A privacy-preserving approach to streaming eye-tracking data

Brendan David-John, *Student Member, IEEE*, Diane Hosfelt,
Kevin Butler *Senior Member, IEEE* and Eakta Jain, *Member, IEEE*

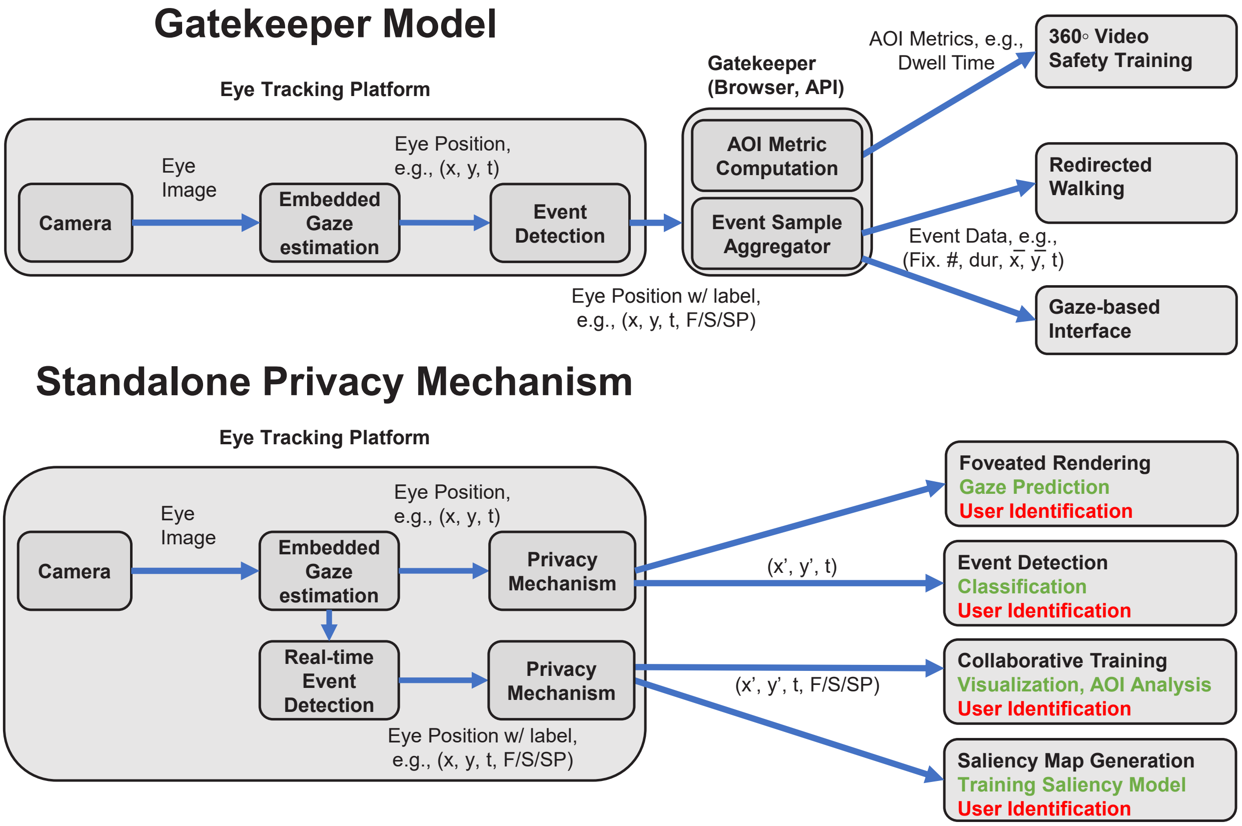

Fig. 1: Top: The *Gatekeeper* model protects identity by delivering relevant data at different levels directly through the API, while withholding raw gaze samples that contain biometric features. This approach cannot be used directly with applications that require raw gaze samples. Bottom: In scenarios where a *Gatekeeper* API cannot be implemented, we instead apply a privacy mechanism to raw gaze samples to serve applications that use gaze samples or event data directly.

Abstract— Eye-tracking technology is being increasingly integrated into mixed reality devices. Although critical applications are being enabled, there are significant possibilities for violating user privacy expectations. We show that there is an appreciable risk of unique user identification even under natural viewing conditions in virtual reality. This identification would allow an app to connect a user's personal ID with their work ID without needing their consent, for example. To mitigate such risks we propose a framework that incorporates gatekeeping via the design of the application programming interface and via software-implemented privacy mechanisms. Our results indicate that these mechanisms can reduce the rate of identification from as much as 85% to as low as 30%. The impact of introducing these mechanisms is less than 1.5° error in gaze position for gaze prediction. Gaze data streams can thus be made private while still allowing for gaze prediction, for example, during foveated rendering. Our approach is the first to support privacy-by-design in the flow of eye-tracking data within mixed reality use cases.

Index Terms—Privacy, Eye Tracking, Eye Movements, Biometrics

1 Introduction

As eye trackers are integrated into mixed reality hardware, data gathered from a user's eyes flows from the mixed reality platform to the applications (apps) that use this data. This data is a critical enabler for a number of mixed reality use cases: streaming optimization [63], foveated rendering [10, 66, 67, 79], redirected walking [49, 50, 54, 99], gaze-based interfaces [34, 84, 107], education [81], and social interaction [26, 61, 64, 70, 74]. The eye-tracking data also contains a variety of information about the user which are not necessarily needed by each

- *Brendan David-John is a PhD student at the University of Florida. E-mail: brendanjohn@ufl.edu.*
- *Diane Hosfelt was a privacy and security researcher at Mozilla at the time of writing. E-mail: dianehosfelt@gmail.com*
- *Dr. Kevin Butler is an Associate Professor at the University of Florida. E-mail: butler@ufl.edu*
- *Dr. Eakta Jain is an Assistant Professor at the University of Florida. E-mail: ejain@cise.ufl.edu*

*Manuscript received 9 Sept. 2020; revised 15 Dec. 2020; accepted 8 Jan. 2021.
Date of publication 22 Mar. 2021; date of current version 7 Apr. 2021.
Digital Object Identifier no. 10.1109/TVCG.2021.3067787*

application. For example, eye movements identify attributes such as gender, bio-markers for various health conditions, and identity. As a result, how this data is handled, and to whom, has privacy and security implications.

The problem of applications receiving data and passing it along to colluding apps or parent companies erodes public trust in technology, and cannot be "regulated away". It has received public attention in the context of similar personal devices, such as smartphones. Recently, The Weather Channel took location data it mined from users' foot traffic at different businesses, and sold it to hedge funds to inform their investments before quarterly income statements were released.[1] Even with regulation, imagine that the weather app collecting location data colludes with an advertising application that belongs to the same parent company. The user will then be served personalized ads based on her location: such as car ads appearing after a visit to the car dealership for an oil change. Now imagine that the parent company also knows which cars she glanced at while waiting, or that she actually spent most of the time looking at the motorcycle parked out front relative to the other vehicles.

This problem becomes even more severe when we recognize that mixed reality headsets are going to have as much enterprise use as personal use. A user might log in at work to do their job-related training with their known real-world identity, but attend labor union meetings as User X to avoid negative repercussions.[2,3] The agent that connects these two identities has the power to "out" the user to her work organization.

In this paper, we have investigated the threat of biometric identification of a user from their eye movements when they are being eye tracked within immersive virtual reality environments. For several mixed reality use cases, raw eye-tracking data does not need to be passed along to the application. As shown in Figure 1, a *Gatekeeper* that resides between the eye tracking platform and applications can alleviate this threat by encapsulating raw data within an application programming interface (API). We have proposed a design for such an API in Section 4.

This philosophy of serving data on a "need-to-know basis" is effective in preventing data from being used for deviant purposes instead of their originally intended purpose. However, there remain certain applications that rely on access to raw gaze data. In this case, we have proposed privacy mechanisms to erase identifying signatures from the raw gaze data before it is passed on to the application. We have evaluated how the proposed privacy mechanisms impact utility, i.e., what the application needs gaze data to do. Finally, we have investigated how the proposed privacy mechanisms impact applications that need access to eye events, i.e., eye-tracking data labeled as fixations, saccades, or smooth pursuits.

Our work is part of a broader thrust in the eye tracking and virtual reality communities on characterizing risks related to unregulated massive scale user eye tracking, and developing technological mitigations for these risks. For risks associated with an adversary gaining access to the eye image itself, we direct readers to the privacy mechanisms presented in [22,44]. For a differential privacy perspective, we direct readers to [45,59,97]. For a differential privacy perspective on the identification of users by colluding apps, we direct readers to the detailed analysis in [15,97], with the caveat that the utility task considered in this body of work is gaze-based document type classification. In contrast, we focus on utility tasks that are specific to mixed reality. Our goal is to provide a foundation for future researchers and developers to organize their thinking around the risks created by the flow of behavioral data in mixed reality, and the proactive rather than reactive design of mitigation strategies.

2 Eye-Tracking Applications in Mixed Reality

We can expect eye tracking to run as a service within a mixed reality device, analogous to the way that location services run on phones today. Eye tracking is a specific case of more general behavioral tracking services in mixed reality, including head, hand, and body tracking. Mixed reality platforms such as Microsoft and Facebook will collect raw data from the native sensors, process it to perform noise removal and event detection, and pass the processed data up the software stack. Because a rich, self-sustaining mixed reality ecosystem will rely on independent content developers, a mixed reality web browser, akin to a conventional web browser, will provide the software interface to access a wide array of content for consumers. In this section, we highlight critical eye-tracking applications for mixed reality that use aggregate-level, individual-level, and sample-level gaze data.

