($\alpha = .05$) to calculate statistical significance, followed by Bonferroni-corrected post-hoc comparisons.

We found a significant Movement×Reference Frame interaction ($F_{6,138}$=17.36, p<.001, η_p^2=.43, Figure 2). The pairwise comparisons did not reveal significant differences between SRFs when standing (4.3±1.12° on average). While walking, Head demonstrated lower accuracy and differed from all other SRFs, HeadDelay — only from World. The rest were not significantly different. When jogging, the only pair that was not significantly different was Head and HeadDelay (p=1.0). Although it may seem counterintuitive, the results are mainly based on differences in precision, which are caused by the fact that various compensatory eye movements, that support a fixation during locomotion, are quite effective in stabilizing gaze on the target located in the Path and World reference frames, thus lowering precision, but 'shake' gaze around the target in the Head reference frame, with HeadDelay being somewhere in between [4, 34].

The results are carried over to the higher-level effects such as Reference Frame ($F_{3,69}$=53.03, p<.001, η_p^2=.7) and Movement ($F_{2,46}$=204.29, p<.001, η_p^2=.9). The only pair of SRFs that did not differ significantly is Path and World (p=.074, the rest p<.001). All levels of Movement significantly differed from each other (p<.001): standing (4.3±1.12°), walking (5.79±1.57°), and jogging (8.01±2.42°), demonstrating that pace has a detrimental effect on gaze pointing accuracy.

[2]As we will see later, in the data the values are not normally distributed. However, we still believe that Equation 1 provides a good approximation of the minimal target size. Thus, we continue using it.

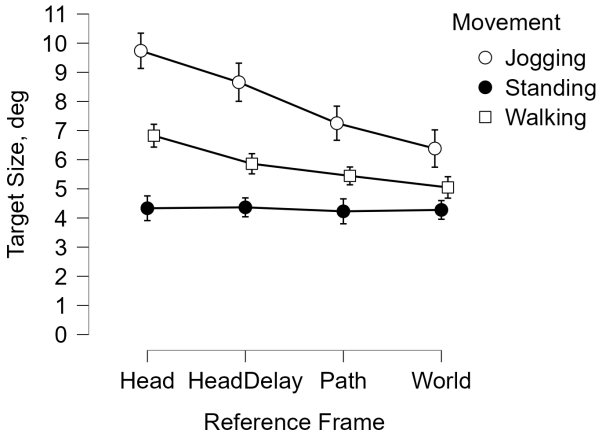

Fig. 2: Mean target sizes for the four SRFs depending on movement pace. Error bars indicate 95% confidence intervals.

5 Comparison of Online Filters

Previous research showed that while attending to the target located within various SRFs on the go, the eye moves differently [4, 34]. Fixating on a target that is more stable relative to the environment, as in the case of the World and Path reference frames, causes the eye to rotate vertically and horizontally with low frequencies of about 1-2 Hz after vertical and lateral head movements [24, 36]. The study by Borg et al. [4] clearly showed that even if the target is firmly affixed to head, the compensatory eye movements such as VOR and others [23] still significantly affect eye rotation leading to, as shown in section 4, lower gaze pointing accuracy. The HeadDelay reference frame falls somewhere in between these two ends of the continuum. The compensatory eye movements are of relatively low frequency, which distinguishes them from high-frequency noise in the gaze signal caused by the inherit imprecision of the eye tracker. This makes the application of low-pass filters a promising solution. Moreover, in the case of the Head reference frame, suppressing even those low-frequency movements can potentially help stabilizing gaze on the target. These considerations informed our choice of filters for our analysis.

5.1 Overview of the Filters

At their core, the filters described below use weighted average or low-pass filters. While the former combined with saccade detection showed itself as a promising solution for gaze filtering [16], its effectiveness is limited by the width of the averaging window. The latter is devoid of such an issue, and thus can potentially more effectively suppress low-frequency VOR movements, which might be useful in the Head SRF. Therefore, the filters we chose to compare are as follows:

- **Weighted average (WA)** [25] as a baseline that is calculated as a multiplication of the previous N points by their respective normalized weights:

$$\hat{X}_t = \sum_{i=0}^{N-1} \frac{w_i}{\sum_j w_j} X_{t-i} \quad (4)$$

where X_t is the data point captured at the time t, t being the current frame, $t-1$ — the previous one, and so on; w_i is the i-th weight, N — the width of the averaging window. It is important to note that the filter is not recursive, i.e. it never uses the filtered values $\hat{X}$ back in the calculation, only raw data. In addition to a *linear kernel* where all weights are equal to 1 making the whole filter identical to moving average [41], we use a *triangular kernel* that is recommended by [26] because it puts more weight on more recent data. The triangular kernel function is $w_i = N - i + 1$. When there are fewer data points than N, for example, when an eye tracker has just been launched or, in our case, at the beginning of a tracking fixation in each trial, the window width is adjusted

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3456153

to the number of available data points and a kernel is regenerated. Thus, the only parameter of this filter that we can control is the desired window width w ($w \in \mathbb{N}$).

- **Weighted average with saccade detection (WA-SD)** [16] is an enhancement of the filter described above that restarts it every time when the gaze shift is detected. Restarting the *WA* filter means adjusting its window width so that it would not include points before the saccade ends, which effectively means that if the current data point is considered to be a part of a saccade, the window width equals 1 and the filter output is equal to its input irregardless of a kernel. Therefore, similar to the previous filter, one of the parameters here is the maximum window width w. Saccades are detected using a simple threshold-based algorithm: if the angle between the previous gaze vector g_{t-1} and the current vector g_t is greater than the threshold, the current data point is considered a part of a saccade. The saccade threshold s is the second parameter of the *WA-SD* filter ($s \in \mathbb{R}_{>0}$).
- **Low-pass filter with saccade detection (LP-SDn)** is an online filter inspired by *WA-SD*. Instead of weighted average, it uses the following first-order low-pass filter in its core:

$$\hat{X}_t = \alpha X_t + (1-\alpha)\hat{X}_{t-1} \tag{5}$$

where X_t is the data point captured at the time t, $\hat{X}_{t-1}$ is the previous filtered value, and α is a smoothing factor which defines how much the new data point affects the filter's output. As opposed to the *WA*-based filters, the low-pass filter is recursive. Following [6], the smoothing factor is defined as follows:

$$\alpha = \frac{1}{1+\frac{1}{2\pi f_c T_e}} \tag{6}$$

where f_c is a cutoff frequency in Hz higher than which the filter attenuates the input signal; T_e is a sampling period which in our case equals to 8.3(3) ms or 1/120 Hz, considering the eye tracker's sampling rate. We propose a modification of the saccade detection algorithm from [16] that looks further into the past, i.e. at n frames back instead of at the previous one. This modification allows for higher values of the saccade threshold, and thus could make the algorithm less susceptible to locomotion-induced noise. Similarly to *WA-SD*, when a gaze shift is detected, the filter is reset, which for *LP-SDn* means that the value of $\hat{X}_t$ is set to X_t. The filter therefore has three parameters: the cutoff frequency f_c, the saccade threshold s, and the comparison frame n ($f_c, s \in \mathbb{R}_{>0}, n \in \mathbb{N}$).
- **Low-pass filter with window-based saccade detection (LP-WSD)** is a combination of the first-order low-pass filter (Equation 5) and the saccade detection algorithm proposed by Kumar et al. [26] that, in contrast to the saccade detection employed in *WA-SD* and *LP-SDn*, compares the current gaze vector g_t with the vector g_f representing a fixation and obtained by averaging the last N data points using the Equation 4 with the triangular kernel. Therefore, the filter has three parameters: cutoff frequency f_c, saccade threshold s, and fixation window width w_f ($f_c, s \in \mathbb{R}_{>0}, w_f \in \mathbb{N}$).
- **1€ filter (1Euro)** [6] is an adaptive filter that uses the same low-pass filter (Equation 5) in its core but the value of f_c is adapted depending on how fast the input signal changes, i.e. in the case of gaze data depending on gaze velocity. The 1Euro has two parameters: the minimum cutoff frequency $f_{c_{min}}$ which defines the threshold lower than which the filter cutoff frequency cannot go, and the adaption rate β that defines how fast f_c changes ($f_{c_{min}}, \beta \in \mathbb{R}_{>0}$). For more details on *1Euro*, please see Casiez et al. [6].

As opposed to working with remote eye trackers where it is customary to deal with 2D coordinates in the screen plane, in XR one works with two 3D vectors: one representing the gaze direction, another gaze origin. In our implementation of the filters, we process both on a per-component basis, i.e. independently filtering the X, Y, and Z coordinates of both vectors. The gaze direction vector is normalized after filtration.

Table 1: The lengths of simulated saccades in degrees of visual angle for 12 conditions based on the 75 percentiles of the target sizes calculated on the raw data.

	Head	*HeadDelay*	*Path*	*World*
Jogging	11.26	9.61	8.36	7.2
Walking	7.84	7.18	6.23	6.02
Standing	5.46	5.27	4.93	5.08

5.2 Data Analysis Procedure

5.2.1 Calculation of the Delay

The application of an online filter introduces a delay, i.e. the filtered signal starts to fall behind with respect to the raw signal [61], and the more aggressive the filtering, the longer the delay. Taking into account the nature of gaze signal, the lag matters the most when a person performs rapid eye movements, i.e. saccades or fast smooth pursuit movements. For that reason, we have chosen the procedure proposed by Feit et al. to assess the amount of delay introduced by filters [16]. It implies simulating a saccade by duplicating and positionally shifting data from a fixation, applying a filter, and measuring how fast the filtered signal catches up with the second shifted fixation. The original procedure was proposed to work with data from a remote eye tracker and had to be adopted to work with gaze data in 3D. The resulting procedure that we used is as follows:

1. First we calculate 75th percentiles of the target sizes for each out of 12 conditions (4 reference frames × 3 movement paces) on the raw data. The value approximates the angular distance between two nearby fixations that one can reliably select using gaze without applying a filter (see Table 1).
2. For each trial, we duplicate a tracking fixation and rotate every vector in it by as many degrees as was calculated on the previous step. The gap between the original and shifted fixations is filled by a simulated saccade by interpolating gaze direction between the last point of the first fixation and the first point of the second fixation using the following Gaussian: (0, 0.05718765, 0.16020404, 0.3569676, 0.63265705, 0.8918506, 1). This made the method more sensitive to the small changes in filters' saccade detection threshold (see end of this subsection for more detail). To account for fixation asymmetry, we create four data sequences by shifting fixations in four directions: pitch up, pitch down, yaw left, and yaw right. All operations are performed with data in the constructed coordinate system discussed above (see section 4).
3. All simulated sequences consisting of a tracking fixation, gaze shift, and another fixation are then filtered.
4. Finally, we calculate how many frames starting from the end of the saccade it takes for the filtered gaze signal to reach the halfway point between the two fixations. The values of the four sequences are averaged to produce a single value of the delay for a trial.

Concerning the original procedure [16], we made the following adaptations: (1) instead of shifting a fixation in a target plane, we rotate the gaze vectors belonging to a fixation in the constructed coordinate system, which, among other things, takes into account movement of the observer [1]; (2) since the rotation happens in 3D, we rotate fixations in four directions instead of only two. Moreover, to get the closest estimate of the performance one can get by applying the filters in a production-ready system, we made the following enhancements to the procedure:

- We decoupled the calculation of delay from the target size by shifting all fixations within one condition by the same angle calculated during step 1. This allowed us to avoid the target size influencing the delay value.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3456153

Table 2: Ranges of the filter parameters that were used in optimization.

WA	*WA-SD*	*1Euro*	*LP-WSD*	*LP-SDn*
Window width: [2, 120]	Max. window width: [2, 120] Saccade threshold: [0.3, 5]	Min. cutoff freq.: [0.01, 100] Adaption rate: [0.01, 100]	Cutoff freq.: [0.01, 100] Saccade threshold: [0.3, 5] Fixation window width: [3, 10]	Cutoff freq.: [0.01, 100] Saccade threshold: [0.3, 5] Comparison frame: [1, 3]

- Instead of simply combining two fixations together, we simulate a saccade using the aforementioned Gaussian. This made the calculation more attuned to produce different delay values for, e.g. filters with saccade detection thresholds of 1°, 1.5°, 2°, etc., when before it would ignore such small differences considering that the distance between the two simulated fixations varies from 4.93° to 11.26° for different conditions (Table 1).

