Information Sciences 675 (2024) 120720

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

DPGazeSynth: Enhancing eye-tracking virtual reality privacy with differentially private data synthesis

Xiaojun Ren [a,b,1], Jiluan Fan [a,b,1], Ning Xu [a,b], Shaowei Wang [a,*], Changyu Dong [a], Zikai Wen [c,*]

[a] *Institute of Artificial Intelligence, Guangzhou University, China*
[b] *Guangdong Provincial Key Laboratory of Blockchain Security, China*
[c] *Department of Computer Science, Virginia Tech, USA*

ARTICLE INFO

Keywords:
Eye-tracking
Data synthesis
Differential privacy

ABSTRACT

As spatial computing devices increasingly integrate eye-tracking technology to enhance virtual reality (VR) experiences, the imperative to protect sensitive eye-tracking data against privacy risks, such as user re-identification, has become more pressing. Existing privacy-preserving mechanisms face challenges in balancing the dual demands of privacy and utility in the context of VR applications. This paper presents *DPGazeSynth*, a novel framework designed to fortify privacy protections while ensuring the utility of eye-tracking data. *DPGazeSynth* addresses the unique requirements of gaze path synthesis, especially the differentiation between fixations and saccades. Our approach introduces a semi-synthetic method based on the Markov Chain model to accurately maintain data correlations for analytical tasks. We demonstrate that *DPGazeSynth* provides robust differential privacy guarantees, and our comprehensive experiments on two real-world datasets validate its effectiveness in safeguarding against re-identification attacks. The results showcase *DPGazeSynth*'s better performance over existing solutions like Kalεido and establish its potential as a reliable foundation for future research aimed at reconciling privacy concerns with the demands of complex trajectory data analysis in eye-tracking applications.

1. Introduction

Recent advancements in spatial computing devices, such as Apple Vision Pro and Meta Quest, have seamlessly integrated eye-tracking technology to enrich interactive experiences, especially within Virtual Reality (VR) environments [1,2]. While these innovations promise more stable user interactions in VR, they also bring to the forefront the pressing issue of safeguarding the privacy of eye-tracking data, given its inherent sensitivity and the potential risks associated with user re-identification [3–5].

In response to these privacy concerns, researchers have made concerted efforts to devise privacy-preserving mechanisms tailored for eye-tracking data [6–9]. Among these, Kalεido [9] stands out as an influential solution, leveraging the (ε, w, r)-differential privacy mechanism, which combines principles of geo-indistinguishability [10] and w-event differential privacy [11]. Nevertheless, it is crucial to acknowledge that Kalεido, while effective in conventional scenarios, exhibits limitations when it comes to machine learning analysis of human visual behavior in VR applications [12].

* Corresponding authors.
E-mail addresses: wangsw@gzhu.edu.cn (S. Wang), zkwen@vt.edu (Z. Wen).
[1] Jiluan Fan and Xiaojun Ren contributed equally in this paper.

https://doi.org/10.1016/j.ins.2024.120720
Received 16 January 2024; Received in revised form 2 March 2024; Accepted 5 May 2024

Available online 8 May 2024

Table 1
Comparison of Privacy and Utility Guarantees for Privacy-preserving Eye-tracking.

Algorithm	Privacy Mechanism	Formal Guarantee	Dynamic Data Support	VR Analysis Utility
k-synth-PD [8]	k-anonymity, γ-PD	No	Yes	Yes
Exponential-DP [7]	ϵ-DP	Yes	No	No
Gaussian [6]	(ϵ, δ)-DP	Yes	No	No
Kalϵido [9]	(ϵ, w, r)-DP	Yes	Yes	No
DPGazeSynth (Ours)	(ϵ, w, r)-DP	Yes	Yes	Yes

In light of these challenges, our paper introduces *DPGazeSynth*, a new approach that seeks to offer more robust privacy protection for the collection and analysis of eye-tracking data. Unlike existing methods primarily centered on synthesizing geographical trajectories [13–15], *DPGazeSynth* focuses on addressing the distinct challenges associated with synthesizing gaze paths, with a particular emphasis on distinguishing between fixations and saccades in eye-tracking data.

Throughout this paper, we propose, evaluate, and showcase the advances of *DPGazeSynth* over Kalϵido in terms of privacy protection and utility analysis. Our main contributions include:

1. The introduction of a semi-synthetic method for eye-tracking data, leveraging the Markov Chain model to preserve correlations and enhance accuracy in analytical tasks.
2. The presentation of the *DPGazeSynth* framework, which provides formal differential privacy guarantees for the collection and analysis of eye-tracking data.
3. The validation of *DPGazeSynth*'s effectiveness, which was conducted through experiments on two real-world datasets.

2. Related work

This section reviews the current landscape of eye-tracking technology, focusing on three key areas: the advancements and applications of eye-tracking, the privacy challenges it presents, the methods developed for privacy-preserving eye-tracking data, and their limitations in supporting human visual behavior analysis for VR applications.

2.1. Advancements and applications in eye-tracking technology

A decade ago, Ye et al. [16] designed a system to detect eye contact between an adult and a child using gaze-tracking glasses. Since then, eye-tracking technology's integration into VR has been recognized for enhancing user experiences [17,18]. Extensive research has established the value of eye-tracking data in analyzing human behavior and cognitive functions [19–24]. Hu et al. investigated visual attention in VR tasks and task recognition using a CNN+BiGRU framework [22]. Kham et al. demonstrated the application of eye tracking in mixed reality driver assistance systems, reducing traffic accident risks associated with driver distraction [23]. Kang et al. combined eye-tracking with electroencephalography to identify children with autism using Support Vector Machines [24]. These studies underscore the significance of eye-tracking data across diverse applications.

2.2. Privacy concerns and protection mechanisms in eye-tracking

However, the increased availability of eye-tracking technologies raises substantial privacy concerns. Researchers have explored biometric identification using eye-tracking data [25–28]. For example, Kasprowski and Ober [25] examined eye movements as a biometric, applying various classifiers to establish the viability of eye-tracking for individual identification. Komogortsev et al. [26] refined the precision of eye movement modeling by using the Oculomotor Plant Mathematical Model to map the dynamics of eye movements. Schröder et al. [27] applied the Indentification by Velocity Threshold (IVT) algorithm to differentiate fixations and saccades in eye-tracking data, which led to more accurate user identification.

As shown in Table 1, researchers have devised various methods to provide privacy protection for eye-tracking data. David-John et al. suggested k-anonymity and plausible deniability as defenses against re-identification attacks [8]. Although their methods were tested in the VR setting, these methods can not provide a formal privacy guarantee leading to potential privacy leakage. Differential Privacy (DP) has been applied to eye-tracking data, ensuring that the inclusion or exclusion of any single user's data does not significantly influence the output of any analysis, thus protecting individual user information from inference attacks [29]. Liu et al. proposed a Gaussian noise mechanism to secure eye-tracking heatmaps [6], while Steil et al. applied DP noise to various eye-tracking features, albeit restricting access to noisy feature sets instead of raw samples [7]. Kalϵido is recognized as a leading solution for privacy in eye-tracking data [9]; however, these DP-based methods have focused on smaller datasets and not specifically on VR contexts. Our research diverges by addressing re-identification challenges within larger eye-tracking datasets for machine learning analysis in VR.

