Mobile Diminished Reality for Preserving 3D Visual Privacy

Salam Tabet
Department of Electrical and Computer Engineering
American University of Beirut
Beirut, Lebanon
sat22@mail.aub.edu

Ayman Kayssi
Department of Electrical and Computer Engineering
American University of Beirut
Beirut, Lebanon
ayman@aub.edu.lb

Imad H. Elhajj
Department of Electrical and Computer Engineering
American University of Beirut
Beirut, Lebanon
ie05@aub.edu.lb

2023 International Conference on Intelligent Metaverse Technologies & Applications (iMETA) | 979-8-3503-2845-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/IMETA59369.2023.10294436

***Abstract*—Privacy is an essential human right. While the world has witnessed massive improvement in immersive technologies and devices in recent years, making them more ubiquitous and accessible in our daily lives, this has led to privacy implications where users' private and sensitive information can easily be exploited in inference attacks. State-of-the-art privacy-preserving approaches are either 2D, focused on protecting a single type of information like faces, or screens, or based on the least privilege paradigm. Preserving privacy in virtual environments, specifically 3D visual privacy in Mixed Reality (MR) applications, remains generally an unsolved problem. To mitigate the risk of private content inadvertently leaking from an MR user's environment, we propose a privacy-preserving Diminished Reality framework to obfuscate 3D objects while preserving the realism of a 3D environment during a live MR feed. We demonstrate the privacy preservation ability during 3D rotation around the sensitive object, real-time performance of our 3D privacy-preserving platform, as well as its implementation using local computations on mobile devices. The proposed framework enables users to select private objects to obfuscate by either enclosing them with 3D virtual objects or detecting their contours. Once identified, the objects are tracked using optical flow and obfuscated in 3D using three inpainting methods. The framework is tested on Android devices and evaluated using frames-per-second (FPS), Peak Signal-to-Noise Ratio (PSNR) and privacy metrics, achieving an average of 15 FPS for 15% mask sizes, PSNR between 25 and 27, and maintaining above 90% privacy.**

***Index Terms*—Diminished Reality, inpainting, tracking, real-time, visual privacy, 3D obfuscation**

I. INTRODUCTION

Recent advances in computer vision, 3D rendering technologies, and artificial intelligence have led to the proliferation of immersive technology applications and devices and noticeable improvement in the machine's ability to interpret visual data [1]. Mixed Reality (MR) applications, in particular, have become ubiquitous in various fields and use cases, such as clinical training [2], smart home control [3], remote collaboration [4] and predictive maintenance and remote surveillance [5].

Most ongoing research in the field of immersive technologies, Augmented Reality/Virtual Reality/Mixed Reality (AR/VR/MR), is focused on making these experiences more immersive and realistic for the users, without fully realizing privacy implications of said technologies. Privacy of users has become a major concern recently, especially following the Cambridge Analytica Scandal, and the creation and implementation of the General Data Protection Regulation (GDPR) in 2018 [6].

That being said, the application environments of these immersive technologies often include sensitive and private information that gets exposed, such as a credit card placed on a table at home, a confidential contract on an office desk, an industrial robot in the background, or family photos on a wall. This private information could be easily picked up by third-party apps that have access to the user's MR system, and get exploited in inference attacks [7]. The privacy concern is further amplified when considering some apps' capabilities to offer remote access for individuals providing support such as in an educational or customer support context. Therefore, there is a need to implement seamless but selective sharing capabilities in MR systems. This is inspired by selective screen-sharing in a video conferencing session (share screen or part of the screen, share application windows, etc.) and the ability to overlay, hide, or blur the speaker's background.

State-of-the-art privacy and security frameworks that address visual privacy are focused on providing an intermediary layer that obfuscates sensitive information before releasing. Most implementations are either 2D, focused on protecting a single type of information like faces [8]–[11] or screens [12], based on the least privilege paradigm [13], [14], or based on context [15]. The few works that provide 3D obfuscation are either too simplistic to provide the needed level of privacy, like placing a 3D bounding box of constant volume in the 3D space to obscure an object [16], or they are based on the least privilege paradigm where they transform raw point clouds to release only needed information to perform a specific task, like producing a 3D line cloud to estimate pose [17], regenerating 3D point clouds to hide the object's identity [18], or releasing detected planes [19] or a number of them [20] for MR anchoring tasks. Preserving 3D visual privacy in MR applications in general remains an unsolved problem. To mitigate the risk of private content inadvertently leaking from an MR user's environment, we propose a privacy-preserving Diminished Reality framework to obfuscate objects in 3D while preserving the realism of the 3D environment.

