Adaptive Mobile Diminished Reality Framework for 3D Visual Privacy

Salam Tabet Ayman Kayssi Imad H. Elhajj
Department of Electrical and Computer Engineering
American University of Beirut
Beirut, Lebanon
sat22@mail.aub.edu, {ayman, ie05}@aub.edu.lb

2024 International Conference on Intelligent Systems and Computer Vision (ISCV) | 979-8-3503-5018-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCV60512.2024.10620084

***Abstract*—Privacy refers to the right of individuals to keep their personal information and activities confidential, limiting access to others. In the context of images and videos, it is referred to as visual privacy. Visual privacy is more than a personal preference; it is an essential component of human dignity, security, and well-being. Recently, immersive technologies: Augmented Reality/Virtual Reality/Mixed Reality (AR/VR/MR), have gained surging popularity. Current research is focused on making these nascent experiences more engaging while overlooking the privacy concerns raised by having their cameras continually capture the user's environment. State-of-the-art visual privacy-preserving techniques focus on anonymizing users, or protecting certain types of information, like faces or bystanders, in a 2-dimensional setting where private information is always considered to be in the background. The few works that provide 3D privacy follow the least privilege paradigm, where only the minimal information required to perform a certain task is released. Moreover, previous frameworks struggle to maintain good performance during significant camera motion. We propose a mobile real-time 3D visual privacy-preserving Diminished Reality (DR) system that adapts to motion patterns. The system tracks the user's movements and modifies the size of the obfuscation region in a Multiplicative Increase Additive Decrease (MIAD) manner. When rapid movements are sensed, the scale of the obfuscated region is increased rapidly (multiplicatively), whereas when the movements are slow, the scale is decreased slowly (additively) to minimize the risk of exposing any part of the private object. Our approach is evaluated during translational and rotational motion patterns, and it significantly reduced the average number of privacy failures from over 18% to less than 1% compared to the state-of-the-art.**

***Index Terms*—Diminished Reality, Mixed Reality, Augmented Reality, immersive technologies, fast motion, adaptation, tracking, real-time, visual privacy, 3D obfuscation**

I. Introduction

We live in a digital world where we expose too much of our personal lives. The amount of visual data collected is proliferating by the second. As of 2023, every year, a staggering 1.8 trillion photos are captured globally, translating to over 57,000 photos per second, or an astonishing 5 billion photos per day [1]. Imagine the amount of privacy-sensitive information present in those images.

Emerging visual data-collecting technologies that have gained a surge in popularity recently are immersive technologies: Augmented Reality/Virtual Reality/Mixed Reality (AR/VR/MR). With an enormous market size of 22.5 billion dollars in 2022, the immersive technology market is attracting more and more customers and is estimated to reach approximately 168 billion dollars by 2032 [2].

The increasing pervasiveness of these technologies, specifically AR and MR applications, poses privacy risks, especially since AR cameras are continually capturing information about users' surroundings, including people and personal places [3]–[5]. Since these technologies are still in their nascent stages, current work is focused on making immersive experiences more realistic and engaging without much considerations for the privacy implications of these technologies. Google, for instance, is working on making remote meetings more realistic through its Starline Project [6]. Apple just released its first MR headset [7] at the time of writing, with promises of seamlessly blending the real and virtual worlds.

A typical risky setting for these application environments might include a credit card on a table at home, a secret contract on an office desk, or an industrial robot in the background. This sensitive information might be readily obtained by third-party apps with access to the user's MR system and abused in inference attacks [8]. As a result, there is a need to provide seamless yet selective sharing capabilities in MR systems, analogous to popular video conference's selective screen-sharing.

Most previous privacy-preserving frameworks are focused on providing face de-identification [9], [10], or individual anonymization [11], [12]. However, anonymization of the user is not applicable in all use cases. Moreover, they either focus on identifying and obfuscating specific types of objects [13], or they are not tailored for 3D environments, always assuming the privacy-sensitive information is in the background [13], [14]. The few works that provide 3D privacy follow the least privilege paradigm, where only the minimal information required to perform a certain task is released [15]–[17]. Moreover, existing frameworks struggle with maintaining acceptable performance during camera motion [13], [18].

Our previous work [19] presented a 3D visual privacy-preserving Diminished Reality (DR) framework for mobile devices that provides support for real-time MR applications. Our framework focused on obfuscating user-selected sensitive information by extracting and tracking a region of interest across frames. We also presented the "Snap-Back" algorithm

979-8-3503-5018-0/24/$31.00 ©2024 IEEE

for enhanced privacy guarantees and continuous obfuscation during 3D rotation around the sensitive object. However, it had limitations when dealing with sudden or fast movements. In this paper, we address this limitation and propose an adaptive algorithm on top of our previous DR system that monitors the camera motion and adapts the size of the obfuscation area by a combination of Multiplicative Increase Additive Decrease (MIAD) and weighted moving average according to the speed and acceleration of the camera in real time, to enhance privacy guarantees even during fast and sudden movements. When rapid movements are sensed, the scale of the obfuscated region is increased sharply (multiplicatively), whereas when the movements are slow, the scale is decreased slowly (additively) to minimize the risk of exposing any part of the private object. A weighted moving average is then applied to the scale for a smoother transition to improve the user experience.

The rest of the paper is organized as follows: Section II describes related work. Sections III and IV describe our system's pipeline and implementation details. Section V shows and discusses the results obtained, and Section VII provides some conclusions.

