Mitigating Privacy Threats Without Degrading Visual Quality of VR Applications: Using Re-identification Attack as a Case Study

Yu-Szu Wei*, Yuan-Chun Sun*, Shin-Yi Zheng*, Hsun-Fu Hsu†, Chun-Ying Huang†, and Cheng-Hsin Hsu*
*National Tsing Hua University, Hsin-Chu, Taiwan
†National Yang Ming Chiao Tung University, Hsin-Chu, Taiwan

***Abstract*—Virtual Reality (VR) applications take users' Head-Mounted Display (HMD) and controller trajectories as inputs for an immersive experience. Leakage of these trajectories threatens user privacy in several aspects, including but not limited to their identities. Existing privacy-preserving approaches, however, overlook the temporal correlation of VR user trajectories, which could be leveraged by attackers. In this paper, we develop a disturber to perturb VR user trajectories in both temporal and spatial domains on the fly. Such trajectory perturbations, unfortunately, could lead to distorted rendered VR viewports. Thus, we develop a compensator to recover from such distortion using efficient image-warping algorithms. Our evaluation results show the merits of our proposed solution: (i) our disturber alone reduces at most 0.42 re-identification rate of VR users compared to the state-of-the-art approach, (ii) our disturber alone outperforms the state-of-the-art approach by 2.43 dB in PSNR, 0.13 in SSIM, and 8.15 in VMAF under the same privacy-preserving settings, and (iii) our compensator further boosts the visual quality of a VR application by at most 6.83 dB in PSNR, 0.45 in SSIM, and 34.57 in VMAF, compared to disturber-only solution.**

***Index Terms*—VR, networks, privacy, attack, quality, re-identification**

I. Introduction

Increasingly more Head Mounted Displays (HMDs) have been released by key manufacturers, like HTC, Pico, Meta, and Apple, which enable novel Virtual Reality (VR) usage scenarios, such as healthcare, education, sports, tourism, and entertainment. In these scenarios, each user could use an HMD and (optionally) two hand-held controllers[1] to interact with VR applications. Both HMDs and controllers come with sensors like accelerometers and gyroscopes, which produce time-series sensor readings on locations (in Cartesian coordinates) and orientations (in quaternions). We collectively refer to the locations and orientations of an HMD (or a controller) at a moment as a *pose* and a time series of poses as a *trajectory*.

In this paper, we consider a networked VR system in which one or multiple VR users are connected to one or multiple servers via the Internet. Every VR user interacts with VR applications hosted by a VR platform in 6 Degrees-of-Freedom (6DoF) through sensors on HMD (and controllers).

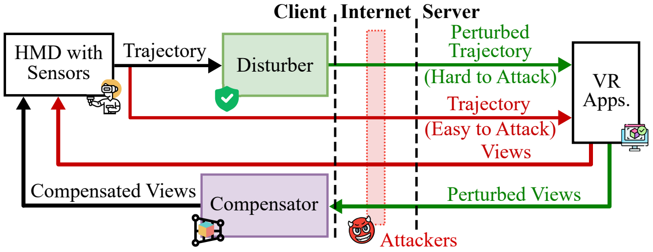


Fig. 1. Our proposed disturber and compensator mitigate the privacy threats to VR applications.

These VR applications often employ Machine Learning (ML) algorithms, which collect sensor data, such as trajectories, from VR users for various optimization objectives for a better user experience. Messages carrying these sensor data may be intercepted by attackers and thus are vulnerable to privacy threats [32], including but not limited to the leakage of a VR user's identity [22], height/fitness [23], and typed text [26].

A straightforward way to cope with such privacy threats is to add a fixed offset to each VR user's trajectory throughout a session. This approach has been adopted by MetaGuard (MG) [23], which applies offsets to telemetry data in various ways, taking into account factors such as the user's height, wingspan, and squat depth. However, such an approach ignores the temporal correlation among poses of each trajectory. In contrast, we propose to sequentially add different offsets drawn from a probability distribution to individual poses over time. While doing so, intuitively, leads to stronger protection against privacy threats, the rendered viewports are likely to be shaky, which causes severe cybersickness for VR users. To deal with this challenge, we apply image warping algorithms [16, 29] to transform every rendered image back to its original pose.

More specifically, Fig. 1 reveals our proposed solution, consisting of: (i) a *disturber* that adds noise, or perturbation, to trajectories to get *perturbed trajectories*, which are less vulnerable to privacy threats; and (ii) a *compensator* that warps the shaky rendered viewports caused by perturbed trajectories back to normal, compensated viewports. Note that the disturbed and compensator are executed on trusted devices with access to (original) trajectories, while only perturbed trajectories are sent over the Internet. Our disturber employs

[1] We acknowledge that VR users may interact with applications using other input modalities. Our proposed solutions can be applied to those modalities in follow-up studies.