The presented research highlights the privacy implications of releasing the captured images/videos containing sensitive information to unwanted parties.

Since most AR/VR/MR technologies are still being realized and not yet as ubiquitous as smartphones for instance, and their privacy implications are still not perceived directly, the most immediate impact that the proposed research can have is on what is already readily available and used by almost every active member of the community, namely video conferencing. The proposed framework can be considered as a first step that can be developed into a tool for the most popular video conferencing platforms like Zoom, Google Meet, Cisco Webex, Microsoft Teams, etc..., that enables users to obfuscate their most sensitive objects on-the-fly during a live feed.

Potential use cases of the proposed framework are not limited to the currently ubiquitous technologies, but also to technologies that are already being applied in advanced settings and will soon take over, like the Metaverse, remote control and guidance, telepresence, and digital twins. The use cases for such applications are countless, whether it is to remotely guide a student through a hands-on workshop, perform a remote surgery, or provide a 3D virtual map of an industrial factory. Privacy risks are inherently present in every single one of these applications, and our proposed platform would provide a potential solution, which to the best of our knowledge, is the first work that demonstrates preserving users' privacy in 3D using a mobile real-time Diminished Reality system.

Our contributions can be summed up as follows:

- A privacy-preserving Diminished Reality system that provides 3D obfuscation of user-defined private objects in real-time while preserving the realism of the background environment.
- Demonstration of privacy preservation throughout 3D motion/ rotation around the private object.
- Demonstration of real-time performance of the framework and its local computation on mobile devices.
- Evaluation of the proposed framework using frames-per-second (FPS), Peak Signal-to-Noise Ratio (PSNR) and privacy metrics, where we achieve an average of 15 fps for 15% mask sizes, PSNR between 25 and 27, and maintain above 90% privacy.

The rest of the paper is organized as follows: Section II describes related work including previous privacy frameworks and Diminished Reality systems. Sections III and IV describe our system's pipeline and implementation details, respectively. Section V discusses the obtained results. Section VI discusses the limitations of our framework, and Section VII provides some conclusions.

II. Related Work

A. Privacy Frameworks

State-of-the-art privacy and security frameworks that address visual privacy are focused on providing an intermediary layer that obfuscates sensitive information before releasing. These privacy-preserving methods have targeted several applications, whether mobile camera applications [21], image sharing in wearable visual life-loggers [15], or even see-through applications [22].

Most previous works focus on obfuscating private information in 2D images based on the least privilege paradigm [13], [14], where only the minimal information required by the application is released. Guzman et al. proposed SafeMR [14] as an intermediary layer between core camera APIs and third-party applications to release only application-relevant data instead of the raw capture. Others are based on context [15], which typically includes the place, individuals or activities taking place, or based on user preferences indicated either by using an accompanying User Interface (UI) [21] or specific physical markers that can be automatically detected by the software [23].

Furthermore, several frameworks have been designed to preserve privacy within real-time 2D videos [24]. Most implementations, however, target specific objects, like faces [8]–[11] or screens [12]. On the other hand, the few works that provide 3D obfuscation are focused on accomplishing that while maintaining acceptable functionality for specific tasks, and thus adopt the least privilege paradigm. Guzman et al. demonstrated the privacy risks of releasing MR spatial data in [19] and proposed conservative plane releasing in [20] for preserving privacy while maintaining some degree of utility specifically for MR anchoring tasks by demonstrating how limiting the number of released planes can achieve a balance between privacy and utility. Speciale et al. proposed replacing 3D point clouds by 3D line clouds in [17] for spatial map representations to estimate the camera pose while hiding the environment's underlying geometry. Nama et al. regenerate 3D point clouds of a specific 3D object in [18] to hide its details, while keeping some of its properties like its bounding box for functionality. In [16], Rueben et al. place a 3D bounding box of constant volume in the 3D space to enclose the private object and blur its insides. While blurring is the most commonly used method to hide sensitive information, Cheung et al. utilized inpainting techniques in 2D surveillance videos in [25].

