Motion Passwords

Christian Rack
Human-Computer Interaction Group, University of Würzburg
Germany
christian.schell@uni-wuerzburg.de

Lukas Schach
Human-Computer Interaction Group, University of Würzburg
Germany
lukas.schach@uni-wuerzburg.de

Felix Achter
Human-Computer Interaction Group, University of Würzburg
Germany
felix.achter@stud-mail.uni-wuerzburg.de

Yousof Shehada
Human-Computer Interaction Group, University of Würzburg
Germany
yousof.shehada@stud-mail.uni-wuerzburg.de

Jinghuai Lin
Human-Computer Interaction Group, University of Würzburg
Germany
jinghuai.lin@uni-wuerzburg.de

Marc Erich Latoschik
Human-Computer Interaction Group, University of Würzburg
Germany
marc.latoschik@uni-wuerzburg.de

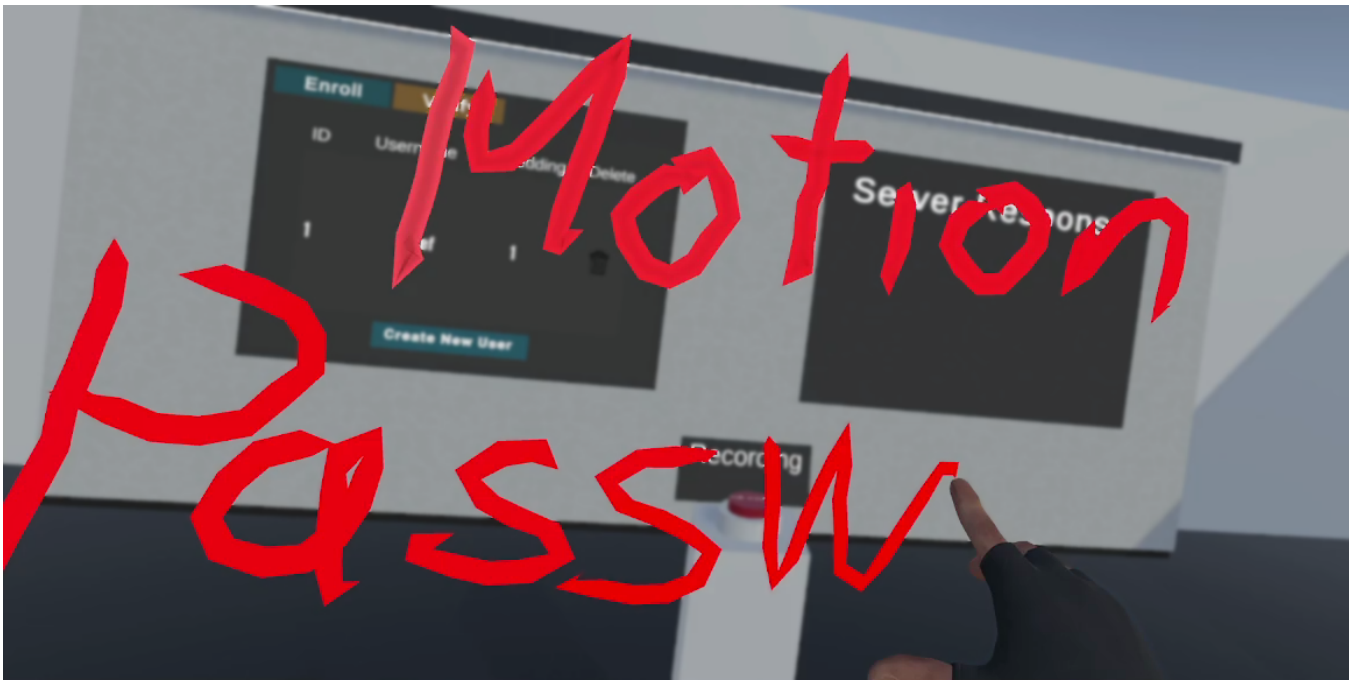

Figure 1: The concept of "Motion Passwords" involves XR users verifying their identity by physically spelling out their password in the air. This screenshot from our VR prototype shows a user writing their Motion Password. This Unity application supports user enrollment and verification, demonstrating the feasibility of motion-based verification.

Abstract

This paper introduces "Motion Passwords", a novel biometric authentication approach where virtual reality users verify their identity by physically writing a chosen word in the air with their hand controller. This method allows combining three layers of verification: knowledge-based password input, handwriting style analysis, and motion profile recognition. As a first step towards realizing this potential, we focus on verifying users based on their motion profiles. We conducted a data collection study with 48 participants, who performed over 3800 Motion Password signatures across two sessions. We assessed the effectiveness of feature-distance and similarity-learning methods for motion-based verification using the Motion Passwords as well as specific and uniform ball-throwing signatures used in previous works. In our results, the similarity-learning model was able to verify users with the same accuracy for both signature types. This demonstrates that Motion Passwords, even

VRST '24, October 09–11, 2024, Trier, Germany

ACM ISBN 979-8-4007-0535-9/24/10
https://doi.org/10.1145/3641825.3687711

when applying only the motion-based verification layer, achieve reliability comparable to previous methods. This highlights the potential for Motion Passwords to become even more reliable with the addition of knowledge-based and handwriting style verification layers. Furthermore, we present a proof-of-concept Unity application demonstrating the registration and verification process with our pretrained similarity-learning model. We publish our code, the Motion Password dataset, the pretrained model, and our Unity prototype on https://github.com/cschell/MoPs

CCS Concepts

• **Security and privacy → Usability in security and privacy**; **Biometrics**; **Graphical / visual passwords**; **Multi-factor authentication**.

Keywords

Authentication, Verification, Extended Reality, Biometrics

ACM Reference Format:

Christian Rack, Lukas Schach, Felix Achter, Yousof Shehada, Jinghuai Lin, and Marc Erich Latoschik. 2024. Motion Passwords. In *30th ACM Symposium on Virtual Reality Software and Technology (VRST '24), October 09–11, 2024, Trier, Germany*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641825.3687711

1 Introduction

Traditional user-verification methods based on username and password combinations face significant challenges in Virtual, Augmented, and Mixed Reality (VR, AR, MR, or XR: eXtended Reality for short). Not only do password theft and brute force attacks also persist for XR, but practical and ergonomic difficulties of using soft or hardware keyboards for password entry pose unique challenges [4, 10, 14, 38]. Such interactions may disrupt the immersive experience fundamental to XR, motivating a need for alternative verification mechanisms.

In response to these challenges, we introduce and evaluate the concept of "Motion Passwords". Motion Passwords are not typed but written in the air, as demonstrated in Figure 1. This allows three distinct layers of verification: first, like traditional passwords this approach enables knowledge-based verification, allowing the system to verify if the correct word has been entered. Second, in-air writing can also reflect the user's accustomed writing style [11], which permits the application of models that verify handwriting characteristics, such as the style and order of strokes. Third, Motion Passwords capture the user's unique motion profile, which has been shown to include highly identifying patterns [19, 26]. Additionally, the in-air writing approach could reduce the ergonomic and usability issues associated with using keyboards in immersive environments.