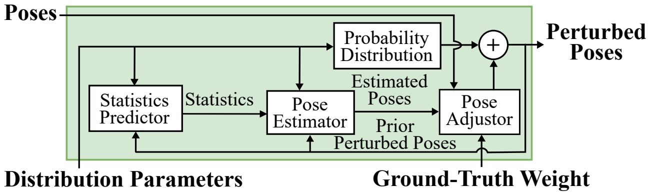


Fig. 2. Our proposed trajectory disturber.

differential privacy [5] to protect each VR user within a set of VR users, called population, from attackers. Differential privacy offers a control knob ϵ for trading off individuals' privacy and population's statistics. In general, higher ϵ values lead to more accurate statistics at the expense of revealing more privacy of individuals. With the guarantees from differential privacy by selecting proper ϵ values, even if an attacker gathers perturbed trajectories from the population, he/she still has a hard time launching attacks.

This paper makes the following contributions:

- (Sec. III-A) We develop a disturber to add perturbations to a time series of poses on the fly to mitigate privacy threats. Prior works either ignored the temporal correlation [23, 12] or added perturbations offline [14, 28, 4, 3].
- (Sec. III-B) We create a compensator to warp rendered viewports to eliminate the shakiness caused by perturbations introduced by the disturber. Doing so allows us to avoid visual quality degradation in VR applications.
- (Sec. IV) We exemplify privacy threats by realizing an ML algorithm for the *re-identification problem* [22] as a case study to evaluate the effectiveness of our solution. A re-identification attack refers to recognizing the user's identity by analyzing his/her trajectories[2]. We conduct extensive evaluations using a public VR dataset [35] captured from 3D virtual worlds. Our evaluation results reveal that: (i) our disturber alone reduces at most 0.42 re-identification rate compared to the state-of-the-art MetaGuard [23] under the same ϵ values, (ii) our disturber alone outperforms the state-of-the-art MetaGuard [23] by 2.43 dB in Peak Signal-to-Noise Ratio (PSNR), 0.13 in Structural Similarity (SSIM), and 8.15 in Video Multimethod Assessment Fusion (VMAF) under the same privacy-preserving settings, and (iii) compared to the disturber-only solution, our compensator further improves the visual quality by at most 6.83 dB in PSNR, 0.45 in SSIM, and 34.57 in VMAF.

II. Related work

To preserve viewers' privacy, perturbations have been added to eye gaze traces collected from HMDs. Most prior studies [28, 4, 3] focused on extracted gaze features, such as fixations, saccades, and blinks. For example, Steil et al. [28] collected an eye-tracking dataset and added noises to some extracted features using exponential differential privacy. Different from these works that considered eye gazes only, our paper considers trajectories from VR users' HMDs.

Adding perturbations to non-eye-gaze data from VR users has only been recently explored. For example, Wei et al. [34] requested noisy tiles close to a VR user's viewport when watching 360° videos. They then empirically derived the tradeoff between viewport prediction accuracy and viewing experience. Such an approach, however, is tightly coupled with VR applications, i.e., 360° tiled video streaming. MetaGuard [23] is probably the closest work to ours, which They added perturbations to HMD trajectories using Laplace differential privacy. However, they did not take the temporal correlation among poses in each trajectory into account when adding perturbations. Moreover, they did not measure how users' experience is affected by perturbing VR trajectories. In contrast to their work, we: (i) consider the temporal correlation among the poses and (ii) introduce a compensator to recover from the shaky rendered viewports.

III. Proposed Solution to Mitigate Privacy Threats

In this section, we first present our trajectory disturber. We then introduce our view compensator.

A. Trajectory Disturber

Design objectives. The purpose of the disturber is to perturb a VR user's trajectory in both temporal and spatial domains on the fly. Doing this in differential privacy is *inherently* challenging because the disturber has no knowledge of the required statistics from the whole trajectory, as some poses happen in the future. Therefore, we need to predict the statistics of the whole trajectory based on prior poses. Another challenge is to find a *good* tradeoff between the incurred perturbations and the degraded visual quality. Hence, we introduce system parameters as control knobs for VR users to exercise the tradeoff, as detailed below.

Overview. Fig. 2 gives the design of our disturber, which sequentially takes *poses* as input and generates a series of *perturbed poses*. Two system parameters, *distribution parameters* and *ground-truth weight*, are used to control the distribution and severity of random perturbations. More specifically, the disturber is composed of four components: (i) probability distribution, (ii) statistics predictor, (iii) pose estimator, and (iv) pose adjuster. Here, the *probability distribution* generates random noise that is added to poses as perturbations. Different probability distributions have been adopted for differential privacy, including Laplace, Gaussian, and Binomial [5]. Each probability distribution takes one or multiple distribution parameters, like the variance. We use the Laplace distribution throughout this paper if not otherwise specified. In addition, the *statistics predictor* predicts the statistics, such as mean and autocorrelation, of the whole trajectory based on prior perturbed poses. These statistics are dictated by the differential privacy and used as input by the *pose estimator*, which estimates the current pose using prior perturbed poses.