Our proposed privacy-preserving Diminished Reality framework aims to provide the user with the capability of selecting and obfuscating any private objects in 3D while preserving the realism of the 3D environment by inpainting their places in real-time, which is suitable for applications that require an MR live feed like video conferencing.

B. Diminished Reality

Diminished Reality (DR) applications have emerged to deal with use cases that require removing real objects from the scene, rather than augmenting virtual objects in the real world, like in AR and MR applications [26]. The way of recovering the occluded background by the object to be diminished in DR systems is either through see-through or inpainting functions. See-through functions aim at recovering the real background, previously observed either by multiple cameras or a different perspective, while inpainting methods try to generate feasible pixels to fill that hole and generate reasonable and consistent frames. Multiple DR frameworks have been developed to serve different purposes including preserving privacy, by removing people from Google Street View pictures [27], redesigning indoor spaces by removing real furniture and replacing them with virtual furniture [28], or even removing clutter from spaces to reduce hoarding behaviors [29].

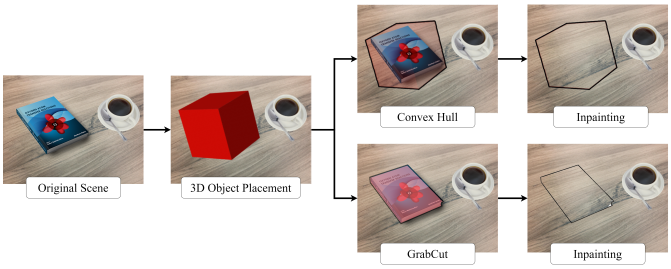

Fig. 1. Selection Pipeline

State-of-the-art techniques for DR face a main challenge: the ability to provide good visual quality and consistency in real-time, including dealing with non-planar backgrounds, non-trivial movement, and lighting changes [30]. The most popular technique, the PixMix approach [31], tries to fill the Region of Interest (ROI) with pixels from the rest of the image, and applies real-time object segmentation and tracking methods.

However, the PixMix technique falls short when dealing with non-planar surfaces or moving objects. Extending PixMix into 3D scenes, 3D PixMix [32] was proposed to improve visual quality, especially when dealing with non-planar backgrounds, by incorporating depth information into the cost functions along with the already existing spatial and texture information in the conventional PixMix cost functions. However, motion is still an issue.

Mobile DR approaches were proposed in [33] and [30], but they require a priori knowledge of the environment, to replace the removed object with its pre-observed background. Moreover, [30] requires heavy processing not suitable for mobile devices.

Our framework presents a lightweight real-time DR system with the specific aim of preserving the privacy of moving users without requiring any form of a priori knowledge of the background.

III. System Design

The pipeline of our DR platform consists of the following stages:

A. Object Selection

The selection criterion is user-defined, where the platform enables the user to place a 3D virtual primitive shape to enclose the private object from all sides.

Once the virtual object is positioned and scaled, the exact area to obfuscate is computed either by taking the entire area enclosed by the virtual object by applying a convex hull to its visible vertices at the current frame, or the GrabCut [34] algorithm is applied to extract the precise contour of the sensitive object. The pipeline for object selection is shown in Fig. 1.

Examples are shown in Fig. 2 and Fig. 3. Figure 2 shows the inpainting of the entire volume enclosed by the placed virtual shapes, which include primitive shapes like cubes, cylinders and spheres. Figure 3 shows the inpainting of the exact volume of the private object detected through exact contour detection by GrabCut.

Typical use cases illustrated in the figures include hiding private objects like a credit card, a personal photo, or personal keys.

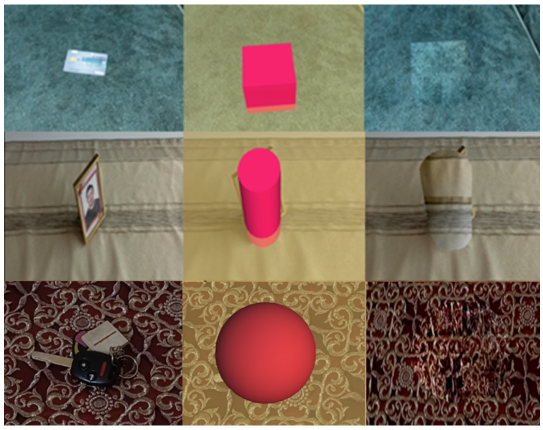

Fig. 2. Object Selection Through 3D Primitive Shapes

Fig. 3. Object Selection Through Exact Contour Detection

Moreover, if the virtual object is placed or moved around in a way that does not enclose the sensitive object entirely, a "Snap-Back" of the virtual object is performed, which automatically positions and scales it accordingly, so that the entire object is enclosed and obfuscated. The Snap-Back procedure is detailed in Algorithm 1. For the object detection model part, we experiment with You Only Look Once (YOLO) and MobileNet Single Shot Detector (SSD).