2.3. Challenges in eye-tracking data synthesis

Data synthesis aims to generate new data that statistically resembles the original dataset while ensuring privacy [30]. This approach, if designed properly, often results in data that maintains more of the original structure and statistical properties compared to adding noise directly to the raw data. The synthetic data can be more useful for machine learning analysis, as it tends to retain key characteristics and patterns of the original dataset.

Eye-tracking paths are akin to trajectories, and preserving the privacy of such data has been a focal point of recent studies. Researchers have delved into creating synthetic location trajectories that offer realistic representations and privacy safeguards [13,14, 31]. Gursoy et al. introduced AdaTrace, a complex four-phase synthesis process for creating differentially private location traces [31]. Wang et al. developed PrivTrace, which adaptively selects crucial Markov transitions for trajectory synthesis [14], while Du et al. proposed LDPTrace, applying local differential privacy to key trajectory patterns using a Markov Chain model [13].

Although these trajectory synthesis methods are well-established for location data, applying them to eye-tracking data is not straightforward. Eye movement trajectories are unique due to their encompassment of both temporal and spatial aspects, posing a significant challenge for directly transferring geographic trajectory synthesis techniques to eye-tracking contexts.

The field of eye-tracking data synthesis has attracted considerable research interest, with various approaches being explored [32–36]. Martin et al. introduced a generative adversarial network for synthesizing gaze data [32]. However, the model was tailored to datasets with low sampling rates, which are not well-suited for capturing quick eye movements like saccades. Assens et al. and Lan et al. developed EyeSyn, which leverages publicly available images and videos to generate eye movement datasets without using actual eye samples [33]. Despite its innovative approach, it falls short of emulating the complex aspects of human visual behavior for machine learning analysis in VR.

In efforts to refine synthesis models, some researchers have incorporated additional data, such as related images [34,35], and considered head orientation at each time step [36]. However, these approaches typically depended on a trusted curator, a setup that is impractical for many real-world scenarios involving data sharing with untrusted vendors. Therefore, we managed to design a local DP algorithm that does not require a trusted data curator and outperforms the leading solution for privacy-preserving eye tracking data, Kalεido, in the human visual behavior analysis for VR applications.

3. Preliminaries

In this section, we revisit the fundamental concepts and theoretical frameworks central to our research. These foundational elements provide the necessary context for the design and validation of our gaze path synthesis algorithm.

3.1. Eye tracking data properties

Eye tracking data is commonly depicted as a scan path, which acts as a sequential depiction to highlight the characteristics and patterns of a user's visual attention [19]. A scan path consists of a chronological series of fixations, periods where the gaze remains relatively steady and interspersed with saccades, the quick eye movements that bridge these fixations [37]. Fixations are defined as dense clusters of gaze points within particular areas of the visual field, denoting intense focus on specific objects or areas of interest. In contrast, saccades are fast gaze shifts moving from one area to another.

3.2. Differential privacy mechanisms

We investigate Local Differential Privacy (LDP) [38] to secure a formal privacy assurance. Unlike the centralized approach, LDP offers a stronger model of trust by ensuring that each value sent to the server is in a distorted form. This method maintains privacy regardless of the server's intentions, even in cases of malicious activity.

Definition 1 *(ε-LDP).* [38]: An algorithm $\mathcal{M}(\cdot)$ adheres to ε-local differential privacy (ε-LDP), with $\varepsilon \geq 0$, if for any two inputs x_1, x_2 and any output o, the following inequality holds:

$$Pr[\mathcal{M}(x_1) = o] \leq e^{\varepsilon} Pr[\mathcal{M}(x_2) = o] \tag{1}$$

This condition specifies the degree of privacy safeguarding, where lower ε values signify enhanced privacy. LDP secures privacy by permitting individuals to submit a modified version of their input, $\mathcal{M}(x)$, rather than the actual x, to the data collector. The ε parameter quantifies the privacy level, with smaller figures representing heightened privacy. The acceptable limit for the privacy budget often depends on the application's unique privacy needs, with a consensus that a budget under 3 is considered acceptable [9, 13,39].

In this study, we applied perturbations to the initial points of each temporal segment according to the principles of geo-indistinguishability within the framework of Local Differential Privacy (LDP). Geo-indistinguishability is a variant of differential privacy tailored for protecting geographic data within a two-dimensional area [10]. Its formal definition is presented as follows:

Definition 2 *((ε,r)-geo-indistinguishability).* [10]: A function $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Z}$ is considered (ε,r)-geo-indistinguishable if for every pair of points $(x, x') \in \mathcal{X} \times \mathcal{X}$ satisfying $d(x, x') \leq r$,

$$\forall S \subset \mathcal{Z}, Pr[\mathcal{M}(x) \in S] \leq e^{\epsilon} Pr[\mathcal{M}(x') \in S] \tag{2}$$

here, $d(\cdot,\cdot)$ is the Euclidean distance.

In our study, we applied the Optimized Unary Encoding (OUE) protocol [40] for estimating mobility models within the framework of Local Differential Privacy (LDP).

Definition 3 *(Optimized Unary Encoding).* [40]: The OUE protocol involves three key phases: encoding, perturbation, and data aggregation.

1. **Encoding**: This phase converts an input x into a binary vector V of length d, where $V[x] = 1$ and all other elements $V[i]$ are set to 0 for $i \neq x$.
2. **Perturbation**: In this step, a vector V is transformed into a perturbed vector V' following the rule:

$$Pr[V'[i] = 1] = \begin{cases} \frac{1}{2}, & \text{if } V[i] = 1 \\ \frac{1}{e^{\epsilon}+1}, & \text{if } V[i] = 0, \end{cases} \tag{3}$$

   where ϵ signifies the privacy budget.
3. **Aggregation**: To accurately estimate the frequency of the actual value from perturbed vectors, the data aggregator compiles and corrects the received data using the equation:

$$\tilde{g}(x) = \frac{g(x) - nq}{\frac{1}{2} - q}, \quad q = \frac{1}{e^{\epsilon} + 1} \tag{4}$$

   In this equation, n is the number of perturbed vectors reported, and $\tilde{g}(x)$ is the aggregated sum of these vectors. Adjusting the data requires maintaining a consistent privacy budget ϵ for all reports.

Empirical evidence [40] has shown that the adjusted estimate $\tilde{g}(x)$ is free from bias. The expected mean and variance for the OUE are given by:

$$E[\tilde{g}(x)] = nf_x, \quad Var[\tilde{g}(x)] = n\frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}, \tag{5}$$

where f_x indicates the proportion of occurrences of the value x.

