De-anonymization Attacks on Metaverse

Yan Meng*, Yuxia Zhan*, Jiachun Li*, Suguo Du†, Haojin Zhu*‡, and Xuemin (Sherman) Shen§
*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
†Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, China
§Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
Email: {yan_meng, dabeidouretriever, jiachunli, sgdu, zhu-hj}@sjtu.edu.cn, sshen@uwaterloo.ca

***Abstract*—Virtual reality (VR) can provide users with an immersive experience in the *metaverse*. One of the most promising properties of VR is that users' identities can be protected by changing their physical world appearances into arbitrary virtual avatars. However, recent proposed de-anonymization attacks demonstrate the feasibility of recognizing the user's identity behind the VR avatar's masking. In this paper, we propose AvatarHunter, a non-intrusive and user-unconscious de-anonymization attack based on victims' inherent movement signatures. AvatarHunter imperceptibly collects the victim avatar's gait information via recording videos from multiple views in the VR scenario without requiring any permission. A Unity-based feature extractor is designed that preserves the avatar's movement signature while immune to the avatar's appearance changes. Real-world experiments are conducted in VRChat, one of the most popular VR applications. The experimental results demonstrate that AvatarHunter can achieve attack success rates of 92.1% and 66.9% in closed-world and open-world avatar settings, respectively, which are much better than existing works.**

***Index Terms*—Virtual reality, De-anonymization attack, Movement signature, Identity inference**

I. Introduction

Facilitated by the development of virtual reality (VR), metaverse is regarded as a disruptive technique to reshape the way that human beings connect with each other via digital avatars. Compared with traditional human-computer interfaces (*e.g.*, keyboard, mouse, touch screen), VR provides users with an immersed experience via leveraging specialized devices such as head-mounted display (HMD), hand-held controller, and full-body tackers. Currently, VR has been deployed not only in industrial scenarios including military training and medical operation, but also in consumer cases such as virtual meetings and immersing games. According to the report published by Grand View Research, the market size of VR was valued at USD 15.81 billion in 2020 and is expected to grow at a compound annual growth rate (CAGR) of 18.0% from 2021 to 2028 [1].

Metaverse intrinsically supports privacy enhancement features. In addition to the traditional privacy supporting techniques (e.g., pseudonym), the user's real identity could be naturally masked by *avatar*, a digital representation of the user in the virtual world [2]. Essentially speaking, the avatar has the shape of a human but may contain extra characteristics configured by the users themselves. For instance, in VR games such as VRChat [3], the user's physical appearance is transformed into avatars with cartoon and fantastic shapes to communicate and interact with each other anonymously in virtual settings. Since users can customize their appearance and sounds in the virtual world, the avatar brings the users a feeling of being *unlinkable* to their real identities and avoiding tracking by the potential external adversary.

The existing research on VR security mainly focuses on VR devices' functional integrity [4], [5] and the VR user's access control [6], [7], while the privacy issues receive less attention [2]. Two recent studies investigate the privacy leakage issues in VR from the perspective of traffic analysis [8] and side-channel-based de-anonymization [9]. Specifically, the former analyzes the traffic payloads originating from Meta Oculus Quest VR devices and shows that the victim's identity may be leaked to malicious VR developers [8], while the latter tries to perform a de-anonymization attack in VR by leveraging the malicious app to obtain the acceleration and gyroscope data from the headset display and conduct the user identification [9]. It is essential to point out that the state-of-the-art works on privacy issues are based on strong adversary models/assumptions (e.g., malicious app developers or requiring data collections from VR devices). Motivated by *Ready Player One*, a famous science fiction adventure film in which protagonist Wade Watts hides his real identity by changing his avatar's appearance while the adversary tries to learn his real identity, we consider a more practical de-anonymization attack: *by only observing a set of VR avatars and real-world identities, is it possible for the external attacker to perform a de-anonymization attack by linking the user's real identities to his avatars, even if the observed avatars could be arbitrarily changed or modified?* The motivating examples for such kind of de-anonymization attack scenario include but are not limited to: tracking users in an avatar-changeable VR game such as VRChat [3] or an anonymous VR meeting [10].

In this study, we propose AvatarHunter, the first practical de-anonymization attack, which aims to reveal the target victim's identity from her counterpart avatar's movement behaviors in the virtual world. Different from the existing de-anonymization schemes in VR [11], [12], [13], [14], AvatarHunter neither requires intruding into the victim's network nor injecting malicious apps. Instead, AvatarHunter adopts a new attack manner, which only needs to record the video clips when the target victim walks in the open virtual world to infer the identity behind the victim's avatar. AvatarHunter is motivated by the following insight: in the metaverse, no matter how

‡ Haojin Zhu is the corresponding author.

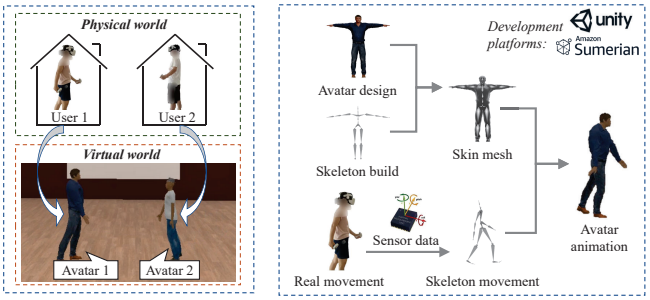
979-8-3503-3414-2/23/$31.00 ©2023 IEEE

the avatars are changed, the victim's inherent and unique movement patterns (*i.e.*, gait information in this study) remain relatively stable, which could be exploited to link the user's avatar to her real identity.

Research challenges. To implement AvatarHunter, we need to address the following three key challenges: *(1)* How to link the user's identity in the physical world to her avatar in the virtual world for de-anonymization in a stable and reliable way? *(2)* How to extract an effective feature that characterizes the victim's unique identity signature and is robust to the avatar changes? Note that, when addressing the gait video clips, existing computer vision-based gait recognition schemes [15], [16], [17] are based on both movement and appearance signatures, in which the latter is unstable since the virtual appearance is frequently changed. *(3)* Considering AvatarHunter is the first non-intrusive and user-unconscious de-anonymization attack to link avatar and the real-world identity in VR and there is no publicly available dataset so far, how can we implement AvatarHunter in the real-world VR application and demonstrate its effectiveness and robustness?