B. Tracking

The masked region is tracked using Lucas-Kanade optical flow [35]. The produced motion vectors are used to fetch the generated inpainting from the previous frame, rather than generating all pixels inside the mask all over again, as well as any newly visible pixels during 3D motion around the obfuscated object. The optical flow tracking procedure is detailed in Algorithm 2.

Algorithm 1 Snap-Back Algorithm

Given raw frames $\mathcal{F} = \{f_0, f_1, \ldots, f_N\}$
for $f_i \in \mathcal{F}$ **do**
  if $i = 0$ **then**
    Apply Object Detection Algorithm/Model to f_0 to produce a set of all detected bounding boxes

$$\mathcal{B} = \{bbox_1, bbox_2, \ldots, bbox_n\}$$

    Extract virtual object 2D screen centroid position

$$c_{obj} = (c_{obj_x}, c_{obj_y})$$

    from 3D world position

$$\mathbf{p} = (p_x, p_y, p_z)$$

    $min\_distance \leftarrow \infty$
    $private\_bbox \leftarrow$ None
    $\mathbf{c}_{\text{private\_bbox}} \leftarrow$ None
    for $bbox_j \in \mathcal{B}$ **do**
      $\mathbf{c_j} \leftarrow$ compute_centroid($bbox_j$)
      distance $\leftarrow \sqrt{(c_{j_x} - c_{obj_x})^2 + (c_{j_y} - c_{obj_y})^2}$
      if distance $< min\_distance$ **then**
        $min\_distance \leftarrow$ distance
        $private\_bbox \leftarrow bbox_j$
        $\mathbf{c}_{\text{private\_bbox}} \leftarrow \mathbf{c_j}$
      end if
    end for
  else
    Perform tracking of *private_bbox* in f_j
    Update

$$\mathbf{c}_{\text{private\_bbox}} \leftarrow \mathbf{c}_{\text{tracked\_bbox}}$$

  end if
  Adjust the position of the virtual object such that

$$\mathbf{p_{new}} = raycast(\mathbf{c}_{\text{private\_bbox}})$$

  Adjust the scale of the virtual object such that

$$\mathbf{size_{new}} = (w, h, max(w,\ h)))$$

  where

$$w = width(private\_bbox),$$

$$h = height(private\_bbox)$$

end for

Tracking is beneficial in two ways: (1) Boosting real-time performance by reducing the time needed to generate feasible inpainting, especially when the inpainting mask is very large, (2) Preserving the coherence of the 3D scene by incorporating newly visible textures into the final result.

C. Inpainting

To preserve the privacy of the user completely, inpainting is performed on the selected private object to cover. We experimented with three real-time inpainting methods: two OpenCV

Algorithm 2 Optical Flow Tracking Algorithm

Given raw frames $\mathcal{F} = \{f_0, f_1, \ldots, f_N\}$, inpainting masks for each frame $\mathcal{M} = \{m_0, m_1, \ldots, m_N\}$
for $f_k \in \mathcal{F}$ **do**
  if $k = 0$ **then**
    Apply inpainting algorithm to f_0 to produce new frame f_0'
  else
    Set motion vector

$$\vec{d} \leftarrow \text{None}$$

    for Every pixel $p_{k-1_{(i,j)}}$ in frame f_{k-1} at row i, column j **do**
      if $m_{k_{(i,j)}} = 1$ **then**
        Find the location of the pixel $p_{k_{(i',j')}}$ in f_k such that

$$p_{k_{(i',j')}} \leftarrow lucas\_kanade\_optical\_flow(p_{k-1_{(i,j)}})$$

        if $p_{k_{(i',j')}}$ is not None **then**
          Update

$$\vec{d} \leftarrow (d_x = i' - i, d_y = j' - j)$$

          break
        end if
      end if
    end for
    for Every pixel $p'_{k_{(i,j)}}$ in new frame f_k' at row i, column j **do**
      if $m_{k_{(i,j)}} = 1$ **then**
        Update

$$p'_{k_{(i,j)}} \leftarrow p'_{k-1_{(i-d_x, j-d_y)}}$$

      end if
    end for
  end if
end for

methods based on Fast Marching Method (FMM) [36] and fluid dynamics (Navier Stokes (NS)) [37], and Siltanen's [38] method.