[2] Our solution can be readily generalized to attacks on other user attributes, such as height, weight, and gender.

The estimated pose is inevitably inaccurate compared to the actual one. The *pose adjuster* takes a system parameter called ground-truth weight and fuses the ground-truth and estimates poses. Last, random noise is added to the adjusted pose for the perturbed one. Among these four components, the probability distribution introduces perturbations in the spatial domain, while the pose estimator does that in the temporal domain.

Notations. Next, we present the operations of the disturber while developing notations. Each pose or perturbed pose at the moment is a 21-dimension vector containing the positions and orientations of an HMD and two controllers. Each dimension is perturbed independently, and we omit some technical math details for brevity in the following. We denote the (input) pose and (output) perturbed pose at time t as V_t and P_t, respectively. The poses and perturbed poses over a time duration with T samples are denoted as $\{V_t\}_{t=1}^T$ and $\{P_t\}_{t=1}^T$. We consider three statistics of the whole trajectory: mean μ, variance σ^2, and autocorrelation ρ. At time t, these statistics are predicted by the statistics predictor given the Laplace distribution parameter var and prior perturbed poses, $\{P_1, P_2, \dots, P_{t-1}\}$. These statistics are fed into the pose estimator for an estimated pose $\hat{V}_t$. $\hat{V}_t$ and V_t are passed to the pose adjuster to be fused into an adjusted pose $A(\hat{V}_t)$ with the ground-truth-weight w. $A(\hat{V}_t)$ has incorporated the temporal-domain perturbations. At time t, a random noise $n_t \sim L(0, var)$ is used to generate the spatial-domain perturbations, where $L(0, var)$ is a zero-mean Laplace random variable. Last, we sum $A(\hat{V}_t)$ and n_t up to get P_t.

Procedure. For tractability, we use a classical linear model to build our pose estimator. Many linear models have been proposed, such as the AutoRegressive (AR), Moving Average (MA), and AutoRegressive Moving Average (ARMA) models [15]. Among them, we choose to model each trajectory as a Gaussian AR process [33] following Zhang et al. [36], and compute the estimated pose $\hat{V}_t$ by:

$$\hat{V}_t = \mu(1 - \rho\frac{\sigma^2}{\sigma^2 + var}) + \rho\frac{\sigma^2}{\sigma^2 + var}P_{t-1}, \tag{1}$$

where P_{t-1} is the preceding (input) pose, var is the distribution parameter, and μ, σ, ρ are statistics of the whole trajectory. Unfortunately, μ, σ, and ρ cannot be computed at time t, and we let $\hat{\mu}_{t-1}$, $\hat{\sigma}_{t-1}$, and $\hat{\rho}_{t-1}$ be the predicted statistics. Among them, $\hat{\mu}_{t-1}$ and $\hat{\sigma}_{t-1}$ can be calculated using $\{P_1, P_2, \dots, P_{t-1}\}$ and $\hat{\rho}_{t-1}$ can be computed following Huitema and McKean [9]. Applying the predicted statistics to Eq. (1), we have:

$$\hat{V}_t = \hat{\mu}_{t-1}(1 - \hat{\rho}_{t-1}\frac{\hat{\sigma}^2_{t-1}}{\hat{\sigma}^2_{t-1} + var}) + \hat{\rho}_{t-1}\frac{\hat{\sigma}^2_{t-1}}{\hat{\sigma}^2_{t-1} + var}P_{t-1}. \tag{2}$$

Inspired by Zhang et al. [36], we compute adjusted pose $A(\hat{V}_t)$ as the following weighted sum:

$$A(\hat{V}_t) = (1 - w)\hat{V}_t + wV_t. \tag{3}$$

Note that by keeping w secret, we create additional burdens for attackers. With the output of the probability distribution, $n_t \sim L(0, var)$, we write the perturbed pose as:

$$P_t = A(\hat{V}_t) + n_t = (1 - w)\hat{V}_t + wV_t + n_t, \tag{4}$$

which concludes our disturber design.

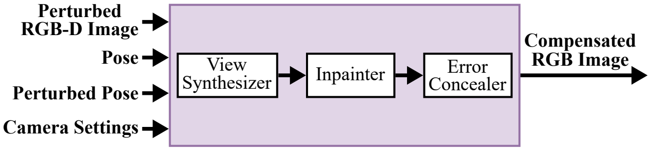


Fig. 3. Our proposed view compensator.

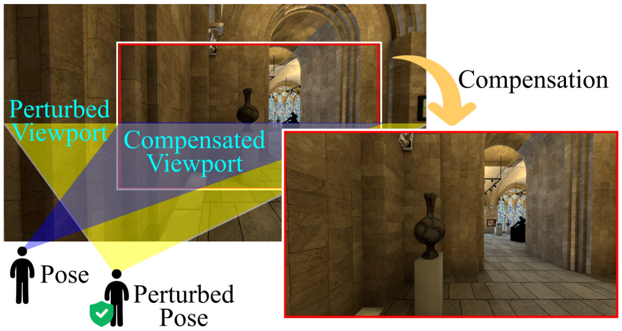


Fig. 4. Sample compensated view to avoid shaky viewports due to perturbations.