II. Related Work

A. Privacy Frameworks

The goal of privacy protection is to keep information that a person wishes to remain private out of the public domain. When referring to images and videos, we call it visual privacy protection [20]. Several techniques for protecting visual privacy have emerged; redaction techniques, which attempt to identify and alter sensitive areas of an image, such as faces, bodies, and number plates, remain the most popular ones [20].

Several methods have been proposed to alter the sensitive parts of an image, that evolved from mere black masks, blurring, or pixelating to more sophisticated and time-consuming transformer-based inpainting methods [21], [22] that can delete objects in the image while preserving its realism.

Another redaction strategy is through face de-identification techniques that focus on altering face images in a way that makes the shown faces unrecognizable. Recent de-identification works have proposed reversible models, allowing authorized parties to recover the correct original faces [9], [10]. Faces, however, are not the only sources of privacy-sensitive information.

Other privacy-preserving frameworks also focused on anonymizing individuals, but not only through altering their faces, such as by replacing them with a wire-frame representation for behavioral analysis [11], [12], or by producing a completely different output video that includes only the information needed for action recognition [23]. However, anonymizing individuals is not always a viable solution, as the identity of the individual needs to be revealed in certain use cases, like in applications of telepresence, the Metaverse, remote control and guidance, and video conferencing.

In a video conferencing context, Chiu et al. proposed in [14] a Generative Adversarial Network (GAN) that synthesizes the user's appearance by capturing real-time low-resolution thermal images instead of real ones, along with a privacy-free image, supplied beforehand of the real scene.

Other works focused on identifying and obfuscating privacy-sensitive regions in the context of AR. The model proposed in [24] focuses on automatically detecting the privacy-sensitive regions in an AR frame, by relying on salient object detection. Also, BYSTANDAR, was proposed by Corbett et al. in [13], as an on-device real-time approach to enhance bystander protection in the AR context. However, not only is it focused on protecting the privacy of bystanders, but it is also very sensitive to motion, where its privacy protection rates drop significantly when the device or bystander is moving. Moreover, those systems are two-dimensional, assuming the private object/bystander is always in the background.

On the other hand, several methods for privacy preservation in a three-dimensional context were proposed, where only the minimal information required for proper functionality is released, i.e., following a least privilege paradigm. In [15], Speciale et al. proposed substituting 3D point clouds with 3D line clouds for spatial map representations, enabling camera pose estimation while concealing the underlying geometry of the environment. Guzman et al. proposed limiting the number of planes released in an MR context in [16] while focusing on enabling the MR anchoring utility. Nama et al. proposed regenerating the 3D point clouds of a specific 3D object, hiding its details in the process while preserving its main attributes for a proper utility like the size of its bounding box [17].

Our proposed visual privacy-preserving DR framework is not specific to faces, humans, or a certain type of object. It provides the user the ability to select and obfuscate any private objects in 3D either by blurring or while preserving the realism of the 3D environment by inpainting in real-time, making it suitable for applications that require an MR live feed, such as video conferencing. Our framework focuses on preserving 3D visual privacy in a 3D environment, where the camera is moving freely.

B. Diminished Reality Frameworks

While AR augments virtual objects into the real world, DR aims to do the exact opposite, enabling users to see-through or even completely remove real and virtual objects from their perceived environment [25].

Various DR frameworks have been created for different use cases, including indoor planning [26], reducing distractions during learning [27] or even attempting to raise one's self-esteem through providing an invisible human experience [28].

Most previous DR systems require previous knowledge of the environment to diminish real objects, either by scanning the room beforehand [29], or by using multiple viewpoints [30], [31]. Some DR systems even require offloading the processing of the frames to a back-end server [32].

Other DR systems were designed to diminish specific types of objects. Kim et al. proposed an online mobile DR system in [18] for detecting and removing real-world human figures from the user's perceived view. However, not only is the detection

and mask generation implemented for pedestrians only, but the system's performance is also sensitive to the camera motion.

Our DR system is specifically designed to run on a mobile device's front end in real-time, to preserve the 3D visual privacy of the user as they move freely in the environment, without any limits on their speed/acceleration, and without requiring any form of prior knowledge of the background.

III. System Design

A. Original System Pipeline

Our adaptive algorithm is implemented and tested on top of the initial DR system pipeline, along with the "Snap-Back" algorithm presented in [19], which allows the user to select an object or area to obfuscate across frames. The pipeline, shown in Fig. 1, consists of the following stages:

1) Object Selection: The user selects the private object by placing one of the 3D primitive shapes—cube, cylinder, or sphere, which are readily available in software implementations and cover most shapes of different objects—as a virtual object that encloses the real-world private object from all sides. The Region of Interest (RoI) is then extracted in every frame as the convex hull of the placed 3D virtual object.
2) Snap-Back: To enhance the system's ability to fully enclose the private object from all sides and prevent an accidental revelation of the private object due to the different perspectives during 3D rotational motion around it, the "Snap-Back" algorithm is applied. Object detection is applied to detect the bounding box of the private object and the detected bounding box is tracked in subsequent frames. The tracked bounding box is used to "snap-back" the position and adjust the scale of the placed 3D virtual object to cover the private object it intends to obfuscate in each frame.
3) Obfuscation: Originally, inpainting was used to fill the extracted RoI with pixels that form a feasible background, and preserve the realism of the environment. However, a DR system's main purpose is to diminish information obtained from the real world, including any feature of the object, so other methods can be used for obfuscation, such as blurring, desaturation, outlining, or reduction of opacity, saliency, contrast, or scale [25].