IV. Implementation Details

The framework is developed using Unity [39] and ARFoundation [40]. We used OpenCVForUnity plugin for OpenCV functions. The application can be built for any platform that supports ARFoundation, i.e., satisfies its hardware requirements, including Android and iOS devices, but it was tested on the Android device Infinix Note 11 Pro. Unity version 2021.3.19f1 and OpenCVForUnity 2.4.7 are used. The frames captured have a 680x460 resolution, and we note that ARFoundation supports higher resolutions as well.

V. Results

The system is evaluated based on three aspects: real-time performance represented by Frames Per Second (FPS), inpainting quality represented by Peek Signal-to-Noise Ratio (PSNR), and the privacy measure achieved.

The privacy measure is referred to as ρ and 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 as follows:

$$\rho = \frac{||A_{priv} \cap A_{mask}||_0}{||A_{priv}||_0} \times 100 \quad (1)$$

In (1), A_{priv} and A_{mask} represent grids/matrices of pixels that have the same dimensions of 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.

To obtain the exact area of the private object A_{priv}, we use GrabCut or Segment Anything [41].

The average FPS is shown in Table 1 for each of the three tested inpainting methods when using optical flow-based tracking and when inpainting in every frame without tracking. The average PSNR (not shown in the table), on the other hand, is similar in both cases, i.e., with and without optical flow for each type, and ranges are between 25 and 27.

TABLE I
AVERAGE FPS FOR ALL INPAINTING TYPES WITH AND WITHOUT OPTICAL FLOW FOR DIFFERENT MASK SIZES

Mask Size	Navier Stokes		Fast Marching		Siltanen Method	
	Without	With	Without	With	Without	With
0-5%	19	**24**	18	**25**	**28**	17
5-10%	10	**20**	11	**18**	**22**	13
10-15%	6	**17**	7	**15**	**19**	11
15-20%	6	**17**	7	**13**	**15**	10
20-25%	5	**13**	4	**12**	**13**	10

The table shows that for the Fast Marching and Navier Stokes methods with optical flow enhanced real-time performance by increasing the average FPS rates even for larger mask sizes. For Siltanen's method, although inpainting in every frame has better overall FPS rates without using optical flow than with using optical flow, using optical flow tracking reduces the rate at which FPS is decreasing with increasing mask size, which is an expected contribution of tracking. With tracking, increasing mask sizes does not affect the frame rates as severely as they would when we are inpainting in every frame.

Figure 4 shows the achieved average privacy percentage measured at specific frames, over 6 runs, with and without "Snap-Back", and Figure 5 shows how the camera (the blue and orange dots) is moving in 3D space around the private object (the red square) that needs to be hidden, in an identical pattern for both cases, in one sample of the experiments.

Figure 6 shows the average privacy obtained, with and without "Snap-Back", as the camera is rotated around the 3D obfuscated object at different azimuth angles, starting from angle 0° and moving, with the same speed of motion, in a counterclockwise direction, as seen from a top-view.