We explore the potential of Motion Passwords, specifically focusing on the third layer, and develop and evaluate a motion-based verification model. We compare two motion-based techniques used by previous works, a feature-distance method used by Li et al. [16] and a similarity-learning method used by Rack et al. [32]. We evaluated both techniques on a new dataset of 48 users executing Motion Passwords and an existing dataset from Miller et al. [23], which includes 41 users performing specific ball-throwing actions. Comparing signatures from both datasets allows us to determine if the increased complexity of 'writing' compared to specific and uniform 'ball-throwing' actions, which have been shown to provide high identifying potential [1, 15, 17, 23], affects verification reliability. Altogether, our contributions include the following:

(1) Introduction of the concept of "Motion Passwords" and release of our new dataset, featuring 48 participants performing over 3800 Motion Passwords across two sessions with a typical VR setup (Meta Quest 2).
(2) Evaluation of Motion Passwords as complex signatures for motion-based verification in comparison to specific and uniform ball-throwing signatures.
(3) Preliminary evaluation of Motion Passwords against shoulder surfing attacks.
(4) Release of our proof-of-concept Unity application featuring our trained similarity-learning model. This application not only demonstrates the motion-based verification process but also offers an interactive experience to try it out hands-on.

Our work marks the first step in exploring Motion Passwords as a reliable method for motion-based user verification in XR. The results show that Motion Passwords provide verification reliability comparable to specific ball-throwing signatures, even when using only motion-based identification techniques. We observed that our model primarily focuses on the user's motion profile rather than the written word itself. This suggests that future work can combine motion-based verification with models that verify the actual word or the user's handwriting to achieve even higher verification reliability. Our Motion Password dataset lays the groundwork for future investigations into these additional verification approaches. Overall, Motion Passwords present a promising alternative to both purely motion-based verification methods, like specific ball-throws, and purely knowledge-based approaches, such as traditional passwords.

2 Related Work

2.1 Handwritten Signatures as Biometric Input

Handwritten signatures have long been recognized as a viable biometric input for identity verification [29]. Their uniqueness stems from the distinct neuromuscular patterns exhibited during the signing process, which are difficult to replicate [5]. This uniqueness encompasses both static and dynamic traits, such as the shape of the signature and the speed, pressure, and rhythm of the signing motion [12]. Signatures maintain relative consistency over time, making them reliable for repeated verification [35].

The integration of handwritten signatures into existing workflows is straightforward due to their widespread acceptance and established use in legal and financial contexts. This ease of integration further supports their viability as a biometric input.

Motion Passwords in virtual reality extend the concept of handwritten signatures into 3D space, making them potentially stronger than traditional 2D signatures. First, the additional dimension allows for the emergence of user-specific writing patterns in 3D space, which should be more complex and hence even harder to replicate. Second, Motion Passwords capture more data points, including the position and orientation of the writing hand, off-hand, and head, not just the position tip of the writing pen. Consequently, Motion Passwords leverage the advantages of traditional signatures while introducing a new level of complexity and source of biometric signals.

2.2 Motion-Based User Verification in XR

Motion-based user verification extends biometric analysis to a broader range of human motions, and there is already a sizable body of literature that discusses using motions for user recognition. In the following, we focus on the context of typical XR systems that track the head and at least one hand. We follow the terminology defined by Jain et al. [7]: biometric user recognition systems can serve either *identification* or *verification* tasks. Identification involves determining a user's identity from a set of known identities, which is typically relevant for access control or surveillance applications. Verification on the other hand confirms or denies a user's claimed identity, like logging into one's account or checking in on an airport with a passport. With the concept of Motion Passwords, we specifically target the verification scenario.

Most of previous works discuss the *identification* task. Rogers et al. [34] were the first to use motions from XR users to explore the feasibility of user identification within a set of 20 users, followed by Pfeuffer et al. [28] who investigated several controlled VR tasks. Subsequently, Miller M. et al. [21] demonstrated that individuals can be re-identified quite accurately even within a larger group of users (N=511). Up to this point, research had focused on whether motion data is identifying at all, and the investigated scenarios were limited to fairly specific and well-defined user actions. Rack et al. [30] collected a dataset from 71 users playing Half-Life: Alyx. The authors showed that deep learning models are capable of identifying users even in contexts, where the user task (i.e., 'play the game') allows a wide spectrum of possible actions and user motions. Recently, Nair et al. [26] demonstrated that motion-based identification is possible even within a substantial user base of 50,000 individuals. Altogether, these works demonstrate that motion data carries a significant identifying signal that can be used as biometric signature for user recognition tasks.

In the context of motion-based *verification*, literature is comparatively sparse and focused on scenarios with very specific user tasks. Li et al. [16] designed a system that required users to nod along to music for a few seconds, wearing Google Glasses, and achieved an average verification accuracy of about 96%. Miller et al. [20, 22] evaluated verification performance with different VR devices by prompting users to throw virtual balls. Both tasks, nodding and throwing, produce motion sequences, where the resulting movement trajectories are highly constrained and very similar. While these user tasks are simple and have been shown to work well for verification, conceptually they may be problematic from a security perspective. XR users can unknowingly reveal a lot of information about themselves, either by accident or through malicious application design [27], and unintentionally perform their verification signatures in front of adversaries. This allows attackers to observe these simple motions and repeat them to gain unauthorized access.

In contrast, by letting users write personalized – potentially more complex – words, Motion Passwords combine the ideas of biometric-based and knowledge-based verification: attackers not only have to know the correct password but also have to somehow infer *how* their victim writes that password in 3D space. Lu et al. [18] also explored verification based on freestyle in-air handwriting, though it relied on a camera and a glove device for data capture and had participants write only two words in a single session.

2.3 Verification Methods

Verification requires a method that determines the similarity between a known and an unknown biometric sample. For motion-based verification, previous works used two types of methods to achieve this: feature-distance and similarity-learning methods. *Feature-distance* methods are conceptually simple as they determine the similarity — or distance — between two samples directly in the feature space, hence they do not require any sort of training phase. Li et al. [16] evaluated three different feature-distance methods and found that Dynamic Time Warping (DTW) worked best for verifying users based on their head nodding. However, in the context of motion sequences, this can only be expected to work if the underlying user action of the two samples is the same, e.g., both samples show a throwing motion. If the underlying actions are different, the distances of the resulting trajectories in the feature space become too large, even if the actions were performed by the same user.

Similarity-learning models use machine learning to learn the motion profile of individuals even within complex and arbitrary motions [22, 32]. Subsequently, the similarity calculation between two samples is done within the representation space learned by the neural networks instead of the feature space. This method can identify users even on arbitrary motions, as demonstrated by Rack et al. [32], but comes at the cost of requiring a pretraining phase.

Against this backdrop, we compare feature-distance and similarity-learning as foundation for the verification task in our analyses. Feature-distance approaches offer themselves as a simple and resource effective method for the verification task, yet have only been evaluated on specific motions. Similarity-learning is a more sophisticated technique that requires pretraining on larger datasets but may yield better results on the comparatively more complex patterns of Motion Passwords.

3 Datasets

We utilized three distinct datasets for our research: our newly created "Motion Passwords" (MoP) dataset, the Ball-Throwing (BaT) dataset from Miller et al. [23], and the "Who Is Alyx?" (WiA) dataset from Rack et al. [30]. Each dataset provides spatial (x, y, z) and rotational (quaternion: x, y, z, w) tracking data of the head-mounted display (HMD) and both hand controllers of the VR users.