2.1 Aggregate-level eye-tracking applications

Aggregate gaze data is collected from many viewers to drive applications such as highlighting salient regions using heatmaps [28,82,95], and learning perceptual-based streaming optimizations for 360° content [63,101]. These applications typically rely on a data collection process conducted in research lab environments for a sample of viewers. Viewer data is then used to train machine-learning models or evaluate the most effective streaming methodology within the dataset. Results from the dataset are then released in aggregate form to inform the deployment of such methods on consumer devices. This provides utility to the consumer without creating privacy risks, however training data for machine-learning models may pose a risk to privacy [25], as well as publicly-released datasets that include the raw gaze data used to generate aggregate representations [1,40,41,57,102].

2.2 Event-level eye-tracking applications

Eye movement behavior captured by eye-tracking events, such as fixations, saccades, and smooth pursuit, contribute to gaze-based interfaces [34,77], evaluating training scenarios [19,30,43], and identifying neurodegenerative diseases [75] and ASD [18]. Detecting eye-tracking events enables improved techniques for redirected walking [49,50,54], a critical application for VR that expands the usable space of virtual environment within a confined physical environment. The most common method to quantify an individual's gaze behavior is to mark Areas of Interest (AOIs) within content and measure how gaze interacts with this region. Typical metrics for these regions depend on fixation and saccade events only, recording dwell times, the number of fixations or glances, and fixation order [55,76]. Event data also poses a privacy risk, as it reveal the viewer's intent and preferences based on how gaze interacts with different stimuli content.

2.3 Sample-level eye-tracking applications

Multiple key mixed reality applications depend on individual gaze samples from an eye-tracker of a sampling rate of at least 60Hz. This includes foveated rendering [10,66,67,79], which is expected to have the biggest impact on deploying immersive VR experiences on low-power and mobile devices. This application relies on gaze samples to determine where the foveal region of the user currently is, and to predict where it will land during an eye movement to ensure that the user does not perceive rendering artifacts [3]. Similarly, gaze prediction models are trained that predict future gaze points while viewing 360° imagery and 3D rendered content [40,41].

Another key set of applications that require sample-level data are gaze guidance techniques [88,89]. Gaze guidance takes advantage of sensitivity to motion in the periphery to present a flicker in luminance that will attract the user's eyes, using eye tracking to remove the flicker before the user can fixate upon the region and perceive the cue [8,38]. This technique enables manipulation of visual attention, and ultimately user behavior. For example, gaze guidance in 2D environments has been shown to improve spatial information recall [7], improve training of novices to identify abnormalities in mammogram images [96], and improve retrieval task performance in real-world environments [12]. Gaze guidance has also been used to enhance redirected walking techniques in VR by evoking involuntary eye movements, and

[1] https://www.nytimes.com/interactive/2019/12/19/opinion/location-tracking-cell-phone.html

[2] https://tcf.org/content/report/virtual-labor-organizing/

[3] https://www.foley.com/en/insights/publications/2015/09/be-careful-what-you-say-during-a-union-organizing

Table 1: State-of-the-art gaze-based biometric methods. Key: RBF = Radial Basis Function Network, RDF = Random Decision Forests, STAT = Statistical test, SVM = Support Vector Machine.

Method	Features	Classifier	Dataset	Results
Schroder et al. [93]	Fixation, Saccade	RBF	BioEye 2015, MIT data set	IR: 94.1%, 86.76%
Schroder et al. [93]	Fixation, Saccade	RDF	BioEye 2015, MIT data set	IR: 90.9%, 94.67%
George&Routray [35]	Fixation, Saccade	RBF	BioEye 2015	IR: 93.5%
Lohr et al. [60]	Fixation, Saccade	STAT	VREM-R1, SBA-ST	EER: 9.98%, 2.04%
Lohr et al. [60]	Fixation, Saccade	RBF	VREM-R1, SBA-ST	EER: 14.37%, 5.12%
Eberz et al. [31]	Fixations, Binocular Pupil	SVM	[31]	EER: 1.88%
Rigas et al. [86]	Fixations, Saccades, Density maps	Multi-score fusion	[86]	EER: 5.8%, IR: 88.6%
Monaco [68]	Gaze Velocity/Acceleration	STAT	EMVIC 2014	IR: 39.6%

taking advantage of saccadic suppression [99]. Guiding gaze through saccades and manipulating the user allows for use of a 6.4m×6.4m virtual space within a 3.5m×3.5m physical space, significantly improving upon the usable area within VR experiences. This application requires an eye tracker sampling rate of 250Hz or more, and requires sample-level data to know precisely when gaze moves towards the periphery cue. Providing sample-level data with high accuracy at this frequency poses a serious risk to user privacy in the form of gaze-based biometric features that can then be extracted from these gaze positions.

3 Related Work

Human eyes reflect their physical attributes. For example, algorithms can estimate the ages of users by monitoring the change in the gaze patterns as they age [73, 106], their gender based on the temporal differences in gaze patterns while viewing faces [92], and their race from the racial classification of faces they tend to look at [9].

Beyond physical attributes, gaze allows rich insights into psychological attributes, such as neurological [56] and behavioral disorders [27,72,80]. The eyes can also reveal whether an individual suffers from an affective disorder—anxious individuals' gaze is characterized by vigilance for threat during free viewing, while depressed individuals' gaze is characterized by reduced maintenance of gaze on positive stimuli [5]. Eye tracking has also been used to investigate gaze behavior in individuals on the autism spectrum, finding that they generally tend to fixate less on faces and facial features [13,23].

Pupillometry, when combined with scene metadata could allow algorithms to infer user sexual orientation, as shown in clinical studies measuring genital responses, offering a less invasive way to infer individual's preferences [85]. In addition to allowing sexual orientation inferences, pupillometry can reveal insight into women's hormonal cycles using similar methodology [52]. Pupil size also reveals the user's cognitive load [29] as well as emotional arousal, as shown in studies with images [17, 53] and videos [83]. Interestingly, pupil response seems to be modulated by subconscious processing, changing when the mind wanders [100].

Body mass index (BMI) status appears to influence gaze parameters that are not under conscious control, allowing BMI estimation when presenting individuals with images of foods of differing caloric content [37]. These risks involve knowledge of both eye position and stimuli, whereas user identification can be applied to raw eye movements without knowledge of what the stimuli was.

3.1 State-of-the-art in user identification based on eye movements

Gaze patterns can be used to identify individuals as they contain unique signatures that are not under a user's voluntary control [47,48]. The Eye Movement Verification and Identification Competitions in 2012 and 2014 challenged researchers to develop algorithms that identified users based on their eye movements when they followed a jumping dot (2012) and when they looked at images of human faces (2014). The best models' accuracy ranged from 58% to 98% for the jumping dot stimuli, and nearly 40% accuracy compared to a 3% random guess probability for viewing faces.

Based on recent surveys on eye movements biometrics [33, 87] as well as our own literature search, we identified algorithms that have been shown to successfully identify individual users from their eye movements in Table 1. These algorithms have been applied to existing gaze-biometric challenge datasets, as well as the natural viewing of image stimuli in 2D (MIT data set). The method with the best biometric performance produces an Equal Error Rate of 1.88% using pupil-based features [31], however the majority of consumer applications in mixed-reality do not require pupil diameter. Thus, we selected to implement the RBF approach proposed by George and Routray [35], as it relies only on fixation and saccade events. This method also produces impressive results with VR eye-tracking data [60] and natural viewing of 2D images [93].

3.2 State-of-the-art in eye-tracking security and privacy

In recent years privacy concerns related to eye-tracking applications has grown significantly [16,42,44,51,58,98]. In response, researchers have developed methods to enhance privacy of aggregate features, like saliency heatmaps [59] and event statistics [15,32,97]. These methods have been shown to reduce performance in classification of gender and identity, however the methods operate only on aggregate gaze data after it has been collected and processed. Recent work by Li et al. has applied formal privacy guarantees to raw streams of gaze designed to obfuscate viewer's gaze relative to AOIs within stimuli over time [58]. The ability to protect biometric identity was was evaluated empirically on the 360_em dataset [1], reducing identification to chance rate. Our work develops a threat model based on the streaming of gaze samples and the privacy risk related to biometric identification within an XR ecosystem.

4 Designing an API for Gaze Privacy

The typical architecture and data flow in an eye-tracking platform is shown in Figure 1. Existing eye trackers process user data in three stages: eye image capture, which images the user's eye, eye position estimation, which infers the point of regard from the eye image, and event detection, which classifies each point of regard as belonging to a fixation, saccade, blink, etc. When eye trackers were specialty equipment, all this data was made available to the application. These applications were typically research data gathering software. The major difference now is that the applications will have a profit-based business model. This model will naturally create incentives to share user gaze data and make inferences by combining data across devices for advertising revenue, for example. We have identified privacy risks created by this ecosystem in Section 3. In this section, we define our threat model and propose the design of an application programming interface (API) which adopts a privacy-preserving approach to passing gaze data to downstream applications.

Threat Model We assume that the components comprising the eye-tracking platform and API are trusted, i.e., the integrity of the hardware and software could be attested through mechanisms such as secure boot [4] and integrity measurement [90], and we assume that the operating system is protected, e.g., through SELinux mandatory access controls [69]. The adversary is capable of examining all data transmitted to the eye-tracking applications, and seeks to use this information to re-identify the user. An adversarial application has the capability to collude with other applications by sharing information through either overt or covert channels [65] in order to re-identify users.

Our privacy-preserving solution is focused on preventing biometric identification of users from their gaze data. First, the eye is imaged by a camera, producing an eye image that is provided to the platform, which processes the image into position coordinates. The platform provides this eye position to trusted applications like the browser, which then pass the eye position on to browser apps that perform tasks such as AOI analysis for performance in training scenarios, saccade detection for redirected walking, and smooth pursuits for gaze-based interaction.