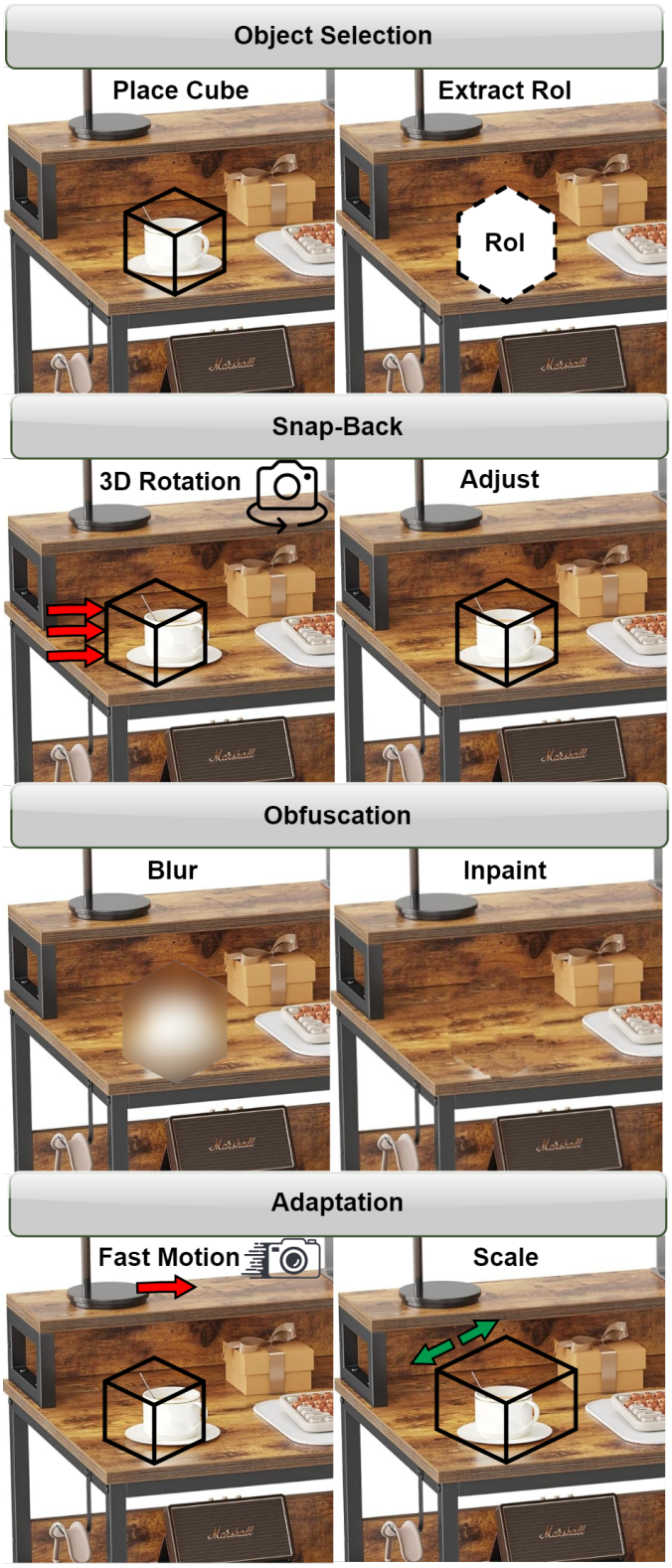

Fig. 1. System Pipeline

B. Adaptive Algorithm

We present the adaptive algorithm in Algorithm 1.

1) Parameters: The adaptive algorithm monitors and reacts to the following parameters, which mainly affect the detection and tracking of private objects and consequently affect the achieved level of privacy:

- Speed: If the speed of motion exceeds a certain threshold ($speed\_flag \leftarrow 1$), the adaptive algorithm increases the area of obfuscation.
- Acceleration: If the magnitude of the acceleration exceeds a certain threshold ($accel\_flag \leftarrow 1$), the area of obfuscation is also increased. This flag also helps detect whenever an abrupt change of direction occurs.

Algorithm 1 Adaptive Algorithm

Given raw frames $\mathcal{F} = \{f_0, f_1, \ldots, f_N\}$
for $f_i \in \mathcal{F}$ **do**
    detection_flag $\leftarrow 0$
    tracking_flag $\leftarrow 0$
    overlap_flag $\leftarrow 0$
    speed_flag $\leftarrow 0$
    accel_flag $\leftarrow 0$
    Apply Snap-Back Algorithm
    Obtain the bounding box of the private object *private_bbox*
    Obtain the bounding box of the current mask *mask_bbox*
    if *private_bbox* = null **then**
        detection_flag $\leftarrow 1$
    end if
    if *AR_Subsystem.TrackingState* $\neq$ Tracking **then**
        tracking_flag $\leftarrow 1$
    end if
    if overlap_area(*private_bbox*, *mask_bbox*) $<$ area(*private_bbox*) **then**
        overlap_flag $\leftarrow 1$
    end if
    if *speed* $\geq$ *speed_threshold* **then**
        speed_flag $\leftarrow 1$
    end if
    if *acceleration* $\geq$ *accel_threshold* **then**
        accel_flag $\leftarrow 1$
    end if
    if *detection_flag* or *overlap_flag* or *tracking_flag* **then**
        aggressive_increase_scale()
    else
        if *speed_flag* or *accel_flag* **then**
            increase_scale()
        else
            decrease_scale()
        end if
    end if
end for

- Detection of the private object: If the object detector fails to detect the private object in a given frame, the area of obfuscation is increased aggressively since the private object can be anywhere and not necessarily overlap with the virtual object, which risks extreme exposure of the private object.
- Plane tracking: If the running AR subsystem fails to properly track the plane where the private and virtual objects are anchored in a given frame, the area of obfuscation is also increased aggressively since the private object can be anywhere relative to the inaccurately detected plane in the current frame and not necessarily overlap with the virtual object, which also risks extreme exposure of the private object.