3.1 Motion Passwords

We conducted a data collection study to create a dataset of Motion Password inputs. Participants were required to attend two separate sessions where they were instructed to write specific words multiple times with either hand. For this, we created a virtual environment with Unity to guide participants through the study. The study has been approved by the Research Ethics Committee of our faculty.

3.1.1 Main Data Collection Study. We recruited 48 participants (9 males, 39 females), aged between 18 and 27 years (average age: 22), via our institute's student participant system. Most participants were right-handed, and only two were left-handed.

Participants were fully briefed on the concept of Motion Passwords and the study's goals and the data collection process. In the VR environment, they followed instructions displayed on a virtual

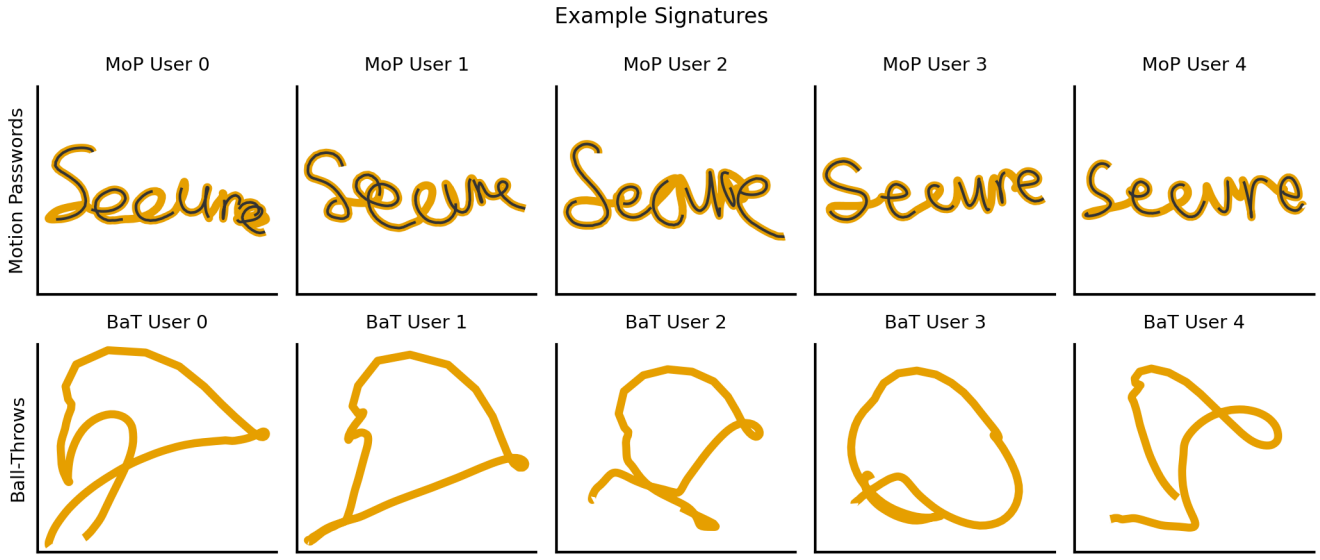

Figure 2: Example 2D projections of signatures from our Motion Password (MoP) dataset and from the Ball-Throwing (BaT) dataset from Miller et al. The black lines in MoP signatures represent trigger button presses.

blackboard using a Meta Quest 2 headset and controllers. Participants attended two separate sessions, writing specific words five times with each hand. The words included 'Motion' (both sessions), 'Secure' (first session), 'Password' (second session), and two random words (same two across both sessions). Participants pressed the trigger button on the controller to write and used virtual buttons within the scene to repeat or proceed with their writing tasks. This resulted in a total of 4 words × 5 repetitions × 2 hands × 2 sessions = 80 individual writing sequences per participant. In total, we collected 3840 Motion Passwords, retaining 3800 after removing non-meaningful inputs (e.g., incomplete words). The average length of a Motion Password was 6.2 seconds, with a minimum of 2.1 seconds and a maximum of 18.2 seconds. To determine when a Motion Password started and when it ended we selected the frames between the first and last trigger button presses. Figure 2 visualizes resulting Motion Passwords from sample users. Note that we did not use the button presses as input for our models in this work.

3.1.2 Fully Informed Attack Data Collection Study. In addition to the previously mentioned attack types, we aimed to include a fully informed attack scenario. To achieve this, we designed a shoulder surfing attack setup within the Unity scene. In this setup, participants watched videos of a hypothetical victim performing their Motion Password. They were then instructed to carefully observe and mimic the victim's password.

However, our initial attempts highlighted several issues. Participants required more detailed instructions and training to understand which aspects to focus on and to apply the necessary diligence expected of a real attacker. For instance, many participants took significantly longer to complete the Motion Password than the victim or failed to notice details such as the order of strokes. These shortcomings indicated that our simulated attacks did not accurately reflect a real-world scenario, leading us to exclude this part of the study from our main analyses.

Despite this, we sought to provide some insight into the system's sensitivity to fully informed attacks. We conducted a follow-up study with six colleagues (1 female, 5 male, all right-handed) from our research group. Before the study, we thoroughly instructed them on the concept of Motion Passwords, showing example videos and explaining key aspects attackers should focus on: matching the writing speed and order of strokes and paying attention to letter size and hand inclination.

We included two victims (both male and right-handed) who wrote the words 'Motion', 'Secure', and 'Motion Password'. Each attacker was tasked with mimicking each word three times using their right hand. While we acknowledge that the small sample size limits the generalizability of our findings, we believe the results are still valuable and can provide first insights into the robustness of Motion Passwords for motion-based verification.

3.2 Ball-Throwing Dataset from Miller et al.

We used the Ball-Throwing (BaT) dataset from Miller et al. [23] to compare the verification reliability of Motion Passwords with very specific motion patterns. This dataset comprises motion data from 41 users, recorded over two sessions using three different VR systems, the HTC Vive, the Oculus Quest, and the HTC Vive Cosmos. For our analyses, we select the signatures recorded with the Oculus Quest to match the device of our Motion Password study. Each user performed a ball-throwing action 10 times per session and device, with each session taking place on a separate day. The task involved throwing a virtual ball at a target, with consistent physical characteristics and locations of the ball, target, and pedestal across sessions. Data was recorded at 90 frames per second and each throwing sequence was cut to be exactly three seconds long.

3.3 Who Is Alyx?

We use the "Who Is Alyx?" (WiA) [30] dataset to pretrain the similarity-learning model, following Rack et al [32]. The dataset contains 71 users who play the VR game "Half-Life: Alyx" over two sessions for about 45 minutes per user and session with a HTC Vive Pro. The game introduces a wide array of different user motions, ranging from calm and subtle motions when users try to solve puzzles or explore the virtual world, to extensive and even hectic motions when users get startled by enemies and have to fight their way out. User motions were recorded with 90 frames per second.

We selected this dataset since it can be used to train versatile similarity-learning models that become able to identify new users from different datasets [32]. The high variety of user motions in "Who Is Alyx?" is the foundation for models to accurately learn user-specific motion signatures amidst a wide array of potential actions. Moreover, since the dataset includes two sessions from different days for each user, models can learn to separate the variance that is specific to individual sessions from the identifying signal.