B. View Compensator

Design objectives. The purpose of the view compensator is to warp each RGB-D image rendered by a VR application with the perturbed pose P_t to an RGB image viewed at the (original) pose V_t. There exist two design objectives for the compensator: (i) high visual quality and (ii) short execution time. Fortunately, perturbations imposed on trajectories are rather small and controllable (via var and w in our proposed disturber), which can be compensated by image warping [16], a.k.a. view synthesis. Our goal is to find a fast synthesizer achieving reasonable visual quality.

Design choices. View synthesizers warp pixels from one or multiple input RGB-D images to a VR user's HMD viewport based on the D (depth) channel. There are quite a few Depth Image-Based Rendering (DIBR)-based [27, 10, 11, 25] and neural-network-based [20, 1, 2, 21] view synthesizers. For a shorter running time, we opt for DIBR-based synthesizers as neural-network-based ones typically take seconds, if not minutes, to synthesize a single frame. Furthermore, neural-network-based synthesizers are trained with limited datasets and may not generalize to other VR applications. Following the findings revealed in Fachada et al. [7] and Sun et al. [29], we build our compensator on Reference View Synthesizer (RVS) [11] for a good tradeoff between visual quality and execution time, while other synthesizers can be easily dropped in if needed.

Overview. Fig. 3 gives the high-level workflow of our compensator. It takes the following inputs: (i) perturbed RGB-D image from the rendering engine, (ii) (original) pose, (iii) perturbed pose, and (iv) camera settings. The key camera settings are the two *Field-of-Views (FoVs)* of the *perturbed* and *compensated viewports*. Fig. 4 reveals the relation between these two viewports. To maintain the quality of the compensated viewport, the perturbed FoV *must* be larger than the compensated FoV, so that most pixels in the compensated viewport fall in the perturbed viewport. In our compensator, we employ the same resolution of input and output RGB(-D) images, denoted as $W \times H$, where W stands for the width and H represents the height of images. We let θ_p and θ_c be the vertical FoVs of perturbed and compensated viewports, respectively. Their horizontal FoVs are calculated by the vertical FoVs with a constant aspect ratio $\frac{W}{H}$.

Procedure. The compensator consists of three components: view synthesizer, inpainter, and error concealer. The *view synthesizer* warps each input RGB-D image utilizing the computed disparity between each (original) pose and its perturbed pose. The next two components handle the exceptions. In particular, the *inpainter* fills in the blank pixels that are disoccluded by interpolating with the surrounding pixels. The *error concealer* deals with missing output RGB images by replaying the preceding successfully compensated RGB image. Missing output RGB images could happen when the compensated viewport falls outside of the perturbed viewport, which may be caused by large perturbations. Another possibility is the input RGB-D images from rendering engines contain imperfect depth values due to occlusions in complex 3D scenes.

IV. Experiments

In this section, we start from a description on our implementation. It is followed by objective and then subject tests.

A. Implementations

We have implemented both our disturber and compensator, as depicted in Figs. 2 and 3, on a VR application that allows VR users to explore a scene rendered by Unity [31]. The disturber was written in Python. It employs Math.NET Numerics [18] from NuGetForUnity [19] to implement the Laplace distributions and adds the random variables to the (original) trajectories. Furthermore, we have implemented our compensator on top of the RVS reference software [6] from MPEG. We captured the trajectories, images, and camera settings–like position, orientation, and FoV–from Unity using Python scripts and FFmpeg [30]. The data files were then sent to the compensator, which subsequently generated the final viewports as the output for visual quality assessment. That is, we execute RVS offline, as it is not optimized for real-timeness.

B. Objective Test Setup

In our experiments, we utilized a 6DoF VR dataset [35] with 24 VR users' trajectories to drive our experiments. In this dataset, VR users continuously traverse through four 3D scenes, which are *City*, *Gallery*, *Nature*, and *Office* [35]. Users can freely interact with the scenes with interactable objects, such as cars, furniture, and plants.

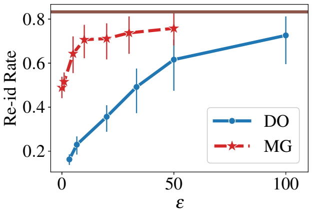


Fig. 5. Re-identification rate under different ϵ.

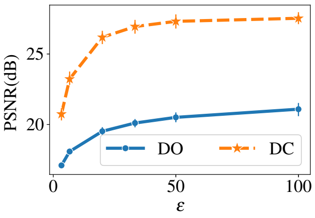
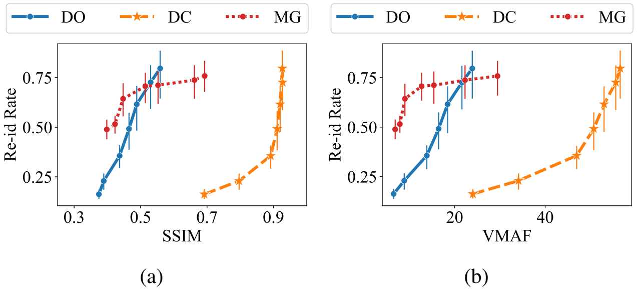


Fig. 6. Visual quality with and without compensation.