5.2.2 Selecting Filter Parameters

By varying filter parameters, one can control (1) how heavily the gaze signal is smoothed and (2) how much lag is introduced. While there is not a single factor at play, the former usually translates to the increase in the gaze pointing accuracy. The latter translates into an increase in the delay in following a saccade [16]. One is a positive change, another is unwanted. Therefore, with any given filter, we trade the target size for the delay, or the other way around, which makes selecting parameters a multiobjective problem. Pareto optimal solutions are those where one of the objectives cannot be improved without hurting another. Finding the pareto optimal solutions for a given filter is not a trivial task because the function between filter parameters and resulting values of the target size and delay is not known a priori and depends on the gaze data. In this work, to select filter parameters that correspond to the pareto optimal solutions, we vary them in wide ranges (Table 2) and for each combination of the parameters calculate the target size and delay. We then find the convex hull of the points in the 2D space and select only those points on the hull that lie between the solution with the lowest delay and the most accurate solution.

The effectiveness of a filter can be determined by comparing its pareto optimal solutions that form a frontier, like in Figure 4, to the pareto optimal solutions of another filter. The filter is considered more effective if it allows for the smaller target size for a given delay. In the following we analyze the filters detailed in subsection 5.1 independently in four SRFs and three movement paces to get insight into their effectiveness in these conditions.

5.3 Results

5.3.1 Filtering in the Target Coordinate System

Filters can be applied to the gaze signal in different coordinate systems. For instance, mobile eye trackers usually have built-in online filters that are applied in the eyes-in-head coordinate system [11]. For immersive environments, it is recommended to apply filters in the gaze-in-world coordinate system [9]. In practice, one can convert gaze data to any coordinate system, perform filtering there, and then convert it back to, for example, World coordinates to use for pointing.

The results of our computational experiment indicate that *filtering eye tracking data in the coordinate system other than the one where the tracked target is located diminishes filters' effectiveness* (Table 3). Indeed, filtering gaze data for the target affixed in the head SRF in the gaze-in-world coordinate system (top right cell) leads to considerably lower performance gains than when it is filtered in the head coordinate system (top left cell)[3]. Similarly, filtering HeadDelay data in any coordinate system other than HeadDelay leads to lower filters' effectiveness (next row), etc. These results happen because filtering gaze data in an inappropriate coordinate system causes the gaze ray to fall out of sync with the target. Let us take a closer look at one illustrative trial from Joggin-Head (see Figure 3). Since we filter both the gaze direction along with the gaze origin, the gaze ray filtered in the world coordinate system starts to follow behind, i.e. its origin, and thus the whole ray starts moving relative to the actual position of the user's head because of the introduced delay. As a result, its movement repeats oscillations of the head but with a time offset that leads the gaze ray to "swing away" from the target (shown as 'Filtered (World)' on Figure 3). It does not happen if the data is filtered in the same coordinate system where the target is located (shown as 'Filtered (Head)' on Figure 3b).

In the following we conduct the analysis by applying the filters in their respective SRFs.

[3]We performed filtering only in the coordinate systems of those objects that were tracked, i.e. the global coordinate system (World), the head, and the target. That is why the values for Head-referenced targets filtered in the HeadDelay coordinate system are missing along with other similar cases.

Table 3: How much the gaze pointing accuracy has improved regarding the raw target sizes in percentage after applying filters in various coordinate systems. Vertical axis represents the SRF data comes from, horizontal axis — the coordinate system within which this data is filtered. Each cell shows the highest improvement within the delay of up to 6 frames for Standing (S), Walking (W), and Jogging (J).

SRF \Filtered in	*Head*	*HeadDelay*	*Path*	*World*
Head	J: 53.95 W: 35.93 S: 14.74	x	x	J: 1.41 W: 0.45 S: 3.71
HeadDelay	J: 4.38 W: 0.87 S: 6.69	J: 37.46 W: 25.81 S: 10.47	x	J: 7.62 W: 2.93 S: 6.09
Path	J: 5.02 W: 0.5 S: -0.12	x	J: 35.14 W: 24.25 S: 12.25	J: 6.32 W: 1.86 S: 10.67
World	J: 9.21 W: 2.22 S: 0.37	x	x	J: 19.28 W: 12.14 S: 12.04

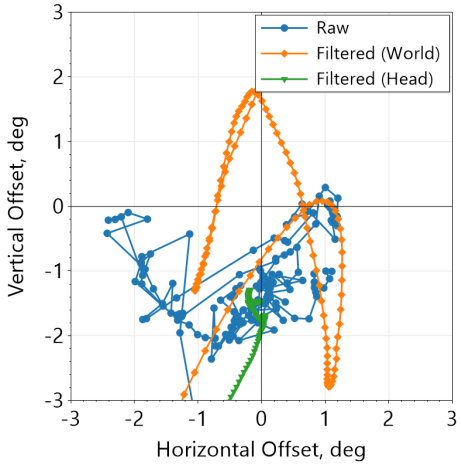

Fig. 3: Scanpaths of the raw signal and the signals filtered in the world and head coordinate systems rendered as the angular distance in degrees relative to the target position (0, 0) for one trial from Jogging-Head.