Naïve API Design The simplest way to provide a gaze API would be to pass along the raw gaze data to applications. At any point in time, the application would be able to request `getGazePosition()`. From this, the application would be able to compute fixations, saccades, and dwell time; in particular, an AOI application would be able to compute fixations in an AOI, time to first saccade into the AOI, and dwell time in the AOI.

Providing raw gaze data also allows for computation of the velocity of eye movements, and other features that are commonly used for identity classification tasks [33,35,93]. Allowing for raw gaze access in an untrusted context, such as the web, allows arbitrary apps the ability to re-identify users.

4.1 Enabling AOI Metrics

However, we can modify the gaze API to be privacy-preserving by acting as a *Gatekeeper*. Privacy vulnerabilities are caused by the design assumption that the application is benign, and the data is used only for the purpose for which it is collected. As discussed previously, applications need not be benign, and connecting user data across devices will allow for richer inferences to be made about that user. This threat motivates our proposed *Gatekeeper* design. An added benefit of our proposed design is that the *Gatekeeper* model provides desired metrics directly to applications, instead of requiring applications to process streamed user gaze data and calculate the metrics themselves.

Advertisers and other AOI applications are interested in the number of fixations and the dwell time of a fixation in a predetermined AOI. Under the *Gatekeeper* framework, instead of passing along raw gaze positions, an API allows requests for this information. For example, a `getFixations` method takes a rectangular area and returns a list of fixations that had occurred in that area, and a `getDwellTime` method takes as input a fixation and returns in milliseconds the dwell time of the fixation. Additionally, we provide a `getSaccades` method that would return a list of saccades into the AOI. Saccades are a strong classifier feature for identity, when raw gaze points are included, however we mitigate this risk by providing only lower dimensional summary data.

It is important to note that this API is designed specifically to provide AOI metrics and summary data of eye movement events. The API does not scale to address applications such as platform foveated rendering, which requires raw gaze samples for utility. The *Gatekeeper* model does support streaming optimizations based on the current gaze position within a discrete set of tiles [20,78], by providing only information about which tile they are currently attending too. This type of optimization is critical for low-power devices to ensure high visual quality while preserving precious network resources.

4.2 Enabling Real-time Event Data

In some situations, such as gaze-based interfaces and redirected walking, applications will need to be notified when a new fixation or saccade occurs, instead of querying for all fixations or saccades.

In this scenario, we can use an `EventListener` model instead of a query-based model. When a new event occurs, the `EventListener` will be notified and given the event data, `(x, y, t)` and a boolean indicating if it is a fixation, saccade, or smooth pursuit. More complex eye movements are difficult to detect in real-time with the sampling rate of mixed reality eye-tracking devices, and typically are not implemented in real-time applications.

Our typical model for streaming event data is to send an event when the eye movement has concluded. For example, in a gaze-based interface the application needs to be notified that a smooth pursuit occurred, and where it landed. In applications such as redirected walking it is critical to know when a saccade begins, to take advantage of saccadic blindness [49,50,54,99]. In this case, one mode of the `EventListener` will be to indicate when a saccade event has started and finished, as opposed to only when the saccade has finished.

Table 2: Privacy mechanism variable definitions.

Variable	Description
x	Horizontal gaze position
y	Vertical gaze position
t	Timestamp
e	Event label: Fix. (F), Sacc. (S), Smooth Pursuit (SP)
X	Input time series of gaze samples
G	Number of gaze positions in time series
X'	Output privacy-enhanced time series
K	Temporal downsample factor relative to sampling rate
L	Spatial downsample factor relative to 3840×2160
M	Number of rows in equirectangular projection
N	Number of columns in equirectangular projection
δ_x	Horizontal step size: $\frac{360}{N}$
δ_y	Vertical step size: $\frac{180}{M}$

4.3 Enabling Privacy-enhanced Sample Data

Most applications will be able to function with the aforementioned API designs; however, two key mixed reality applications that will require sample-level data are foveated rendering and subtle gaze guidance.

Foveated rendering is critical for performance on next generation wearable VR headsets. In an ideal situation, platforms will use GPU-based foveated rendering—where gaze information is sent to the graphics driver, informing it to do fewer calculations for the parts of the screen that are away from the center of view. This requires cooperation with the graphics hardware driver for optimal performance. Experiments on native platforms show up to a 2.71 times speed up in frames per second [66]. This will not be possible in all cases, so platforms and browsers will also need to leverage software-based foveated rendering and streaming optimization [71]. In this scenario, gaze samples are transmitted directly to the content or webpage, which then knows where it should render objects in more detail. However, this exposes the raw gaze data to the application and allows the content to perform further processing on the raw gaze information, whether that is user identification or inferring sensitive characteristics.

In these scenarios the eye-tracking platform must stream sample-level data, and it is impossible to simply abstract data using a privacy-preserving API. Therefore, we propose the use of a privacy mechanism to manipulate gaze samples as they are streamed to increase privacy.

5 Methodology

In this section, we propose, implement, and evaluate three privacy mechanisms with the goal of mitigating the threats identified in Section 4. Our goal is to reduce the accuracy of user identification based on features derived from common eye events, such as fixations and saccades. We consider the following privacy mechanisms: addition of Gaussian noise to raw gaze data, temporal downsampling, and spatial downsampling. We implement these mechanisms and evaluate them against the baseline identification rate when raw gaze data is passed to the application as is. For each of the privacy mechanisms, we also evaluate the utility of the data that is passed downstream.

5.1 Privacy Mechanism Definitions

We define the data received by the privacy mechanism to be a time series where each tuple is comprised of horizontal and vertical gaze positions (x,y), a time stamp t, and the event label assigned to the sample e: $X = \{(x_1,y_1,t_1,e_1),(x_2,y_2,t_2,e_2),...,(x_G,y_G,t_G,e_G)\}$, a set of G gaze positions. This data is processed via a privacy mechanism and the processed output as a time series X', with additional variables defined in Table 2. The following three privacy mechanisms are explored in this paper.

Table 3: Dataset characteristics.

Dataset	Participants	# Stimuli	Avg. # Stimuli	Stimuli Duration	Stimuli Type	Task
ET-DK2 (ours)	18	50	50	25s	360° Images	Free Viewing
VR-Saliency [95]	130	23	8	30s	360° Images	Free Viewing
VR-EyeTracking [102]	43	208	148	20s-70s	360° Videos	Free Viewing
360_em [1]	13	14	14	38s-85s	360° Videos	Free Viewing
DGaze [40]	43	5	2	180s-350s	3D Rendered Scene	Free Viewing

Additive Gaussian Noise Noise is sampled from a Gaussian distribution of zero mean and standard deviation σ defined in visual degree and added to the gaze positions. Noise is independently sampled for horizontal and vertical gaze positions as $X' = \{(x_1 + N(0,\sigma), y_1 + N(0,\sigma), t_1, e_1), (x_2 + N(0,\sigma), y_2 + N(0,\sigma), t_2, e_2), ..., (x_G + N(0,\sigma), y_G + N(0,\sigma), t_G, e_G)\}$.

Temporal Downsampling Temporal downsampling reduces the temporal resolution of the eye-tracking data stream. Downsampling is implemented by streaming the data at a frequency of the original sampling rate divided by a scaling parameter K. The output time series is defined as $X' = \{(x_{(K\cdot p)+1}, y_{(K\cdot p)+1}, t_{(K\cdot p)+1}, e_{(K\cdot p)+1}), ...\}$ for all integers $p \in [0, \frac{G}{K}]$. For example, with a scaling parameter of two, the private gaze positions are defined as $X' = \{(x_1, y_1, t_1, e_1), (x_3, y_3, t_3, e_3), (x_5, y_5, t_5, e_5), ...\}$, retaining only every other gaze sample. For a scaling parameter of three, $X' = \{(x_1, y_1, t_1, e_1), (x_4, y_4, t_4, e_4), (x_7, y_7, t_7, e_7), ...\}$.

Spatial Downsampling Spatial downsampling reduces the resolution of eye-tracking data down to a discrete set of horizontal and vertical gaze positions. Intuitively, the scene is divided into a grid and each gaze sample is approximated by the grid cell that it lies within. Spatial downsampling is performed by defining a target equirectangular domain spanning 180° vertically and 360° horizontally with M rows and N columns. For smaller values of M and N there are less possible positions, and thus reduced spatial resolution. Raw gaze positions $(x \in [0, 360^\circ), y \in [0, 180^\circ), t)$ are transformed by first computing the horizontal step size $\delta_y = \frac{180}{M}$ and vertical step size $\delta_x = \frac{360}{N}$. Downsampled gaze positions are then computed as $(\lfloor \frac{x}{\delta_x} \rfloor \cdot \delta_x, \lfloor \frac{y}{\delta_y} \rfloor \cdot \delta_y, t)$, where $\lfloor \cdot \rfloor$ represents the floor function that rounds down to the nearest integer.

For the results presented in this paper, we parameterize spatial downsampling as a factor L relative to an equirectangular domain of $M = 2160$ and $N = 3840$, mapping to a domain of $M = \frac{2160}{L}$ and $N = \frac{3840}{L}$. For example, an input downsampling factor of L equals two will result in $M = 1080$ and $N = 1920$, a factor of L equals three will result in a resolution of $M = 720$ and $N = 1280$, and so on.