To address the above three challenges, we *firstly* introduce a novel biometric signature, coined as *movement signature*, to link the victim's identity in the physical world and the avatar in the virtual world. We further validate its feasibility and robustness under the setting of changeable avatars via a motivation example. *Secondly*, considering existing video-based user recognition schemes, which concentrate on gait samples in the physical world, are sensitive to distortions caused by avatars' changeable appearances, we design a novel feature extractor when extracting the movement signature. The insight is to let the extractor own prior knowledge about various avatars. To achieve it, we leverage Unity, the largest platform for developing VR applications [18], to automatically generate abundant gait videos from various avatars. After constructing the feature extractor based on abundant gait videos originating from various avatars in Unity, AvatarHunter is immune to the avatar's appearance changes. *Finally*, we implement AvatarHunter on VRChat, one of the most popular VR applications owning 4 million users [19]. We deploy multiple VR accounts as *cameras* to record gait video trials containing 100 volunteer-avatar pairs (*i.e.*, 10 recruited volunteers with 10 different avatars). The experimental results show AvatarHunter can achieve the attack success rate of 92.1% and 66.9% in the closed-world and open-world avatar settings, respectively. AvatarHunter is robust to various factors, including the number of cameras, input video lengths, and the prior knowledge of target victims. Our main contributions are summarized below:

- *New attack paradigm.* We present AvatarHunter, a non-intrusive and user-unconscious de-anonymization attack on the avatars in the VR scenario. AvatarHunter neither requires access to the victim's VR device nor the approval/permission of the victim.
- *Novel de-anonymization method.* We design a novel manner to extract the identity-related movement signature from the avatar's video clips. By leveraging numerous gait samples originating from various avatars in the Unity platform, AvatarHunter is robust to the changeable appearances of avatars.
- *Open-source dataset.* A dataset containing video trials from 10 users with 10 avatars collected in the VR application will be available to researchers, vendors, and developers to assess the privacy risks in metaverse and design countermeasures for de-anonymization attacks.



(a) Multi-user application. (b) Avatar generation process.

Fig. 1. Overview of VR application and avatar.

The remainder of this paper is organized as follows. Section II introduces the necessary preliminary knowledge. Section III illustrates the threat model and motivation. Section IV presents the system design of AvatarHunter, followed by evaluations, discussion, and related work in Section V, Section VI, and Section VII, respectively. Finally, Section VIII concludes this work.

II. Preliminaries

In this section, we first review the framework of multi-user VR applications. Then, we elaborate on the VR device and the workflow of avatars in VR applications, respectively.

A. Multi-user VR Application Framework

With the popularity of VR technologies, the VR applications attract an increasing number of users to participate. The VR platform provides various scenes, animations, and communication functions for the participants. To ensure the anonymization of users in the virtual world, the current multi-user VR application framework employs the mechanism of *avatar*. As shown in Fig. 1(a), in the multi-user VR application scenario, the users on the client side first enter the virtual world with the assistance of VR devices. Then, the users can conduct various actions including chatting, movement, and gaming in the VR application. To protect the user's identity and increase entertainment, the VR platform allows the users to transform themselves into various avatars to hide their appearances in reality. Besides, the users can also change their voices to enhance their anonymity.

B. VR Device

There are many VR devices in the consumer market. To improve the user's interactive experience, current popular VR devices (*e.g.*, Meta Quest 2 [20], HTC VIVE pro 2 [21]) equip with not only general purpose sensors (*e.g.*, gyroscope, accelerator) but also human interactive sensors (*e.g.*, controller

handles). In this study, we choose a specific stand-alone device named Meta Quest 2, as our study object.[1] Quest is developed by Meta which owns 90% of the global VR headset market [22]. It consists of a system on chip (SoC) to perform computation, an LCD to display visual contents, and a microphone-loudspeaker pair to interact with the human via the voice channel. Quest not only supports VR applications in its official stores but also supports games from third-party VR application platforms (*e.g.*, Steam VR, SideQuest). Besides, Quest can be connected to the laptop/desktop and provide the developer with a view of its LCD screen (*i.e.*, the view watched by the user) for game video recording or application debugging.

C. The Workflow of Avatars in VR Applications

As illustrated in Fig. 1(b), users can leverage the avatars to change their appearance and hide their real-world identities in the virtual world. In this subsection, we introduce the avatar's generation procedure, movement mechanism, and development platforms.

1) Avatar Generation Procedure: In most VR applications, the avatar is generated based on the skeleton animation mechanism [23] consisting of three steps. *(i) Designing avatar model*: the designer first draws the static character sketch of the avatar and then creates the corresponding model consisting of abundant mesh structures in the 3D virtual space. *(ii) Building skeleton structure*: for the generated 3D model of the avatar, the skeleton is defined for action execution. The avatar's skeleton has the parent-child structure, which can be regarded as the avatar's bone and joint during the action period. More specifically, when the parent node moves, the child node will execute the corresponding movement according to the mapping rules. *(iii) Mapping skeleton and skins:* skin is the external appearance of the avatar in the virtual world. After building the skeleton, the properties of the skin (*e.g.*, positions, colors, materials, ornaments) related to the skeleton are determined. When the avatar's skeleton moves, the skin will generate the related movements.

2) Mapping VR's Sensor Data to Avatar Movements: Different from traditional desktop applications in which the user input devices are keyboard and mouse, in the VR scenario, the avatar's movement is highly coupled with the user's physical movement behaviors. During the user playing in the VR application, the data collected from the equipped VR helmet and controller handles are mapped to the avatar's skeleton. Then, the skeleton is driven by the motion data and its related skins are rendered in the virtual world.

3) VR Application Development Platforms: Several platforms assist VR developers in generating and activating avatars quickly. Unity is the typical platform by which 60% of popular VR games [24] are developed. Unity also allows the developer to set up the virtual scenario of VR applications and easily construct and control the virtual appearance of avatars in C# based programs [25]. We utilize Unity platform when extracting the victim's features as described in Section IV-C.

[1]In the following sections, we utilize the terminology *Quest* to refer to the Meta Quest 2.

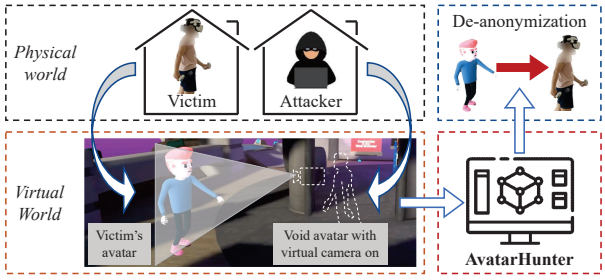


Fig. 2. Attack scenario.

III. Attack Overview and Motivation Example

In this section, we first give a picture of AvatarHunter's attack scenario and elaborate on the adversary's capabilities. Then, we demonstrate the feasibility of AvatarHunter via a motivation example. Finally, we formally define the attack scenarios in this study.

