Combining Real-World Constraints on User Behavior with Deep Neural Networks for Virtual Reality (VR) Biometrics

Robert Miller* Natasha Kholgade Banerjee† Sean Banerjee‡

Clarkson University, Potsdam, NY

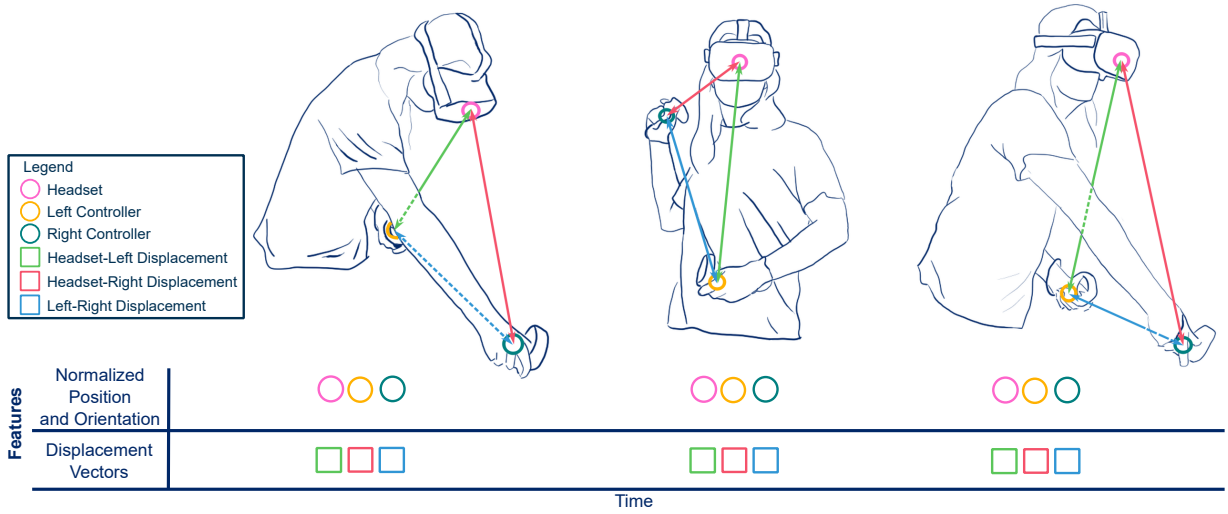

Figure 1: We provide user identification and authentication using behavioral biometrics in virtual reality by augmenting orientation and normalized position features, expressed within the local coordinate systems of the hand controllers and headset devices, with inter-device displacement vectors. We demonstrate that using inter-device displacement vectors provides maximum success rate more often than baseline methods.

Abstract

Deep networks have demonstrated enormous potential for identification and authentication using behavioral biometrics in virtual reality (VR). However, existing VR behavioral biometrics datasets have small sample sizes which can make it challenging for deep networks to automatically learn features that characterize real-world user behavior and that may enable high success, e.g., high-level spatial relationships between headset and hand controller devices and underlying smoothness of trajectories despite noise. We provide an approach to perform behavioral biometrics using deep networks while incorporating spatial and smoothing constraints on input data to represent real-world behavior. We represent the input data to neural networks as a combination of scale- and translation-invariant device-centric position and orientation features, and displacement vectors representing spatial relationships between device pairs. We assess identification and authentication by including spatial relationships and by performing Gaussian smoothing of the position features. We evaluate our approach against baseline methods that use the raw data directly and that perform a global normalization of the data. By using displacement vectors, our work shows higher success over baseline methods in 36 out of 42 cases of analysis done by varying user sets and pairings of VR systems and sessions.