5.2 Datasets

In order to evaluate the privacy mechanisms on how effectively they prevented an adversary from re-identifying the user, we selected five existing datasets of VR eye-tracking data. Table 3 presents characteristics of each dataset included in analysis. Datasets were selected to have diversity in the number of participants, the number of stimuli presented, and the task being performed. Four of the datasets are publicly available, while ET-DK2 consists of data previously collected by the authors.[4]

5.2.1 ET-DK2

The ET-DK2 dataset consists of twenty participants viewing fifty 360° images using an Oculus-DK2 HMD with integrated SMI 60Hz binocular eye tracker. Data was collected under an IRB approved protocol in December 2017 for the purpose of generating saliency maps from gaze data. Two participants were not included in analysis, as one participant got motion sickness, and the data collection software did not log data from all 50 images for one participant. The remaining 18 individuals were made up of five females and thirteen males with an average age of 32, and an age range of 23 to 52 years. Each participant viewed 40 images from the Salient360! [82] dataset and ten additional images from construction sites in random order. Participants were seated in a swivel chair so they could rotate and explore each 360° scene while eye and head movements were recorded.

All participants performed a 9-point calibration at the beginning of the experiment, and eye-tracking accuracy was validated to less than 2° visual angle before image viewing. Each 360° image was shown for 25 seconds, following the Salient360! [82] protocol. In contrast to their protocol, we varied the starting orientation of the participant within the 360° image across eight orientations instead of being held constant. Halfway through the experiment participants were given a five minute break, after which the eye tracker was re-calibrated before viewing the rest of the images. The entire data collection process took approximately 40 minutes, including informed consent and a post-study demographics survey.

5.2.2 VR-Saliency

The VR-Saliency [95] dataset includes gaze data collected from participants viewing 360° images on a 2D display, in VR while seated in a swivel chair, and in VR while standing. We analyze only the seated VR condition, as it is the only VR condition with raw data available at 120Hz for all stimuli. Free-viewing data was collected in a similar manner to ET-DK2 for the purpose of saliency map generation, however only eight 360° images were viewed by each participant.

5.2.3 VR-EyeTracking

The VR-EyeTracking [102] dataset includes gaze data collected at 100Hz from participants viewing 360° videos. The dataset application is to train a deep network model for predicting gaze within dynamic VR environments. The video stimuli did not have a fixed duration, as in ET-DK2 and VR-Saliency, however participants viewed many videos and took many breaks to avoid motion sickness.

5.2.4 360_em

The 360_em [1] dataset includes gaze data collected at 120Hz from participants viewing 360° videos. Fourteen of the stimuli consisted of typical 360° videos from YouTube, while one stimuli was created by the authors to elicit specific eye and head movements. The dataset application is to train and evaluate event detection algorithms, classifying fixation, saccade, smooth pursuit, and OKN events in VR viewing data. For our analysis we only consider the fourteen stimuli downloaded from YouTube.

5.2.5 DGaze

The DGaze [40] dataset includes gaze data collected at 100Hz from participants that explore and navigate various 3D rendered scenes. Within each environment multiple animals dynamically move around, attracting visual attention of the participant. Gaze data is used to train and evaluate the DGaze model for gaze prediction. DGaze can predict gaze position given head orientation, or predict the next gaze position given the current gaze position. Gaze prediction by DGaze has been demonstrated in the context of foveated rendering, and can help account for latency in the eye-tracking and rendering pipeline [3, 40, 79].

5.3 Metrics

For each dataset metrics are computed to identify privacy risks, and evaluate the impact of privacy mechanisms on application utility. Utility measures depend on the application of eye-tracking within the datasets, ranging from AOI analysis to gaze prediction. We define a utility metric for each dataset depending on the type of stimuli and application.

[4] The dataset will be released publicly when the manuscript is published

5.3.1 Privacy

In our context, privacy refers to how effectively the mechanism prevents an adversary from identifying an individual. Identification is defined as a classification task: an algorithm matches the input to the database and return the closest match. If the algorithm matches the input to the ground truth identity, then the comparison is counted as a True Positive, otherwise it is considered a False Negative. The Identification Rate (IR), is the total number of True Positive classifications divided by the total number of comparisons [47,48,93]. A high IR indicates accurate classification of identity, and therefore, low privacy.

5.3.2 Utility

Predicting future gaze position from eye-tracking data is a critical area of research that has yet to be solved [40,41]. Using the DGaze dataset we evaluate the ability to predict ground truth gaze position 100 ms into the future when gaze data output from a privacy mechanism is used as the testing data, and as both the training and testing data. Utility is measured as angular gaze prediction error for each input gaze sample, with lower values indicating higher accuracy.

The most common form of eye-tracking analysis is performed using static AOIs defined within image content [55,76]. AOI analysis is used to study gaze behavior during social interaction [11], while viewing websites [103], and to evaluate content placement in 3D environments [2], among many other applications. A key AOI metric that is robust to fixation detection parameters is dwell time [76]. Dwell time measures how long a viewer's gaze fell within an AOI, and allows for comparison between which AOIs attracted the most attention. We evaluate the loss in utility between ground truth and gaze data output by a privacy mechanism by computing the Root Mean Squared Error (RMSE) between AOI dwell times. AOI utility is measured for the ET-DK2 dataset, as two rectangular AOIs are marked within each image that correspond with a salient object, such as people or natural landmarks, to measure individual viewing behavior within the scene.

Eye-tracking data is also used to generate saliency maps, which represent a probability distribution over visual content that highlights regions most likely to be looked at by a viewer [55]. Saliency maps are generated from aggregate eye-tracking data from many viewers and are used to train and evaluate deep learning models for saliency and scanpath prediction [6,24]. Saliency metrics are computed for both 360° images (VR-Saliency), and 360° video (VR-EyeTracking and 360_em). We compute KL-Divergence [55] to measure the impact on aggregate-level gaze measures and saliency modeling.

5.4 Implementation Details: Biometric Re-identification

We define two classifiers for biometric identification using a Radial Basis Function (RBF) network [35,60], with one network to classify fixation events and one to classify saccade events. This method is analogous to a traditional neural network with an input layer representing a feature vector $\vec{x} \in \mathbb{R}^p$ containing p fixation or saccade features from a single event, one hidden layer consisting of m nodes, and an output layer containing c class scores, one for each unique individual in the dataset. The output class scores are used to measure which individual the input feature vector is most similar to. Thus, larger scores indicate a higher probability of the fixation or saccade event being from that class, or individual. Each node in the hidden layer is defined by an activation function $\phi_i(\vec{x})$ and a set of real-valued activation weights $w_{i,c}$, where $i \in [1,2,\ldots,m]$ and $j \in [1,2,\ldots,C]$. The similarity score for a given class c in the output layer is computed as a weighted sum of all activation functions in the hidden layer,

$$Score_c(\vec{x}) = \sum_{i=1}^{m} w_{i,c} \cdot \phi_i(\vec{x}). \tag{1}$$

The activation function of each hidden node takes the form of a Gaussian distribution centered around a prototype vector $\vec{\mu}_i$ with spread coefficient β_i. The function is defined as

$$\phi_i(\vec{x}) = e^{-\beta_i ||\vec{x} - \vec{\mu}_i||^2}, \tag{2}$$

with shape coefficient β_i and prototype feature vector $\vec{\mu}_i$ defined prior to training the network. Thus, an RBF network must be constructed in two stages by first defining the prototypes and then optimizing the activation weights.

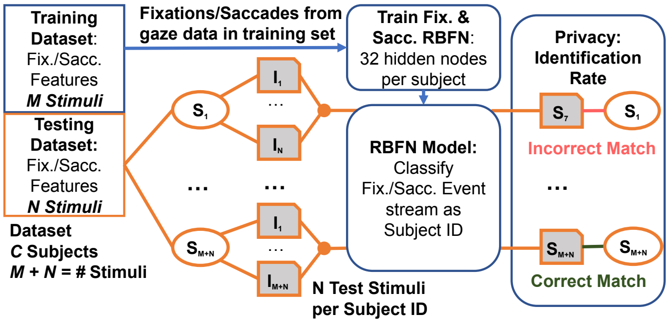
Training Dataset: Fix./Sacc. Features M Stimuli
Fixations/Saccades from gaze data in training set
Train Fix. & Sacc. RBFN: 32 hidden nodes per subject
Privacy: Identification Rate
Testing Dataset: Fix./Sacc. Features N Stimuli
S_1
I_1
I_N
S_{M+N}
I_{M+N}
Dataset
C Subjects
M + N = # Stimuli
RBFN Model: Classify Fix./Sacc. Event stream as Subject ID
N Test Stimuli per Subject ID
S_7
S_1
Incorrect Match
S_{M+N}
S_{M+N}
Correct Match

Fig. 2: Evaluation procedure for the gaze-based biometric classifier.

First, k-means clustering is applied to a training set of n feature vectors to determine k representative feature vectors per individual [35, 60]. Through this process β_i and $\vec{\mu}_i$ are defined for each of the $m = k \cdot c$ hidden nodes. The activation function $\phi_i(\vec{x})$ is then defined using the cluster centroid as $\vec{\mu}_i$, and β_i as $\frac{1}{2\sigma}$, where σ is the average distance between all points in the cluster and the centroid $\vec{\mu}_i$.

Second, the activation weights $w_{i,c}$ are learned from the same set of training data used to define the activation functions. Weights are trained using only fixation or saccade features from the training set. Training can be implemented using gradient descent [94], or by the Moore–Penrose inverse when setting up the network as a linear system [35]. The latter method is implemented in this work by defining the RBF network using an activation output matrix $A_{n\times m}$, where rows consist of the n training feature vectors input to the m previously defined activation functions, weight matrix $W_{m\times c}$ comprised of activation weights $w_{i,c}$, and an output matrix $Y_{n\times c}$ generated as a one-hot encoding of the ground truth identity labels. Using matrix multiplication the following system defines the RBF Network $A \cdot W = Y$.