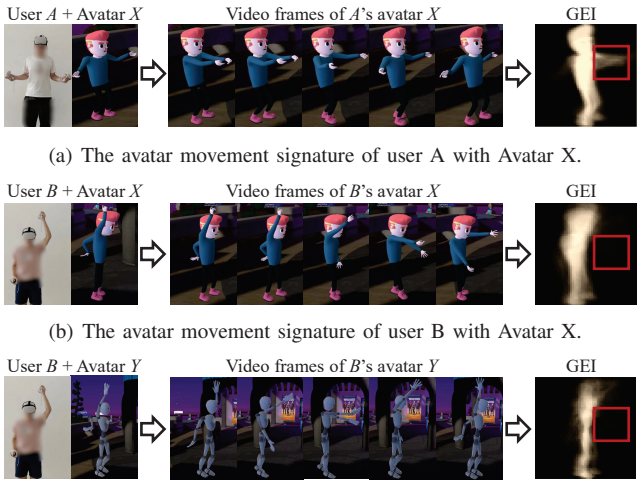
A. Attack Model and Adversary Capabilities

1) Attack Model: Fig. 2 illustrates a typical attack scenario of AvatarHunter. A victim is playing a VR game (*e.g.*, VRChat [3]) and transforms her appearance into an avatar. Meanwhile, the adversary also logins the same VR app in which the victim plays and starts monitoring the victim's behaviors via recording the video displayed on the VR device's screen. Since the adversary can do the recording procedure without any participation from the victim, the victim cannot recognize it and believe her identity is well hidden behind the avatar. However, from the adversary's perspective, after finishing the collection of the victim's video clips, the adversary can launch the de-anonymization attack based on the extracted biometric signature related to the victim's identity.

2) Adversary's Capabilities: We assume the adversary in AvatarHunter has two capabilities during the attack preparation and implementation phases.

Attack preparation phase: acquiring avatar's video with identity label (*i.e.*, gallery construction). It is well known that in de-anonymization attacks [26], [9], the adversary has to obtain the victim's prior knowledge such as video clips and identity labels in advance. Note that, in practice, there are several ways to obtain the victim's gait video clips. For instance, when the target victim shows her identity in the nickname, the adversary can actively record the video of the victim's avatar walking movement. The attacker can also obtain such video clips from the victim's social network contents. We regard the pre-collected video clips as *gallery* in this study.

Attack implementation phase: recording avatar's video without identity label. The adversary can enter the same VR application as the target victim and record the video. Note that, recording a video does not require any permission and approval from the victim. Take a popular VR chatting application, VRChat, as an example. The adversary can focus



(a) The avatar movement signature of user A with Avatar X.

(b) The avatar movement signature of user B with Avatar X.

(c) The avatar movement signature of user B with Avatar Y.

Fig. 3. The relationship between avatar movements and user identities.

her sight on the victim and record the sight view (*i.e.*, recording the contents of the LCD screen of Quest or the remote synchronization on the desktop). Furthermore, to avoid raising any suspicion from the victim, the adversary can transform herself into an invisible camera by employing the avatar with a void appearance (*i.e.*, the avatar becomes invisible).

B. Motivation Example

In this subsection, we utilize a motivational case study to demonstrate the insight of AvatarHunter: *when victims choose avatars with various appearances to hide their identities, it is possible to link their inherent movement signatures to their identities*.

In the case study, two volunteers (*i.e.*, user A and user B) are recruited and required to login into a VR application (*i.e.*, VRChat). Firstly, these two volunteers use the same avatar (*i.e.*, avatar X) to walk for several seconds while we deploy a virtual camera to record their gait video simultaneously. Fig. 3(a) and Fig. 3(b) illustrate the collected video frames and their extracted gait energy images (GEIs), which are representations to characterize human walking properties [27]. It is observed that even though two users have the same appearance in VR, their different gait behaviors cause their GEIs to have significant differences, as illustrated in red rectangles. Then, user B employs another avatar (*i.e.*, avatar Y) and walks again. Fig. 3(c) shows the collected video frames and GEI. We observe that the GEI in this case is similar (although slightly different) to that when user B employs avatar X and is quite different from that of user A as shown in red rectangles. Therefore, it is feasible for AvatarHunter to reveal the avatar's identity based on the observed gait video clips during walking.

C. Closed-world and Open-world Avatar Settings in Attack

According to the adversary's capabilities and the challenges caused by avatars' changeable appearances, we divide AvatarHunter's attack scenarios into two types.

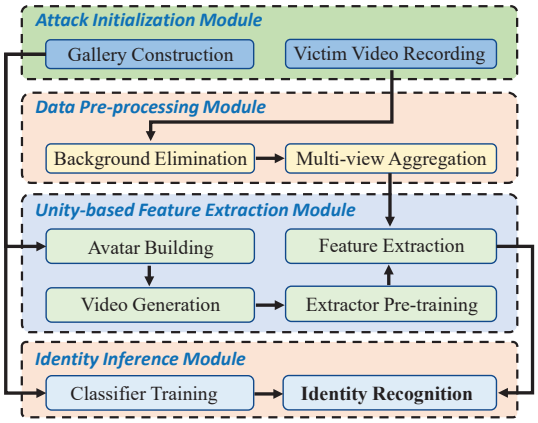


Fig. 4. System overflow.

Scenario-1: closed-world avatar setting. Before launching the attack, the adversary collects the victim's gait videos (gallery) in which a set of several avatars are utilized. Then, the victim hides her external information (*e.g.*, nickname) and changes her avatar into an avatar inside the gallery. For the adversary, the goal is to identify the victim's identity after observing the new arriving gait video clip.

Scenario-2: open-world avatar setting. In this scenario, the victim can hide her external information and utilize a *novel* avatar outside the gallery in the closed-world setting. Compared with scenario-1, it is harder for AvatarHunter to infer the real-world identity since the pre-collected gallery contains no prior knowledge about the victim's gait behaviors in this novel avatar. Note that, as described in Section III-A, we assume the adversary can learn the details about the novel avatar used by the victim after investigating the collected video clips.

IV. The Design of AvatarHunter

As shown in Fig. 4, AvatarHunter consists of four modules: *Attack Initialization Module*, *Data Prepossessing Module*, *Unity-based Feature Extraction Module*, and *Identity Recognition Module*. In this section, we elaborate on each module's details.

A. Attack Initialization Module

In this module, AvatarHunter constructs a *gallery* before launching the de-anonymization attack. Then, during the victim plays VR applications, AvatarHunter collects the video clips imperceptibly.

1) Gallery Construction during Attack Preparation: As described in Section III-A, constructing a gallery is essential in de-anonymization attacks because without it, AvatarHunter does not have prior knowledge of the target victim's identity. In this study, to build the gallery, a common method for the adversary is to login into the VR application where the victims stay and expose their identities to obtain the gait video clips and identity labels. Finally, AvatarHunter collects a gallery with several users containing the target victim before launching de-anonymization attacks. The i-th element of gallery G_i

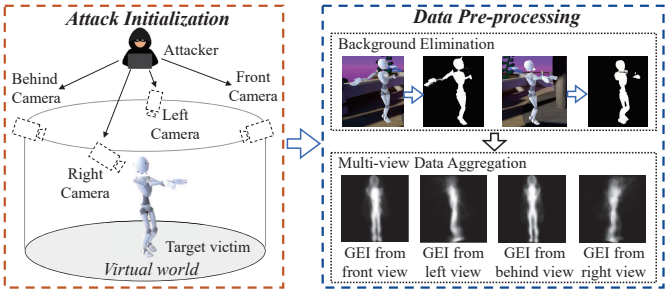
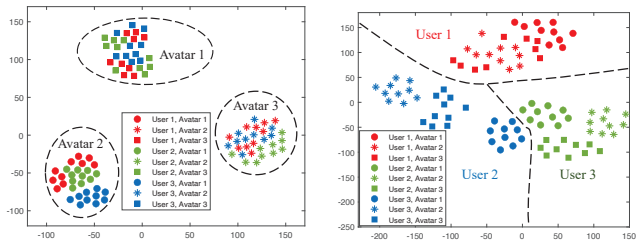


Fig. 5. An illustration of attack initialization module and data pre-processing module.