5.3.2 Filters Comparison

The Figure 4 shows the pareto optimal solutions for the studied filters which were plotted using the procedure discussed in subsection 5.2. The values of the *LP-SDn*'s third parameter (Table 2) are presented as separate plots to make the effect of the comparison frame *n* on the

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3456153

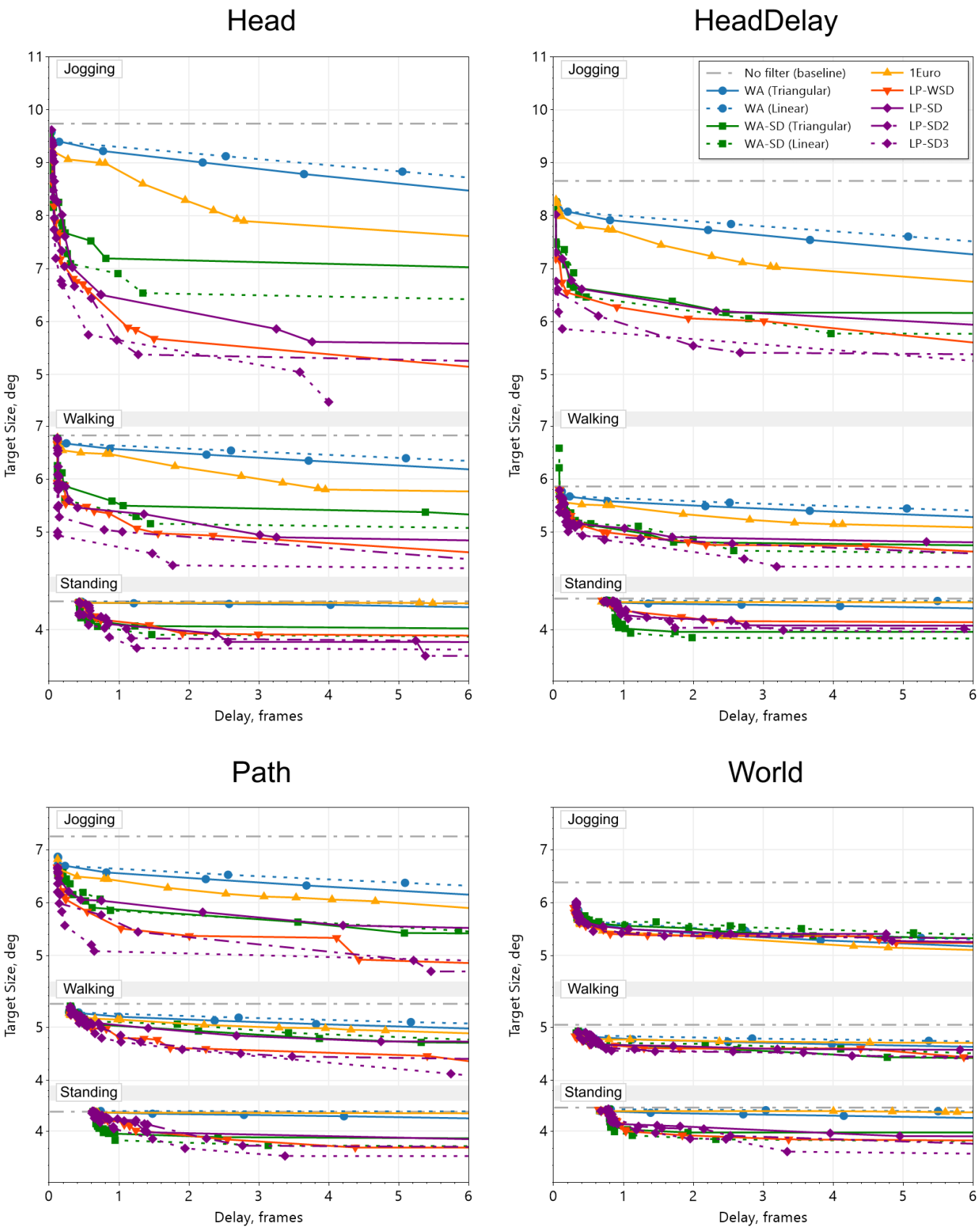

Fig. 4: Pareto frontier plots for all the filters displayed separately for the four SRFs and three movement paces. To produce the plots, the filters were applied in the same coordinate system in which the visual target was located. For comparison, a baseline accuracy, i.e. target size with no filter applied, is represented by a gray dash-dotted horizontal line.

filter's effectiveness clearer. The values of filters' parameters for each data point on the plots can be found in the supplementary materials.

It is not surprising that *WA* shows low effectiveness in all conditions, with the triangular kernel being a bit more effective. Since the filter lacks saccade detection or any adaptive mechanisms, its effectiveness grows slower than that of the other filters.

1Euro demonstrates performance comparable to *WA* in all standing conditions. However, on the go it is in between *WA* and more effective filters with saccade detection. A possible explanation for this is that the *1Euro*'s adaptive mechanism does not allow it to catch up with changes as fast as saccade detection does in *WA-SD* and *LP-SDn*, no matter its adaptation rate. It becomes clearer when looking at the filter's parameters, for example, for Walking-Head (the filter's parameters can be found in the supplementary materials). The application of the filter leads to more accurate pointing when both minimum cutoff frequency and adaption rate are at their lowest (both 0.01), however, at the price of too long of a delay (112.1 frames). The middle range with still acceptable delays of less than 12 frames (100 ms at 120 Hz) is dominated by the adaptation rate of 1 which is not fast enough to quickly react to a saccade but too fast to effectively suppress low frequency movements, even when paired with the values of minimum cutoff frequency lower than 0.5 Hz. The picture is similar in other on-the-go conditions. Interestingly, the 1€ filter's exhibiting poorer effectiveness in all standing conditions has nothing to do with movement pace. As mentioned above, we use different angular distances to simulate a saccade in the calculation of the delay (Table 1). Shorter saccades have a lower velocity, making it harder to detect by the filter's adaptation mechanism. This does not happen to saccade detection-equipped filters, because they do not use the input signal's derivative to identify a gaze shift but instead rely simply on the path went by gaze similar to I-DT algorithms [50].

WA-SD consistently demonstrates high effectiveness in all conditions. Taking a closer look at the filter's parameters, it becomes clear that, with few exceptions, the higher the saccade threshold s, the more accurate gaze pointing gets. Notably, the range of optimal parameters changes among various paces. It fills with the higher values of s, the higher the pace. It makes sense, considering that the noise level in the raw gaze output grows with the higher pace. To test this conjecture, we conducted an additional analysis by calculating the angular distances between successive gaze vectors in all tracking fixations separately for each movement pace. Indeed, the results show that both the median and 95th percentile grow at a higher pace (see Table 4 for detail). Thus, the higher the saccade threshold, the less frequently the filter restarts, i.e. its window size shrinks to 1 frame, during tracking fixation, and therefore the filter becomes more effective.