The weight matrix W is then learned by computing $W = A^* \cdot Y$, where A^* is the Moore-Penrose inverse of A computed using MATLAB's *pinv* implementation. Class score predictions $\hat{Y}$ are then generated for the testing data $\hat{A}$ by computing $\hat{A} \cdot W = \hat{Y}$. Every sample in the testing set is then classified as the class label with the maximum score. To classify a stream of events the class scores from all events are first summed together, and then the class with the maximum value returned. Scores from the fixation RBF and saccade RBF are combined by summing the average of scores from each network for equal contribution to the final classification.

5.5 Evaluation Protocol

The evaluation protocol for the RBF-based biometric, illustrated in Figure 2, is derived from [93], where a stream of gaze data collected from multiple participants viewing numerous static images is used for training and testing the identity classification. The size of the training and testing sets are defined by the number of stimuli from which gaze data is used. For example, with a train/test split of 50%/50%, gaze data from half of the dataset is selected at random and used for training and the other half for testing. Fixation and saccade events data from all C participants are aggregated from the training stimuli and are then used to train the fixation and saccade RBF networks for classifying identity, as described in Section 5.4. Fixation and saccade events from the testing set are input to the trained RBF networks to classify the identity of each participant. Each participant is present in both the training set and the testing set. Identification rate is then computed as the number of correct matches divided by the number of comparisons.

6 Results

In this section we will compute privacy and utility metrics to evaluate the proposed privacy mechanisms from Section 5.1 for each dataset

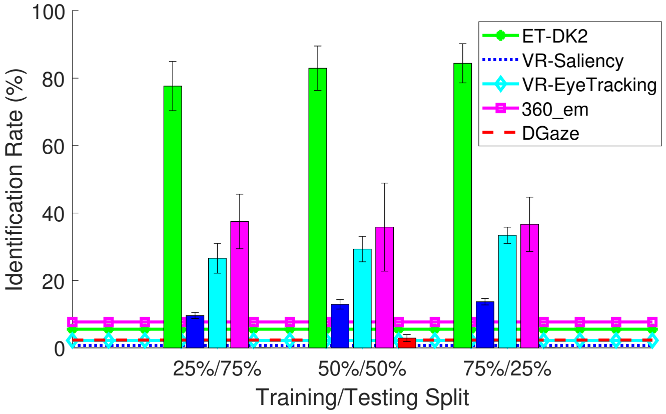

Fig. 3: Mean and standard deviations of identification rates across datasets of 360° images (ET-DK2, VR-Saliency), 360° videos (VR-EyeTracking, 360_em), and 3D rendered scenes (DGaze). Lines for each dataset indicate a baseline of random guessing for the given number of subjects.

listed in Table 3. In Section 6.1, we first compute identification rate using the RBF biometric for each dataset without modification, to establish a baseline privacy risk. Then, we compute identification rate for the privacy mechanisms for different parameter values and discuss observed effects. Last, in Section 6.2 we explore the privacy achieved by each mechanism, and the measured impact on eye-tracking utility.

6.1 Gaze-based Biometric

We evaluate the RBF biometric by splitting gaze data from stimuli viewed by each participant into training and testing sets as described in Section 5.5. For each dataset we evaluate a 75%/25%, 50%/50%, and 25%/75% train/test split, except for DGaze as each participant only saw two stimuli. Identification rate is computed over ten runs with random stimuli selected as part of the training and test set, to account for variance in stimuli content.

Figure 3 presents the mean and standard deviation of identification rates for each dataset, along with a baseline rates corresponding to random guessing. For all datasets, identification rate were highest when there was more training data than testing data, i.e., a 75%/25% split. ET-DK2 produced the highest identification rate with 85% on average, where participants viewed 50 static 360° images. VR-Saliency used a similar protocol with 130 participants, however only eight images were shown to each individual on average. A lower identification rate of 9% was observed in this dataset, compared to a baseline guess rate of 0.77%. Further analysis comparing identification rates for ET-DK2 using only eight stimuli, and VR-Saliency with eighteen random subjects closed the gap, producing identification rates of 47% and 22% respectively. Identification rates for the VR-EyeTracking and 360_em datasets are lower on average than the ET-DK2 dataset, reporting rates of 33% and 47%. We observed that DGaze produced an identification rate of 2.7%, showing only slight improvement over a baseline rate of 2.3%. This dataset differs in that participants moved through two 3D rendered virtual scenes using a controller for teleportation for several minutes at a time, instead of viewing many 360° scenes from a fixed viewpoint.

In summary, we observe that using more data for training and viewing many different stimuli produces higher identification rates. Thus, it will become easier and easier to re-identify an individual as a large volume of gaze data is collected in a variety of contexts. Identification rates are as high as 85% depending on the circumstances, highlighting the need to enforce privacy in future mixed reality applications.

Figure 4 presents the mean and standard deviations achieved when privacy mechanisms are applied to each dataset. A training/testing split of 75%/25% is used to generate these results. We observe that Gaussian noise achieves the most privacy, reducing the identification rate of ET-DK2 from 85% to 30% on average. Temporal downsampling is not recommended, as it had the least observed impact on identification rate and event detection is degraded at sampling rates less than 120Hz [104].

6.2 Utility Evaluation

The utility of eye-tracking data depends on the context of the application, thus we evaluate the impact of our privacy mechanisms at three different scales: sample-level gaze points, individual-level gaze behavior, and aggregate-level gaze behavior over many individuals. First, we evaluate sample-level utility by computing gaze prediction error using the DGaze neural network architecture, then, individual-level utility by computing dwell time for AOIs defined in the ET-DK2 dataset, and finally, we compute aggregate-level utility measures for generating saliency heatmaps of 360° images and video by computing KL-Divergence for the VR-Saliency, VR-EyeTracking, and 360_em datasets. Tables 4, 5, and 6 present the impact of privacy mechanisms on utility based on the parameter that provided the largest decrease in identification rate.

Gaze Prediction Evaluating gaze prediction accuracy involved configuring the DGaze neural network to predict gaze position 100ms into the future, which as a baseline produces an average gaze prediction error of 4.30°. Gaze prediction error was as high as 9.50° for the Gaussian mechanism, more than double the baseline gaze prediction error reported in [40]. Next, we evaluated performance by re-training the DGaze model from scratch and applying privacy mechanisms to both training and testing data dataset. This resulted in much lower prediction errors, with results as low as 5.44° (Table 4), which are comparable to the 4.30° reported in [40].

Introducing the privacy mechanism to both training and testing data implies that raw gaze data is not shared with any party during model training and deployment. Our experiments indicate that it is still possible to learn a reasonable gaze prediction model without access to the raw gaze data. Withholding raw gaze data from the training dataset is desirable, as it removes the need to safeguard additional data and alleviates the risk of membership inference attacks [25]. We expect future gaze prediction models will improve in performance, and in turn decrease the absolute gaze prediction error when using gaze data output from the privacy mechanisms.

AOI Analysis The impact of privacy mechanisms on area of interest (AOI) analysis is measured as the Root Mean Squared Error (RMSE) between AOI metrics. There are several popular AOI metrics, suitable for different analyses, such as number of visits to an AOI [103], time to first fixation, and number of visits to an AOI [43]. For an overview of AOI analysis, see the discussion by Le Meur and Baccino [55]. For an investigation into privacy mechanisms, we select Dwell Time as a representative AOI metric. Dwell time is the amount of time spent by a user on an AOI, computed as the sum of the durations of all the fixations inside that AOI. The key logical operation is checking whether a fixation location falls within the bounding box that demarcates the AOI, which is the typical first step in all AOI metrics.

If the fixation location is perturbed, such as with the privacy mechanisms proposed above, then we can anticipate an error being introduced in the dwell time computation. We report the RMSE computed between AOI Dwell Time for each individual on the original dataset and after privacy mechanisms are applied, averaged across all stimuli in the dataset. RMSE in dwell time computation for additive Gaussian noise and temporal downsampling is below 40ms (Tables 4 and 5), which is insignificant for the practical application of AOI metrics, as a fixation itself typically lasts 200ms [91, 105]. However, for spatial downsampling, an RMSE of 247ms is introduced, which is greater than the length of one visual fixation. While being a few fixations off on average may not have a large effect on AOI applications such as evidence-based user experience design, it may be noticeable in scenarios with multiple small AOIs close together, such as figuring out which car the user spent longest looking at on a virtual visit to a car dealership.

Saliency Map Generation Saliency maps represent a spatial probability distribution of attention over an image or video. Maps are generated by aggregating fixations from eye-tracking data of multiple observers to highlight regions that attract the most attention in the stimulus [46]. Saliency maps are used directly for gaze prediction [24] and to opti-

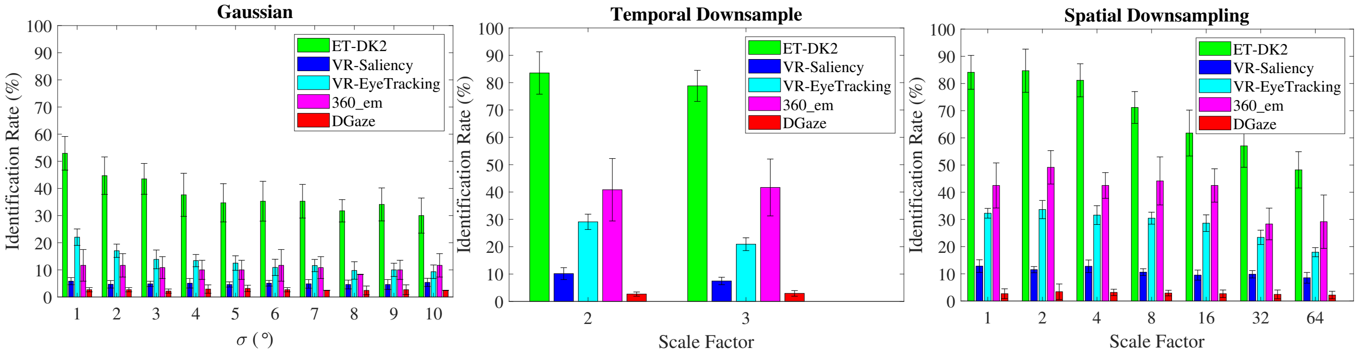

Fig. 4: Mean and standard deviation of identification rate for each privacy mechanism with different internal parameters. Gaussian noise generates the lowest observed identification rates across all datasets, while temporal downsampling has the least impact.