(a) Feature visualization of benchmark feature extractor. (b) Feature visualization of Unity-based feature extractor.

Fig. 6. Illustrations of the performance improvement when employing the Unity-based feature extractor.

can be denoted as a 3-tuple $\langle V_{G,i}, A_{G,i}, I_{G,i} \rangle$, where $V_{G,i}$ is the collected video clips, $A_{G,i}$ is the avatar employed in this video, and $I_{G,i}$ is the gallery member's identity.

2) Recording the Victim's Video in the Virtual World: For the targeted victim, after building the gallery, AvatarHunter deploys *cameras* to collect her movement video clips in the VR applications. As illustrated in Fig. 5, the victim enters the VR application and hides her identity by making her external information (*e.g.*, nicknames, user profiles) anonymous and employing various avatars. AvatarHunter utilizes N_V accounts to login into the VR application to monitor the victim's behaviors with N_V views following the manners described in Section III-A and Fig. 2. Besides, to record videos imperceptibly, AvatarHunter sets the appearances of N_V avatars as invisible. As shown in Fig. 5, the collected videos contain N_V views, which can be denoted as *gait video trial* $V = \{V_1, V_2, \cdots, V_{N_V}\}$, where V_i represents the video from the i-th invisible camera. Besides, V_i contains $M = T \times S$ frames, where T is the recording duration time and S is the frames per second.

B. Data Pre-processing Module

To improve the de-anonymization performance, AvatarHunter pre-processes the collected video trial V before sending it to the next module.

1) Background Elimination: For a given video trial V with $M \times N_V$ frames, to eliminate the distortion of the VR application background on extracting movement signature, AvatarHunter first eliminates the background of these frames and only preserves the information related to the avatar movement. First, for the video with a specific view, AvatarHunter collects a background image B_I in which the victim's avatar does not stay during the data collection step. Then, based on the collected B_I, for each frame in this video, AvatarHunter applies the background matting scheme proposed by Lin et al. [28] to obtain its corresponding silhouette. Finally, for collected V, AvatarHunter gets a sequence of silhouettes S with the dimension of $M \times N_V$, and each element is:

$$s_{i,j} = B_E(v_{i,j}, B_I), \tag{1}$$

where $v_{i,j}$ is the i-th frame from the j-th view's video V_j, $s_{i,j}$ is $v_{i,j}$'s corresponding silhouette, and $B_E(\cdot)$ is the background elimination operation described in [28]. As shown in Fig. 5, the victim avatar's movement behaviors are obviously exposed after conducting background elimination.

2) Multi-view Data Aggregation: As mentioned in Section IV-A, AvatarHunter employs multiple cameras to collect data with increased diversity to improve the de-anonymization attack's performance. Thus, AvatarHunter aggregates the data from multi-view as the input for the next module.

Since multiple cameras have different locations, sight views, and angles, AvatarHunter needs to address silhouettes from different cameras uniformly. Given silhouette $s_{i,j}$, AvatarHunter firstly calculates its centroid $c_{i,j}$. Then, AvatarHunter cuts the margins of $s_{i,j}$ to obtain a new silhouette $s'_{i,j}$ in which $c_{i,j}$ locates in the center.

To demonstrate that utilizing multiple cameras can provide more fine-grained information, for the i-th camera, we calculate the GEI G_i from silhouettes $\{s'_{1,i}, s'_{2,i}, \cdots, s'_{M,i}\}$ as:

$$G_i = \frac{1}{M} \sum_{j=1}^{M} s'_{j,i}. \tag{2}$$

As shown in Fig. 5, deploying multiple cameras could provide diversified information reflected by the avatar movement. For instance, when deploying $N_V = 4$ cameras, from the front perspective, we can observe the existence of the leg movements, but only from the side perspective (*i.e.*, GEIs from the left and right cameras) can the amplitude of the leg movements be marked clearly. Finally, for the collected video trial V with the dimension of $M \times N_V$, AvatarHunter utilizes the pre-processed silhouettes S' as the input of the feature extraction module and S' can be represented as:

$$S' = \begin{bmatrix} s'_{1,1}, s'_{1,2}, \cdots, s'_{1,N_V} \\ s'_{2,1}, s'_{2,2}, \cdots, s'_{2,N_V} \\ \cdots \\ s'_{M,1}, s'_{M,2}, \cdots, s'_{M,N_V} \end{bmatrix}. \tag{3}$$

C. Unity-based Feature Extraction Module

After pre-processing the collected video trial, AvatarHunter needs to obtain the feature from the victim avatar's gait for final identity inference. Note that, to characterize the gait feature, existing video-based gait feature extractors (*e.g.*, GaitSet [17]) usually leverage deep learning models, which are pre-trained using a public gait video dataset (CASIA-B gait dataset [29]). However, the user's gaits for pre-training

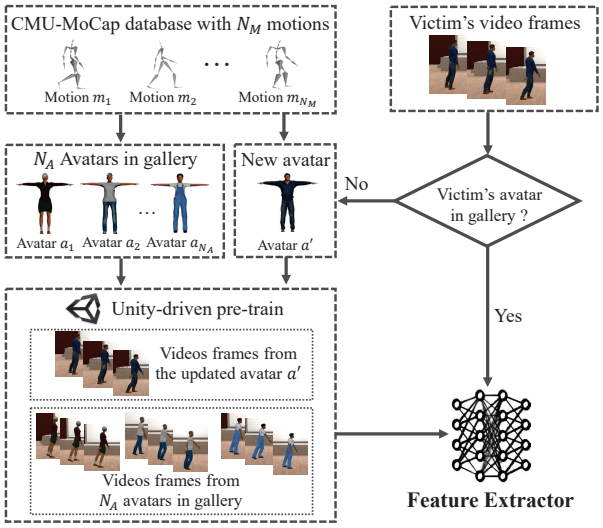


Fig. 7. The pipeline of the Unity-based feature extraction module.

existing extractors (*e.g.*, GaitSet) are collected in the physical world (*i.e.*, none of the avatars are employed, and appearances of different real human volunteers vary broadly) rather than virtual world. Thus, when applying the existing feature extractor on the input S', the extracted biometric signature contains both appearance and movement signatures in which the former is unstable when changing avatars. Fig. 6(a) visualizes the features extracted by GaitSet, the benchmark feature extractor in this study when three volunteers employ three different avatars. Note that features are projected into 2D space using the t-Distributed Stochastic Neighbor Embedding (t-SNE) method [30]. It is observed that features extracted by the benchmark extractor are clustered depending on the avatars' appearances rather than volunteers' movement signatures.