(a)

(b)

Fig. 7. Privacy-quality tradeoff achieved by different privacy-threat mitigation approaches, where visual quality is in: (a) SSIM and (b) VMAF.

We compare two of our proposed solutions, *Disturber Only (DO)* and *Disturber with Compensator (DC)*, against the state-of-the-art MetaGuard (MG) [23]. Notice that, MetaGuard adds perturbations to multiple attributes (rather than trajectories directly), and then projects the perturbed attributes back to perturbed locations (rather than trajectories). Since MetaGuard only adds perturbations to locations, we do the same for our proposed solution for meaningful comparisons. In MetaGuard, we consider attributes that are more relevant to VR user locations, i.e., height, room size, squat depth, wingspan, and arm length, and apply $\epsilon \in \{0.000001, 1, 5, 10, 20, 30, 50\}$ to these attributes. For our solutions, based on some pilot tests, we select $\epsilon \in \{3.33,\ 6.67,\ 20,\ 33.33,\ 50,\ 100,\ 200\}$. These ϵ values are chosen to roughly align the resulting visual quality from our solutions and MetaGuard. Our solutions take a few additional parameters: (i) ground-truth weight $w \in \{0.1,\ \mathbf{0.3}\}$, (ii) perturbed FoV $\theta_p \in \{\mathbf{115}^\circ,\ 125^\circ\}$, and (iii) compensated FoV $\theta_c = \mathbf{104}^\circ$, where bold font indicates the default values.

We consider three popular visual quality metrics: Peak Signal-to-Noise Ratio (PSNR) [8], Structural Similarity (SSIM) [8], and Video Multimethod Assessment Fusion (VMAF) [24]. We compare the quality of VR users' HMD viewports from DO, DC, and MG against the ground-truth ones without perturbations. Computing visual quality for all VR users is prohibitively time-consuming. Hence, we chose six diverse users for each of the four scenes to compute their visual quality. In particular, for each scene, we calculate the moving distances of individual users. In ascending order, we sorted the users based on their moving distances and then chose the 1st, 6th, 10th, 15th, 19th, and 24th users to approximate the overall visual quality.

We implemented a Random Forest (RF) based re-

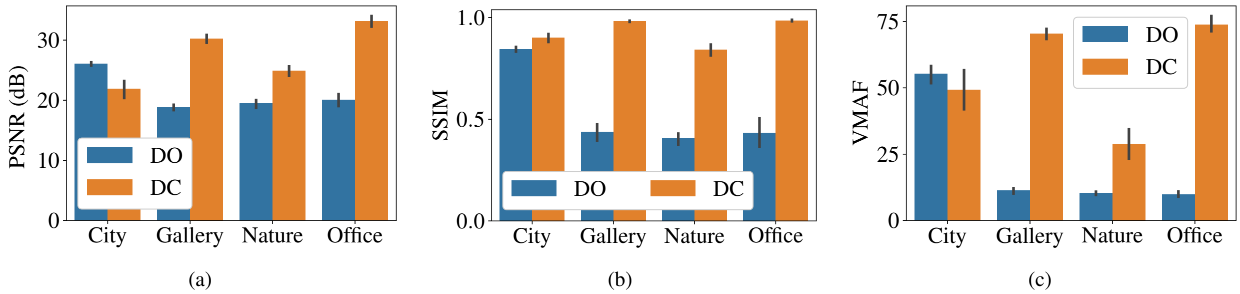


Fig. 8. Visual quality comparison among the four considered scenes: (a) PSNR, (b) SSIM, and (c) VMAF.

identification classifier to evaluate the protection provided by different mitigation solutions. We consider trajectories with longer than 2000 poses in our experiments (about 4% of trajectories were removed) and apply a 50-pose sliding window (equivalent to 1 s, as the dataset was collected at 50 Hz). By doing so, each trajectory can be turned into 1956 feature vectors, which are divided into five folds for cross-validation. We select 75 features from prior-arts [22, 17, 13] on user identification and authentication in VR applications, including: (i) the velocity and angular velocity of each HMD and its controllers, (ii) the minimum, average, and maximum distances between each HMD and its controllers, (iii) the minimal, mean, and maximal locations/orientations of each HMD and its controllers. Through some pilot tests, we empirically set the number of estimators to be 150 and the maximum depth to be 15.

We run all experiments on a workstation with an Intel i9-9920X CPU, 64 GB RAM, and an NVIDIA GeForce RTX 3080 Ti GPU. We give average performance results with 95% confidence intervals whenever possible. For re-identification (re-id), rates across five runs of the five-fold cross-validations are reported for statistically meaningful results.