3.4 Dataset Alignment

When working with motion data from more than one dataset, it is important to align all recordings to use the same coordinate system, representation of rotations, time encoding, and units of measurement [33]. While all datasets use quaternions, they use different coordinate systems and units of measurement, as WiA had been recorded with Steam OpenVR and MoP and BaT both with Unity. Hence, we pay attention to ensuring the same format by converting recordings from the MoP Dataset to use the Steam OpenVR coordinate system (X: right, Y: up, Z: forward) and 'centimeters' for positions. After these preprocessing steps, we inspected recordings from both datasets with the visualization tool from Rack et al. [33] to visually verify the correct alignment of all datasets.

4 Methodology

In this section, we describe our methodology for motion-based verification, focusing on specific signatures from the BaT dataset and Motion Passwords from the MoP dataset.

4.1 Input Data

The input data to our verification models is a motion sequence representing the user's entered signature, as visualized in Figure 2. This sequence consists of frames containing 3D coordinates for positions (x, y, z), as well as quaternions (x, y, z, w) for each peripheral (HMD, left & right controller). Since the original framerate can vary, we resample every sequence to a constant framerate of 30 frames per second (fps). This framerate balances computational costs and fidelity, retaining sufficient information without excessive data size [31]. For Motion Passwords, the final sequence length varies according to the duration of the user input. The ball-throwing signatures from the BaT dataset are all exactly 90 frames (i.e., 3 seconds) long.

Next, we convert the resampled motion data with the Body-Relative Acceleration (BRA) encoding from Rack et al. [30, 31]. This encoding removes irrelevant information (e.g., user's position or orientation within the scene), preventing overfitting by ensuring models focus on actual identifying signals. First, we transform the motion sequences into the body-relative (BR) encoding, making each frame's positions and rotations relative to the HMD's local coordinate system. This step also removes the HMD's position, as it is always the origin (0,0,0) in its local coordinate system. Then, we compute the second derivative between the frames, producing the positional and angular accelerations based on the BR data. After these steps, the preprocessed input sequence consists of 18 features per frame: (pos-x, pos-y, pos-z, rot-x, rot-y, rot-z, rot-w) for each controller (left and right) and (rot-x, rot-y, rot-z, rot-w) for the HMD.

4.2 Feature-Distance Model

For the feature-distance approach, we utilize Dynamic Time Warping (DTW) [36]. DTW measures the similarity between two temporal sequences that may vary in speed or timing. The algorithm aligns sequences in the time dimension using a dynamic programming approach that minimizes the cumulative distance between them.

DTW calculates an optimal match by adjusting the time indices of the points in one sequence to align with the corresponding points in another sequence. This involves constructing a distance matrix where each element represents the distance between points in the two sequences. From this matrix, DTW determines the shortest path that best aligns the sequences, corresponding to the minimum cumulative distance, which provides the degree of similarity.

4.3 Similarity-Learning Model

Our similarity-learning model employs Deep Metric Learning [24], which learns a function that maps input data to an embedding space where distances reflect semantic similarities between samples. The model is trained to reduce the distance between embeddings of samples from the same class (i.e., the same user) while increasing the distance between embeddings from different classes. This approach facilitates effective measurement of similarity directly from the learned embeddings. We followed the methodology from Rack et al. [32] but implemented an updated architecture.

4.3.1 Architecture. Our architecture processes input sequences through a Gated Recurring Unit (GRU) layer first, followed by a transformer encoding unit before the final output layer. This architecture yielded superior results in our preliminary experiments on the WiA dataset compared to the original architecture from Rack et al. [32], who used a single GRU unit with several layers. Our architecture exposes several hyperparameters and given the variability of optimal configurations for individual use cases, we implemented a hyperparameter search (see Section 4.3.3). Detailed documentation of the model architecture and the exposed hyperparameters can be found in the code repository.

4.3.2 Pretraining. We pretrained the similarity-learning model on the WiA dataset. Notably, the WiA and our MoP datasets do not share any users. Our training procedure generally aligns with the methods outlined by Rack et al. [32], reserving 11 users for validation purposes and the remainder for training. Throughout training, we monitored the 'R Precision', measuring retrieval accuracy by quantifying the rate of relevant items retrieved within the top positions of the ranking. Training checkpoints were saved upon achieving new high scores in R Precision. The process was terminated when no improvement was observed for several consecutive epochs.

4.3.3 Hyperparameter Search. Our similarity-learning model's architecture involves several hyperparameters. Given the unpredictability of optimal configurations, a hyperparameter search was conducted. This search employed the 'sweeps' function of the machine learning monitoring service "Weights and Biases", using the Bayesian search strategy. We iteratively refined our search parameters, expanding or narrowing the search space based on intermediate results on the WiA validation set, and initiated new sweeps accordingly. Ultimately, we performed 2,750 individual trainings and selected the model configuration yielding the highest R Precision on the validation set. The investigated search space, the final hyperparameter configuration, and the trained model can be found in the code repository.

4.4 Model Implementations

All code was implemented in Python and can be found in the accompanying repository. For the similarity-learning model, we used PyTorch Lightning [6] together with the PyTorch Metric Learning library [25]. For the feature-distance model, we used the 'tslearn' package [39].

4.5 User Verification

Our verification system comprises two primary stages: registration and verification. In the registration stage, users submit their *reference* signature, for the system to store as template for their identity. During verification, a user claims their supposed identity, and the system assesses the authenticity of this claim. This involves the user providing their *query* signature, which the system then compares against the stored reference template of the claimed identity. The query is exclusively compared to the reference of the claimed identity, ignoring any other references in the database – otherwise, this would be considered identification, not verification [8].

The feature-distance and the similarity-learning method both yield a similarity score between any two given items. For final verification, this score is compared against a predefined threshold to determine acceptance or rejection. Using DTW, the feature-distance approach directly compares the query's temporal sequence with the reference and immediately generates a distance value that quantifies the dissimilarity between the two sequences.

Unlike the feature-distance approach, the similarity-learning method initially processes references and queries independently and requires a second step to retrieve a similarity score. During registration, the user provides one or more reference signatures, each of which gets encoded as an embedding. For instance, in our MoP dataset, each user provided five iterations per session of the same word with the same hand, resulting in five reference embeddings, and five query embeddings. To create a single, representative reference embedding, we compute the kernel embedding, which is positioned at the center of all provided reference embeddings. This accounts for variations in the user's reference signatures, aiming to capture a more robust and representative motion profile. Then, in the verification phase, the retrieved query embedding is compared to the reference embedding. This comparison yields a similarity score, which is evaluated against a predefined threshold to determine whether the user's identity is verified. Since our training approach uses the ArcFace loss, we use the cosine similarity between the two embeddings as the similarity score.

5 Experimental Setup

In our experimental setup, we first compared feature-distance (FD) and similarity-learning (SL) models both in combination with signatures from the MoP and the BaT dataset, so we ended up with four conditions: FD+BaT, SL+BaT, FD+MoP, and SL+MoP. Subsequently, we analyzed the potential of Motion Passwords using the SL+MoP condition. This section explains the setup and how we evaluate the verification performances.

5.1 Verification Scenario

We simulated the following scenario: users register on their first day using our hypothetical XR application by providing their signature, either a ball throw or a Motion Password. After a few days, users attempt to log in and get verified by the system, either as themselves (genuine) or as another user (impostor). For genuine cases, reference and query signatures are from the same user. For impostor cases, they are from different users. The system should ideally accept genuine attempts and reject impostor attempts.