If none of the monitored parameters sense a privacy failure, the area of obfuscation is gradually reverted to its original size.

2) Output Action:

- Multiplicative Increase: The area of obfuscation is increased by increasing the scale of the placed virtual object that covers the private object. To minimize the risk of exposing any part of the private object during rapid movements, the scale is increased sharply by multiplying it by a factor (β) that is greater than 1, in each frame.

$$Scale_{adaptive} = \beta \times Scale_{old} \tag{1}$$
$$\beta > 1$$

  The scale of aggressiveness of the adaptive algorithm is controlled by β. The larger β is, the sharper the increase in the area of obfuscation, and the more aggressive the algorithm is.
- Additive Decrease: The area of obfuscation is decreased by decreasing the scale of the placed virtual object that covers the private object. Since protecting privacy is the adaptive method's main aim, the scale is decreased slowly by decreasing a small factor (γ) from it, in each frame.

$$Scale_{adaptive} = Scale_{old} - \gamma \tag{2}$$
$$\gamma \ll Scale_{old}$$

- Weighted Average: To ensure a smooth change in the blurred area for improved human perception, the updated scale is a weighted average of the scale in the previous frame ($Scale_{old}$) and the new scale ($Scale_{adaptive}$) that is computed by the adaptive algorithm using the Multiplicative Increase or Additive Decrease.

$$Scale_{new} = \alpha \times Scale_{old} + (1 - \alpha) \times Scale_{adaptive} \tag{3}$$
$$\alpha \in [0, 1]$$

IV. Implementation Details

The system is implemented using Unity [33] and ARFoundation [34]. For OpenCV functions, we utilized the OpenCVForUnity plugin [35]. Although the program was tested on the Android handset Infinix Note 11 Pro, it can be developed for any platform that supports ARFoundation, meaning any device that meets its hardware requirements, including iOS and Android devices. We utilized the following versions: OpenCVForUnity 2.4.7 and Unity 2021.3.19f1. The collected frames have a resolution of 680x460, and the system achieves a frames-per-second (fps) rate that is above 15, on average.

A. Hyper-Parameters

The implemented framework has the same pipeline described above, with the following hyper-parameters:

- In the Snap-Back stage:
  The You Only Look Once version 4 (YOLOv4) model [36] is used for object detection, and the Kernelized Correlation Filters (KCF) [37] method is used for tracking. The detected bounding box of the object is then inflated by 10% to reduce the impact of improper detection on privacy levels.

- In the obfuscation stage:
  We used blurring as our obfuscation method, since it maintains higher fps rates than inpainting, enabling the adaptive method to react just in time to prevent any exposure of the private object.
- In the adaptive algorithm:
  For the additive decrease, we used an $\alpha = 0.6$ and $\gamma = 0.006$ when the mask size is below 75%, otherwise, $\gamma = 0.05$ to speed up the reduction of the mask size. Also, we used $\alpha = 0.4$ and $\beta = 1.11$ in case of a normal multiplicative increase, and in case of an aggressive increase, β also increases multiplicatively by a scale factor ζ, i.e.,

$$\beta_{new} = \zeta \times \beta_{old} \tag{4}$$

  where $\zeta = 1.2$ in case *overlap_flag* $\leftarrow 1$, and $\zeta = 1.1$ otherwise.
  The per-frame acceleration is measured directly through the mobile device's built-in accelerometer and gyroscope. Our acceleration threshold is $0.25g$. The per-frame speed is measured as the difference between the camera-estimated positions by ARFoundation, divided by the time passed between every pair of consecutive frames. Our speed threshold is 0.7 (in units of ARFoundation coordinates per second).
  All the aforementioned hyper-parameters were determined experimentally. They mainly control the trade-off between how much we are obfuscating versus the achieved level of privacy, and can be tuned according to different requirements.

B. Experiment Protocol

Several experiments were conducted to test the implemented framework, such that each setup contained one selection of the following:

- Three different objects representing the private object of increasing shape complexity: a bottle, a mug, and a banana
- Two types of movements: slow and fast
- Two types of motion patterns: translation and rotation.

For each private object, a different virtual shape is used: a cylinder for the bottle, a cube for the mug and a sphere for the banana.

The translational motion runs are performed 20 times, where the mobile device is held by hand and moved sharply to the left and right repeatedly. Fig. 2 shows the camera position for four sample runs during a translational motion. On the other hand, the rotational motion runs are performed 15 times, where the mobile device is rotated around the private object repeatedly by a remote-controlled rotating platform with various speeds. Fig. 3 shows the camera position for four sample runs during a rotational motion. Each run consists of 500 frames, including either slow or fast speeds and accelerations, with sharp direction changes.