LP-SDn's effectiveness varies depending on the value of n. It is useful to look at *LP-SDn* for what it is — a combination of the low-pass filter, which can deliver heavy smoothing of the input signal, and the n-back saccade detection algorithm. The base version — *LP-SD* — with n=1 demonstrates effectiveness comparable to *WA-SD* everywhere except for Head where *LP-SD* pulls ahead. In Head SRF, due to more aggressive smoothing than the WA filter with the widest window can deliver on this data, it removes the adverse low-frequency compensatory eye movements and delivers the higher performance. However, with higher values of n, *LP-SDn*'s performance is predicated on the fact that its n-back saccade detection is less susceptible to the noise caused by physical locomotion. Thus, combining this saccade detection algorithm with other filters, for instance *WA*, can increase their performance too.

LP-WSD demonstrates higher effectiveness than *LP-SDn* with the comparison frame n of 1 across multiple conditions (Figure 4). By averaging out the last N data points to detect a saccade, it handles the noise better, i.e. it does not restart the filter as frequently as *LP-SD1* does. The other side of using window-based saccade detection is that it introduces more delay compared to *LP-SD2* and *LP-SD3*. In terms of gaze pointing accuracy, *LP-WSD* catches up with them around the 14th frame and even goes ahead in such conditions as Walking-Path and Walking-HeadDelay. However, 14 frames translate to 116 ms of delay at 120 Hz which might be considered as quite high.

To outline more general trends in the analysis:

- *Filtering is effective in increasing gaze pointing accuracy, i.e. in lowering target size, in both stationary and on-the-go conditions*.
- *Filters that distinguish between saccades and fixations perform better* as is evident from the fact that effectiveness of *1Euro* and all filters with saccade detection grows faster compared to *WA*.
- *Noise sensitivity of saccade detection is crucial on the go* causing more resilient *LP-WSD* and *LP-SDn* with higher values of n to outperform *LP-SD* that is more susceptible to noise.

5.3.3 Effects of Pace, Distance, and Reference Frame

Our data indicates that:

- *Filtering gets more effective the higher the movement pace* (see Table 3). We attribute this to the higher level of noise caused by physical locomotion. According to Table 4, the higher the pace, the greater the angular difference between two consecutive gaze vectors which, in turn, leads to a lower tracking precision which is supported by [34]. Heavily smoothing a less precise signal gives a larger gain in the gaze pointing accuracy than smoothing an already quite precise gaze signal.
- *The range of compensatory eye movements affects filters' performance*. The difference between the World and Path SRFs in both on-the-go conditions is that on average the target is closer in Path (1m) than in World (3.3m), thus leading to bigger angular movement of the gaze vector for Path caused by head oscillations. That might explain the differences in the filters' performance between these two SRFs. Notably, it is not the case with World while Standing where the targets were located at 1m depth leading to a comparable performance gain in respect to Path.
- *Head in both on the go conditions favors heavy filtering* (Table 3). In this SRF, as was shown in prior work [4,34], the low-frequency eye movements degrade tracking precision. Suppressing them gives a significant boost in the gaze pointing accuracy, and that is why in this SRF *WA-SD* with the linear kernel and *LP-SDn* with the minimum cutoff frequencies below 0.5 Hz demonstrate better effectiveness than their counterparts.

6 Discussion

While we cannot directly compare our results with the results of Feit et al. [16], where filters were applied to the 2D coordinates output by remote eye trackers, we can still draw parallels. Our Standing condition is the closest to the setup the authors had in their experiment. In there, relative performance of *WA* and *WA-SD* resembles Feit et al.'a results, however, the *1Euro*'s effectiveness is lower. The reason for this, as was mentioned in subsection 5.3, is the fact that we simulated a saccade using the Gaussian function that combined with smaller angular distance for this condition led to the poorer performance of the *1Euro*'s adaptation mechanism which is not the case for Feit et al.'s work. In the on-the-go conditions, specifically in the Head reference frame as the one that resembles the authors' setup the most because of the absence of head rotation relative to the target, *WA*, *WA-SD*, and *1Euro* comparative performance is similar to what was demonstrated by Feit et al. showing that these online filters are quite robust to changes in application environment.

6.1 Recommendations on Gaze Data Filtering in XR

Our findings suggest that when developing immersive applications that use gaze direction as their primary means of input and are designed to be used while moving, one should take great care of gaze signal filtering. The issue of diminished filter performance when applied outside the target's SRF can be solved by relegating filtering to the application. This means turning off the filtering built into an eye tracker, because it is applied in the eyes-in-head coordinate system, and applying filters in the SRF of the application's UI. In case of a complex spatial UI distributed across several SRFs, the coordinate system for filtering should be chosen dynamically. Imagine an application where a gaze-activated contextual menu is attached to the user's hand [45,49],

Table 4: Angular distances in degrees between consecutive gaze vectors for all tracking fixations and gaze shifts in the dataset depending on the movement pace.

	1-back						3-back					
	Tracking Fixations			*Gaze Shifts*			*Tracking Fixations*			*Gaze Shifts*		
Percentiles	*Median*	*75th*	*95th*	*5th*	*25th*	*Median*	*Median*	*75th*	*95th*	*5th*	*25th*	*Median*
Jogging	0.33	0.53	1.02	0.28	1.11	2.58	0.49	0.75	1.47	2.05	4.68	7.79
Walking	0.17	0.32	0.69	0.3	1.5	2.63	0.28	0.46	1	2.45	5.32	7.72
Standing	0.03	0.07	0.32	0.29	1.59	2.53	0.07	0.13	0.51	2.51	5.47	7.56

and after interacting with it, the user switches to work with World-anchored content while approaching it. In this situation, the developer can, using relative positions of the interactive zones of the menu and virtual content, identify which one is closer to the raw gaze vector, and then dynamically switch filtering from hand to world coordinate system. The whole procedure should look like this:

1. Get an unfiltered output from an eye tracker in the eyes-in-head coordinate system.
2. Fuse the output with the head position, getting two vectors: gaze origin and direction (a.k.a. cyclopean eye), both in the gaze-in-world coordinate system.
3. Identify at what part of the spatial UI the user is looking at using both vectors from step 2, and thus identify the target coordinate system for filtering.
4. Convert the gaze origin and the gaze direction from the gaze-in-world to the target's coordinate system.
5. Apply an online filter to both.
6. Convert both back to the gaze-in-world coordinate system and use them for pointing.