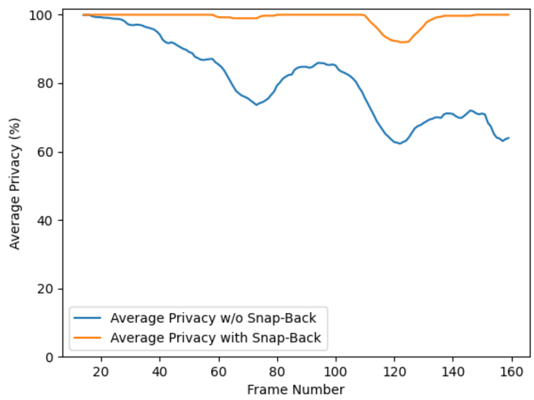

Fig. 4. Privacy Percentage (ρ) vs. Frame Number with and without Snap-Back

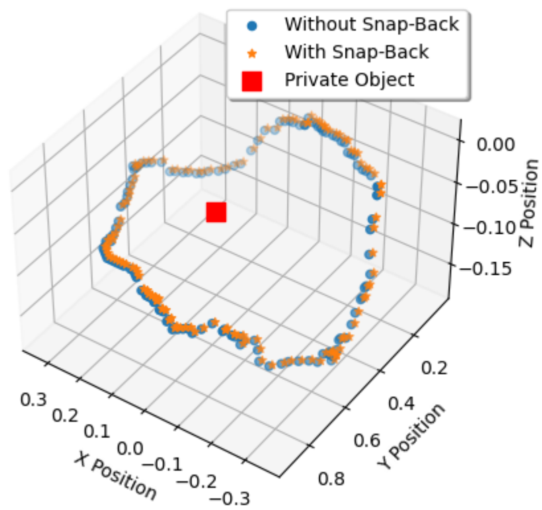

Fig. 5. Camera Position in 3D Space

The azimuth angle φ is computed as in the cylindrical coordinate system using:

$$\varphi = \arctan \frac{y}{x} \quad (2)$$

where x and y represent the coordinates of the virtual object in the horizontal xy-plane, with x being the displacement of the object to the left or right, and y being the depth, or how far the object is from the camera. These coordinates are fixed, even as the camera is moving around the object, since they refer to the fixed anchor's position in 3D space, which was produced when the virtual object was first instantiated, and is

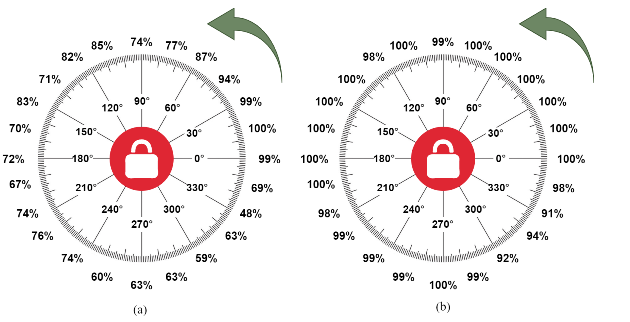

Fig. 6. Privacy Percentage (ρ) vs. Angle: (a) Without Applying "Snap-Back" to Object (b) With Applying "Snap-Back"

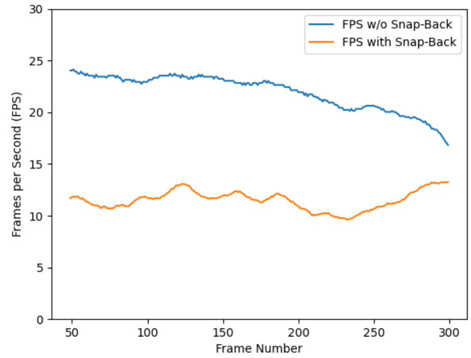

Fig. 7. FPS vs. Frame Number without Applying "Snap-Back" to Object and with Applying "Snap-Back"

tracked across frames. The fixed anchor position is used as a proxy to the private object's actual position in the azimuth angle computation, since it is either equal to the position of the private object or very close to it.

Figure 6 (a) shows the privacy results without our "Snap-Back" feature, whereas (b) shows results while using "Snap-Back", averaged over 6 runs. Both Fig. 4 and Fig. 6 (a) show a degradation in privacy with time as we move around the object and view it from different sides, followed by a slight increase in privacy again, as we approach the angle 360°, i.e., the initial perspective that was visible to the user once the virtual object was placed over the private object as shown in Fig. 6.

In Fig. 4 the privacy trend fluctuates between high and low depending on the current perspective from which the camera is viewing the private object, when "Snap-Back" is not used. Such degradation is expected for two reasons: (1) the user is relying on pure ARFoundation tracking, which is not robust to 3D motion, and (2) the user might mistakenly enclose the private object with a virtual 3D object that perfectly encloses it from the current view, and reveals parts of it from another view hidden from the user at the moment of 3D object placement. Depending on the environment and textures in view, the results using this approach can be better, where ARFoundation tracking in 3D rotation can provide almost a perfect obfuscation of the private object (above 95% privacy level), but here we show the worst case, where the ARFoundation tracking loses the object gradually during the 3D motion around it.