To handle the challenges incurred by the avatar's changeable appearances, as shown in Fig. 7, during the feature extractor's pre-training procedure, instead of using the public gait dataset collected in physical worlds, we employ the gait dataset in the virtual world built by ourselves. In our VR gait dataset, various avatars are designed and driven by the user's movement data in the Unity platform following steps described in Section II-C. Therefore, the retrained scheme could focus on the identity behind the avatar since it has prior knowledge about the user's movement signature under various avatar appearances. The detailed steps are shown below.

1) Avatar Construction in Unity Platform: In Unity platform, AvatarHunter constructs N_A avatars occurring in the *gallery*, which are denoted as $A = \{a_1, a_2, \cdots, a_{N_A}\}$. For i-th target avatar a_i, if it is open-source, we can directly import its source file (*e.g.*, .FBX file) into the Unity platform. Otherwise, we utilize a popular 3D character making tool, MakeHuman [31], to design the closed-source avatar a_i in Unity.

2) Gait Video Clips Generation: To activate the avatar's movement in Unity, we utilize Blender [32] to map the sensor data collected during the human's walking into the avatar's skeleton movement. The sensor data in this study are selected from a third-party CMU-CoMap dataset [33], which utilizes 41 contacted sensors to track the participant during movement. Since AvatarHunter is based on the victim's gait information, we only select sensor data of $N_M = 136$ walking samples involving 29 participants conducting walking behaviors from CMU-CoMap. These N_M motion data samples are denoted as $M = \{m_1, m_2, \cdots, m_{N_M}\}$. Besides, we revise the avatars' skeleton in Unity to conform with the sensor data. To record the video clips from multiple views, we deploy eight cameras and the angles between avatar and cameras are set as $0°$, $72°$, $90°$, $144°$, $180°$, $216°$, $270°$, and $288°$ respectively. Finally, after applying the N_M sensor data on N_A avatars in the gallery, we totally generate the $N_M \times N_A$ eight-view videos P_{pre} as below:

$$P_{pre} = \Phi(A, M) = \begin{bmatrix} P_{1,1}, P_{1,2}, \cdots, P_{1,N_A} \\ P_{2,1}, P_{2,2}, \cdots, P_{2,N_A} \\ \cdots \\ P_{N_M,1}, P_{N_M,2}, \cdots, P_{N_M,N_A} \end{bmatrix}, \quad (4)$$

where $P_{i,j}$ is the video with eight views when applying i-th motion m_i data on j-th avatar a_j.

3) Feature Extractor Pre-training: We regard the generated $N_M \times N_A$ eight-view videos as the pre-training dataset. Note that, each video in the dataset lasts for several minutes, which is far greater than that in the gallery (*e.g.*, usually several seconds). We divide the pre-training dataset into training and validation components following the ratio of 9:1. During the pre-training, the video clips will be transformed into silhouettes as described in Section IV-B and we set the model structure of the feature extractor as the same with GaitSet. Note that, the feature extractor of AvatarHunter can receive the silhouette with arbitrary camera views and output the feature with a uniform dimension.

4) Extracting Feature from Inputting: For a given collected gait video trial V and its pre-processed silhouettes S', if the avatar in V exists in the gallery (*i.e.*, closed-world avatar setting), AvatarHunter takes S' as input and obtains feature F. Otherwise, in the open-world avatar setting, we add the novel avatar a' in the Unity platform and repeat the above-mentioned three steps. The updated pre-training dataset can be represented as:

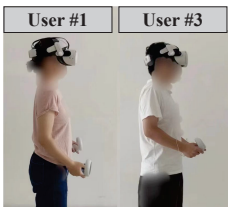
$$P'_{pre} = \Phi(A \cup a', M). \quad (5)$$

Finally, AvatarHunter obtains feature F from the updated feature extractor pre-trained by P'_{pre}.

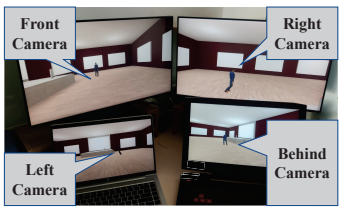
Fig. 6(b) illustrates the features generated by our Unity-based feature extractor. Compared with the benchmark feature extractor (*i.e.*, GaitSet) as shown in Fig. 6(a), AvatarHunter is robust to the avatar appearance changes and characterizes the user's inherent movement signature.

D. Identity Inference.

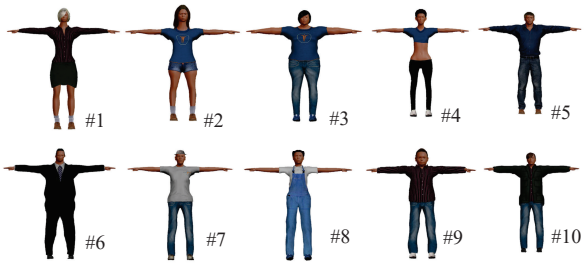
Finally, after obtaining the feature F from the target user, which characterizes the inherent movement signature AvatarHunter generates a classifier Ψ based on the gallery G. Then, it determines the identity of F.



(a) Real-world scenario.



(b) Multiple camera views.



(c) Avatars employed in evaluations.

Fig. 8. Evaluation setup.

1) Gallery-based Classifier Generation: For the gallery G contains N_u users' identities, we denote the i-th element $G_i = \langle V_{G,i}, A_{G,i}, I_{G,i} \rangle$, where $V_{G,i}$ is video clip, $A_{G,i} \in \{a_1, a_2, \cdots, a_{N_A}\}$ is avatar, and $I_{G,i} \in \{u_1, u_2, \cdots, u_{N_U}\}$ is the identity. To build the classifier, AvatarHunter transforms G into G', in which the i-th element G_i is represented as:

$$G'_i = H(G_i) = \langle E(V_{G,i}), I_{G,i} \rangle, \tag{6}$$

where $H(\cdot)$ is the transform operation and $E(\cdot)$ is the feature extraction function defined in Section IV-C. After that, we leverage a random forest in which G' is the input to generate the classifier Ψ.

2) Identity Recognition: Finally, for the F extracted from V, AvatarHunter utilizes Ψ to determine its identity J as below:

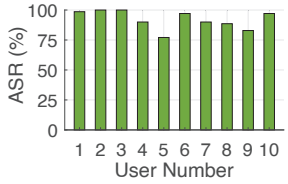
$$\Psi(F) = J, \tag{7}$$

where $J \in \{u_1, u_2, \cdots, u_{N_U}\}$. If J is the victim's real identity, AvatarHunter conducts the de-anonymization attack successfully.