3.3. Markov chain model

For creating synthetic datasets, the Markov Chain model [41] is a favored approach. It is a probabilistic framework that maps out a series of potential events, making it ideal for the examination of sequential data, such as paths of location movement cited in literature [13,14,31]. This model operates on the premise that the likelihood of any given event depends exclusively on the state achieved in the immediately preceding step, embodying the concept of memorylessness. The Markov Chain model is formally introduced as follows:

Definition 4 *(Markov Chain model).* [41]: Consider a set of discrete states $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, ..., \sigma_k\}$. A series $\mathcal{S} = \{s_1, s_2, s_3, ..., s_l\}$ adheres to a kth-order Markov process if for $k \leq i \leq l-1$ and for every $s \in \Sigma$,

$$Pr[s_{i+1} = \sigma_j | s_{i+1} ... s_1] = Pr[s_{i+1} = \sigma_j | s_{i+1} ... s_{i-k}], \tag{6}$$

where the probability of transitioning to s_{i+1} from a preceding sequence of states is denoted as the transition probability.

In a dataset D, for a subsequence r within any $S \in D$, the empirical transition probability $Pr[\sigma|r]$ is determined by:

$$Pr[\sigma|r] = \frac{\sum_{\forall S \in D} N_S(r\sigma)}{\sum_{\forall S \in D} \sum_{\forall x \in \Sigma} N_S(rx)} = \frac{N_D(r\sigma)}{\sum_{\forall x \in \Sigma} N_D(rx)}, \tag{7}$$

where the sequence rx indicates r followed by any state x, and $N_S(rx)$ counts how often rx appears within $\mathcal{S}$. The notation $\sum_{\forall S \in D} N_S(rx)$ is represented as $N_D(rx)$. In our research, we employed one-stage Markov transitions in conjunction with comprehensive mobility models.

4. Algorithmic design

As illustrated in Fig. 1, our proposed *DPGazeSynth* workflow integrates local differential privacy into the generation of synthetic gaze paths. The workflow involves two main entities: users and a data curator, and it unfolds in two primary stages: (1) data perturbation and aggregation, and (2) synthetic data generation.

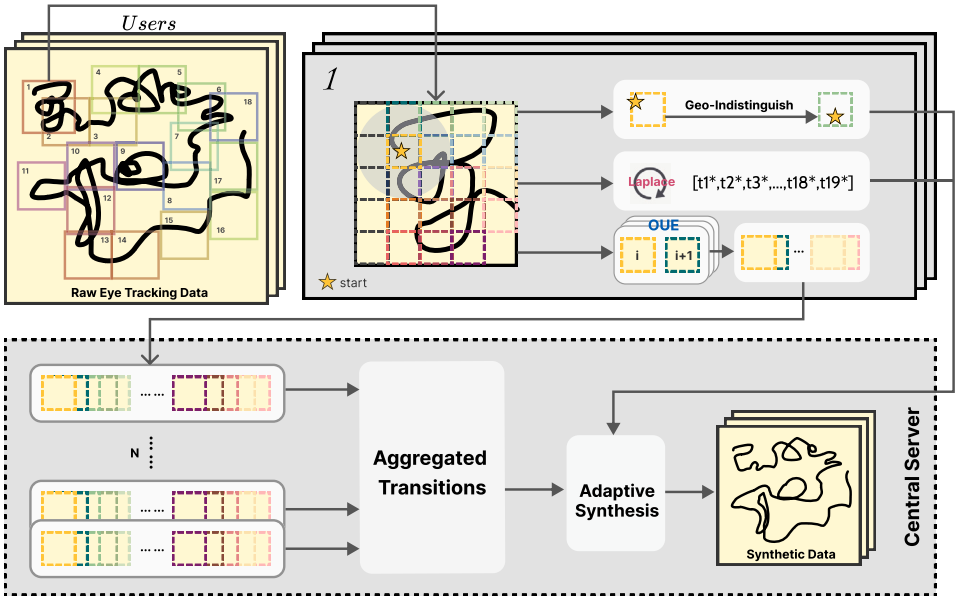

Fig. 1. Workflow of Privacy-Preserving Eye-Tracking Data Synthesis in VR Environments.

During the first stage, users' gaze data undergo spatial discretization into grids reflecting their temporal sequence and fixation duration. To accurately generate synthetic gaze paths, we develop a transition model that learns the user transition behaviors between grid cells. In the second stage, the curator uses the perturbed initial grids, time series data, and the transition model to recreate synthetic versions of the users' eye-tracking paths.

4.1. Gaze spatial data discretization

VR devices capture raw gaze data as a stream of directional vectors $v = [x, y, z]$. These vectors, after conversion into horizontal (θ_H) and vertical (θ_V) rotation angles of the eye globe, can be recorded as a sequence of tuples $S_t = [(\theta_{H_1}, \theta_{V_1}), (\theta_{H_2}, \theta_{V_2}), ..., (\theta_{H_t}, \theta_{V_t})]$ with each tuple paired with a timestamp.

Informed by studies [42,43] that correlate eye gaze patterns with spatial trajectories, we apply geospatial discretization to model gaze data. This involves translating camera angles into screen coordinates based on the virtual distance from the screen and then segmenting the visual field into a grid pattern according to a chosen granularity size G. Each gaze point is assigned to a grid cell, creating a sequence $L = C_1, C_2, ..., C_{|L|}$, with two points considered overlapping if they fall within the same cell.

If a gaze point falls outside the screen boundary, the nearest on-screen cell is selected as its representative. The screen is uniformly divided into $G \times G$ grids, with the selection criteria for G detailed subsequently.

4.2. Initial grids of temporal order windows

Unlike traditional trajectories, gaze paths do not have defined start and end points but are influenced by various visual stimuli over time. Eye-tracking datasets thus often include diverse stimuli across different spatial scenes, each with its own temporal sequence. For instance, a study involving video viewing tasks [44] recorded participants' eye movements as they watched a film, noting that the viewers' gaze accelerations matched the on-screen action. These initial points in each time window are critical for replicating natural eye movements, including saccades.

To model the spatial-temporal characteristics of eye movements, we preserve the initial grid position for each temporal window. Using Geo-indistinguishability with a privacy budget ε_1, we obscure the exact starting points of gaze paths within a radius r. This process is formalized as:

$$r = \frac{min(H, V)}{6 \times G}, \tag{8}$$

Here, r is expressed as a percentage of the screen's smaller dimension, whether height H or width V. This ensures that subsequent gaze locations also receive the same level of differential privacy protection.

4.3. Fixation time of temporal order windows

The complexity of gaze paths extends to variations in fixation durations and eye movement acceleration, which differ according to the visual stimuli encountered. To capture these nuances, we distinguish between fixations—where the gaze remains on the same grid—and movements to new grids. Time within a window is divided between these two states, with the total equaling the window's length W.

Due to the sensitivity of timing information [12,45], we apply the Laplace mechanism with parameter ε_2 to the fixation time series, adding noise to the data while preserving overall window duration. Post-perturbation, values are normalized to sum to W, with floating-point results converted into integers while maintaining the total.