5.2 Performance Measures

To measure the performance of our verification system, we are interested in the ratio of genuine verification successes (i.e., True Acceptance Rate (TAR)) and the ratio of impostor verification successes (i.e., False Acceptance Rate (FAR)). For example, a TAR of 95% and a FAR of 1% indicates that the system succeeds in accepting 95 in 100 genuine attempts, and fails to reject 1 in 100 impostor attempts.

Adjusting the similarity threshold facilitates a trade-off: a stricter system enhances security by requiring higher similarity, resulting in lower TAR and FAR, whereas a more lenient system increases both TAR and FAR, reducing security. To analyze and compare this trade-off we employ the 'Receiver Operating Characteristic' (ROC) curve. The ROC curve plots the TAR against the FAR at various threshold settings, providing a visualization of the trade-offs between sensitivity and specificity.

Additionally, we evaluate overall verification accuracy using the corresponding 'Area Under the Curve' (AUC). The AUC provides a single scalar value that summarizes the performance across all possible threshold settings. A higher AUC value indicates a better overall ability of the system to distinguish between genuine and impostor attempts, irrespective of any specific threshold. An AUC of 1.0 represents a perfect system that completely separates genuine and impostor samples, while an AUC of 0.5 suggests a performance no better than random guessing.

Another metric often used in the verification context is the Equal Error Rate (EER), which is the point at which the FAR and the False Rejection Rate are equal. Like the AUC, EER provides a single value that summarizes the overall accuracy of the system. A lower EER indicates better performance.

5.3 Verification Thresholds

The choice of a verification threshold depends on the individual use case. To demonstrate system performance across different levels of

verification strictness, we determined four exemplary thresholds that yield the following FARs in our initial analysis in Section 6.1: we selected a *strict* threshold at FAR = 0.1%, a *moderate* threshold at FAR = 1%, a *lenient* threshold at FAR = 10%, and a *permissive* threshold at FAR = 25%. We used these threshold values in subsequent analyses to compare how different conditions, such as attacks, impact FAR and TAR when either threshold is applied for verification. Overall, these four thresholds are intended to demonstrate the verification system's performance across different levels of strictness.

6 Results

6.1 Genuine Attempts & Uninformed Attacks

In this section, we evaluate the verification accuracy using genuine attempts and uninformed attacks across four conditions: FD+BaT, SL+BaT, FD+MoP, and SL+MoP. The primary goal is to determine the efficacy of our Motion Passwords in comparison to traditional ball-throwing signatures under different verification models.

For Motion Passwords, we included the three words each participant wrote in both sessions (i.e., 'Motion' and two random words). For genuine attempts, we paired the same words from the same user as reference and query. For impostor attempts, we paired different words from different users, representing an uninformed attack scenario where the attacker does not know the correct word or the victim's writing style.

The ROC curves for these conditions are shown in Figure 3. The SL model outperformed the FD model for both signature types. For the BaT dataset, the SL model achieved an AUC of 0.941, slightly higher than the FD model's AUC of 0.931. However, the performance gap widened for Motion Passwords, with the SL model achieving an AUC of 0.93 compared to the FD model's 0.75. This indicates that the DTW algorithm struggles to find similarities in Motion Passwords when they originate from the same user. The scored EERs are 12.4% for SL+BaT, 14.5% for SL+MoP, 14.6% for FD+BaT, and 32.3% for FD+MoP.

Figure 3 also provides a detailed view at four FAR values selected for the exemplary thresholds discussed in 5.3. The TARs achieved by SL+MoP, SL+BaT, and FD+BaT lie within each other's confidence intervals, suggesting that differences between these three conditions are insignificant at the selected thresholds. When the verification threshold is set to allow 1% of impostor attempts to succeed, approximately half of the genuine attempts are accepted, indicating that genuine users typically need two attempts for successful verification with the FD+BaT and SL+BaT/MoP approaches. The FD+MoP combination performs significantly worse, requiring a more lenient threshold that accepts around 25% of impostor attempts to achieve a similar TAR.

6.2 Correct vs. Incorrect Passwords

In this subsection, we examine the impact of using correct versus incorrect passwords on the verification performance. The goal is to understand how the similarity-learning model handles genuine attempts with incorrect passwords and attacks with correct passwords.

In our previous experiment, we paired the same words for genuine users and different words for attacks. In this experiment, we reversed the setup by pairing different words for genuine users

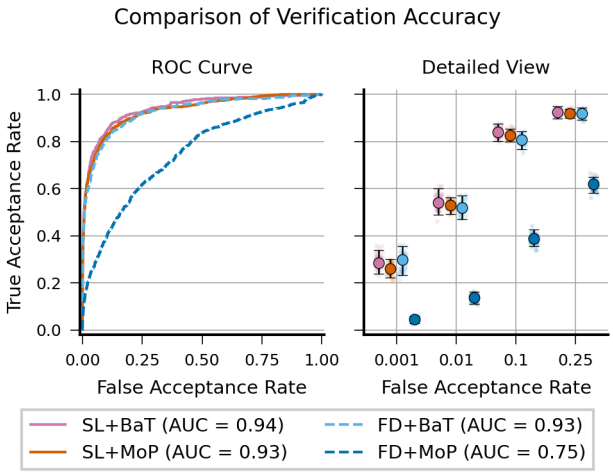

Figure 3: Trade-offs between TAR and FAR for the four combinations of model types (Similarity-Learning (SL), Feature-Distance (FD)) and signature types (Motion Passwords (MoP), Ball-Throws (BaT)). The right figure provides a detailed view of the ROC curve on the left for the FARs discussed in Section 5.3.

Table 1: Resulting TARs and cosine similarities of the SL+MoP condition based on the results in Section 6.1 for the four representative thresholds we selected in Section 5.3.

Threshold	FAR	TAR	Similarity
strict	0.1%	28%	0.903
moderate	1%	52%	0.850
lenient	10%	81%	0.746
permissive	25%	91%	0.660

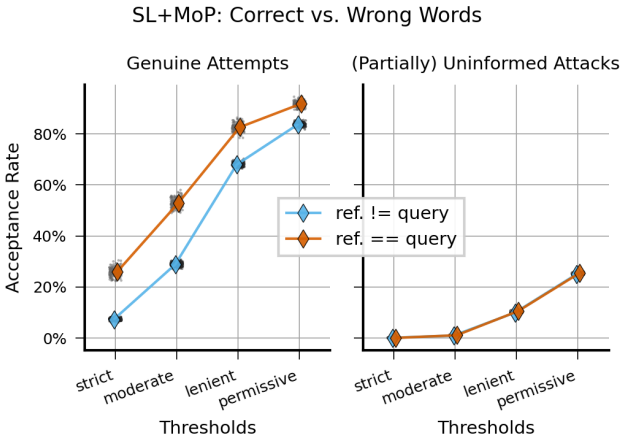

Figure 4: Comparison of TAR and FAR if the query word is correct or incorrect, i.e., equal or unequal to the reference.

and the same words for attacks. The results of this experiment are shown in Figure 4.

For genuine users, the TAR drops significantly when the reference and query words do not match. The TAR decreases by 14 to 24 percentage points for the strict, moderate, and lenient thresholds, indicating a substantial impact on verification accuracy. Despite these reductions, the model still detects genuine users to some extent, indicating that the model can still pick up on the users' motion profiles.