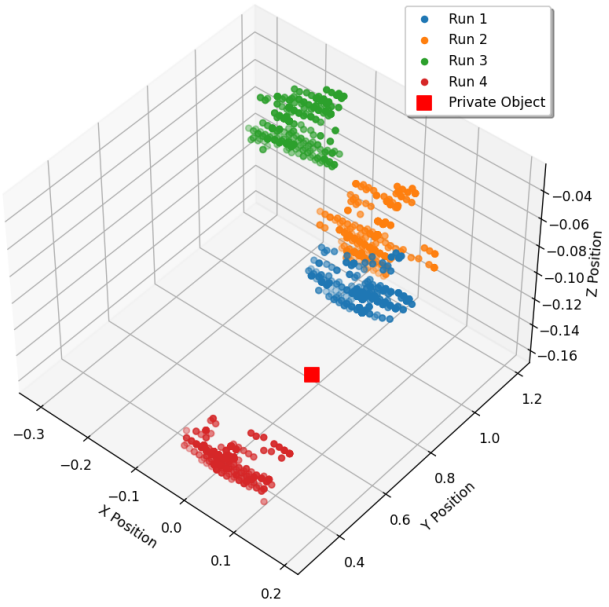

Fig. 2. Camera Position in 3D Space During Translational Motion

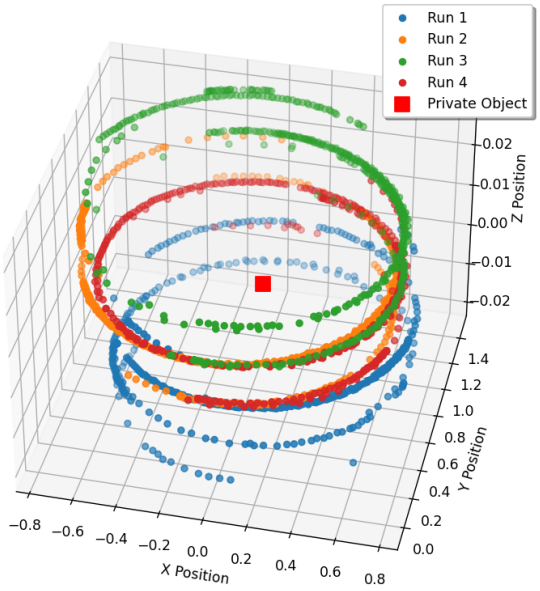

Fig. 3. Camera Position in 3D Space During Rotational Motion

V. Results

A. Privacy Evaluation

The system is evaluated by a privacy measure, denoted by ρ. It is defined as the percentage of the private object's area (A_{priv}) that is contained within the inpainting mask (A_{mask}), and thus obfuscated and inpainted. It is computed by the following equation:

$$\rho = \frac{||A_{priv} \cap A_{mask}||_0}{||A_{priv}||_0} \times 100 \tag{5}$$

In (5), A_{priv} and A_{mask} represent grids or matrices of pixels that have the same dimensions as the initial frame, but A_{priv} is populated with pixels of the private object only and contains zeros otherwise, and A_{mask} is populated with 1 where the inpainting mask is present and 0 otherwise. The overlap ($A_{priv} \cap A_{mask}$) between the two matrices is computed by performing an element-wise AND operation between them. L0-norm ($||\ \ ||_0$) is used to count the number of nonzero elements inside the matrices, and thus the number of nonzero pixels.

B. Adaptive Plots

Figs. 4 and 5 show how the Snap-Back privacy, adaptive privacy, and mask sizes vary with instantaneous acceleration and speed, respectively, for one of the sample runs, where the camera was moving slowly at certain frames, and fast at other frames, around a private object. The plots clearly show how the adaptive mask size increases sharply (multiplicatively) when any of the aforementioned speed and acceleration thresholds are bypassed, and how it decreases slowly (additively) when none of the thresholds are exceeded. While the Snap-Back privacy levels drop significantly to reach 0 at high speeds and high accelerations, after frame 120, the adaptive algorithm reacts to the drastic increase in the camera movement and increases the obfuscation area sharply to maintain a perfect privacy level at 100%. Once the motion becomes slow again, and the acceleration and speed drop below their thresholds, the adaptive mask size drops slowly and approaches the mask size of the Snap-Back algorithm, which approximates the real size of the private object. The adaptive mask size will eventually reach the Snap-Back mask size if no sharp movements occur.

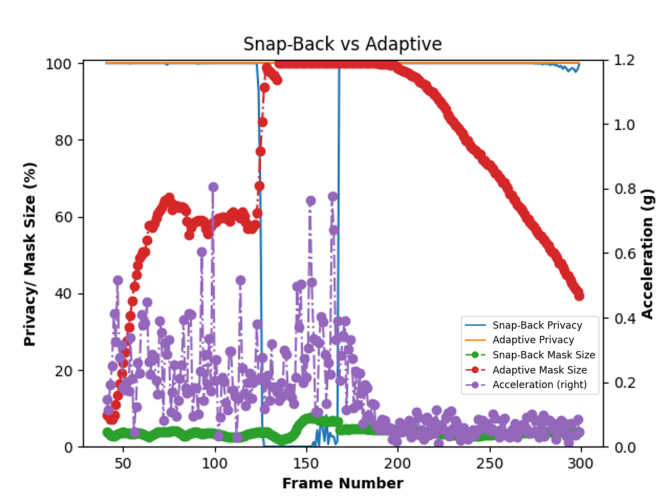

Fig. 4. The Variation of Snap-Back Privacy, Adaptive Privacy, and Mask Size With Acceleration

C. Translational Motion

The privacy results of the experiments of the translational motion pattern are averaged and the privacy failure count for two privacy thresholds: 95% and 99%, defined as the number of times the privacy drops below 95% or 99%, are aggregated for every set of 20 experiments, for every setting, and tabulated in Table I, for the adaptive and for the Snap-Back algorithms.