*e-mail: romille@clarkson.edu
†e-mail:nbanerje@clarkson.edu
‡e-mail:sbanerje@clarkson.edu

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Security and privacy—Security services—Authentication—Biometrics

1 Introduction

With the future potential for virtual reality (VR) in consumer domains with critical data such as surgical training [30, 33], remote teleoperation and driving [29, 34, 44, 45, 52], healthcare [7, 10, 12, 32, 35, 46, 53], education [9, 26, 28, 51, 54, 55], retail [50, 57], and personal banking [8, 56], a large number of approaches have emerged to investigate security provision in VR applications so as to avoid compromise by malicious users. Some approaches incorporate traditional credentials in the VR environment, such as password entries via screens displayed on 2D planes or unique 3D arrangements of objects [14, 18–21, 48, 58]. Once traditional credentials are acquired by a malicious user, the application is rendered insecure. Additionally, applications that depend solely on traditional credentials are difficult to embed into a continuous authentication approach without compromising system usability, as every time a credential-entry screen shows up, the user needs to stop their interaction. Stopping of interactions can be detrimental during activities such as test-taking or completion of physical therapy routines in VR, and hazardous in cases such as VR teleoperation of drones or vehicles. Traditional biometrics, such as iris, have been explored as a means of VR authentication [4–6], however iris cameras are not available in current consumer VR devices on the market. To overcome the limitations of passwords in ensuring continuous security, a growing body of work has emerged on using user behavior in VR as a biometric, where the motion of hand controllers and headsets is tracked as users perform VR interactions [1, 27, 31, 36–43, 47, 49]. Recent approaches use

deep neural networks [31, 36, 42] given their success at learning user behavior from input data with minimal pre-processing.

Existing VR datasets are small, since unlike work in keystroke and gesture-based biometrics, where data can be collected at scale using web-based keyloggers or downloadable smartphone applications [11, 13], VR systems still remain within the hands of a few niche users. Release of VR applications to collect data at large relies on the availability of permissions from application hosting platforms to acquire user information. So far all studies on VR biometrics have involved lab collections using in-house devices, with most datasets having 41 users or below, and only one having 511 users [39]. The small sample size of current datasets compromises the ability of deep neural networks to leverage raw data directly for learning real-world information that may contribute to improved success, e.g., high-level spatial relationships or underlying smoothness. In this work, we contribute two forms of input data pre-processing that incorporate real-world constraints on user behavior into learning algorithms for VR behavior-based identification and authentication using deep networks.

- We model spatial relationships between pairs of devices comprising a VR system, i.e., between the two controllers, and between each controller and the headset. Physical characteristics of users, e.g., height, weight, dexterity, and body part measurements, as well as approaches to task performance, e.g., arm hyperextension versus elbow bend for throwing, pointing, or resting, may induce user-specific spatial relationships between the headset and hand controllers. To incorporate spatial relationships while leveraging the strengths of normalized data in training neural networks [25], we augment normalized device-centric input trajectory positions from each device with displacement vectors from device pairings.

- Real-world motions exhibited by users are smooth, with sudden sharp changes being rare and largely intentional, e.g., the cusp induced in a golf swing trajectory due to a pause at the top of the swing. Errors in tracking mechanisms employed by current VR systems, using outward facing cameras or lighthouses, may cause recorded data to be non-smooth in regions where the motion is too fast or the device moves out of the view of the tracker. Corruption from tracking noise may cause trajectories across trials or VR systems to appear different, and may reduce identification and authentication success. To incorporate smooth motions, we filter input trajectory positions using Gaussian kernels.

While the concept of using spatial relationships was explored in the work of Pfeuffer et al. [49], their work uses random forests and provides low recognition accuracies, with a maximum of 63.55%. By including inter-device displacement vectors as neural network inputs, our is the first to include explicit spatial relationships in a deep learning approach to VR biometrics without requiring pre-interaction phases [31]. We evaluate our approach of combining device-centric trajectory representations with inter-device spatial relationships against baseline models that directly operate on the raw data as in Mathis et al. [36], that perform per-device normalization as in Miller et al. [42], and that perform a global normalization of the data without treating per-device data independently. Our work demonstrates higher identification accuracy over baseline models in 36 out of 42 cases of testing using Siamese networks and N-class networks over various system and session pairings from the 41-user and 3-system dataset of Miller et al. [41, 42]. We have made all code and data for our work public at `https://git.io/J9GIW`.