4.4. Transition model learning

To accurately generate synthetic gaze paths, we develop a transition model that reflects real transition behaviors between grid cells. Drawing from [42], a first-order Markov Chain model is employed to describe the likelihood of moving from one grid cell $L[i]$ to the next $L[i+1]$. This probabilistic model posits that the future state depends only on the current state, not the sequence of events that preceded it:

$$Pr(L[n+1] = C | L[1], ..., L[n]) = Pr(L[n+1] = C | L[n]). \tag{9}$$

This assumption simplifies the dependency chain, focusing only on the immediate predecessor $L[n]$ to determine the next grid cell $L[n+1]$. We define the set of probabilities for moving from any given grid cell to the next as the transition pattern.

Considering the eye's saccadic movements, where the next fixation point $L[n+1]$ might not be adjacent to the current one $L[n]$, we introduce a technique that selects the subsequent grid cell from the same direction as the saccade. The transition probability from a grid cell C_i to another C_j is defined as s_{ij} and is formulated as follows:

$$Pr(s_{ij}) = \begin{cases} Pr(L[n+1] = C | L[n]) & \text{if } C_j \text{ is adjacent to } C_i \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

Here, $Pr(s_{ij})$ denotes the transition probability from C_i to C_j, and S is the transition model capturing all such probabilities from the aggregated data.

Each user encodes their gaze path transitions into a binary vector of length $|S|$, which approximates eight times the number of grid cells, considering the possible movements to adjacent cells. During the perturbation phase, each transition is noised and subsequently uploaded by individual users. The curator then aggregates these transitions to estimate the overall model. By converting each mode of transition into a binary vector, it becomes possible to apply disturbances to these vectors using the Optimized Unary Encoding (OUE) protocol. To account for the unequal frequency of transitions in different temporal windows, we distribute the budget ε_3 equally over each window's duration, assigning $\varepsilon_3/(w-1)$ to each transition within a window. This ensures consistent privacy protection across all time windows.

4.5. Gaze path synthesis

Algorithm 1 Synthesizing Gaze Paths.

Initialization: Empty synthetic gaze path set P_{syn}
Input: Transition model $\mathcal{M}$, fixation times $\{S_1, ..., S_T\}$, initial cells $\{C_1, ..., C_T\}$
Output: Synthetic gaze path P_{syn}

Initialize P_{syn} as an empty set
for each time window t from 1 to T **do**
  for each fixation f in S_t **do**
    Step 1: Initial Grid. If $f = 1$, set C_{now} to C_t.
    Step 2: Random Transition. If $f > 1$, sample C_{next} from $\mathcal{N}^*_{C_{now}}$ using $\mathcal{M}$, update C_{now}.
    Step 3: Gaze Point Sampling. Sample a point p_{now} in C_{now}, replicate it $S_t[f]$ times, add to P_{syn}.
  end for
end for
return P_{syn}

The Gaze Path Synthesis algorithm utilizes the noisy fixation times, initial cell sequences, and the transition model to generate synthetic gaze paths as detailed in Algorithm 1. The inputs include the transition model $\mathcal{M}$, fixation times $\{S_1, ..., S_T\}$, and initial cell sequence $\{C_1, ..., C_T\}$.

For each time window t, we iteratively process the fixation times in S_t. The initial cell C_t serves as the starting point. Subsequent transitions are predicted using the model $\mathcal{M}$, and gaze points are sampled from the predicted grid cells, replicating them based on fixation duration and appending them to P_{syn}.

The synthesis process leverages user-provided data, including initial cell sequence and fixation times, while incorporating aggregated transition patterns to generate the majority of gaze points. Differential privacy and data discretization further safeguard user privacy against re-identification, enhancing utility for subsequent analysis.

4.6. Differential privacy definition and analysis

We define a specialized form of differential privacy that provides privacy guarantees for the eye-tracking trajectory **and its fixation time** based on the definition for gaze stream prefixes [9].

Definition 1 *((ε,w,r,t)-differential privacy for gaze stream trajectory and its fixation time).* A mechanism $\mathcal{M}$ mapping the domain of all eye-tracking data stream prefixes and their corresponding fixation times, is defined to be (ε,w,r,t)-indistinguishable. The parameter w presents the transition window, r represents the fixation radius, and t represents the fixation time.

Formally, for any two (w,r,t)-neighboring sets of eye-tracking data, denoted as $<\mathcal{X},\mathcal{T}>_k^g$ and $<\mathcal{X},\mathcal{T}>_k^{g'}$, the probability of the mechanism $\mathcal{M}$ producing an output from $<\mathcal{X},\mathcal{T}>_k^g$ is bounded by e^ϵ times the probability of producing the same output from $<\mathcal{X},\mathcal{T}>_k^{g'}$. This is mathematically represented as:

$$Pr[\mathcal{M}(<\mathcal{X},\mathcal{T}>_k^g)=O] \le e^\epsilon Pr[\mathcal{M}(<\mathcal{X},\mathcal{T}>_k^{g'})=O] \tag{11}$$

where O represents the possible outputs of the mechanism $\mathcal{M}$.

Based on the refined DP definition, we offer an analysis of the privacy guarantees provided by *DPGazeSynth*, elaborating on the budget allocation strategy.

Theorem 1. *DPGazeSynth ensures event-level ε-Local Differential Privacy (ε-LDP), where* $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$.

Proof. *DPGazeSynth* divides gaze data into multiple time windows based on their timestamps, treating each window as an event. *DPGazeSynth* consists of five components: gaze space discretization, initial grid acquisition, fixation time series perturbation, mobility model learning, and gaze path synthesis. We assess the privacy budget consumption of each component as follows:

Gaze Spatial Discretization. DPGazeSynth first discretely points to grids. Note that the granularity size of discretization grids G impacts the level of granularity in the discretized gaze path. When G is small, multiple points in the original gaze path are aggregated into a single grid, leading to a loss of detailed mobility patterns. Conversely, when G is large, each grid becomes smaller, increasing the risk of revealing individual biometrics. Nonetheless, discretization alone doesn't provide formal privacy guarantees and doesn't consume the privacy budget.

Initial Grid Acquisition. The initial point of each time window is perturbed using the geo-indistinguishability mechanism, and then discretizes into the grid. The adversary cannot distinguish between all points within radius r from the initial point and the privacy budget consumed is ε_1. It's worth noting that the protection radius of the initial point used by *DPGazeSynth* is related to the size of the discretization grid.

Fixation Time Series Perturbation. The fixation time series is used to control the time and number of transitions, where the number of transitions can represent fixation and saccade movements. *DPGazeSynth* utilizes the Laplacian mechanism with a privacy budget of ε_2 to report a noisy version, which makes it impossible for adversaries to distinguish between any two fixation time series with the same number of transitions. Regarding sensitivity G_f, we consider the two most extreme scenarios: one where there are no transitions within the window W (i.e., $[W]$) and the other where a transition at every timestamp within the window (i.e., $[t_1,...,t_W]$).

$$\begin{aligned} GS_f &= max_{D\simeq D'} ||f(D)-f(D')||_1 \\ &= |W-t_1| + |0-t_2| + ... + |0-t_W| \\ &= 2W-2 \end{aligned} \tag{12}$$

Based on the calculation, we normalize the perturbed version to form a series with a sum equal to W. Importantly, this step does not consume any privacy budget by the post-processing property of DP.