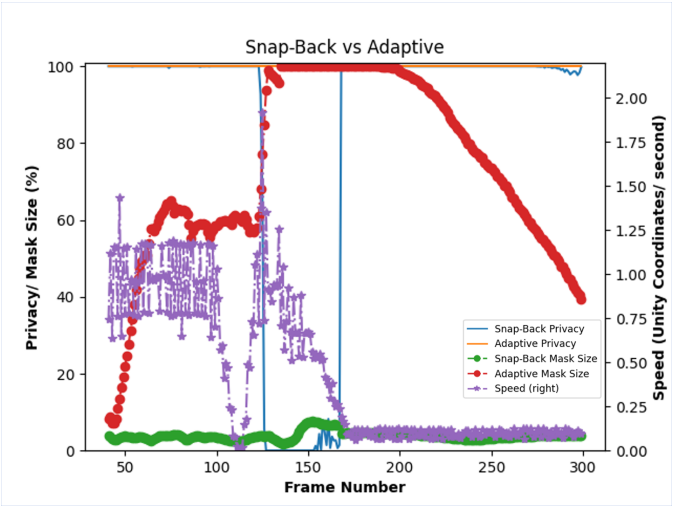

Fig. 5. The Variation of Snap-Back Privacy, Adaptive Privacy, and Mask Size With Speed

D. Rotational Motion

The privacy results of the experiments of the rotational motion pattern are averaged and the privacy failure count for two privacy thresholds: 95% and 99%, are aggregated for every set of 15 experiments, for every setting, and tabulated in Table II, for the adaptive and the Snap-Back algorithms.

E. Discussion

The results show that for slow motion, both methods achieve high privacy levels (above 97% for Snap-Back and above 99% for adaptive) while keeping a relatively low mask size, except for the rotational experiment for the banana object, where the adaptive algorithm shows a significantly larger adaptive mask size, even during slow motion. This is due to the highly irregular and asymmetric shape of the banana, which makes it difficult for the object detector to detect it correctly from different views as we rotate around it. Such inaccurate object detection triggered the adaptive method to perform an aggressive increase and caused more privacy failures. In contrast, for fast motion, while Snap-Back privacy levels drop significantly to reach a minimum of around 43%, the adaptive privacy levels are barely altered, remaining above 99% for all cases. This is accompanied by a sharp increase in the mask size.

It is evident from the privacy failure count results (for both thresholds) that the adaptive method has a significant advantage over the Snap-Back method, especially during fast motion. In translational fast motion experiments, while the Snap-Back algorithm fails more than 42% of the time, the adaptive method reduces failures to less than 0.2% of the

TABLE I
AVERAGE PRIVACY, MASK SIZE, AND PRIVACY FAILURE COUNT FOR TRANSLATIONAL MOTION EXPERIMENTS

Object	Virtual Shape	Motion	Privacy		Mask Size		95% Privacy Failure Count		99% Privacy Failure Count	
			SnapBack	Adaptive	SnapBack	Adaptive	SnapBack	Adaptive	SnapBack	Adaptive
Bottle	Cylinder	Slow	99.75 ± 2.24	100 ± 0.11	19.21 ± 15.04	20.77 ± 24.27	133 (1.34%)	0 (0%)	187 (1.88%)	8 (0.08%)
		Fast	53.94 ± 41.31	99.99 ± 0.22	20.57 ± 17.06	95.80 ± 17.48	6009 (68.35%)	4 (0.05%)	6387 (72.65%)	7 (0.08%)
Mug	Cube	Slow	99.99 ± 0.14	99.99 ± 0.14	8.73 ± 1.67	8.73 ± 1.67	0 (0%)	0 (0%)	50 (0.5%)	50 (0.5%)
		Fast	43.7 ± 41.73	99.98 ± 0.78	10.23 ± 3.65	95.55 ± 17.03	7172 (77.61%)	11 (0.12%)	7559 (81.80%)	16 (0.17%)
Banana	Sphere	Slow	100 ± 0.02	100 ± 0.02	8.66 ± 2.29	8.66 ± 2.29	1 (0.01%)	1 (0.01%)	1 (0.01%)	1 (0.01%)
		Fast	78.37 ± 32.48	100 ± 0.02	11.54 ± 4.67	94.17 ± 20.26	1327 (42.85%)	0 (0%)	1518 (49.02%)	0 (0%)