On the other hand, using "Snap-Back" maintains privacy levels above 90% in Fig. 4, and an average privacy above 90% in Figure 6 (b) throughout the full 360° rotation around the object, giving "Snap-Back" a major advantage over just relying on pure ARFoundation tracking. This is due to the fact that the virtual object is adjusted to fully enclose the private object, at any given view.

Figure 7 shows the running average of FPS with each frame when applying "Snap-Back" and without applying "Snap-Back". Since "Snap-Back" includes further processing of the frame, i.e., more operations, the FPS is expected to be lower than when it is not used. However, this decline in FPS is not severe, and the FPS goes from ranging between 17 and 24 without "Snap-Back" to ranging between 10 and 14 with "Snap-Back".

VI. Limitations

Although the proposed system can effectively hide any 3D object in real-time, it has some limitations, as the tracking and inpainting are not robust when dealing with fast motion and different lighting conditions. Another aspect that can be improved is increasing the average FPS for the private object contour selection using GrabCut algorithm to make it more reliable in real-time use cases.

VII. Conclusions

We presented a Diminished Reality system that enables users to obfuscate any private object in 3D space and inpaint the background that was previously occluded, to preserve the realism of the environment. We demonstrated our framework's ability to occlude any private object from all sides during 3D motion around it. Our system is real-time, achieving 15 fps for mask sizes of 15% on average, near perfect levels of privacy, and implemented locally on a mobile device. Future work includes enhancing robustness of the system to higher resolutions, different lighting conditions and fast motion, as well as further improving frame rates, particularly for precise contour detection of the selected 3D private object.

Acknowledgment

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

References

[1] 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.

[2] R. Monterubbianesi, V. Tosco, F. Vitiello, G. Orilisi, F. Fraccastoro, A. Putignano, and G. Orsini, "Augmented, virtual and mixed reality in dentistry: A narrative review on the existing platforms and future challenges," *Applied Sciences*, vol. 12, no. 2, 2022.

[3] M. Sivý, B. Sobota, Korecko, and M. Hudák, "Smart environment control in virtual and mixed reality based on cognitive user abilities," in *2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 600–605, 2020.

[4] S. Kim, G. Lee, W. Huang, H. Kim, W. Woo, and M. Billinghurst, "Evaluating the combination of visual communication cues for hmd-based mixed reality remote collaboration," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), p. 1–13, Association for Computing Machinery, 2019.

[5] D. Mourtzis, J. Angelopoulos, and N. Panopoulos, "Intelligent predictive maintenance and remote monitoring framework for industrial equipment based on mixed reality," *Frontiers in Mechanical Engineering*, vol. 6, 2020.

[6] M. Salnitri, J. Jürjens, H. Mouratidis, L. Mancini, and P. Giorgini, eds., *Visual Privacy Management - Design and Applications of a Privacy-Enabling Platform*, vol. 12030 of *Lecture Notes in Computer Science*. Springer, 2020.

[7] 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.

[8] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "A scalable and privacy-aware iot service for live video analytics," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, (New York, NY, USA), p. 38–49, Association for Computing Machinery, 2017.

[9] J. Shu, R. Zheng, and P. Hui, "Cardea: Context-aware visual privacy protection for photo taking and sharing," in *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, (New York, NY, USA), p. 304–315, Association for Computing Machinery, 2018.

[10] Z. Ren, Y. J. Lee, and M. S. Ryoo, "Learning to anonymize faces for privacy preserving action detection," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 639–655, Springer International Publishing, 2018.

[11] 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.

[12] M. Korayem, R. Templeman, D. Chen, D. Crandall, and A. Kapadia, "Enhancing lifelogging privacy by detecting screens," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), p. 4309–4314, Association for Computing Machinery, 2016.

[13] S. Jana, A. Narayanan, and V. Shmatikov, "A scanner darkly: Protecting user privacy from perceptual applications," in *2013 IEEE Symposium on Security and Privacy*, pp. 349–363, 2013.

[14] J. A. de Guzman, K. Thilakarathna, and A. Seneviratne, "Safemr: Privacy-aware visual information protection for mobile mixed reality," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, pp. 254–257, 2019.