Table 4: This table illustrates the impact of introducing the Gaussian Noise privacy mechanism on the identification rate as well as on three use cases. The reported numbers are for $\sigma = 10°$. The second column shows how the identification rate falls after the privacy mechanism is applied. The fourth column reports an error metric that is relevant to that use case.

Mechanism	Identif. Rate	Utility	Impact on Utility	Dataset
Gaussian Noise	3% → 2%	Gaze Prediction	Avg. Prediction Error Difference = 1.14°	DGaze (Re-trained)
Gaussian Noise	85% → 30%	AOI Analysis	Dwell Time RMSE = 0.0359s	ET-DK2 (360° images)
Gaussian Noise	33% → 9%	Generate Saliency Map	KL-Divergence = 0.0367	VR-EyeTracking (360° videos)

Table 5: This table illustrates the impact of introducing the Temporal Downsample privacy mechanism on the identification rate as well as on three use cases. The reported numbers are for $K = 3$. The second column shows how the identification rate falls after the privacy mechanism is applied. The fourth column reports an error metric that is relevant to that use case.

Mechanism	Identif. Rate	Utility	Impact on Utility	Dataset
Temporal Downsample	3% → 3%	Gaze Prediction	Avg. Prediction Error Difference = 0.22°	DGaze (Not Re-trained)
Temporal Downsample	85% → 79%	AOI Analysis	Dwell Time RMSE = 0.006s	ET-DK2 (360° images)
Temporal Downsample	9% → 7%	Generate Saliency Map	KL-Divergence = 0.0019	VR-Saliency (360° images)

Table 6: The lowest achievable identification rate (IR) for the Spatial Downsample was at $L = 64$, and the corresponding impact on utility are reported below. The arrow indicates the IR before and after the privacy mechanism is applied.

Mechanism	Identif. Rate	Utility	Impact on Utility	Dataset
Spatial Downsample	3% → 2%	Gaze Prediction	Avg. Prediction Error Difference = 0.51°	DGaze (Re-trained)
Spatial Downsample	85% → 48%	AOI Analysis	Dwell Time RMSE = 0.2473s	ET-DK2 (360° images)
Spatial Downsample	47% → 29%	Generate Saliency Map	KL-Divergence = 0.1293	360_em (360° videos)

mize streaming [63, 101] or rendering [62]. We compute error as the KL-Divergence between a saliency map generated from the original gaze data and the saliency map generated by gaze data after the privacy mechanisms have been applied. KL-Divergence measures the relative entropy between the two saliency maps and is commonly used in loss functions to train deep saliency prediction models and to evaluate learned models [21, 24, 39, 55]. The spatial errors introduced by the privacy mechanism may cause regions highlighted by the saliency map to shift or spread out, leading to larger KL-Divergence values. A recent survey revealed the best performing model in predicting human fixations produced a KL-Divergence of 0.48 for the MIT300 dataset, with baseline models producing values of 1.24 or higher [14]. We observed that spatial downsampling produces the largest KL-Divergence on average of 0.1293, while Gaussian and temporal downsampling mechanisms produces much smaller values of 0.0367 and 0.0019 respectively. Spatial downsampling introduced errors that are approximately a fourth of the existing gap in fixation prediction. Errors of this magnitude will cause saliency maps generated from spatially downsampled gaze data to deviate from ground truth, and negatively impact performance of models that use the maps for training.

7 Conclusions and Future Work

As eye-tracking technology is built into mixed reality devices, they open up possibilities for violating user privacy. In this paper, we have examined a specific threat to user privacy: unique user identification based on their eye movement data. This identification would enable colluding applications to connect a user logged in "anonymously" with their work ID, for example.

We first determine biometric identification rates across five datasets of eye movements in immersive environments. We show that identification rates can reach as high as 85% depending on the type of stimulus used to elicit the eye movements, and the amount of eye movement data collected in total. Our highest identification rates were achieved when viewing many 360° images with short duration (ET-DK2), with all datasets having an identification rate higher than chance except DGaze. We hypothesize this is the result of the DGaze dataset providing viewers only two scenes to explore, containing sparse environments with animals that they can follow around by using teleporting to navigate. In the context of saliency Borji [14] describes the role that stimuli plays in eye movements elicited by viewers, suggesting that datasets from more diverse stimuli is needed to improve generalized performance of saliency prediction models. In the context of privacy, this suggests that the presence of biometric features within gaze data collected in environments differs for photorealistic, static, and dynamic stimuli. Given enough eye movement data collected from the right stimuli, there is an appreciable risk for identification.

We propose a *Gatekeeper* model to alleviate biometric authentication by apps that need AOI metrics or event specific data for their utility. This model provides API calls that return desired metrics and summary information of fixation and saccades to applications without providing streams of raw gaze data, which suffices for certain classes of mixed reality use cases. However, in the case of use cases such as foveated rendering, streaming gaze data is required. We propose that in this case, privacy mechanisms be applied to the raw data stream to reduce identification rate, while maintaining the utility needed for the given application. We evaluated three privacy mechanisms: additive Gaussian noise, temporal downsampling, and spatial downsampling. Our best results used additive Gaussian noise to reduce an identification rate of 85% to 30% while supporting AOI analysis, gaze prediction, and saliency map generation.

Implications Imagine the scenario described earlier of a worker that anonymously attends labor union meetings as User X. The eye-tracking data collected during a VR union meeting attended by User X is exposed through a database breach or collusion with the employer, who then discovers a match between User X and their real identity at a rate greater than chance. Even though they were not the only worker to attend this meeting, biometric data suggested they were the most likely employee to have attended, turning User X into a scapegoat for the entire group. The individual may then have their reputation tarnished in retaliation by their employer. Our investigations are a first step towards protecting such a user. Though the proposed mechanisms lower identification rates, they do not eliminate the possibility of weak identification. More work is needed to create and evaluate mechanisms that allow users, organizations, and platforms to trust eye tracking, and more broadly, behavioral tracking, within mixed reality use cases.

Limitations Our threat model assumes a trusted platform. In cases where the platform itself cannot be trusted, there is a need for user-implementable solutions, similar in spirit to the user-implementable optical defocus in [42]. Our characterization of the proposed privacy mechanisms is based on one biometric authentication approach (RBFN). As newer methods are developed, we will likely need new privacy mechanisms that can applied as a software patch for the mixed reality headset. This work also considers each privacy mechanism individually. We expect there will be greater gains in terms of privacy when applying a combination of different privacy mechanisms.

Future Work In addition to exploring combinations of privacy mechanisms, future work might draw inspiration from research in location privacy, and investigate adapting location k-anonymity schemes for gaze [36]. It would also be interesting to characterize stimuli as being dangerous from the perspective of biometric signatures, akin to "click-bait". More broadly, while our work considers the user privacy, future work might also consider security from a platform's perspective. Consider the case of an attacker injecting gaze positions to fool an AOI metric into thinking that an AOI has been glanced at (for monetization of advertisements). One potential solution to this problem is direct anonymous attestation in a trusted platform module (TPM) to assure gaze consumers that there have been no injections.

Acknowledgments

Authors acknowledge funding from the National Science Foundation (Awards FWHTF-2026540, CNS-1815883, and CNS-1562485), the National Science Foundation GRFP (Awards DGE-1315138 and DGE-1842473), and the Air Force Office of Scientific Research (Award FA9550-19-1-0169).

References

[1] I. Agtzidis, M. Startsev, and M. Dorr. A ground-truth data set and a classification algorithm for eye movements in 360-degree videos. *arXiv preprint arXiv:1903.06474*, 2019.

[2] R. Alghofaili, M. S. Solah, and H. Huang. Optimizing visual element placement via visual attention analysis. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 464–473. IEEE, 2019.

[3] E. Arabadzhiyska, O. T. Tursun, K. Myszkowski, H.-P. Seidel, and P. Didyk. Saccade landing position prediction for gaze-contingent rendering. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[4] W. A. Arbaugh, D. J. Farber, and J. M. Smith. A secure and reliable bootstrap architecture. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pp. 65–71, 1997.

[5] T. Armstrong and B. O. Olatunji. Eye tracking of attention in the affective disorders: A meta-analytic review and synthesis. *Clinical psychology review*, 32(8):704–723, 2012.

[6] M. Assens, X. Giro-i Nieto, K. McGuinness, and N. E. O'Connor. Pathgan: visual scanpath prediction with generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

[7] R. Bailey, A. McNamara, A. Costello, S. Sridharan, and C. Grimm. Impact of subtle gaze direction on short-term spatial information recall. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 67–74, 2012.

[8] R. Bailey, A. McNamara, N. Sudarsanam, and C. Grimm. Subtle gaze direction. *ACM Transactions on Graphics (TOG)*, 28(4):1–14, 2009.

[9] Y. Bar-Haim, T. Ziv, D. Lamy, and R. M. Hodes. Nature and nurture in own-race face processing. *Psychological science*, 17(2):159–163, 2006.