Mobility Model Learning. The mobility model aggregates mobility rules using First-Order Markov models and allocates budget ε_3 for each time window using the OUE protocol. Since the OUE protocol requires allocating the same privacy budget for each transition and a time window of length W can have at most $W-1$ grid transitions, *DPGazeSynth* considers the most extreme scenario by evenly distributing the privacy budget based on the maximum possible number of transitions. In other words, it allocates $\varepsilon_3/(W-1)$ privacy budget to each transition. We can observe that, for any two transitions x_1, x_2, and an output o, where F represents the OUE protocol, the following relation holds:

$$Pr(F(x_1)=o) \le e^{\varepsilon_3/(W-1)} Pr(F(x_2)=o). \tag{13}$$

Then, it follows from the sequential composition property of DP that for any two adjacent transition sequences $X=\{x_1,x_2,...,x_m\}$ with a length of m and $X'=\{x_1',x_2',...,x_n'\}$ with a length of n, where $max(m,n) \le W-1$, and for any possible output set O, the following holds:

$$\begin{aligned} Pr(F(X)\in O) &\le e^{\varepsilon_3 max(m,n)/(W-1)} Pr(F(X')\in O) \\ &\le e^{\varepsilon_3} Pr(F(X')\in O). \end{aligned} \tag{14}$$

In other words, it is challenging for an adversary to distinguish between any two adjacent transition sequences. It's noted that actual privacy consumption is always less than the privacy budget ε_3 because realistically it's hardly ever the extreme situation.

Gaze Path Synthesis. The gaze path synthesis method leverages the mobility model and perturbation data from the previous components. This approach does not impact the privacy budget owing to the post-processing attribute of Differential Privacy (DP) [51]. Specifically, *DPGazeSynth* uses the Markov Chain model to forecast all movements within the current time window based on the

noise-added initial grid and fixation time series. As such, the synthesis of gaze paths is considered a post-processing operation, and thus does not consume any privacy budget.

Based on the sequential composition property of DP, *DPGazeSynth* satisfies ε-DP, where $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$.

5. Evaluation

We undertook a thorough evaluation of *DPGazeSynth* across three dimensions: (1) its effectiveness in mitigating re-identification attacks, (2) its utility in practical applications, and (3) its performance based on utility metrics. *DPGazeSynth* was benchmarked against the leading state-of-the-art method, examining its defensive capabilities against potential attackers and its usefulness when applied to publicly available VR eye-tracking datasets.

5.1. Experimental setup

Datasets. Our experiments were performed using two benchmark eye-tracking datasets:

- The 360_em dataset [46] comprised recordings from 13 participants, each equipped with a VR headset. Their gaze data was captured as they engaged with a variety of YouTube videos and one self-generated video.
- GazebaseVR [44], noted as the most extensive longitudinal, binocular eye-tracking dataset, was collected at 250 Hz using a VR headset with eye-tracking technology. It consists of 5,020 binocular recordings from 407 diverse college-aged participants.

Baseline. We juxtaposed our approach with Kalεido [9], a cutting-edge method known for its strict adherence to Local Differential Privacy (LDP). For a fair assessment, both mechanisms were compared under the same conditions of protection scope, time window, and privacy budget.

Experimental Settings. The privacy mechanisms were implemented on a computing platform powered by an Intel(R) Xeon(R) Gold 5218R CPU at 2.10 GHz, equipped with 100 GB of RAM. We varied the privacy budget ε from 1 to 3 in our experiments, allocating ε_1 as 0.6ε, and ε_2 and ε_3 each as 0.2ε. In line with Kalεido, a uniform time window of $W = 0.5$ seconds was adopted. Each set of experiments was replicated 10 times with varying random seeds, and the mean values were reported.

Utility Metrics. Following the testing convention [13], we adopted two metrics to assess the congruence between the original dataset and the synthetic dataset: Root Mean Squared Error (RMSE) and Density Error, which are described in detail as follows:

- The RMSE for the trajectories was computed to gauge the precision of the synthetic trajectories. We calculated the RMSE for each pair of gaze paths to obtain the overall average.

$$RMSE = \sqrt{\sum_{i=1}^{N}(x_i - x_i')^2 + (y_i - y_i')^2}, \tag{15}$$

  where i indicates the sampling point, x_i, y_i denote the actual coordinate points, x_i', y_i' are the corresponding synthetic points, and N represents the total number of trajectory points.
- Density error measures the spatial density discrepancy between the synthetic gaze path set T_{syn} and the actual gaze path set T.

$$Density\ Error = JSD(D(T), D(T_{syn})), \tag{16}$$

  where $D(\cdot)$ indicates the spatial density distribution, and $JSD(\cdot)$ is the Jenson-Shannon divergence between the two distributions. To determine the spatial density distribution, we utilized a 60×60 uniform grid for screen space discretization.

5.2. Re-identification

Setup. Our objective was to construct predictive attacks that could potentially reveal users' identities. We selected 12 participants from the 360_em dataset for this purpose. Notably, each participant contributed at least 13 samples, with 10 designated for training and the remaining three for testing. Utilizing a Radial Basis Function network (RBFN), we extracted various features to train two distinct classifiers for saccades and fixations, setting their ratio at [0.4, 0.6]. The settings applied to the Identification by Velocity Threshold (IVT) algorithm included a velocity threshold (VT) of 30 degrees per second and a minimum fixation duration of 200 milliseconds, aligning with the traditional configuration [47]. The raw gaze samples formed the basis of our model training, and we evaluated performance by contrasting these with noise-augmented gaze samples, reporting the F1 scores as the metric for re-identification attack success.

Results. The F1 scores, as seen in Fig. 2, indicate that while raw gaze data facilitated nearly complete re-identification of users, the introduction of *DPGazeSynth* and Kalεido significantly diminished the attacker's classification accuracy, particularly in low-privacy settings. Intriguingly, *DPGazeSynth*'s performance was almost on par with that of a random baseline. However, it is important to note that when the parameter r was set to 1, Kalεido struggled to maintain the anonymity of all user identities.

Additionally, we also sought to compromise user identities within the GazebaseVR dataset. This endeavor, however, was met with challenges owing to the limited sample availability for each participant, with many providing only two samples for the same stimulus task.