C. Objective Test Results

Our disturber leads to lower re-identification rates. Fig. 5 shows the re-id rates of DO and MG under different ϵ values. The re-id rate without any privacy-threat mitigation is 0.83, as shown as the horizontal dashed line in the figure. Compared to the state-of-the-art MG, our DO reduces the re-id rate by up to 0.42. Such improvement may be attributed to the fact that the re-identification classifier considers temporal correlation in trajectories, which is not protected by MG. We conclude that perturbing the trajectories in the temporal domain preserves more user privacy. Note that when ϵ approaches 0, the re-id rate of DO also approaches 0.1; in contrast, the re-id rate of MG is still above 0.45.

Our compensator mitigates the degradation of visual quality due to perturbations. Fig. 6 compares the overall visual quality in PSNR from DO and DC across all four 3D scenes under different ϵ values. We observe that our compensator improves the visual quality by 6.83 dB at most and 5.87 dB on average. Although their figures are omitted due to space limitation, the SSIM and VMAF boosts are up to 0.45 and 34.57, respectively, which are significant. Hence, we conclude that our compensator successfully improves the visual quality.

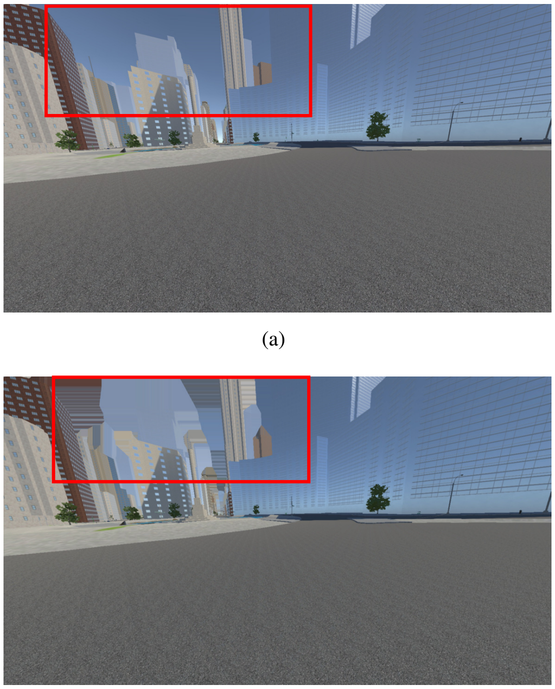
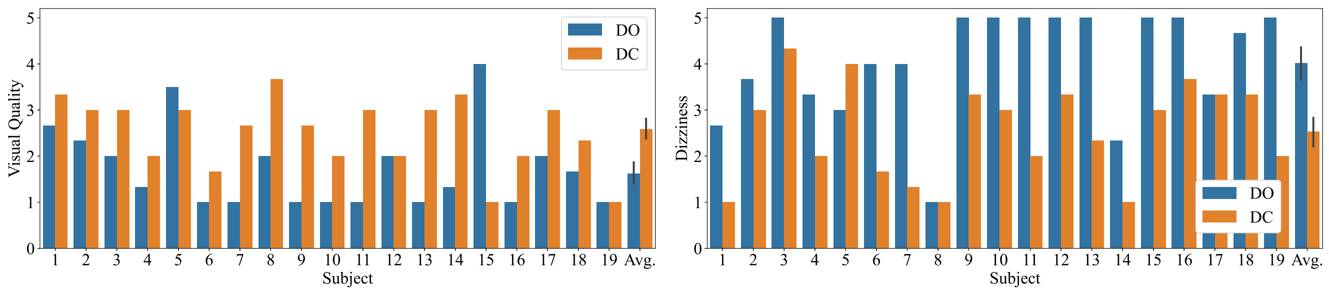


Fig. 9. Rendered sample viewports from City scene: (a) with and (b) without compensation.

Our solution provides strong protection while delivering good visual quality. We plot the *privacy-quality tradeoff* in Fig. 7, which maps the privacy level to visual quality. This figure shows that DO achieves better visual quality than MG when the re-id rate is between 0.10 and 0.73. Even though MG delivers better visual quality when the re-id rate is higher than 0.73; such high re-id rates reveal that MG is quite vulnerable to privacy attacks. In fact, DO lowers the re-id rate by almost half compared to MG under the same visual quality. We also observe that with compensation, DC achieves both strong privacy protection while achieving very good visual quality. Compared to DO, DC improves by up to 6.83 dB in PSNR (PSNR figure is not shown due to the space limit), 0.45 in



(a)

(b)

Fig. 10. Mean opinion scores of subjects across all scenes: (a) visual quality, where higher is better, and (b) dizziness, where lower is better.

SSIM, and 34.57 in VMAF at the same re-id rate. The curves reported here validate the excellent privacy-quality tradeoff of our solutions.