For impostors, the TAR does not change when they attack using the correct word. This result implies that merely knowing the correct word does not significantly increase the impostor's chances of success. In other words, attackers cannot improve their success rate solely by knowing the correct word without replicating the victim's writing style.

Additionally, we paired reference and query signatures of the same words written with different hands. In this scenario, the SL model's performance comes close to random guessing, with an AUC score near 0.5. This result indicates that the attackers must use the correct hand to have any chance of success.

6.3 Fully Informed Attack with Motion Passwords

With our preliminary attack study, we investigated the robustness of the similarity-learning model against fully informed attacks, where attackers are aware of both the correct password and the victim's writing style. Participants were tasked to replicate the Motion Passwords of the two victims after observing video recordings of the correct writing actions. The results of these fully informed attacks are depicted in Figure 5, which shows the ranking of each verification attempt by both genuine users and attackers for the three passwords: 'Motion', 'Secure', and 'Motion Password'.

Overall, the similarity-learning model achieved an AUC score of 0.95 across all three words, indicating a slight improvement in performance compared to scenarios with partially informed or completely uninformed attacks. A5 was the only attacker able to achieve higher similarity scores than victim V2 for two of the words and very similar scores across all attempts. In contrast, attacks on victim V1 were significantly less successful, with no attacker reaching the average similarity scores of V1.

Figure 5 also illustrates the different verification thresholds discussed in Section 5.3. The 'strict' and 'moderate' thresholds were breached only by attacker A5 when imitating V2: the 'strict' threshold was surpassed once with the word "Motion Password", and the 'moderate' threshold with the other two words. The AUC scores remained consistent across the three passwords, indicating that the length of the word did not significantly affect verification reliability.

7 Discussion

In this work, we explored the efficacy of Motion Passwords for user verification in XR environments, focusing on the motion-based layer of security. Our findings indicate that Motion Passwords, when combined with a similarity-learning model, can achieve a level of verification reliability on par with previous methods using specific motion signatures. The similarity-learning model demonstrated robust performance by effectively distinguishing genuine users from impostors, leveraging the unique motion profiles of users. This performance was consistent across various attack scenarios, highlighting the potential of Motion Passwords as a viable biometric verification method.

We observed that the similarity-learning model can effectively handle the complexity of Motion Passwords. The model focuses on the user's underlying motion profile rather than the specific trajectory of the motion. This capability was demonstrated by the model's high verification success rate for genuine users, even when paired with a wrong word. This finding suggests that Motion Passwords can encapsulate unique motion profiles, making them a viable biometric signature for motion-based user verification.

One significant advantage of Motion Passwords seems to be their potential resistance against shoulder-surfing attacks as demonstrated by our preliminary fully informed attack study. The overall achieved similarity scores of these attacks were predominantly below the scores of genuine attempts and the AUC was comparable to the AUC of the uninformed impostor scenarios. Particularly, the 'strict' and 'moderate' thresholds were just breached by one attacker. While these findings are promising, they are based on a limited number of participants and should be interpreted with caution.

The comparison of Motion Passwords with specific ball-throwing signatures revealed that the latter could be effectively verified using both feature-distance and similarity-learning models. However, the feature-distance model failed with the more complex Motion Passwords. This failure underscores the necessity of more advanced techniques, such as similarity-learning models, to handle complex and individualized motion patterns.

8 Future Work

We believe that our current Motion Password dataset represents a conservative estimate of the potential similarity of genuine verification attempts between reference and query signatures. In trials with our proof-of-concept application, we observed that the consistency of signatures improved with practice. This observation suggests that with increased experience and muscle memory, users will be able to produce more consistent Motion Passwords over time. As users develop more consistent signatures, the similarity between registration and verification signatures for genuine users is expected to increase. If future work can confirm this learning effect, it would allow for higher verification thresholds, thereby reducing the likelihood of impostor success. To improve the learning process of users, implementing feedback mechanisms during the registration phase providing measures of similarity between entered and stored signatures could be beneficial. This allows users to refine their Motion Passwords and could lead to more consistent and distinctive motion profiles, potentially improving verification accuracy.

Technical advancements also offer avenues for future research. While our study focused on the fundamental evaluation of Motion Passwords, exploring optimal deep-learning architectures and more sophisticated input sequence encodings could yield better results. Additionally, training methodologies that include fine-tuning with specialized datasets might further enhance the model's ability to identify unique motion patterns.

Future work should also investigate further potential failure modes to enhance the robustness and reliability of Motion Passwords. This includes examining other attack vectors, system errors

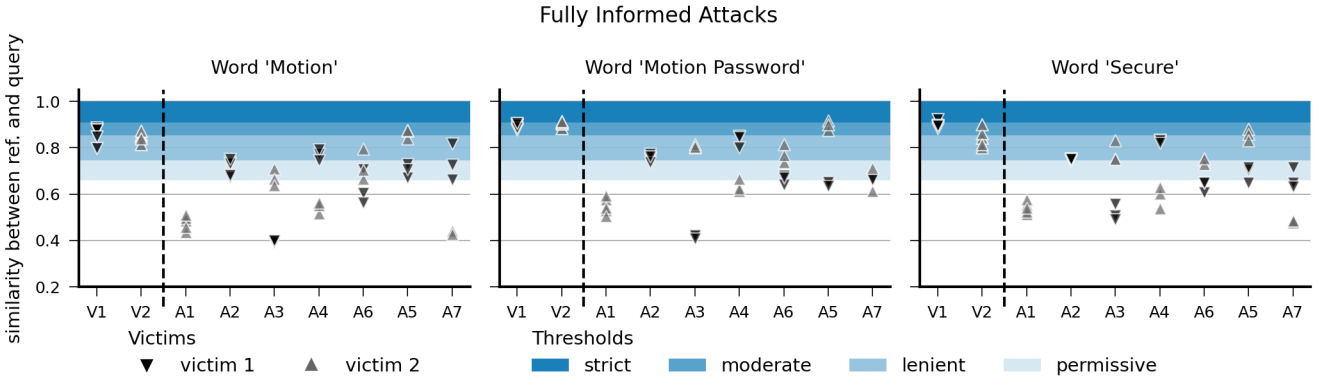

Figure 5: Similarity scores between reference and query for fully informed attacks on Motion Passwords. The three subplots represent each of the three passwords we included in the study. Victims (V1, V2) are shown on the left, followed by attackers (A1-A7). Note that V1 and V2 are also A1 and A2, which is why they only have scores shown for the other victim. Thresholds are indicated by shaded areas. Higher similarity scores indicate more successful attempts.

such as tracking failures or hardware malfunctions, and user errors, such as inconsistent motion due to fatigue or stress. Mitigation strategies could involve incorporating error-correction algorithms and designing fallback methods, like traditional keyboard-based passwords.