The results of our experiment highlight the advantages of using filters with saccade detection in step 5. Although the results detailed in subsection 5.3 can guide XR developers in selecting the filters' parameters, we caution against using too high values of the saccade threshold. Ideally, the saccade threshold should be high enough not to restart a filter because of the noise during tracking fixation but low enough to identify the smallest gaze shifts as early as possible. The data from Table 4 provides guidance in balancing the two. Using data on real, not simulated gaze shifts in the dataset from [34], we calculated the angular distances between consecutive gaze vectors for 1-back and 3-back saccade detection. When using the former, because the 95th percentile for tracking fixations overlaps with the 5th percentile for gaze shifts, we recommend using the values between the 75th and 25th percentiles, respectively (highlighted in Table 4). Choosing the higher end of the interval leads to higher gaze accuracy (see the filter parameters in the supplementary materials). Therefore, if, for instance, jogging is anticipated, one should not select a saccade threshold higher than 1.11°. For 3-back saccade detection, we recommend using the 95th percentile for tracking fixations and the 5th percentile for gaze shifts, thus going as high as 2.45° if walking is anticipated, and 2.05° if jogging is expected. The saccade threshold can even be dynamically set depending on the current movement pace of the user.

An alternative approach to gaze pointing is motion correlation [60]. For gaze-based interaction, the concept implies using smooth pursuit [14,62] or vergence [51] eye movements to infer the object the user is looking at. This approach can be advantageous during locomotion because it is less reliant on the tracking trueness. If an XR developer chooses to employ this type of interaction on the go, we believe our results can still be used as guidance on selecting filter parameters. However, we recommend picking Pareto optimal solutions that lead to less heavy filtering and lower delay to not suppress low-frequency pursuit and vergence eye movements. That being said, we acknowledge that filtering eye tracking data for use with the eye-based motion correlation on the go requires further investigation.

6.2 Limitations and Future Work

The limitations of this work are threefold. Firstly, while we believe that the conclusions we draw apply to all mobile video-based eye trackers, the data was gathered using the specific hardware, i.e. the Tobii eye tracker built into the HTC Vive Pro Eye VR headset. Secondly, our estimation of the delay introduced by an online filter relies on saccade simulation. While closely approximating a saccade, the Gaussian function we employed lacks variability of this oculomotor event observed naturally. Thirdly, we have chosen a computational experiment as the methodological approach for this work. Although the results of our simulation should closely resemble filter performance in real time, added complexity of more naturalistic contexts may influence their efficacy. Moreover, the dataset we used contains data collected in a controlled lab environment devoid of external distractions and the activities people usually perform during physical locomotion such as wayfinding and navigating complex dynamic environments. More research is needed to investigate the filters' effectiveness in more ecologically valid settings.

7 Conclusion

In this work, we investigated whether the accuracy of gaze pointing performed during walking and jogging can be improved by filtering. For this purpose, we simulated the performance of well-known online filters and the proposed low-pass filter with n-back saccade detection on the dataset of participants fixating on targets positioned in the four SRFs in an offline computational experiment. Our results clearly show that applying filters in the target coordinate system increases the effectiveness. Furthermore, they demonstrate that SRF and movement pace affect filter performance. We argue that these considerations should be taken into account when developing XR applications that are intended to be used on the go.

Acknowledgments

The current work was partially supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 101021229 GEMINI) as well as the Pioneer Centre for Artificial Intelligence (P1). We would like to thank Stasia Manakhova for her help with illustrations for the paper.

References

[1] N. Alinaghi and I. Giannopoulos. Consider the head movements! saccade computation in mobile eye-tracking. In *2022 Symposium on Eye Tracking Research and Applications*, ETRA '22, article no. 2, 7 pages. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3517031.3529624 3, 4, 5

[2] K. Binaee, B. Shankar, B. Szekely, M. Greene, and P. MacNeilage. Evaluating data stability during active head-eye tracking: A comparison of dynamic gaze error between two custom-built head-mounted devices. *Journal of Vision*, 22(14):4469–4469, 2022. doi: 10.1167/jov.22.14.4469 2

[3] P. Blignaut and T. Beelders. The precision of eye-trackers: a case for a new measure. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, 4 pages, p. 289–292. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2168556.2168618 2

[4] O. Borg, R. Casanova, and R. J. Bootsma. Reading from a Head-Fixed Display during Walking: Adverse Effects of Gaze Stabilization Mechanisms. *PLOS ONE*, 10(6):e0129902, June 2015. doi: 10.1371/journal.pone.0129902 1, 2, 4, 8

[5] A. Bulling and H. Gellersen. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing*, 9(4):8–12, 2010. doi: 10.1109/MPRV.2010.86 2

[6] G. Casiez, N. Roussel, and D. Vogel. 1 € filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, 4 pages, p. 2527–2530. ACM, 2012. doi: 10.1145/2207676.2208639 1, 2, 5

[7] S. Chartier and P. Renaud. An online noise filter for eye-tracker data recorded in a virtual environment. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, 4 pages, p. 153–156. Association for Computing Machinery, New York, NY, USA, 2008. doi: 10.1145/1344471.1344511 2

[8] S. Davari, F. Lu, and D. A. Bowman. Validating the benefits of glanceable and context-aware augmented reality for everyday information access tasks. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 436–444. IEEE, 2022. doi: 10.1109/VR51125.2022.00063 2

[9] G. Diaz, J. Cooper, D. Kit, and M. Hayhoe. Real-time recording and classification of eye movements in an immersive virtual environment. *Journal of Vision*, 13(12):5–5, 10 2013. doi: 10.1167/13.12.5 3, 6

[10] J. Drewes, G. S. Masson, and A. Montagnini. Shifts in reported gaze position due to changes in pupil size: Ground truth and compensation. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, 4 pages, p. 209–212. ACM, 2012. doi: 10.1145/2168556.2168596 2

[11] A. T. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer, Cham, Switzerland, 2017. doi: 10.1007/978-3-319-57883-5 2, 3, 6

[12] L. A. Elkin, M. Kay, J. J. Higgins, and J. O. Wobbrock. An aligned rank transform procedure for multifactor contrast tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, 15 pages, p. 754–768. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3472749.3474784 4