TABLE II
AVERAGE PRIVACY, MASK SIZE, AND PRIVACY FAILURE COUNT FOR ROTATIONAL MOTION EXPERIMENTS

Object	Virtual Shape	Motion	Privacy		Mask Size		95% Privacy Failure Count		99% Privacy Failure Count	
			SnapBack	Adaptive	SnapBack	Adaptive	SnapBack	Adaptive	SnapBack	Adaptive
Bottle	Cylinder	Slow	99.99 ± 0.15	99.99 ± 0.04	10.03 ± 1.89	16.15 ± 18.42	1 (0.01%)	0 (0%)	2 (0.03%)	0 (0%)
		Fast	99.36 ± 7.05	100 ± 0	8.91 ± 1.78	81.97 ± 22.29	55 (0.83%)	0 (0%)	254 (3.84%)	0 (0%)
Mug	Cube	Slow	99.99 ± 0.11	99.99 ± 0.10	5.06 ± 1.25	6.38 ± 8.99	0 (0%)	0 (0%)	20 (0.27%)	16 (0.21%)
		Fast	95.31 ± 19.34	100 ± 0.04	4.94 ± 1.32	72.40 ± 24.81	453 (6.13%)	0 (0%)	484 (6.55%)	2 (0.03%)
Banana	Sphere	Slow	97.99 ± 5.21	99.73 ± 1.50	5.24 ± 1.07	52.92 ± 44.44	909 (13.21%)	134 (1.95%)	1565 (22.74%)	344 (5.00%)
		Fast	95.69 ± 14.97	100 ± 0.01	5.71 ± 1.33	93.08 ± 23.40	817 (12.37%)	0 (0%)	1204 (18.23%)	0 (0%)

time for those same cases. Whereas for rotational fast motion experiments, when the Snap-Back algorithm fails more than 6% of the time on average, the adaptive method records zero failures for the 95% threshold.

We summarize the overall average improvement of privacy levels during slow and fast motions of the adaptive method versus Snap-Back in Table III. The adaptive technique greatly enhances privacy in all cases. When the Snap-Back method fails to obfuscate more than 95% of the private object more than 18% of the time, the adaptive method significantly reduces the failures to less than 0.2%. By further constraining the privacy level requirements, the adaptive method reduces Snap-Back's failure to obfuscate more than 99% of the private object in more than 21% of the time to less than 0.6%.

Overall, we have demonstrated how privacy can still be achieved in real-time, even in non-trivial motion patterns. However, since covering the entire screen can also maintain perfect privacy in real-time, we need a way to distinguish our method from such drastic measures, as we are only covering the most critical parts of the screen. This ambiguity in the application's functionality when using our adaptive method needs to be clearly defined and measured, highlighting the need for defining a suitable privacy-utility trade-off for our particular use case. We aim to define and measure utility in our future work. We also aim to extend our implementation to obfuscate dynamic objects, as our current implementation is tested only on static objects.

TABLE III
PERCENT OF TIME WHEN PRIVACY DROPS BELOW THE 95% AND 99% THRESHOLDS

Motion	Privacy Dropping Below 95%		Privacy Dropping Below 99%	
	SnapBack	Adaptive	SnapBack	Adaptive
Rotational	5.43%	0.33%	8.61%	0.87%
Translational	31.69%	0.03%	34.31%	0.14%

VI. CONCLUSION

We presented an on-device real-time 3D visual privacy-preserving DR system that adapts to fast motion patterns through an adaptive algorithm. The algorithm monitors the user's movement and adjusts the size of the obfuscation area through MIAD and weighted moving average. We demonstrated our framework's ability to enhance privacy guarantees and reduce the average frequency of privacy failures from over 18% of the time to less than 1% as compared to the Snap-Back algorithm. Our future work includes extending implementation to obfuscate dynamic objects and defining utility for our use-case and evaluating the privacy-utility trade-off in our framework, to find a suitable balance for both objectives.

ACKNOWLEDGMENT

This research was funded by TELUS Corp. and the American University of Beirut University Research Board.

REFERENCES

[1] P. Suciu, "Dying for the perfect photo or video – is social media claiming lives?." https://www.forbes.com/sites/petersuciu/2023/02/03/dying-for-the-perfect-photo-or-video--is-social-media-claiming-lives, Feb 2023.
[2] Precedence Research, "Immersive technology market size, trends, growth, report 2032." https://www.precedenceresearch.com/immersive-technology-market, Aug 2023.
[3] F. Roesner, T. Kohno, and D. Molnar, "Security and privacy for augmented reality systems," *Commun. ACM*, vol. 57, p. 88–96, apr 2014.
[4] X. Wang, L.-H. Lee, C. Bermejo Fernandez, and P. Hui, "The dark side of augmented reality: Exploring manipulative designs in ar," *International Journal of Human–Computer Interaction*, pp. 1–16, 2023.
[5] J. A. De Guzman, K. Thilakarathna, and A. Seneviratne, "Security and privacy approaches in mixed reality: A literature survey," *ACM Comput. Surv.*, vol. 52, oct 2019.
[6] C. Bavor, "Project starline: Feel like you're there, together." https://blog.google/technology/research/project-starline/, May 2021.
[7] Apple, Inc., "Apple vision pro." https://www.apple.com/apple-vision-pro/.
[8] J. A. d. Guzman, A. Seneviratne, and K. Thilakarathna, "Unravelling spatial privacy risks of mobile mixed reality data," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, mar 2021.

[9] X. Gu, W. Luo, M. S. Ryoo, and Y. J. Lee, "Password-conditioned anonymization and deanonymization with face identity transformers," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 727–743, Springer International Publishing, 2020.

[10] Y. Yang, Y. Huang, M. Shi, K. Chen, and W. Zhang, "Invertible mask network for face privacy preservation," *Information Sciences*, vol. 629, pp. 566–579, 2023.

[11] A. Kunchala, M. Bouroche, and B. Schoen-Phelan, "Towards a framework for privacy-preserving pedestrian analysis," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4370–4380, January 2023.

[12] C. Zou, D. Yuan, L. Lan, and H. Chi, "Privacy-preserving action recognition," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2175–2179, 2022.