[15] E. Zarepour, M. Hosseini, S. S. Kanhere, and A. Sowmya, "A context-based privacy preserving framework for wearable visual lifeloggers," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 1–4, 2016.

[16] M. Rueben, F. J. Bernieri, C. M. Grimm, and W. D. Smart, "Evaluation of physical marker interfaces for protecting visual privacy from mobile robots," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 787–794, 2016.

[17] 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.

[18] 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.

[19] J. A. de Guzman, K. Thilakarathna, and A. Seneviratne, "A first look into privacy leakage in 3d mixed reality data," in *Computer Security – ESORICS 2019* (K. Sako, S. Schneider, and P. Y. A. Ryan, eds.), (Cham), pp. 149–169, Springer International Publishing, 2019.

[20] 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.

[21] P. Aditya, R. Sen, P. Druschel, S. Joon Oh, R. Benenson, M. Fritz, B. Schiele, B. Bhattacharjee, and T. T. Wu, "I-pic: A platform for privacy-compliant image capture," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, (New York, NY, USA), p. 235–248, Association for Computing Machinery, 2016.

[22] M. Hayashi, R. Yoshida, I. Kitahara, Y. Kameda, and Y. Ohta, "An installation of privacy-safe see-through vision," *Procedia - Social and Behavioral Sciences*, vol. 2, no. 1, pp. 125–128, 2010. The 1st International Conference on Security Camera Network, Privacy Protection and Community Safety 2009.

[23] N. Raval, A. Srivastava, K. Lebeck, L. Cox, and A. Machanavajjhala, "Markit: Privacy markers for protecting visual secrets," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, (New York, NY, USA), p. 1289–1295, Association for Computing Machinery, 2014.

[24] J. Padilla-López, A. Chaaraoui, F. Gu, and F. Flórez-Revuelta, "Visual privacy by context: Proposal and evaluation of a level-based visualisation scheme," *Sensors*, vol. 15, pp. 12959–12982, 06 2015.

[25] S.-C. Cheung, M. Venkatesh, J. Paruchuri, J. Zhao, and T. Nguyen, *Protecting and Managing Privacy Information in Video Surveillance Systems*, pp. 11–33. London: Springer London, 2009.

[26] S. Mori, S. Ikeda, and H. Saito, "A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, 2017.

[27] A. Flores and S. Belongie, "Removing pedestrians from google street view images," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pp. 53–58, 2010.

[28] P. Kulshreshtha, N. Lianos, B. Pugh, and S. Jiddi, "Layout aware inpainting for automated furniture removal in indoor scenes," in *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, (Los Alamitos, CA, USA), pp. 839–844, IEEE Computer Society, oct 2022.

[29] S. W. T. Chan, B. Ryskeldiev, and S. Nanayakkara, "Declutterar: Mobile diminished reality and augmented reality to address hoarding by motivating decluttering and selling on online marketplace," in *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 870–874, 2022.

[30] C. Kunert, T. Schwandt, and W. Broll, "An efficient diminished reality approach using real-time surface reconstruction," in *2019 International Conference on Cyberworlds (CW)*, pp. 9–16, 2019.

[31] J. Herling and W. Broll, "High-quality real-time video inpaintingwith pixmix," *IEEE Transactions on Visualization amp; Computer Graphics*, vol. 20, pp. 866–879, jun 2014.

[32] S. Mori, J. Herling, W. Broll, N. Kawai, H. Saito, D. Schmalstieg, and D. Kalkofen, "3d pixmix: Image inpainting in 3d environments," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 1–2, 2018.

[33] G. Queguiner, M. Fradet, and M. Rouhani, "Towards mobile diminished reality," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 226–231, 2018.

[34] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, p. 309–314, aug 2004.

[35] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, (San Francisco, CA, USA), p. 674–679, Morgan Kaufmann Publishers Inc., 1981.

[36] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, 01 2004.

[37] M. Bertalmio, A. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–I, 2001.

[38] S. Siltanen, "Texture generation over the marker area," in *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 253–254, 2006.

[39] "Unity real-time development platform — 3D, 2D, VR & AR engine." https://unity.com/. Accessed: 2023-01-01.

[40] "Unity-technologies/arfoundation-samples: Example content for unity projects based on ar foundation." https://github.com/Unity-Technologies/arfoundation-samples. Accessed: 2023-01-30.

[41] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.