[13] A. Erickson, N. Norouzi, K. Kim, J. J. LaViola, G. Bruder, and G. F. Welch. Effects of depth information on visual target identification task performance in shared gaze environments. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1934–1944, 2020. doi: 10.1109/TVCG.2020.2973054 2

[14] A. Esteves, E. Velloso, A. Bulling, and H. Gellersen. Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, 10 pages, p. 457–466. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2807442.2807499 2, 9

[15] R. Fares, S. Fang, and O. Komogortsev. Can we beat the mouse with magic? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, 4 pages, p. 1387–1390. ACM, 2013. doi: 10.1145/2470654.2466183 2

[16] A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, and M. R. Morris. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, 13 pages, p. 1118–1130. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025599 1, 2, 4, 5, 6, 8

[17] A. S. Fernandes, T. S. Murdison, and M. J. Proulx. Leveling the Playing Field: A Comparative Reevaluation of Unmodified Eye Tracking as an Input and Interaction Modality for VR. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2269–2279, May 2023. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi: 10.1109/TVCG.2023.3247058 1, 2

[18] D. R. Flatla, C. Gutwin, L. E. Nacke, S. Bateman, and R. L. Mandryk. Calibration games: Making calibration tasks enjoyable by adding motivating game elements. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, 10 pages, p. 403–412. ACM, 2011. doi: 10.1145/2047196.2047248 2

[19] J. C. for Guides in Metrology. International vocabulary of metrology, fourth edition – committee draft (vim4 cd), 1 2021. 2

[20] A. R. Gomez and H. Gellersen. Smooth-i: Smart re-calibration using smooth pursuit eye movements. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, article no. 10, 5 pages. ACM, 2018. doi: 10.1145/3204493.3204585 2

[21] S.-T. Graupner, M. Heubner, S. Pannasch, and B. M. Velichkovsky. Evaluating requirements for gaze-based interaction in a see-through head mounted display. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, 4 pages, p. 91–94. Association for Computing Machinery, New York, NY, USA, 2008. doi: 10.1145/1344471.1344495 2

[22] J. Grubert, T. Langlotz, S. Zollmann, and H. Regenbrecht. Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1706–1724, 2017. doi: 10.1109/TVCG.2016.2543720 2

[23] Y. H. Han, A. N. Kumar, M. F. Reschke, J. T. Somers, L. F. Dell'Osso, and R. J. Leigh. Vestibular and non-vestibular contributions to eye movements that compensate for head rotations during viewing of near targets. *Experimental Brain Research*, 165:294–304, 2005. 1, 4

[24] T. Imai, S. T. Moore, T. Raphan, and B. Cohen. Interaction of the body, head, and eyes during walking and turning. *Experimental brain research*, 136:1–18, 2001. 1, 4

[25] J. Jimenez, D. Gutierrez, P. Latorre, and U. De Zaragoza. Gaze-based interaction for virtual environments. *Journal of Universal Computer Science*, 14(19):3085–3098, 2008. 1, 4

[26] M. Kumar, J. Klingner, R. Puranik, T. Winograd, and A. Paepcke. Improving the accuracy of gaze input for interaction. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, 4 pages, p. 65–68. Association for Computing Machinery, New York, NY, USA, 2008. doi: 10.1145/1344471.1344488 1, 2, 4, 5

[27] M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, and M. Billinghurst. Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, 14 pages, p. 1–14. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173655 2

[28] M. Lamb, M. Brundin, E. Perez Luque, and E. Billing. Eye-tracking beyond peripersonal space in virtual reality: validation and best practices. *Frontiers in Virtual Reality*, 3:864653, 2022. 4

[29] O. Lappi. Eye tracking in the wild: the good, the bad and the ugly. *Journal of Eye Movement Research*, 8(5), Oct. 2015. doi: 10.16910/jemr.8.5.1 3

[30] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, 2017. 1

[31] F. Lu, S. Davari, L. Lisle, Y. Li, and D. A. Bowman. Glanceable ar: Evaluating information access methods for head-worn augmented reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 930–939. IEEE, 2020. doi: 10.1109/VR46266.2020.00113 2

[32] Y. Lu, B. Gao, H. Tu, H. Wu, W. Xin, H. Cui, W. Luo, and H. B.-L. Duh. Effects of physical walking on eyes-engaged target selection with ray-casting pointing in virtual reality. *Virtual Reality*, Aug. 2022. doi: 10.1007/s10055-022-00677-9 1

[33] M. N. Lystbæk, P. Rosenberg, K. Pfeuffer, J. E. Grønbæk, and H. Gellersen. Gaze-hand alignment: Combining eye gaze and mid-air pointing for interacting with menus in augmented reality. *Proceedings of the ACM on Human-Computer Interaction*, 6(ETRA):1–18, 2022. 2

[34] P. Manakhov, L. Sidenmark, K. Pfeuffer, and H. Gellersen. Gaze on the go: Effect of spatial reference frame on visual target acquisition during physical locomotion in extended reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, 16 pages. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3613904.3642915 1, 2, 4, 8, 9

[35] R. G. Marteniuk, C. J. Ivens, and C. P. Bertram. Evidence of motor equivalence in a pointing task involving locomotion. *Motor Control*, 4(2):165–184, 2000. Publisher: Human Kinetics, Inc. doi: 10.1123/mcj.4.2.165 3

[36] S. T. Moore, E. Hirasaki, T. Raphan, and B. Cohen. The human vestibulo-ocular reflex during linear locomotion. *Annals of the New York Academy of Sciences*, 942(1):139–147, 2001. doi: 10.1111/j.1749-6632.2001.tb03741.x 1, 4

[37] M. H. Mughrabi, A. K. Mutasim, W. Stuerzlinger, and A. U. Batmaz. My eyes hurt: Effects of jitter in 3d gaze tracking. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 310–315, 2022. doi: 10.1109/VRW55335.2022.00070 2

[38] T. Mustonen, M. Berg, J. Kaistinen, T. Kawai, and J. Häkkinen. Visual task performance using a monocular see-through head-mounted display (HMD) while walking. *Journal of Experimental Psychology: Applied*, 19:333–344, 2013. doi: 10.1037/a0034635 3