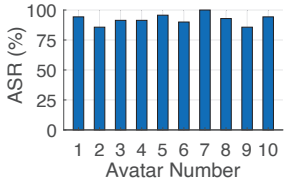
V. Evaluation

A. Evaluation Setup

Experimental conditions. In this study, we evaluate the de-anonymization performance of AvatarHunter in VRChat, a popular VR application that is developed for real-time social interactions. 10 volunteers are recruited for the experiments. As shown in Fig. 8(a), each volunteer is required to wear the Oculus Quest 2 device and then enter VRChat. To launch the attack, when the volunteer walks in the virtual environment, four cameras are deployed in the front, left, right, and behind the volunteer to record the video simultaneously. Fig. 8(b) illustrates frames from four sight views at a given time. During the data collection period, each volunteer is asked to employ 10 avatars as shown in Fig. 8(c) and walks 10 times when employing each avatar.



(a) ASRs on each user.



(b) ASRs on each avatar.

Fig. 9. Performance on closed-world avatar setting.

Dataset Description and Metric. We totally obtain 10 users $\times 10$ avatars $\times 10$ trials $= 1000$ gait video trials in which each trial contains 4 videos with different views. For each view, the video lasts at least 4.5 seconds with 30 frames per second. During the evaluation, for each avatar used by each volunteer, we randomly choose 3 trials as gallery and regard rest the other 7 trials as AvatarHunter's testing dataset. Thus the gallery and testing dataset have 300 and 700 trials respectively. In the experiments, we choose attack success rate (ASR) as the evaluation metric, which is defined as $\frac{N_S}{N_T}$, where N_T is the total attack times and N_S is the times when AvatarHunter successfully reveals the victim's identity.

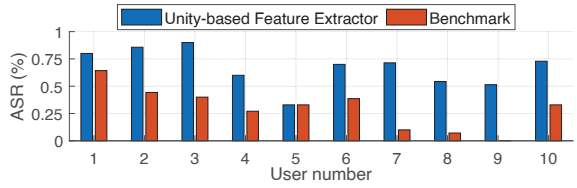
B. Overall Performance of AvatarHunter

In this subsection, we introduce AvatarHunter's performance under closed-world and open-world avatar settings.

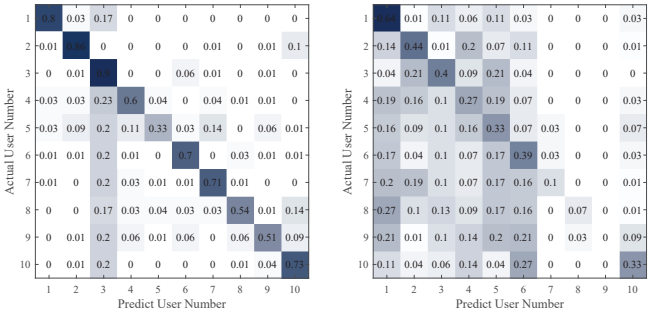
1) Performance under Closed-world Avatar Setting: In this scenario, all avatars existing in the testing dataset are covered by the gallery. To evaluate AvatarHunter's performance, we train the classifier based on the gallery containing 300 trials and conduct identity inference using the testing dataset containing 700 trials. For a given trial containing four views, 30 frames (*i.e.*, 1 second) from each view are chosen as the input of AvatarHunter. Fig. 9(a) illustrates the ASR among volunteers. It is observed that the overall ASR is 92.1%, and ASRs among different volunteers vary from 77.1% at user #5 to 100.0% at user #2 and user #3. Even in the worst case, the ASR is far larger than the random guess (*i.e.*, 10% in this experiment setting). From the aspects of avatar, it is observed from Fig. 10(b) that AvatarHunter's performance varies when choosing different avatars. ASRs among different avatars vary from 85.7% to 100%. In the worst case, the ASR is still far larger than the random guess. Therefore, it proves the effectiveness of AvatarHunter in de-anonymizing the user's identity in VR scenarios.

2) Performance under Open-world Avatar Setting: In the open-world setting, the victim's avatar does not exist in the gallery. Therefore, for each avatar, we exclude its corresponding 30 trials in the gallery and train AvatarHunter's classifier based on the updated gallery. Then, we apply the classifier to the remaining 70 trials of this avatar. AvatarHunter achieves an overall ASR of 66.9%. The performance is worse than that in the closed-world setting because the gallery lacks any prior knowledge about the avatar employed by the victim.

We also compare AvatarHunter with an existing popular gait-based identity inference scheme (*i.e.*, GaitSet [17]). As shown in Fig. 10(a), among 10 employed users, AvatarHunter



(a) ASRs on each avatar.



(b) AvatarHunter's confusion matrix. (c) Benchmark's confusion matrix.

Fig. 10. Performance on open-world avatar setting.

TABLE I
INFERENCE SUCCESS RATE OF AVATARHUNTER

Frame length	10	20	30	60	90
ASR in scenario-1 (%)	87.0	91.6	92.1	93.0	93.4
ASR in scenario-2 (%)	63.6	65.3	66.9	65.9	65.0

achieves the best ASR of 90.0% in user #3 and the worst ASR of 32.9% in user #5. Meanwhile, the best and worst ASRs of the benchmark attack are 64.3% in user #1 and 0.0% in user #9 respectively. Furthermore, Fig. 10(b) and Fig. 10(c) show the confusion matrix between AvatarHunter and the benchmark attack. It is observed that compared with the benchmark, for AvatarHunter, most darker areas locate at the diagonal of the matrix. AvatarHunter achieves better performance because its feature extractor is pre-trained by various avatars' videos generated in Unity platform, which improves its robustness to appearance changes. The results successfully demonstrate the superiority of our proposed Unity-based feature extractor in improving de-anonymization performance.

3) Time overhead: The time overhead consists of three components: feature extractor pre-training time, classifier training time, and real-time inference time. For a server with 2 GPUs (GeForece RTX 2080 Ti), Ubuntu 18.04.6 LTS OS, Intel Xeon E5-2678 v3 CPU, and 128 GB RAM, the time overheads of three components are 8.05 hours, 2.3 seconds, and 123.5 ms respectively. Note that, since the training procedures can be done ahead of the attack, the real-time inference time of 123.5 ms is acceptable in real-world scenarios.

C. Impact of Various Factors on AvatarHunter

1) Length of Recording Video: In Section V-B, for each view of a given trial, 30 frames (*i.e.*, 1 second) are chosen as the input of AvatarHunter. We evaluate the impact of the frame length on AvatarHunter's performance. As listed in TABLE I, we set the frame length of each view as 10, 20, 30, 60, and 90, respectively. It is observed that when increasing frame length, the ASRs on both open-world and closed-world avatar settings increase slightly. Especially, even with 10 frames (0.33 seconds), the ASRs at the closed-world and open-world scenarios are higher than 87% and 63% respectively, which is far larger than the random guess (*i.e.*, ASR of 10%). The experimental results show that AvatarHunter works well even when the collected video has a small length.