[13] M. Corbett, B. David-John, J. Shang, Y. C. Hu, and B. Ji, "Bystandar: Protecting bystander visual data in augmented reality systems," in *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, MobiSys '23, (New York, NY, USA), p. 370–382, Association for Computing Machinery, 2023.

[14] S.-Y. Chiu, Y. ting Huang, C.-T. Lin, Y.-C. Tseng, J.-J. Chen, M.-H. Tu, B. Tung, and Y.-C. Nieh, "Privacy-preserving video conferencing via thermal-generative images," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9478–9485, 2023.

[15] P. Speciale, J. L. Schonberger, S. Kang, S. N. Sinha, and M. Pollefeys, "Privacy preserving image-based localization," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 5488–5498, IEEE Computer Society, jun 2019.

[16] J. A. de Guzman, K. Thilakarathna, and A. P. Seneviratne, "Conservative plane releasing for spatial privacy protection in mixed reality," *ArXiv*, vol. abs/2004.08029, 2020.

[17] A. Nama, A. Dharmasiri, K. Thilakarathna, A. Y. Zomaya, and J. A. de Guzman, "User configurable 3d object regeneration for spatial privacy," *ArXiv*, vol. abs/2108.08273, 2021.

[18] T. Kim and G. J. Kim, "Real-time and on-line removal of moving human figures in hand-held mobile augmented reality," *The Visual Computer*, vol. 39, no. 7, p. 2571–2582, 2023.

[19] S. Tabet, A. Kayssi, and I. H. Elhajj, "Mobile diminished reality for preserving 3d visual privacy," in *2023 International Conference on Intelligent Metaverse Technologies Applications (iMETA)*, pp. 01–07, 2023.

[20] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "Visual privacy protection methods: A survey," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4177–4195, 2015.

[21] Z. Wan, J. Zhang, D. Chen, and J. Liao, "High-fidelity pluralistic image completion with transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4672–4681, 2021.

[22] G. Yang, J. Cao, D. Wang, P. Qi, and J. Li, "Eraser: adversarial sensitive element remover for image privacy preservation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 14584–14592, 2023.

[23] M. Li, X. Xu, H. Fan, P. Zhou, J. Liu, J.-W. Liu, J. Li, J. Keppo, M. Z. Shou, and S. Yan, "STPrivacy: Spatio-Temporal Privacy-Preserving Action Recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5106–5115, 2023.

[24] G. Ramajayam, T. Sun, C. C. Tan, L. Luo, and H. Ling, "Saliency-aware privacy protection in augmented reality systems," in *Proceedings of the First Workshop on Metaverse Systems and Applications*, MetaSys '23, (New York, NY, USA), p. 1–6, Association for Computing Machinery, 2023.

[25] Y. F. Cheng, H. Yin, Y. Yan, J. Gugenheimer, and D. Lindlbauer, "Towards understanding diminished reality," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[26] G. Albanis, V. Gkitsas, N. Zioulis, S. Onsori-Wechtitsch, R. Whitehand, P. Ström, and D. Zarpalas, "An ai-based system offering automatic dr-enhanced ar for indoor scenes," in *Advanced Intelligent Virtual Reality Technologies* (K. Nakamatsu, S. Patnaik, R. Kountchev, R. Li, and A. Aharari, eds.), (Singapore), pp. 187–199, Springer Nature Singapore, 2023.

[27] I. Murph, K. Richardson, and A. McLaughlin, "Methods of training to overcome distraction via diminished reality," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 66, no. 1, pp. 1844–1848, 2022.

[28] M. Sasaki, H. Ishii, K. Ueda, and H. Shimoda, "Development of an invisible human experience system using diminished reality," in *Virtual, Augmented and Mixed Reality: Design and Development* (J. Y. C. Chen and G. Fragomeni, eds.), (Cham), pp. 485–494, Springer International Publishing, 2022.

[29] E. Andre and H. Hlavacs, "Diminished reality based on 3d-scanning," in *Entertainment Computing and Serious Games* (E. van der Spek, S. Göbel, E. Y.-L. Do, E. Clua, and J. Baalsrud Hauge, eds.), (Cham), pp. 3–14, Springer International Publishing, 2019.

[30] L. Overmeyer, L. Jütte, and A. Poschke, "A real-time augmented reality system to see through forklift components," *CIRP Annals*, vol. 72, no. 1, pp. 409–412, 2023.

[31] C. Lin and V. Popescu, "Fast intra-frame video splicing for occlusion removal in diminished reality," in *Virtual Reality and Mixed Reality* (G. Zachmann, M. Alcañiz Raya, P. Bourdot, M. Marchal, J. Stefanucci, and X. Yang, eds.), (Cham), pp. 111–134, Springer International Publishing, 2022.

[32] T. Kikuchi, T. Fukuda, and N. Yabuki, "Diminished reality using semantic segmentation and generative adversarial network for landscape assessment: evaluation of image inpainting according to colour vision," *Journal of Computational Design and Engineering*, vol. 9, pp. 1633–1649, 07 2022.

[33] "Unity real-time development platform — 3D, 2D, VR & AR engine." https://unity.com/.

[34] "Unity-technologies/arfoundation-samples: Example content for unity projects based on ar foundation." https://github.com/Unity-Technologies/arfoundation-samples, 2023.

[35] "Enoxsoftware/opencvforunity: Opencv for unity (untiy asset plugin)." https://github.com/EnoxSoftware/OpenCVForUnity, 2023.

[36] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020.

[37] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *CoRR*, vol. abs/1404.7584, 2014.