[39] D. C. Niehorster, T. Santini, R. S. Hessels, I. T. C. Hooge, E. Kasneci, and M. Nyström. The impact of slippage on the data quality of head-worn eye trackers. *Behavior Research Methods*, 52(3):1140–1160, Jun 2020. doi: 10.3758/s13428-019-01307-0 2

[40] N. Norouzi, A. Erickson, K. Kim, R. Schubert, J. LaViola, G. Bruder, and G. Welch. Effects of shared gaze parameters on visual target identification task performance in augmented reality. In *Symposium on Spatial User Interaction*, SUI '19, article no. 12, 11 pages. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3357251.3357587 2

[41] A. Olsen. The tobii i-vt fixation filter, 3 2012. 3, 4

[42] S. Pastel, C.-H. Chen, L. Martin, M. Naujoks, K. Petri, and K. Witte. Comparison of gaze accuracy and precision in real-world and virtual reality. *Virtual Reality*, 25(1):175–189, Mar 2021. doi: 10.1007/s10055-020-00449-3 2

[43] K. Pfeuffer, Y. Abdrabou, A. Esteves, R. Rivu, Y. Abdelrahman, S. Meitner, A. Saadi, and F. Alt. Artention: A design space for gaze-adaptive user interfaces in augmented reality. *Computers & Graphics*, 95:1–12, 2021. 2

[44] K. Pfeuffer, B. Mayer, D. Mardanbegi, and H. Gellersen. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, SUI '17, 10 pages, p. 99–108. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3131277.3132180 2

[45] K. Pfeuffer, J. Obernolte, F. Dietz, V. Mäkelä, L. Sidenmark, P. Manakhov, M. Pakanen, and F. Alt. Palmgazer: Unimanual eye-hand menus in augmented reality. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction*, SUI '23, article no. 10, 12 pages. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3607822.3614523 2, 8

[46] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, and H. Gellersen. Pursuit calibration: Making gaze calibration less tedious and more flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, 10 pages, p. 261–270. ACM, 2013. doi: 10.1145/2501988.2501998 2

[47] R. Piening, K. Pfeuffer, A. Esteves, T. Mittermeier, S. Prange, P. Schröder, and F. Alt. Looking for info: Evaluation of gaze based information retrieval in augmented reality. In *Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part I 18*, pp. 544–565. Springer, 2021. 2

[48] D. Purves, G. J. Augustine, D. Fitzpatrick, L. C. Katz, A.-S. LaMantia, J. O. McNamara, and M. S. Williams, eds. *Neuroscience*. Sinauer Associates, Sunderland (MA), 2nd ed., 2001. 2

[49] K. Reiter, K. Pfeuffer, A. Esteves, T. Mittermeier, and F. Alt. Look & turn: One-handed and expressive menu interaction by gaze and arm turns in vr. In *2022 Symposium on Eye Tracking Research and Applications*, ETRA '22, article no. 66, 7 pages. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3517031.3529233 2, 8

[50] D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, 8 pages, p. 71–78. Association for Computing Machinery, New York, NY, USA, 2000. doi: 10.1145/355017.355028 2, 8

[51] L. Sidenmark, C. Clarke, J. Newn, M. N. Lystbæk, K. Pfeuffer, and H. Gellersen. Vergence matching: Inferring attention to objects in 3d environments for gaze-assisted selection. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, article no. 257, 15 pages. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3580685 9

[52] L. Sidenmark, C. Clarke, X. Zhang, J. Phu, and H. Gellersen. Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, 13 pages, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376438 2

[53] L. Sidenmark and A. Lundström. Gaze behaviour on interacted objects during hand interaction in virtual reality for eye tracking calibration. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, article no. 6, 9 pages. ACM, 2019. doi: 10.1145/3314111.3319815 2

[54] L. Sidenmark, M. Parent, C.-H. Wu, J. Chan, M. Glueck, D. Wigdor, T. Grossman, and M. Giordano. Weighted pointer: Error-aware gaze-based interaction through fallback modalities. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3585–3595, 2022. doi: 10.1109/TVCG.2022.3203096 2

[55] L. Sidenmark, D. Potts, B. Bapisch, and H. Gellersen. Radi-eye: Hands-free radial interfaces for 3d interaction using gaze-activated head-crossing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, article no. 740, 11 pages. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445697 2

[56] S. Stuart, B. Galna, S. Lord, L. Rochester, and A. Godfrey. Quantifying saccades while walking: Validity of a novel velocity-based algorithm for mobile eye tracking. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5739–5742, 2014. doi: 10.1109/EMBC.2014.6944931 3

[57] E. M. Taranta, S. L. Koh, B. M. Williamson, K. P. Pfeil, C. R. Pittman, and J. J. LaViola. Pitch pipe: An automatic low-pass filter calibration technique for pointing tasks. In *Proceedings of the 45th Graphics Interface Conference on Proceedings of Graphics Interface 2019*, GI'19, article no. 27, 8 pages. Canadian Human-Computer Communications Society, Waterloo, CAN, 2019. doi: 10.20380/GI2019.27 2

[58] M. Tomasi, S. Pundlik, A. R. Bowers, E. Peli, and G. Luo. Mobile gaze tracking system for outdoor walking behavioral studies. *Journal of Vision*, 16(3):27, Feb. 2016. doi: 10.1167/16.3.27 2

[59] E. Velloso, M. Wirth, C. Weichel, A. Esteves, and H. Gellersen. Ambigaze: Direct control of ambient devices by gaze. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, 6 pages, p. 812–817. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2901790.2901867 2

[60] M. Vidal, A. Bulling, and H. Gellersen. Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, 10 pages, p. 439–448. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2493432.2493477 9

[61] O. Špakov. Comparison of eye movement filters used in hci. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, 4 pages, p. 281–284. ACM, 2012. doi: 10.1145/2168556.2168616 2, 5

[62] O. Špakov, P. Isokoski, J. Kangas, D. Akkil, and P. Majaranta. Pursuitadjuster: An exploration into the design space of smooth pursuit –based widgets. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, 4 pages, p. 287–290. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2857491.2857526 9

[63] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, 4 pages, p. 143–146. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/1978942.1978963 4

[64] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, 8 pages, p. 246–253. ACM, New York, NY, USA, 1999. doi: 10.1145/302979.303053 2