[10] B. Bastani, E. Turner, C. Vieri, H. Jiang, B. Funt, and N. Balram. Foveated pipeline for AR/VR head-mounted displays. *Information Display*, 33(6):14–35, 2017.

[11] J. K. Bennett, S. Sridharan, B. John, and R. Bailey. Looking at faces: autonomous perspective invariant facial gaze analysis. In *Proceedings of the ACM Symposium on Applied Perception*, pp. 105–112. ACM, 2016.

[12] T. Booth, S. Sridharan, A. McNamara, C. Grimm, and R. Bailey. Guiding attention in controlled real-world environments. In *Proceedings of the ACM Symposium on Applied Perception*, pp. 75–82, 2013.

[13] Z. Boraston and S.-J. Blakemore. The application of eye-tracking technology in the study of autism. *The Journal of physiology*, 581(3):893–898, 2007.

[14] A. Borji. Saliency prediction in the deep learning era: Successes and limitations. *IEEE TPAMI*, 2019.

[15] E. Bozkir, O. Günlü, W. Fuhl, R. F. Schaefer, and E. Kasneci. Differential privacy for eye tracking with temporal correlations. *arXiv preprint arXiv:2002.08972*, 2020.

[16] E. Bozkir, A. B. Ünal, M. Akgün, E. Kasneci, and N. Pfeifer. Privacy preserving gaze estimation using synthetic images via a randomized encoding based framework. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 1–5, 2020.

[17] M. Bradley. Natural selective attention: orienting and emotion. *Psychophysiology*, 46(1):1–11, 2009.

[18] J. Bradshaw, F. Shic, A. N. Holden, E. J. Horowitz, A. C. Barrett, T. C. German, and T. W. Vernon. The use of eye tracking as a biomarker of treatment outcome in a pilot randomized clinical trial for young children with autism. *Autism Research*, 12(5):779–793, 2019.

[19] A. Burova, J. Mäkelä, J. Hakulinen, T. Keskinen, H. Heinonen, S. Siltanen, and M. Turunen. Utilizing vr and gaze tracking to develop ar solutions for industrial maintenance. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020.

[20] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan. Viewport-driven rate-distortion optimized 360º video streaming. In *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE, 2018.

[21] F.-Y. Chao, L. Zhang, W. Hamidouche, and O. Deforges. Salgan360: Visual saliency prediction on 360 degree images with generative adver-

sarial networks. In *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 01–04. IEEE, 2018.

[22] A. K. Chaudhary and J. B. Pelz. Privacy-preserving eye videos using rubber sheet model. In *ACM Symposium on Eye Tracking Research & Applications*, pp. 1–5, 2020.

[23] K. Chawarska and F. Shic. Looking but not seeing: Atypical visual scanning and recognition of faces in 2 and 4-year-old children with autism spectrum disorder. *Journal of autism and developmental disorders*, 39(12):1663, 2009.

[24] D. Chen, C. Qing, X. Xu, and H. Zhu. Salbinet360: Saliency prediction on 360° images with local-global bifurcated deep network. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 92–100. IEEE, 2020.

[25] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang. When machine unlearning jeopardizes privacy. *arXiv*, pp. arXiv–2005, 2020.

[26] S. Cho, S.-w. Kim, J. Lee, J. Ahn, and J. Han. Effects of volumetric capture avatars on social presence in immersive virtual environments. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 26–34. IEEE, 2020.

[27] K. M. Dalton, B. M. Nacewicz, T. Johnstone, H. S. Schaefer, M. A. Gernsbacher, H. H. Goldsmith, A. L. Alexander, and R. J. Davidson. Gaze fixation and the neural circuitry of face processing in autism. *Nature neuroscience*, 8(4):519–526, 2005.

[28] E. J. David, J. Gutiérrez, A. Coutrot, M. P. Da Silva, and P. L. Callet. A dataset of head and eye movements for 360 videos. In *Proceedings of the 9th ACM Multimedia Systems Conference*, pp. 432–437. ACM, 2018.

[29] A. T. Duchowski, K. Krejtz, I. Krejtz, C. Biele, A. Niedzielska, P. Kiefer, M. Raubal, and I. Giannopoulos. The index of pupillary activity: Measuring cognitive load vis-à-vis task difficulty with pupil oscillation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2018.

[30] A. T. Duchowski, V. Shivashankaraiah, T. Rawls, A. K. Gramopadhye, B. J. Melloy, and B. Kanki. Binocular eye tracking in virtual reality for inspection training. In *Proceedings of the Symposium on Eye Tracking Research & Applications*, pp. 89–96, 2000.

[31] S. Eberz, G. Lovisotto, K. B. Rasmussen, V. Lenders, and I. Martinovic. 28 blinks later: Tackling practical challenges of eye movement biometrics. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1187–1199, 2019.

[32] W. Fuhl. Reinforcement learning for the manipulation of eye tracking data. *arXiv preprint arXiv:2002.06806*, 2020.

[33] C. Galdi, M. Nappi, D. Riccio, and H. Wechsler. Eye movement analysis for human authentication: a critical survey. *Pattern Recognition Letters*, 84:272–283, 2016.

[34] C. Gebhardt, B. Hecox, B. van Opheusden, D. Wigdor, J. Hillis, O. Hilliges, and H. Benko. Learning cooperative personalized policies from gaze data. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 197–208, 2019.

[35] A. George and A. Routray. A score level fusion method for eye movement biometrics. *Pattern Recognition Letters*, 82:207–215, 2016.

[36] A. Gkoulalas-Divanis, P. Kalnis, and V. S. Verykios. Providing k-anonymity in location based services. *ACM SIGKDD explorations newsletter*, 12(1):3–10, 2010.

[37] R. Graham, A. Hoover, N. A. Ceballos, and O. Komogortsev. Body mass index moderates gaze orienting biases and pupil diameter to high and low calorie food images. *Appetite*, 56(3):577–586, 2011.

[38] S. Grogorick, M. Stengel, E. Eisemann, and M. Magnor. Subtle gaze guidance for immersive environments. In *Proceedings of the ACM Symposium on Applied Perception*, pp. 1–7, 2017.

[39] J. Gutiérrez, E. J. David, A. Coutrot, M. P. Da Silva, and P. Le Callet. Introducing un salient360! benchmark: A platform for evaluating visual attention models for 360 contents. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–3. IEEE, 2018.

[40] Z. Hu, S. Li, C. Zhang, K. Yi, G. Wang, and D. Manocha. Dgaze: Cnn-based gaze prediction in dynamic scenes. *IEEE transactions on visualization and computer graphics*, 26(5):1902–1911, 2020.

[41] Z. Hu, C. Zhang, S. Li, G. Wang, and D. Manocha. SGaze: A data-driven eye-head coordination model for realtime gaze prediction. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2002–2010, 2019.

[42] B. John, S. Jorg, S. Koppal, and E. Jain. The security-utility trade-off for iris authentication and eye animation for social virtual avatars. *IEEE transactions on visualization and computer graphics*, 2020.

[43] B. John, S. Kalyanaraman, and E. Jain. Look out! a design framework for safety training systems a case study on omnidirectional cinemagraphs. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 147–153. IEEE, 2020.

[44] B. John, S. Koppal, and E. Jain. EyeVEIL: degrading iris authentication in eye tracking headsets. In *ACM Symposium on Eye Tracking Research & Applications*, p. 37. ACM, 2019.

[45] B. John, A. Liu, L. Xia, S. Koppal, and E. Jain. Let it snow: Adding pixel noise to protect the user's identity. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 1–3, 2020.

[46] B. John, P. Raiturkar, O. Le Meur, and E. Jain. A benchmark of four methods for generating 360° saliency maps from eye tracking data. *International Journal of Semantic Computing*, 13(03):329–341, 2019.

[47] P. Kasprowski and K. Harezlak. The second eye movements verification and identification competition. In *IEEE International Joint Conference on Biometrics*, pp. 1–6. IEEE.

[48] P. Kasprowski, O. V. Komogortsev, and A. Karpov. First eye movement verification and identification competition at btas 2012. In *2012 IEEE fifth international conference on biometrics: theory, applications and systems (BTAS)*, pp. 195–202. IEEE, 2012.

[49] M. Keyvanara and R. Allison. Transsaccadic awareness of scene transformations in a 3d virtual environment. In *ACM Symposium on Applied Perception 2019*, pp. 1–9, 2019.

[50] M. Keyvanara and R. Allison. Effect of a constant camera rotation on the visibility of transsaccadic camera shifts. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 1–8, 2020.

[51] J. L. Kröger, O. H.-M. Lutz, and F. Müller. What does your gaze reveal about you? on the privacy implications of eye tracking. In *IFIP International Summer School on Privacy and Identity Management*, pp. 226–241. Springer, 2019.

[52] B. Laeng and L. Falkenberg. Women's pupillary responses to sexually significant others during the hormonal cycle. *Hormones and behavior*, 52(4):520–530, 2007.

[53] P. Lang, M. Greenwald, M. M. Bradley, and A. O. Hamm. Looking at pictures: affective, facial, visceral, and behavioral reactions. *Psychophysiology*, 30(3):261–73, 1993.

[54] E. Langbehn, F. Steinicke, M. Lappe, G. F. Welch, and G. Bruder. In the blink of an eye: Leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality. *ACM Transactions on Graphics (TOG)*, 37(4):66, 2018.

[55] O. Le Meur and T. Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013.

[56] R. J. Leigh and D. S. Zee. *The neurology of eye movements*. Oxford University Press, USA, 2015.

[57] C. Li, M. Xu, X. Du, and Z. Wang. Bridge the gap between vqa and human behavior on omnidirectional video: A large-scale dataset and a deep learning model. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 932–940, 2018.