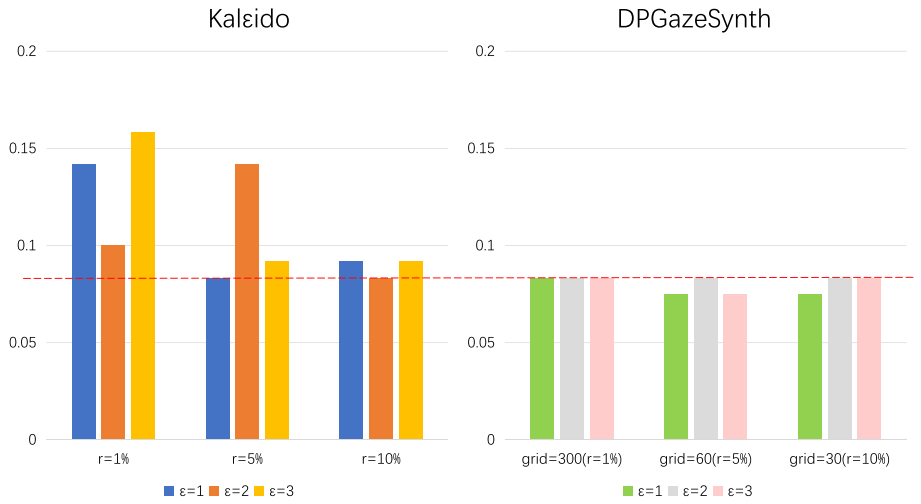

Fig. 2. Comparative Analysis of Re-identification Risk: F1 Scores for Kalεido versus *DPGazeSynth* Across Various Privacy Budgets and Parameters.

Table 2
Comparison of Task Classification Accuracy between *DPGazeSynth* and Kalεido Using the GazebaseVR Dataset.

Parameters		Accuracy
Original Data		98.5%
Kalεido	$r = 5\%$	$r = 10\%$
$\varepsilon = 3$	32.17%	32.95%
$\varepsilon = 2$	32.57%	32.80%
$\varepsilon = 1$	34.04%	33.91%
DPGazeSynth	Grid = 60 (r = 5%)	Grid = 30 (r = 10%)
$\varepsilon = 3$	**68.28%**	**65.02%**
$\varepsilon = 2$	**68.86%**	**64.46%**
$\varepsilon = 1$	**68.04%**	**64.44%**

5.3. Application utility

The analysis of human visual attention in virtual reality (VR) is vital for numerous applications. One of the most insightful approaches is examining and contrasting human gaze behaviors across different tasks, which enriches our understanding of the cognitive processes underpinning visual attention, as evidenced by prior research [20–22]. To this end, we utilized task-based gaze analysis as a criterion for assessing the practicality of eye-tracking data post-perturbation.

Setup. Our experiment focused on the three most common tasks within the GazebaseVR dataset: (1) a video viewing task, (2) a reading task, and (3) a random saccade task. Each task was represented by 1004 data instances, 704 for training and the remainder for testing. We opted for a Random Forest approach [27,48], commonly used for task recognition [48–50], and trained the model using gaze features. The velocity threshold was set at 30 degrees per second with a minimum fixation duration of 200 ms, served as our standard parameter. The models were trained using the original gaze data, and we compared the performance of *DPGazeSynth* with Kalεido, focusing on the utility of data that had been altered while ensuring complete privacy.

Results. As shown in Table 2, *DPGazeSynth* consistently outperformed Kalεido in terms of classification accuracy under various privacy settings. While Kalεido's performance was close to random guessing, *DPGazeSynth* achieved near 75% accuracy in most cases and maintained substantial utility even at stricter privacy settings, exemplified by a grid size of 30 yielding an accuracy of approximately 85%.

5.4. Utility metrics analysis

We compared the utility metrics between *DPGazeSynth* and Kalεido, examining their performance across varying privacy budgets, denoted by ε. Each experimental iteration was conducted five times to ensure statistical reliability, and the mean results are reported.

Setup. Our evaluation protocol was stringent, considering only the data points that fell within the screen's boundaries. This was straightforward for the 360_em dataset, as all points were on-screen. However, the GazebaseVR dataset included some data points

Table 3
Comparison of RMSE between *DPGazeSynth* and Kalεido on the 360_em and GazebaseVR Datasets.

Metric		360_em Dataset		GazebaseVR Dataset	
		Kalεido	*DPGazeSynth*	Kalεido	*DPGazeSynth*
r = 5%	$\varepsilon = 3$	5.461×10^5	$\mathbf{4.170 \times 10^4}$	1.728×10^4	$\mathbf{1.252 \times 10^4}$
	$\varepsilon = 2$	6.482×10^5	$\mathbf{4.172 \times 10^4}$	1.729×10^4	$\mathbf{1.251 \times 10^4}$
	$\varepsilon = 1$	9.076×10^5	$\mathbf{4.175 \times 10^4}$	1.728×10^4	$\mathbf{1.247 \times 10^4}$
r = 10%	$\varepsilon = 3$	1.206×10^6	$\mathbf{4.273 \times 10^4}$	1.729×10^4	$\mathbf{1.296 \times 10^4}$
	$\varepsilon = 2$	1.441×10^6	$\mathbf{4.278 \times 10^4}$	1.728×10^4	$\mathbf{1.300 \times 10^4}$
	$\varepsilon = 1$	1.917×10^6	$\mathbf{4.284 \times 10^4}$	1.730×10^4	$\mathbf{1.250 \times 10^4}$

Table 4
Comparison of Density Error between *DPGazeSynth* and Kalεido on the 360_em and GazebaseVR Datasets.

Metric		360_em Dataset		GazebaseVR Dataset	
		Kalεido	*DPGazeSynth*	Kalεido	*DPGazeSynth*
r = 5%	$\varepsilon = 3$	0.358	**0.120**	0.310	**0.009**
	$\varepsilon = 2$	0.329	**0.122**	0.237	**0.009**
	$\varepsilon = 1$	0.295	**0.122**	0.125	**0.011**
r = 10%	$\varepsilon = 3$	0.375	**0.130**	0.255	**0.016**
	$\varepsilon = 2$	0.345	**0.130**	0.182	**0.016**
	$\varepsilon = 1$	0.319	**0.136**	0.051	**0.017**

beyond the screen, which were duly omitted from our analysis. We utilized RMSE and density error to measure and compare the performance of *DPGazeSynth* and Kalεido quantitatively.

Results. Overall, *DPGazeSynth* consistently surpassed Kalεido across both datasets on the utility metrics (as shown in Tables 3 and 4). Notably, in many instances, *DPGazeSynth* showed an enhancement in performance by an order of magnitude compared to Kalεido, showcasing its robustness and superior ability to preserve the utility of the data. It is important to highlight that the density error for Kalεido did not show a straightforward correlation with the privacy budget ε. This can be attributed to the fact that as ε was reduced, there was an increase in the perturbation of points by Kalεido that fell outside the screen's area, thereby broadening the density distribution and inadvertently increasing the margin for error.

6. Discussion

Prevailing methodologies struggled to meet both privacy protection and utility. *DPGazeSynth*, with its semi-synthetic approach, represents a significant leap forward, finely balancing this dichotomy in the context of eye-tracking data sharing. It ensures formal local differential privacy by introducing noise to the initial gaze points through geo-indistinguishability, to transition frequencies via the OUE protocol, and to fixation time series using the Laplacian mechanism.