Lastly, our motion-based verification model represents only one layer of security for Motion Passwords. Incorporating a knowledge-based approach with an Optical Character Recognition (OCR) model allows comparison of the actual words between reference and query signatures, thereby preventing impostor success when the correct password is not known. OCR is a well-established field with mature techniques and readily available solutions for 2D handwriting such as Tesseract [37], OCRopus [2], or Kraken [13]. The challenge of applying OCR in this context lie in adapting these solutions from 2D to 3D space. Additionally, training a model to specifically recognize handwriting styles can complement the motion-based approach. This might seem redundant to the motion-based verification used in this work because this approach also analyzes how a word has been written. However, analyzing false positives from our SL+MoP condition revealed that many confused signatures do not actually look the same if visualized side by side. Therefore, incorporating techniques that focus on writing style and stroke order should significantly reduce the FAR. Recent studies have successfully used machine learning techniques for handwriting verification [3, 9], indicating the feasibility of this approach for Motion Passwords. Similar to OCR, the challenge here is to extend these 2D solutions to 3D space. By combining these three methods, an attacker would need to replicate the victim's motion profile, know the correct password, and accurately mimic their handwriting style, including the order of strokes and fine details, to succeed. Future work should investigate integrating these multi-layered verification methods to fully exploit the potential of Motion Passwords in 3D space.

9 Limitations

Despite these promising results, our work has several limitations that need addressing in future research.

Firstly, the small scale of our attack study limits the generalizability of our findings regarding security against shoulder-surfing. Future studies should include a larger number of victims and attackers to obtain more robust insights.

Secondly, the narrow set of ethnicities represented in our datasets may not capture the full diversity of motion profiles across different cultures and populations. Including a more diverse participant pool will help to understand how cultural differences may affect motion-based verification.

Finally, our study primarily focused on technical feasibility without extensive consideration of user experience factors, such as ease of use and user acceptance, which are crucial for practical deployment. Future research should explore these aspects to ensure broader applicability and effectiveness of the proposed verification system.

10 Unity Prototype

To demonstrate the feasibility of motion-based verification in XR environments, we developed a prototype Unity application. This prototype integrates our similarity-learning model, as used in the analyses presented in this paper. To our knowledge, this represents the first published application incorporating motion-based recognition in an XR setting.

The prototype supports two primary modes. In registration mode, new users can be created, and they provide multiple repetitions of the same signature to build a robust profile for future verification attempts. In verification mode, a user can claim a registered identity and make verification attempts. The system displays the achieved similarity score and, based on the selected threshold, either confirms or rejects the attempt.

The application comprises two main components: a Python server and the Unity application. The Python server manages user data and handles the registration and verification of motion sequences through an HTTP API. It executes the similarity-learning model and processes the motion data accordingly. The Unity scene provides the VR environment and communicates with the Python

server. Detailed setup and usage instructions are available in the accompanying Readme file in the code repository.

11 Conclusion

In this study, we have demonstrated the viability of using Motion Passwords for user verification in XR environments. Our findings show that motion-based verification works effectively with Motion Passwords. The similarity-learning model, in particular, has proven to be a reliable method for distinguishing genuine users from impostors, even under various attack scenarios.

However, while the motion-based layer has shown promising results, there remains significant potential for enhancing verification reliability by integrating additional layers of security, like OCR or handwriting style recognition techniques. These enhancements can further establish Motion Passwords as a robust and versatile verification method, combining ease of use with additional security.

Acknowledgments

This research was supported by the Bavarian State Ministry For Digital Affairs in the project 'XR Hub' (Grant A5-3822-2-16) and by the German Federal Ministry of Labour and Social Affairs (DKI.00.00030.21).

References

[1] Ashwin Ajit, Natasha Kholgade Banerjee, and Sean Banerjee. 2019. 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)*. IEEE, 9–16. https://doi.org/10.1109/AIVR46125.2019.00012

[2] Thomas M. Breuel. 2008. The OCRopus Open Source OCR System. In *Document Recognition and Retrieval XV*, Berrin A. Yanikoglu and Kathrin Berkner (Eds.), Vol. 6815. SPIE / International Society for Optics and Photonics, 68150F. https://doi.org/10.1117/12.783598

[3] Moises Diaz, Miguel A. Ferrer, Donato Impedovo, Muhammad Imran Malik, Giuseppe Pirlo, and Réjean Plamondon. 2019. A Perspective Analysis of Handwritten Signature Technology. *Acm Computing Surveys* 51, 6, Article 117 (Jan. 2019). https://doi.org/10.1145/3274658

[4] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2019. Text Entry in Virtual Reality: A Comprehensive Review of the Literature. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 11567 LNCS. Springer International Publishing, Cham, 419–437. https://doi.org/10.1007/978-3-030-22643-5_33/TABLES/5

[5] M.C. Fairhurst and E. Kaplani. 2003. Perceptual Analysis of Handwritten Signatures for Biometric Authentication. *IEE Proceedings - Vision, Image, and Signal Processing* 150, 6 (2003), 389. https://doi.org/10.1049/ip-vis:20031046

[6] William Falcon, Jirka Borovec, Adrian Wälchli, Nic Eggert, Justus Schock, Jeremy Jordan, Nicki Skafte, Ir1dXD, Vadim Bereznyuk, Ethan Harris, Tullie Murrell, Peter Yu, Sebastian Præsius, Travis Addair, Jacob Zhong, Dmitry Lipin, So Uchida, Shreyas Bapat, Hendrik Schröter, Boris Dayma, Alexey Karnachev, Akshay Kulkarni, Shunta Komatsu, Martin.B, Jean-Baptiste SCHIRATTI, Hadrien Mary, Donal Byrne, Cristobal Eyzaguirre, Cinjon, and Anton Bakhtin. 2020. PyTorchLightning/Pytorch-Lightning: 0.7.6 Release. Zenodo. https://doi.org/10.5281/ZENODO.3828935

[7] Anil K. Jain, Arun A. Ross, and Karthik Nandakumar. 2011. *Introduction to Biometrics*. Springer US. https://doi.org/10.1007/978-0-387-77326-1

[8] Eakta Jain, Lisa Anthony, Aishat Aloba, Amanda Castonguay, Isabella Cuba, Alex Shaw, and Julia Woodward. 2016. Is the Motion of a Child Perceivably Different from the Motion of an Adult? *ACM Transactions on Applied Perception* 13, 4 (July 2016), 1–17. https://doi.org/10.1145/2947616

[9] Jiajia Jiang, Songxuan Lai, Lianwen Jin, and Yecheng Zhu. 2022. DsDTW: Local Representation Learning with Deep Soft-DTW for Dynamic Signature Verification. *IEEE Transactions on Information Forensics and Security* 17 (2022), 2198–2212. https://doi.org/10.1109/TIFS.2022.3180219

[10] Florian Kern, Peter Kullmann, Elisabeth Ganal, Kristof Korwisi, René Stingl, Florian Niebling, and Marc Erich Latoschik. 2021. Off-The-Shelf Stylus: Using XR Devices for Handwriting and Sketching on Physically Aligned Virtual Surfaces. *Frontiers in Virtual Reality* 2 (June 2021). https://doi.org/10.3389/frvir.2021.684498

[11] Florian Kern, Jonathan Tschanter, and Marc Erich Latoschik. 2024. Handwriting for Text Input and the Impact of XR Displays, Surface Alignments, and Sentence Complexities. *IEEE Transactions on Visualization and Computer Graphics* 30, 5 (2024), 2357–2367. https://doi.org/10.1109/TVCG.2024.3372124

[12] Salman H. Khan, Zeashan Khan, and Faisal Shafait. 2013. Can Signature Biometrics Address Both Identification and Verification Problems?. In *2013 12th International Conference on Document Analysis and Recognition*. 981–985. https://doi.org/10.1109/ICDAR.2013.198

[13] Benjamin Kiessling. 2019. Kraken - a Universal Text Recognizer for the Humanities. https://doi.org/10.34894/Z9G2EX

[14] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. 2018. Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Conference on Human Factors in Computing Systems - Proceedings*, Vol. 2018-April. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3173574.3173919

[15] Alexander Kupin, Benjamin Moeller, Yijun Jiang, Natasha Kholgade Banerjee, and Sean Banerjee. 2019. Task-Driven Biometric Authentication of Users in Virtual Reality (VR) Environments. In *MultiMedia Modeling*, Ioannis Kompatsiaris, Benoit Huet, Vasileios Mezaris, Cathal Gurrin, Wen-Huang Cheng, and Stefanos Vrochidis (Eds.). Springer International Publishing, Cham, 55–67.