Implications of diverse characteristics of 3D scenes. Fig. 8 shows the quality of each scene using $\epsilon = 100$ with and w/o compensation. Considering DO, among the four scenes, the visual quality of City outperforms the others at all times. This is due to the large size of the City scene (which is $128 \times 50 \times 128\ m^3$), including wide roads and a vast sky, in which the nearby pixels are more similar as shown in Fig. 9. Therefore, the difference between the original and the perturbed images of City is small, leading to better visual quality than other scenes. Interestingly, applying compensation to City lowers the visual quality. This is because there are many buildings in front of the sky. RVS uses the texture of adjacent pixels for inpainting. Most of the disoccluded sky areas in City are inpainted with texture from the buildings, as shown with red boxes in Fig. 9, causing degraded visual quality. In contrast, the compensator improves the visual quality of all three other scenes. This reveals that a geometry-aware compensator would further improve the visual quality, which is among our future works.

D. Subjective Test Setup

We also conducted a user study to determine the effectiveness of our compensator with 19 subjects. The subjects are between 19–22 years old, and five of them are female. Fifteen of them have used HMDs before. We generate and record the HMD viewports into videos based on our perturbed trajectories, with a medium $\epsilon = 33.33$. All four scenes are considered, hence there are four DC and four DO videos for each subject to watch. We use a laptop with a 13.3-inch monitor and synchronously play the ground-truth video along with impaired video side-by-side (from either DO or DC). The order of the two videos is random. After watching each pair of videos, we ask each subject to answer the following questions:

- Which viewport is worse in overall quality?
- How would you rate the worse viewport's user experience in visual quality? Between 1 (unacceptable) and 5 (as good as the better one).
- How is the dizziness when you watch the worse viewport? 1 (none) to 5 (severe).

We filter out outlier scores by checking the correctness of the first question. Particularly, 1.78% of the scores were dropped.

E. Subjective Test Results

Our compensator successfully improves perceived user experience. Fig. 10 reports the mean opinion scores of individual users across all scenes in our user study. This figure reveals that after performing compensation, the subjective visual quality and dizziness scores are both improved. Only two subjects report that DC is worse than DO (10.53%) in visual quality, and only one subject reports DC is worse than DO in dizziness (5.26%). In summary, the mean opinion scores for visual quality are increased by 0.6 (out of 5) on average, and the scores for dizziness are decreased by 1.45 (out of 5), showing that our compensator can improve VR user experience.

V. Conclusion

In this paper, we proposed a privacy-threat mitigation approach to protect VR users with HMDs and controllers. We achieve that in two steps. First, we developed a disturber to add perturbations in both temporal and spatial domains to the trajectories. Next, we built a compensator to warp rendered viewports of perturbed poses to eliminate the shakiness caused by the perturbations. Extensive objective and subjective experiments demonstrated the merits of our disturber and compensator: they achieved an excellent tradeoff between privacy and visual quality. In particular, our evaluation results showed that our disturber alone reduced at most 0.42 re-identification rate compared to the state-of-the-art MetaGuard [23] while improving the visual quality by 2.43 dB in PSNR, 0.13 in SSIM, and 8.15 in VMAF. Additionally, our user study revealed the effectiveness of our compensator: it improved the mean opinion score of visual quality by 12% and that of dizziness by 29%.

Our work can be extended in multiple dimensions. For instance, more comprehensive linear and non-linear models can be adopted by the pose estimator in the disturber, and multiple RGB-D images from different locations/orientations can be leveraged for view synthesis to avoid artifacts caused by occlusions in the compensator.

References

[1] J. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV'21)*, pages 5855–5864, Montreal, Canada, October 2021.

[2] J. Barron, B. Mildenhall, D. Verbin, P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pages 5470–5479, New Orleans, LA, 2022.

[3] E. Bozkir, O. Gunlu, W. Fuhl, R. Schaefer, and E. Kasneci. Differential privacy for eye tracking with temporal correlations. *Plos One*, 16(8):1–22, August 2021.

[4] B. David, K. Butler, and E. Jain. For your eyes only: Privacy-preserving eye-tracking datasets. In *Proc. of ACM Symposium on Eye Tracking Research & Applications*, pages 1–6, Seattle, USA, June 2022.

[5] C. Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12, 2006.

[6] S. Fachada, D. Bonatto, A. Schenkel, B. Kroon, and B. Sonneveldt. RVS Software, 2023. https://gitlab.com/mpeg-i-visual/rvs/-/tree/master.

[7] S. Fachada, D. Bonatto, A. Schenkel, and G. Lafruit. Free navigation in natural scenery with DIBR: RVS and VSRS in MPEG-I standardization. In *Proc. of International Conference on 3D Immersion*, pages 1–6, Brussels, Belgium, 2018.

[8] A. Horé and D. Ziou. Image quality metrics: PSNR vs. SSIM. In *Proc. of IEEE International Conference on Pattern Recognition*, pages 2366–2369, Istanbul, Turkey, August 2010.