6.1. Privacy semantics

We utilized the composition theorem of Differential Privacy (DP) to distribute the privacy budget among various data components judiciously. Under the same overall budget, *DPGazeSynth* opts for a conservative allocation approach in the OUE protocol, assuming each transition as an independent event, which is a rare occurrence in practice. This conservative stance means we often consume less of the privacy budget than allocated. Conversely, Kalεido fully expends its budget. Methodologically, *DPGazeSynth*'s semi-synthesis only ties the initial point of each time window to user data, while subsequent points are generated by a synthetic Markov model, thus substantially masking individual data. The re-identification experiments corroborate *DPGazeSynth*'s defensive efficacy, showing it to be on par with or better than Kalεido.

6.2. Practical implementation

DPGazeSynth offers enhanced practicality over existing state-of-the-art methods, safeguarding shared eye-tracking data with differential privacy. This system could pave the way for a broader data sharing scope, potentially benefiting data miners and advertisers while being underpinned by users' increased willingness to share their data.

6.3. Limitations

DPGazeSynth is not without its limitations. It does not fully generate gaze paths independently but relies partially on perturbed user information for transition synthesis. Moreover, the exploration of *DPGazeSynth* in real-time settings was not part of our study.

Our research was also confined to three main tasks commonly associated with VR settings. Future investigations could expand to include diverse VR tasks like memory or interaction-based exercises, which could yield further insights.

6.4. Future work

We suggest the following prospective research avenues. For instance, developing fully autonomous privacy-preserving gaze path synthesis methods that do not depend on user-specific data is an intriguing prospect. Merging low-frequency gaze data generation with privacy techniques for initial point data in each time window might offer a solution. Further, employing a higher-order Markov model could capture more complex gaze patterns, reflecting the nuanced temporal sequences and variable behaviors across extended scenes. Given the Markov Chain model's simplicity and fewer parameters, the potential for its real-time application in *DPGazeSynth* is an avenue worth investigating. For example, data collectors could implement Markov Chain model-based generation locally, depending on the type of task, to offer adaptive privacy safeguards. Furthermore, with the increasing integration of eye-tracking in spatial computing devices, such as Apple Vision Pro, for interacting with operating systems, the application of *DPGazeSynth* within such eye-tracking operating systems emerges as a compelling field of exploration. Finally, we aim to explore the potential of applying our system in eye movement data analytics, ensuring user privacy remains intact.

7. Conclusion

Our research navigated the precarious path between privacy and utility in eye-tracking data analysis. We introduced *DPGazeSynth*, a framework that not only adheres to a strict LDP protocol but also maintains high data utility. Our experimental evaluation across two distinct datasets highlighted *DPGazeSynth*'s efficacy. The outcomes confirm our method's validity and open the door to future work, especially in complex trajectory data collection scenarios, reinforcing the foundation for future research in this domain.

CRediT authorship contribution statement

Xiaojun Ren: Resources. **Jiluan Fan:** Writing – original draft, Methodology, Investigation, Conceptualization. **Ning Xu:** Validation, Investigation, Conceptualization. **Shaowei Wang:** Validation, Writing – review & editing. **Changyu Dong:** Conceptualization, Resources, Supervision, Validation. **Zikai Wen:** Conceptualization, Formal analysis, Investigation, Supervision, Validation, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The databases are open sourced.

References

[1] C. Warin, D. Reinhardt, Vision: usable privacy for xr in the era of the metaverse, in: Proceedings of the 2022 European Symposium on Usable Security, EuroUSEC '22, Association for Computing Machinery, 2022, pp. 111–116.
[2] P. Casey, I. Baggili, A. Yarramreddy, Immersive virtual reality attacks and the human joystick, IEEE Trans. Dependable Secure Comput. 18 (2) (2021) 550–562, https://doi.org/10.1109/TDSC.2019.2907942.
[3] Z. Liang, F. Tan, Z. Chi, Video-based biometric identification using eye tracking technique, in: Signal Processing, Communication and Computing (ICSPCC), 2012.
[4] C.D. Holland, O.V. Komogortsev, Complex eye movement pattern biometrics: the effects of environment and stimulus, IEEE Trans. Inf. Forensics Secur. 8 (12) (2013) 2115–2126.
[5] S. Eberz, G. Lovisotto, A. Patanè, M. Kwiatkowska, V. Lenders, I. Martinovic, When your fitness tracker betrays you: quantifying the predictability of biometric features across contexts, in: 2018 IEEE Symposium on Security and Privacy (SP), 2018, pp. 889–905.
[6] A. Liu, L. Xia, A. Duchowski, R. Bailey, K. Holmqvist, E. Jain, Differential privacy for eye-tracking data, in: Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA '19, 2019.
[7] J. Steil, I. Hagestedt, M.X. Huang, A. Bulling, Privacy-aware eye tracking using differential privacy, in: Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, 2019, pp. 1–9.
[8] B. David-John, D. Hosfelt, K. Butler, E. Jain, A privacy-preserving approach to streaming eye-tracking data, IEEE Trans. Vis. Comput. Graph. 99 (2021) 1.
[9] J. Li, A.R. Chowdhury, K. Fawaz, Y. Kim, Kalεido: real-time privacy control for eye-tracking systems, in: USENIX Security Symposium, 2021.
[10] M.E. Andrés, N.E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, Geo-indistinguishability: differential privacy for location-based systems, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, CCS '13, Association for Computing Machinery, 2013, pp. 901–914.
[11] G. Kellaris, S. Papadopoulos, X. Xiao, D. Papadias, Differentially private event sequences over infinite streams, Proc. VLDB Endow. 7 (12) (2014) 1155–1166, https://doi.org/10.14778/2732977.2732989.
[12] C. Schäler, T. Hütter, M. Schäler, Benchmarking the utility of w-event differential privacy mechanisms - when baselines become mighty competitors, Proc. VLDB Endow. 16 (8) (2023) 1830–1842, https://doi.org/10.14778/3594512.3594515.
[13] Y. Du, Y. Hu, Z. Zhang, Z. Fang, L. Chen, B. Zheng, Y. Gao, Ldptrace: locally differentially private trajectory synthesis, Proc. VLDB Endow. 16 (8) (2023) 1897–1909, https://doi.org/10.14778/3594512.3594520.