[58] J. Li, A. R. Chowdhury, K. Fawaz, and Y. Kim. Kalεido: Real-time privacy control for eye-tracking systems. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.

[59] A. Liu, L. Xia, A. Duchowski, R. Bailey, K. Holmqvist, and E. Jain. Differential privacy for eye-tracking data. In *ACM Symposium on Eye Tracking Research & Applications*, p. 28. ACM, 2019.

[60] D. J. Lohr, S. Aziz, and O. Komogortsev. Eye movement biometrics using a new dataset collected in virtual reality. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 1–3, 2020.

[61] S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)*, 37(4):68, 2018.

[62] P. Longhurst, K. Debattista, and A. Chalmers. A gpu based saliency map for high-fidelity selective rendering. In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pp. 21–29, 2006.

[63] P. Lungaro, R. Sjöberg, A. J. F. Valero, A. Mittal, and K. Tollmar. Gaze-aware streaming solutions for the next generation of mobile VR experiences. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1535–1544, 2018.

[64] A. MacQuarrie and A. Steed. Perception of volumetric characters' eye-gaze direction in head-mounted displays. In *Proceedings of 2019 IEEE*

Virtual Reality (VR), vol. 2019. IEEE, 2019.

[65] C. Marforio, H. Ritzdorf, A. Francillon, and S. Capkun. Analysis of the communication between colluding applications on modern smartphones. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 51–60, 2012.

[66] X. Meng, R. Du, and A. Varshney. Eye-dominance-guided foveated rendering. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1972–1980, 2020.

[67] X. Meng, R. Du, M. Zwicker, and A. Varshney. Kernel foveated rendering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–20, 2018.

[68] J. V. Monaco. Classification and authentication of one-dimensional behavioral biometrics. In *IEEE International Joint Conference on Biometrics*, pp. 1–8. IEEE, 2014.

[69] J. Morris, S. Smalley, and G. Kroah-Hartman. Linux security modules: General security support for the linux kernel. In *Proceedinsg of the 2002 USENIX Security Symposium*, 2002.

[70] C. Mousas, A. Koilias, D. Anastasiou, B. Hekabdar, and C.-N. Anagnostopoulos. Effects of self-avatar and gaze on avoidance movement behavior. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 726–734. IEEE, 2019.

[71] J. H. Mueller, P. Voglreiter, M. Dokter, T. Neff, M. Makar, M. Steinberger, and D. Schmalstieg. Shading atlas streaming. In *SIGGRAPH Asia 2018 Technical Papers*, p. 199. ACM, 2018.

[72] P. Mundy. A review of joint attention and social-cognitive brain systems in typical development and autism spectrum disorder. *European Journal of Neuroscience*, 47(6):497–514, 2018.

[73] D. Munoz, J. Broughton, J. Goldring, and I. Armstrong. Age-related performance of human subjects on saccadic eye movement tasks. *Experimental brain research*, 121(4):391–400, 1998.

[74] M. Murcia-López, T. Collingwoode-Williams, W. Steptoe, R. Schwartz, T. J. Loving, and M. Slater. Evaluating virtual reality experiences through participant choices. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 747–755. IEEE, 2020.

[75] J. Orlosky, Y. Itoh, M. Ranchet, K. Kiyokawa, J. Morgan, and H. Devos. Emulation of physician tasks in eye-tracked virtual reality for remote diagnosis of neurodegenerative disease. *IEEE Transactions on Visualization and Computer Graphics*, 23(4):1302–1311, 2017.

[76] J. L. Orquin, N. J. Ashby, and A. D. Clarke. Areas of interest as a signal detection problem in behavioral eye-tracking research. *Journal of Behavioral Decision Making*, 29(2-3):103–115, 2016.

[77] Y. S. Pai, B. I. Outram, B. Tag, M. Isogai, D. Ochi, and K. Kunze. Gazesphere: Navigating 360-degree-video environments in VR using head rotation and eye gaze. In *ACM SIGGRAPH 2017 Posters*, p. 23. ACM, 2017.

[78] G. Papaioannou and I. Koutsopoulos. Tile-based caching optimization for 360 videos. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 171–180, 2019.

[79] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35(6):179, 2016.

[80] K. A. Pelphrey, J. P. Morris, and G. McCarthy. Neural basis of eye gaze processing deficits in autism. *Brain*, 128(5):1038–1048, 2005.

[81] Y. Rahman, S. M. Asish, N. P. Fisher, E. C. Bruce, A. K. Kulshreshth, and C. W. Borst. Exploring eye gaze visualization techniques for identifying distracted students in educational vr. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 868–877. IEEE, 2020.

[82] Y. Rai, J. Gutiérrez, and P. Le Callet. A dataset of head and eye movements for 360 degree images. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pp. 205–210. ACM, 2017.

[83] P. Raiturkar, A. Kleinsmith, A. Keil, A. Banerjee, and E. Jain. Decoupling light reflex from pupillary dilation to measure emotional arousal in videos. In *Proceedings of the ACM Symposium on Applied Perception*, pp. 89–96, 2016.

[84] V. Rajanna and J. P. Hansen. Gaze typing in virtual reality: impact of keyboard design, selection method, and motion. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, p. 15. ACM, 2018.

[85] G. Rieger, B. M. Cash, S. M. Merrill, J. Jones-Rounds, S. M. Dharmavaram, and R. C. Savin-Williams. Sexual arousal: The correspondence of eyes and genitals. *Biological Psychology*, 104:56–64, 2015.

[86] I. Rigas, E. Abdulin, and O. Komogortsev. Towards a multi-source fusion approach for eye movement-driven recognition. *Information Fusion*, 32:13–25, 2016.

[87] I. Rigas and O. V. Komogortsev. Current research in eye movement biometrics: An analysis based on bioeye 2015 competition. *Image and Vision Computing*, 58:129–141, 2017.

[88] S. Rothe, F. Althammer, and M. Khamis. Gazerecall: Using gaze direction to increase recall of details in cinematic virtual reality. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*, pp. 115–119, 2018.

[89] S. Rothe, D. Buschek, and H. Hußmann. Guidance in cinematic virtual reality-taxonomy, research status and challenges. *Multimodal Technologies and Interaction*, 3(1):19, 2019.

[90] R. Sailer, X. Zhang, T. Jaeger, and L. Van Doorn. Design and Implementation of a TCG-based Integrity Measurement Architecture. In *Proceedings of the 2004 USENIX Security Symposium*, 2004.

[91] D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the Symposium on Eye Tracking Research & Applications*, pp. 71–78, 2000.

[92] N. Sammaknejad, H. Pouretemad, C. Eslahchi, A. Salahirad, and A. Alinejad. Gender classification based on eye movements: A processing effect during passive face viewing. *Advances in cognitive psychology*, 13(3):232, 2017.

[93] C. Schröder, S. M. K. Al Zaidawi, M. H. Prinzler, S. Maneth, and G. Zachmann. Robustness of eye movement biometrics against varying stimuli and varying trajectory length. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–7, 2020.

[94] F. Schwenker, H. A. Kestler, and G. Palm. Three learning phases for radial-basis-function networks. *Neural networks*, 14(4-5):439–458, 2001.

[95] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein. Saliency in VR: How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1633–1642, 2018.

[96] S. Sridharan, R. Bailey, A. McNamara, and C. Grimm. Subtle gaze manipulation for improved mammography training. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 75–82, 2012.

[97] J. Steil, I. Hagestedt, M. X. Huang, and A. Bulling. Privacy-aware eye tracking using differential privacy. In *ACM Symposium on Eye Tracking Research & Applications*. ACM, 2019.

[98] J. Steil, M. Koelle, W. Heuten, S. Boll, and A. Bulling. Privaceye: privacy-preserving head-mounted eye tracking using egocentric scene image and eye movement features. In *ACM Symposium on Eye Tracking Research & Applications*, p. 26. ACM, 2019.

[99] Q. Sun, A. Patney, L.-Y. Wei, O. Shapira, J. Lu, P. Asente, S. Zhu, M. Mcguire, D. Luebke, and A. Kaufman. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM Transactions on Graphics (TOG)*, 37(4):67, 2018.

[100] S. Uzzaman and S. Joordens. The eyes know what you are thinking: eye movements as an objective measure of mind wandering. *Consciousness and cognition*, 20(4):1882–1886, 2011.

[101] M. Xu, C. Li, S. Zhang, and P. Le Callet. State-of-the-art in 360 video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):5–26, 2020.

[102] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao. Gaze prediction in dynamic 360° immersive videos. In *Proceedings of IEEE CVPR 2018*, pp. 5333–5342, 2018.

[103] C. Yangandul, S. Paryani, M. Le, and E. Jain. How many words is a picture worth? attention allocation on thumbnails versus title text regions. In *ACM Symposium on Eye Tracking Research & Applications*, pp. 1–5, 2018.

[104] R. Zemblys and O. Komogortsev. Developing photo-sensor oculography (PS-OG) system for virtual reality headsets. In *ACM Symposium on Eye Tracking Research & Applications*, p. 83. ACM, 2018.

[105] R. Zemblys, D. C. Niehorster, O. Komogortsev, and K. Holmqvist. Using machine learning to detect events in eye-tracking data. *Behavior research methods*, 50(1):160–181, 2018.

[106] A. T. Zhang and B. O. Le Meur. How old do you look? inferring your age from your gaze. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2660–2664. IEEE, 2018.

[107] G. Zhang and J. P. Hansen. Accessible control of telepresence robots based on eye tracking. In *ACM Symposium on Eye Tracking Research & Applications*, p. 50. ACM, 2019.