[9] B. Huitema and J. McKean. Autocorrelation estimation and inference with small samples. *Psychological Bulletin*, 110(2):291–304, 1991.

[10] J. Jung and P. Boissonade. VVS: Versatile View Synthesizer for 6-DoF Immersive Video. working paper or preprint, 2020.

[11] B. Kroon and G. Lafruit. Reference view synthesizer (RVS) 2.0 manual, 2018.

[12] J. Li, A. Roy, K. Fawaz, and Y. Kim. Kalεido: Real-time privacy control for eye-tracking systems. In *Proc. of USENIX Security Symposium*, pages 1793–1810, Virtual, August 2021.

[13] J. Liebers, M. Abdelaziz, L. Mecke, A. Saad, J. Auda, U. Gruenefeld, F. Alt, and S. Schneegass. Understanding user identification in virtual reality through behavioral biometrics and the effect of body normalization. In *Proc. of ACM CHI Conference on Human Factors in Computing Systems*, pages 1–11, Yokohama, Japan, May 2021.

[14] A. Liu, L. Xia, A. Duchowski, R. Bailey, K. Holmqvist, and E. Jain. Differential privacy for eye-tracking data. In *Proc. of ACM Symposium on Eye Tracking Research & Applications*, pages 1–10, Denver, USA, June 2019.

[15] Z. Liu, Z. Zhu, J. Gao, and C. Xu. Forecast methods for time series data: A survey. *IEEE Access*, 9:91896–91912, June 2021.

[16] W. Mark. *Postrendering 3D Image Warping: Visibility, Reconstruction, and Performance for Depth-image Warping*. PhD thesis, The University of North Carolina, 1999.

[17] F. Mathis, H. Fawaz, and M. Khamis. Knowledge-driven biometric authentication in virtual reality. In *Proc. of ACM CHI Conference on Human Factors in Computing Systems*, pages 1–10, Honolulu, USA, April 2020.

[18] Math.NET. Math.net numerics, 2023. https://numerics.mathdotnet.com/.

[19] P. McCarthy. Nugetforunity, 2023. https://tinyurl.com/3u446efc.

[20] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, December 2021.

[21] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4), July 2019.

[22] M. Miller, F. Herrera, H. Jun, J. Landay, and J. Bailenson. Personal identifiability of user tracking data during observation of 360-degree VR video. *Scientific Reports*, 10(1):1–10, October 2020.

[23] V. C. Nair, G. Munilla-Garrido, and D. Song. Going incognito in the Metaverse: Achieving theoretically optimal privacy-usability tradeoffs in vr. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.

[24] Netflix. VMAF - video multi-method assessment fusion, 2021.

[25] B. S. Salahieh, J. Jung, and A. Dziembowski. Test model 10 for mpeg immersive video. July 2021.

[26] C. Slocum, Y. Zhang, N. Abu-Ghazaleh, and J. Chen. Going through the motions: AR/VR keylogging from user head motions. In *Proc. of USENIX Security Symposium*, Anaheim, USA, August 2023.

[27] O. Stankiewicz, K. Wegner, M. Tanimoto, and M. Domański. Enhanced view synthesis reference software (vsrs) for free-viewpoint television. 2013.

[28] J. Steil, I. Hagestedt, X. Huang, and A. Bulling. Privacy-aware eye tracking using differential privacy. In *Proc. of ACM Symposium on Eye Tracking Research & Applications*, pages 1–9, Denver, USA, June 2019.

[29] Y.-C. Sun, S.-M. Tang, C.-T. Wang, and C.-H. Hsu. On objective and subjective quality of 6DoF synthesized live immersive videos. In *Proc. of ACM Workshop on Quality of Experience in Visual Multimedia Applications*, pages 49–56, Lisboa, Portugal, October 2022.

[30] S. Tomar. Converting video formats with FFmpeg, 2006. https://tinyurl.com/3tr3nuvt.

[31] Unity. Unity, 2023. https://unity.com/.

[32] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. Luan, and X. Shen. A survey on Metaverse: Fundamentals, security, and privacy. *IEEE Communications Surveys & Tutorials*, 25(1):319–352, September 2022.

[33] W. Wei. *Time Series Analysis*, volume 2. Oxford University Press, 2013.

[34] X. Wei and C. Yang. FoV privacy-aware VR streaming. In *Proc. of IEEE Wireless Communications and Networking Conference*, pages 1515–1520, Austin, USA, April 2022.

[35] Y.-S. Wei, X. Wei, S.-Y. Zheng, C.-H. Hsu, and C. Yang. A 6DoF VR dataset of 3D virtual world for privacy-preserving approach and utility-privacy tradeoff. In *Proc. of ACM Multimedia Systems*, pages 444–450, Vancouver Canada, June 2023.

[36] X. Zhang, M. Khalili, and M. Liu. Differentially private real-time release of sequential data. *ACM Transactions on Privacy and Security*, 26(1):1–29, November 2022.