TABLE II
PERFORMANCE WHEN CHANGING THE GALLERY SIZE

Gallery size	10%	30%	50%	70%	90%
ASR in scenario-1 (%)	78.8	92.1	94.6	94.0	98.0
ASR in scenario-2 (%)	63.0	66.9	64.0	62.7	66.0

TABLE III
PERFORMANCE WHEN CHANGING CAMERA COMBINATION

Camera combination	F	FB	FR	FRL	FBRL
ASR in scenario-1 (%)	72.9	90.1	91.9	88.4	92.1
ASR in scenario-2 (%)	51.3	62.3	68.6	66.4	66.9

2) Gallery Size: In this study, the gallery serves as the role of "training dataset" of the classifier. It is well known that the abundance of "training samples" would cause a great influence on the final inference performance. We evaluate the impact of gallery size by adjusting the proportion of trials in the gallery. The results are listed in Table II. It is observed that the ASR increases from 78.8% to 98.0% when the gallery proportion increases from 10% to 90%. Note that, After the gallery proportion exceeds 30%, the performance has limited improvement when increasing gallery size. When selecting gallery proportion as 30%, since each trial contain the victim's behaviors for only 1 second, building a gallery containing 30 trials from each victim is easy to achieve. Thus, AvatarHunter causes a light burden for the adversary.

3) The Number of Cameras: As mentioned in Section IV-B, to improve the de-anonymization performance, AvatarHunter deploys four cameras which are in the front (F), left (L), right(R), and behind (B) of the victim respectively. We evaluate the robustness of AvatarHunter when choosing different camera combinations. As listed in Table III, when increasing the number of cameras, the ASR will increase in the open-world avatar setting. Besides, different angles also cause different ASRs. For instance, the performance when using front and behind cameras (FB) is less than that of front and right cameras (FR). However, even utilizing only one camera, it is possible for AvatarHunter to achieve effective performance (*i.e.*, ASR of 72.9% in the closed-avatar setting when utilizing only the front camera). In summary, utilizing multiple cameras improves the performance of AvatarHunter, and AvatarHunter is still robust in the single-camera scenario.

VI. DISCUSSIONS

A. Countermeasures

1) Detecting Suspicious Users in VR: For AvatarHunter, recording the victim avatar's behaviors is essential during the attack phase. Thus, we can detect suspicious users who focus their insights on a fixed user or choose an avatar with a void

appearance. By excluding these suspicious users, it can reduce the victim's possibility of privacy leakage.

2) Adding Noises when Mapping Sensor Data to Avatar's Movements: As described in Section II-C, the movement of the avatar is driven by the data collected by the sensor of VR devices. Therefore, by intentionally adding noises into sensor data during the avatar movement generation process, it is feasible to cause AvatarHunter to extract no meaningful information about the target victim's movement signature.

3) Restricted Access Control: One of the most straightforward solutions is enhancing the access control of the VR application. For the user who requires high demands of anonymity, they can only share their VR scenarios with the users they trust. However, this countermeasure only works when users can control their VR applications (*e.g.*, the user is the administrator of the chatting room of VRChat).

B. Limitations and Future Works

To apply AvatarHunter to more broad VR scenarios, the following limitations need to be addressed.

1) Enlarging Dataset Including Avatars with Non-humanoid Appearance and More Users: In this study, only humanoid avatars as shown in Fig. 8(a) are studied. However, for the avatars with the appearance of non-humanoid appearance, since their skeletons are different from that of humans, it is hard for the feature extractor to extract human movement signatures. Involving the animal-like avatars into the Unity-based feature extraction module, or proposing more advanced techniques are potential solutions and we leave them for future works.

2) Victims not in Gallery: Acquiring prior knowledge about the target user is an essential prerequisite for user recognition or authentication schemes. In this study, AvatarHunter builds a gallery to store the victim's avatar gait information in VR scenarios. For users who are failed to extract their gaits in VR, extracting the features from their gaits in the physical world is a potential solution and we will study it in the future.

3) Other Action Types: In this study, we only collect the videos about the victim's walking behaviors and extract the gait-based features. Although gait is one of the most popular features used by existing user identification schemes, there are other actions (*e.g.*, running, throwing) users can conduct in VR scenarios. Building a de-anonymization attack based on universal actions is a promising future research direction.

4) Other VR Device Types: For other VR device types such as HTC Vive Pro 2 and Valve Index, AvatarHunter should also work since they have the same working principles as Quest used in our evaluation. Besides, by incorporating advanced full-body tracking and hand gesture tracking sensors into VR devices, AvatarHunter is expected to achieve more effective performance, which will be studied in the future.

VII. Related Work

Sensor-based user recognition. There are a bunch of works about using sensor data to conduct user recognition, which can be applied to designing secure and efficient authentication schemes in VR scenarios [34], [6], [35], [36], [37], [38], [39]. OcuLock [6] and GaitLock [35] leverage users' eye globe movement triggered by immersive 3D visual content and gait signature recorded by onboard IMUs to recognize login users respectively. Other interactions like pointing, grabbing, typing with controllers, bowling, and shooting arrows [12], [40], [41] can also be utilized to build authentication schemes. However, these user recognition schemes highly depend on accurate sensor data extracted from built-in sensors in VR devices, which limits their implementations in remote attacking scenarios.

Video-based user recognition. The research community also looked into the computer-vision-based user recognition task using the videos of users walking in different scenarios like lab-setting[29], market[42] and, university campus[43]. [15], [16], [17] achieve great performance on these specific tasks. However, as analyzed in Section IV-C, these videos (image sequences) are captured in the physical world and thus contain both the appearance and movement signature of users, which will make it challenging to put them into use in VR scenarios, especially when users can change their appearance by putting on different avatars. ReAvatar [26] leverages videos in VR to perform user recognition. However, it requires the adversary to lure the user to explicitly conduct actions, which easily incurs the user's suspicion and limits its practicability.

Other privacy breaches in VR. Recent studies point out that privacy information including text inputs, locations, and speeches can be revealed by attacks. Arafat et al. [44] leverage the channel state information (CSI) existing in the VR device's wireless communication to track the user's gesture and infer the text input. Kotaru et al. [45] also utilize CSI to track the user's location in the VR scenario. Shi et al. [9] utilize the vibration data collected by sensors in the VR device to infer the user's speeches. However, these methods either require intruding on the target victim's network or injecting malicious applications.

VIII. Conclusion

In this study, we have proposed AvatarHunter, a non-intrusive and user-unconscious de-anonymization attack in the VR scenario. AvatarHunter imperceptibly collects victim's gait videos in VR applications and leverages a Unity-based feature extractor to characterize the victim's movement signature which is immune to avatar appearance changes. We deploy AvatarHunter in a real-world VR application and the experimental results show that AvatarHunter can effectively reveal the identity behind the employed avatar in both closed-world and open-world avatar settings. AvatarHunter reveals a severe privacy threat for the stakeholders of VR-related areas, which is expected to inspire studies of countermeasures defending against it in the future.

Acknowledgment

This research was supported in part by National Natural Science Foundation of China under Grants No. 61972453, 72171145, 62132013, and NSERC of Canada.

REFERENCES

[1] G. V. Research, "Virtual reality market share & trends report, 2021-2028," https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market, 2021.

[2] Y. Wang, Z. Su, N. Zhang, D. Liu, R. Xing, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *arXiv preprint arXiv:2203.02662*, 2022.