[14] H. Wang, Z. Zhang, T. Wang, S. He, M. Backes, J. Chen, Y. Zhang, Privtrace: differentially private trajectory synthesis by adaptive Markov model, in: USENIX Security Symposium 2023, 2023.
[15] M.E. Gursoy, L. Liu, S. Truex, L. Yu, W. Wei, Utility-aware synthesis of differentially private and attack-resilient location traces, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 196–211.
[16] Z. Ye, Y. Li, A. Fathi, Y. Han, A. Rozga, G.D. Abowd, J.M. Rehg, Detecting eye contact using wearable eye-tracking glasses, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12, Association for Computing Machinery, 2012, pp. 699–704.
[17] P.A. Rauschnabel, R. Felix, C. Hinsch, H. Shahab, F. Alt, What is xr? Towards a framework for augmented and virtual reality, Comput. Hum. Behav. 133 (2022) 107289.
[18] M. Meißner, J. Pfeiffer, T. Pfeiffer, H. Oppewal, Combining virtual reality and mobile eye tracking to provide a naturalistic experimental environment for shopper research, J. Bus. Res. 100 (2019) 445–458, https://doi.org/10.1016/j.jbusres.2017.09.028.
[19] R. Pieters, E. Rosbergen, M. Wedel, Visual attention to repeated print advertising: a test of scanpath theory, J. Mark. Res. 36 (4) (1999) 424–438.
[20] Z. Hu, A. Bulling, S. Li, G. Wang, Fixationnet: forecasting eye fixations in task-oriented virtual environments, IEEE Trans. Vis. Comput. Graph. 27 (5) (2021) 2681–2690, https://doi.org/10.1109/TVCG.2021.3067779.
[21] J. Hadnett-Hunter, G. Nicolaou, E.J. O'Neill, M.J. Proulx, The effect of task on visual attention in interactive virtual environments, ACM Trans. Appl. Percept. 16 (3) (2019).
[22] Z. Hu, A. Bulling, S. Li, G. Wang, Ehtask: recognizing user tasks from eye and head movements in immersive virtual reality, IEEE Trans. Vis. Comput. Graph. 29 (4) (2023) 1992–2004, https://doi.org/10.1109/TVCG.2021.3138902.
[23] M.Q. Khan, S. Lee, Gaze and eye tracking: techniques and applications in adas, Sensors 19 (24) (2019), https://doi.org/10.3390/s19245540.
[24] J. Kang, X. Han, J. Song, Z. Niu, X. Li, The identification of children with autism spectrum disorder by svm approach on eeg and eye-tracking data, Comput. Biol. Med. 120 (2020) 103722, https://doi.org/10.1016/j.compbiomed.2020.103722.
[25] P. Kasprowski, J. Ober, Eye movements in biometrics, in: International Workshop on Biometric Authentication, Springer, 2004, pp. 248–258.
[26] O.V. Komogortsev, S. Jayarathna, C.R. Aragon, M. Mahmoud, Biometric identification via an oculomotor plant mathematical model, in: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, 2010, pp. 57–60.
[27] C. Schröder, S.M.K. Al Zaidawi, M.H. Prinzler, S. Maneth, 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, 2020, pp. 1–7.
[28] A. George, A. Routray, A score-level fusion method for eye movement biometrics, Pattern Recognit. Lett. 82 (2016) 207–215.
[29] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (3–4) (2014) 211–407.
[30] Z. Zhang, T. Wang, N. Li, J. Honorio, M. Backes, S. He, J. Chen, Y. Zhang, {PrivSyn}: differentially private data synthesis, in: 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 929–946.
[31] M.E. Gursoy, L. Liu, S. Truex, L. Yu, W. Wei, Utility-aware synthesis of differentially private and attack-resilient location traces, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18, Association for Computing Machinery, 2018, pp. 196–211.
[32] D. Martin, A. Serrano, A.W. Bergman, G. Wetzstein, B. Masia, Scangan360: a generative model of realistic scanpaths for 360° images, IEEE Trans. Vis. Comput. Graph. 28 (5) (2022) 2003–2013, https://doi.org/10.1109/TVCG.2022.3150502.
[33] G. Lan, T. Scargill, M. Gorlatova, Eyesyn: psychology-inspired eye movement synthesis for gaze-based activity recognition, in: 2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2022, pp. 233–246.
[34] M. Assens, X. Giro-i Nieto, K. McGuinness, N.E. O'Connor, Pathgan: visual scanpath prediction with generative adversarial networks, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018.
[35] M. Assens Reina, X. Giro-i Nieto, K. McGuinness, N.E. O'Connor, Saltinet: scan-path prediction on 360 degree images using saliency volumes, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, 2017.
[36] Z. Hu, C. Zhang, S. Li, G. Wang, D. Manocha, Sgaze: a data-driven eye-head coordination model for realtime gaze prediction, IEEE Trans. Vis. Comput. Graph. 25 (5) (2019) 2002–2010, https://doi.org/10.1109/TVCG.2019.2899187.
[37] Pichet Termsarasab, Thananan Thammongkolchai, Janet C. Rucker, Steven J. Frucht, The diagnostic value of saccades in movement disorder patients: a practical guide and review, J. Clin. Mov. Disord. 2 (1) (2015).
[38] S.P. Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, A. Smith, What can we learn privately?, SIAM J. Comput. 40 (3) (2011) 793–826, https://doi.org/10.1137/090756090.
[39] V. Clay, P. König, S. Koenig, Eye tracking in virtual reality, J. Eye Mov. Res. 12 (1) (2019).
[40] T. Wang, J. Blocki, N. Li, S. Jha, Locally differentially private protocols for frequency estimation, in: 26th USENIX Security Symposium (USENIX Security 17), 2017, pp. 729–745.
[41] S.P. Meyn, R. Tweedie, Computable bounds for geometric convergence rates of Markov chains, Ann. Appl. Probab. 4 (4) (1994) 981–1011.
[42] N.C. Anderson, F. Anderson, A. Kingstone, W.F. Bischof, A comparison of scanpath comparison methods, Behav. Res. Methods 47 (2015) 1377–1392.
[43] H. Jarodzka, K. Holmqvist, M. Nyström, A vector-based, multidimensional scanpath similarity measure, in: Symposium on Eye-Tracking Research and Applications, 2010.
[44] D. Lohr, S. Aziz, L. Friedman, O.V. Komogortsev, Gazebasevr, a large-scale, longitudinal, binocular eye-tracking dataset collected in virtual reality, Sci. Data 10 (1) (2023).
[45] T. Allard, G. Hébrail, F. Masseglia, E. Pacitti, Chiaroscuro: transparency and privacy for massive personal time-series clustering, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD '15, Association for Computing Machinery, 2015, pp. 779–794.
[46] I. Agtzidis, M. Startsev, M. Dorr, 360-degree video gaze behaviour: a ground-truth data set and a classification algorithm for eye movements, in: Proceedings of the 27th ACM International Conference on Multimedia, MM '19, Association for Computing Machinery, 2019, pp. 1007–1015.
[47] S.M.K. Al Zaidawi, M.H. Prinzler, C. Schröder, G. Zachmann, S. Maneth, Gender classification of prepubescent children via eye movements with reading stimuli, in: Companion Publication of the 2020 International Conference on Multimodal Interaction, 2020, pp. 1–6.
[48] L. Hua, D. Weihua, H. Haosheng, G. Georg, L. Huiping, Inferring user tasks in pedestrian navigation from eye movement data in real-world environments, Int. J. Geogr. Inf. Sci. (2018) 1–25.
[49] J. Hild, M. Voit, C. Kühnle, J. Beyerer, Predicting observer's task from eye movement patterns during motion image analysis, in: Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ETRA '18, Association for Computing Machinery, 2018.
[50] J.F. Boisvert, N.D. Bruce, Predicting task from eye movements: on the importance of spatial distribution, dynamics, and image features, Neurocomputing 207 (2016) 653–668, https://doi.org/10.1016/j.neucom.2016.05.047.
[51] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (3–4) (2014) 211–407.