[16] Sugang Li, Ashwin Ashok, Yanyong Zhang, Chenren Xu, Janne Lindqvist, and Macro Gruteser. 2016. Whose Move Is It Anyway? Authenticating Smart Wearable Devices Using Unique Head Movement Patterns. *2016 IEEE International Conference on Pervasive Computing and Communications, PerCom 2016* (2016), 1–9. https://doi.org/10.1109/PERCOM.2016.7456514

[17] Jonathan Liebers, Mark Abdelaziz, and Lukas Mecke. 2021. Understanding User Identification in Virtual Reality through Behavioral Biometrics and the Effect of Body Normalization. In *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3411764.3445528

[18] Duo Lu, Yuli Deng, and Dijiang Huang. 2021. Global Feature Analysis and Comparative Evaluation of Freestyle In-Air-Handwriting Passcode for User Authentication. In *Annual Computer Security Applications Conference*. ACM, Virtual Event USA, 468–481. https://doi.org/10.1145/3485832.3485906

[19] Mark Roman Miller, Fernanda Herrera, Hanseul Jun, James A. Landay, and Jeremy N. Bailenson. 2020. Personal Identifiability of User Tracking Data during Observation of 360-Degree VR Video. *Scientific Reports* 10, 1 (2020), 1–10. https://doi.org/10.1038/s41598-020-74486-y

[20] Robert Miller, Ashwin Ajit, Natasha Kholgade Banerjee, and Sean Banerjee. 2019. Realtime Behavior-Based Continual Authentication of Users in Virtual Reality Environments. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 253–2531. https://doi.org/10.1109/AIVR46125.2019.00058

[21] Robert Miller, Natasha Kholgade Banerjee, and Sean Banerjee. 2020. Within-System and Cross-System Behavior-Based Biometric Authentication in Virtual Reality. *Proceedings - 2020 IEEE Conference on Virtual Reality and 3D User Interfaces, VRW 2020* (2020), 311–316. https://doi.org/10.1109/VRW50115.2020.00070

[22] Robert Miller, Natasha Kholgade Banerjee, and Sean Banerjee. 2021. Using Siamese Neural Networks to Perform Cross-System Behavioral Authentication in Virtual Reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 140–149. https://doi.org/10.1109/VR50410.2021.00035

[23] Robert Miller, Natasha Kholgade Banerjee, and Sean Banerjee. 2022. Combining Real-World Constraints on User Behavior with Deep Neural Networks for Virtual Reality (VR) Biometrics. In *Proceedings - 2022 IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2022*. Institute of Electrical and Electronics Engineers Inc., 409–418. https://doi.org/10.1109/VR51125.2022.00060

[24] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. A Metric Learning Reality Check. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 681–699.

[25] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. PyTorch Metric Learning. arXiv:2008.09164 [cs]

[26] Vivek Nair, Wenbo Guo, Justus Mattern, Rui Wang, James F. O'Brien, Louis Rosenberg, and Dawn Song. 2023. Unique Identification of 50,000+ Virtual Reality Users from Head & Hand Motion Data. (Feb. 2023).

[27] Vivek Nair, Louis Rosenberg, James F. O'Brien, and Dawn Song. 2023. Truth in Motion: The Unprecedented Risks and Opportunities of Extended Reality Motion Data. (2023). https://doi.org/10.48550/ARXIV.2306.06459

[28] Ken Pfeuffer, Matthias J. Geiger, Sarah Prange, Lukas Mecke, Daniel Buschek, and Florian Alt. 2019. Behavioural Biometrics in VR: Identifying People from Body Motion and Relations in Virtual Reality. In *2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, Vol. 12. Association for Computing Machinery, New York, NY, USA, 1–12.

[29] S. Prabhakar, S. Pankanti, and A.K. Jain. 2003. Biometric Recognition: Security and Privacy Concerns. *IEEE Security & Privacy* 1, 2 (2003), 33–42. https://doi.org/10.1109/MSECP.2003.1193209

[30] Christian Rack, Tamara Fernando, Murat Yalcin, Andreas Hotho, and Marc Erich Latoschik. 2023. Who Is Alyx? A New Behavioral Biometric Dataset for User

Identification in XR. *Frontiers in Virtual Reality* 4 (2023). https://doi.org/10.3389/frvir.2023.1272234

[31] Christian Rack, Andreas Hotho, and Marc Erich Latoschik. 2022. Comparison of Data Encodings and Machine Learning Architectures for User Identification on Arbitrary Motion Sequences. In *2022 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2022*. IEEE.

[32] Christian Rack, Konstantin Kobs, Tamara Fernando, Andreas Hotho, and Marc Erich Latoschik. 2023. Versatile User Identification in Extended Reality Using Pretrained Similarity-Learning. arXiv:2302.07517 [cs]

[33] Christian Rack, Vivek Nair, Lukas Schach, Felix Foschum, Marcel Roth, and Marc Erich Latoschik. 2024. Navigating the Kinematic Maze: Analyzing, Standardizing and Unifying XR Motion Datasets. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE.

[34] Cynthia E. Rogers, Alexander W. Witt, Alexander D. Solomon, and Krishna K. Venkatasubramanian. 2015. An Approach for User Identification for Head-Mounted Displays. *ISWC 2015 - Proceedings of the 2015 ACM International Symposium on Wearable Computers* (2015), 143–146. https://doi.org/10.1145/2802083.2808391

[35] Osvaldo A. Rosso, Raydonal Ospina, and Alejandro C. Frery. 2016. Classification and Verification of Handwritten Signatures with Time Causal Information Theory Quantifiers. *PLOS ONE* 11, 12 (Dec. 2016), 1–19. https://doi.org/10.1371/journal.pone.0166868

[36] H. Sakoe and S. Chiba. 1978. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1 (1978), 43–49. https://doi.org/10.1109/TASSP.1978.1163055

[37] Ray Smith. 2007. An Overview of the Tesseract OCR Engine. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*. IEEE Computer Society, Washington, DC, USA, 629–633.

[38] Sophie Stephenson, Bijeeta Pal, Stephen Fan, Earlence Fernandes, Yuhang Zhao, and Rahul Chatterjee. 2022. SoK: Authentication in Augmented and Virtual Reality. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 267–284. https://doi.org/10.1109/SP46214.2022.9833742

[39] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. 2020. Tslearn, a Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research* 21, 118 (2020), 1–6.