[3] V. Inc., "Vrchat," https://hello.vrchat.com/, 2021.

[4] P. Casey, I. Baggili, and A. Yarramreddy, "Immersive virtual reality attacks and the human joystick," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 550–562, 2021.

[5] W.-J. Tseng, E. Bonnail, M. McGill, M. Khamis, E. Lecolinet, S. Huron, and J. Gugenheimer, "The dark side of perceptual manipulations in virtual reality," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI)*, 2022, pp. 612:1–612:15.

[6] S. Luo, A. Nguyen, C. Song, F. Lin, W. Xu, and Z. Yan, "Oculock: Exploring human visual system for authentication in virtual reality head-mounted display," in *27th Annual Network and Distributed System Security Symposium (NDSS)*, 2020.

[7] R. Miller, N. K. Banerjee, and S. Banerjee, "Using siamese neural networks to perform cross-system behavioral authentication in virtual reality," in *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, 2021, pp. 140–149.

[8] R. Trimananda, H. Le, H. Cui, J. T. Ho, A. Shuba, and A. Markopoulou, "OVRseen: Auditing network traffic and privacy policies in oculus VR," in *31st USENIX Security Symposium (USENIX Security)*, 2022.

[9] C. Shi, X. Xu, T. Zhang, P. Walker, Y. Wu, J. Liu, N. Saxena, Y. Chen, and J. Yu, "Face-mic: Inferring live speech and speaker identity via subtle facial dynamics captured by ar/vr motion sensors," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2021, p. 478490.

[10] D. Edward, "6 best meeting & collaboration tools in vr for oculus quest 2," https://allvirtualreality.com/review/best-meeting-collaboration-tools-vr-quest.html, 2021.

[11] M. R. Miller, F. Herrera, H. Jun, J. A. Landay, and J. N. Bailenson, "Personal identifiability of user tracking data during observation of 360-degree vr video," *Scientific Reports*, vol. 10, no. 1, pp. 1–10, 2020.

[12] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt, "Behavioural biometrics in vr: Identifying people from body motion and relations in virtual reality," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI)*, 2019, pp. 110:1–110:12.

[13] T. Mustafa, R. Matovu, A. Serwadda, and N. Muirhead, "Unsure how to authenticate on your vr headset? come on, use your head!" in *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics (IWSPA)*, 2018, pp. 23–30.

[14] C. George, M. Khamis, D. Buschek, and H. Hussmann, "Investigating the third dimension for authentication in immersive virtual reality and in the real world," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2019, pp. 277–285.

[15] D. Fu, D. Chen, J. Bao, H. Yang, L. Yuan, L. Zhang, H. Li, and D. Chen, "Unsupervised pre-training for person re-identification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2021, pp. 14 750–14 759.

[16] T. He, X. Jin, X. Shen, J. Huang, Z. Chen, and X.-S. Hua, "Dense interaction learning for video-based person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 1490–1501.

[17] H. Chao, K. Wang, Y. He, J. Zhang, and J. Feng, "Gaitset: Cross-view gait recognition through utilizing gait as a deep set," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3467–3478, 2022.

[18] B. Sinha, "Unity: The biggest platform for creating vr content," 2017. [Online]. Available: https://digital.hbs.edu/platform-digit/submission/unity-the-biggest-platform-for-creating-vr-content/

[19] VRChat, "Thanks to our community for making 2018 vrchat's best year yet!" https://twitter.com/VRChat/status/1086389685268635648, 2019.

[20] L. Facebook Technologies, "Oculus quest 2," https://www.oculus.com/quest-2/, 2022.

[21] H. Corporation, "Vive - vr headsets, games, and metaverse life," https://www.vive.com/us/, 2022.

[22] T. Baker, "The best vr headset in 2022: all the latest devices compared," https://www.gamesradar.com/the-best-vr-headset/, 2022.

[23] M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate, and M. Botsch, "The effect of avatar realism in immersive social virtual realities," in *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2017, pp. 39:1–39:10.

[24] D. Takahashi, "59% of vr developers use unity, but devs make more money with unreal," https://uploadvr.com/vr-developers-unity-unreal/, 2017.

[25] U. Technologies, "Getting started with vr development in unity," https://docs.unity3d.com/Manual/VROverview.html, 2021.

[26] B. Falk, Y. Meng, Y. Zhan, and H. Zhu, "Poster: Reavatar: Virtual reality de-anonymization attack through correlating movement signatures," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021, pp. 2405–2407.

[27] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[28] S. Lin, A. Ryabtsev, S. Sengupta, B. L. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Real-time high-resolution background matting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8762–8771.

[29] C. for Biometrics and S. Research, "Gait databases," http://www.cbsr.ia.ac.cn/english/Gait\%20Databases.asp, 2022.

[30] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[31] M. Community, "Makehuman," https://github.com/makehumancommunity/makehuman, 2022.

[32] BLENDER., "Blender," https://www.blender.org/, 2022.

[33] C. G. Lab, "Motion capture database," http://mocap.cs.cmu.edu, 2021.

[34] C. E. Rogers, A. W. Witt, A. D. Solomon, and K. K. Venkatasubramanian, "An approach for user identification for head-mounted displays," in *Proceedings of the 2015 ACM International Symposium on Wearable Computers (ISWC)*, 2015, pp. 143–146.

[35] Y. Shen, H. Wen, C. Luo, W. Xu, T. Zhang, W. Hu, and D. Rus, "Gaitlock: Protect virtual and augmented reality headsets using gait," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, pp. 484–497, 2019.

[36] F. Mathis, J. H. Williamson, K. Vaniea, and M. Khamis, "Fast and secure authentication in virtual reality using coordinated 3d manipulation and pointing," *ACM Transactions on Computer-Human Interaction*, vol. 28, no. 1, pp. 6:1–6:44, 2021.

[37] S. Eberz, K. B. Rasmussen, V. Lenders, and I. Martinovic, "Evaluating behavioral biometrics for continuous authentication: Challenges and metrics," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (AsiaCCS)*, 2017, pp. 386–399.

[38] A. Kupin, B. Moeller, Y. Jiang, N. K. Banerjee, and S. Banerjee, "Task-driven biometric authentication of users in virtual reality (VR) environments," in *MultiMedia Modeling - 25th International Conference (MMM)*, 2019, pp. 55–67.

[39] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1961–1974, 2015.

[40] A. Ajit, N. K. Banerjee, and S. Banerjee, "Combining pairwise feature matches from device trajectories for biometric authentication in virtual reality environments," in *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, 2019, pp. 9–16.

[41] 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 *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI)*, 2021, pp. 517:1–517:11.

[42] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1116–1124.

[43] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European conference on computer vision*, 2016, pp. 17–35.

[44] A. A. Arafat, Z. Guo, and A. Awad, "Vr-spy: A side-channel attack on virtual key-logging in vr headsets," in *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, 2021, pp. 564–572.

[45] M. Kotaru and S. Katti, "Position tracking for virtual reality using commodity wifi," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2671–2681.