2 Related Work

A number of approaches address VR security by translating the concept of passwords to work in virtual environments using 2D patterns [20, 48, 58], unique arrangements of 3D virtual objects through

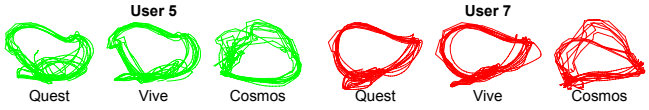

Figure 2: Right hand controller trajectories for users 5 and 7 performing ball-throwing using an Oculus Quest, HTC Vive, and HTC Vive Cosmos. The external lighthouse tracking of the HTC Vive provides cleaner trajectories, unlike the Oculus Quest and HTC Vive Cosmos which track using onboard cameras.

controller- or gaze-based selections [14, 18, 19], and sequences of actions [21] as inspired by analyses in desktop graphical environments [2, 3]. While several methods evaluate resistance to shoulder surfing [14, 18, 47], security is greatly diminished if the attacker gains direct access to the password combination by other mechanisms. As alternate biometrics for VR, the work of Boutros et al. [4–6] evaluates iris and periocular data by retraining pre-existing deep networks, DeepIrisNet [16], MobileNetV3 [23], ResNet [22], and DenseNet [24] using iris images in the OpenEDs dataset of Garbin et al. [17]. Their work relies on the availability of iris-facing cameras, which at the current juncture are lacking in off-the-shelf VR headsets. Additionally, since all images were acquired using a single camera [17], it is unclear how the work will translate across future potential iris cameras in VR systems upon deployment.

Recognizing the limitations of traditional credentials for VR authentication, a number of approaches have arisen to investigate using the behavior of a user in VR as a biometric signature. These approaches are summarized in Table 1. As shown in the table, approaches that use traditional learning techniques such as nearest neighbors and random forests with hand-crafted features are largely outperformed by methods that use deep learning. Miller et al. [39], provide the largest dataset consisting of 511 subjects viewing 5 360-degree videos and answering multiple choice questions using the HTC Vive. However, users exhibit movement during the study. As a result, the height of the user and distance from the VR content become the most discriminating feature in the dataset, reducing protection when a malicious attacker of the same physical dimensions and placement as the genuine user gains access.

Amongst approaches that use deep learning for VR behavior biometrics, Mathis et al. [36] do not report performing any data pre-processing apart from using sliding windows from the original data to provide classification of behavior trajectories. Miller et al. [42] perform a per-device normalization of the raw input data as per current deep network training guidelines that recommend layer input normalization for faster and more reliable convergence of network weights [25]. Within each headset or hand controller, they perform a zero mean and unit variance adjustment of the data, and subtract the bounding box center to remove the effect of variable point densities along the trajectory that influence the centroid. Their approach loses spatial relationships among the headset and controller devices in the VR system. With smaller sample sizes, e.g., 41 users as in our work, we demonstrate that using the raw data alone as in Mathis et al. [36] or using device-centric normalization alone as in Miller et al. [41] provides lower success at identification and authentication in most cases than using the spatial relationships encoded by our approach. The approach of Liebers et al. [31] performs normalization of arm length and height, and report highest results of 90% with height normalization. However, their normalization method requires users to engage in a pre-interaction phase where the user observes their virtual hand, and the front-facing cameras on the headset are used to compute arm length and height. Such a pre-interaction phase may prove cumbersome for the user, limiting fluid usability. In contrast, our approach does not require a preparatory phase, making it seamless for users to use the environment.

Study	Classifier	Users	Activities	VR System	Features	Success
Mustafa et al. [43]	SVMs	23	Listen to Music	Cardboard	Head movement patterns	7% (EER)
Kupin et al. [27]	Nearest Neighbor	14	Ball-throwing	HTC Vive	Position of right controller	92.86%
Ajit et al. [1]	Perceptron	33	Ball-throwing	HTC Vive	Position & orientation for both controllers and headset	93.03%
Miller et al. [41]	Perceptron	41	Ball throwing	HTC Vive, Oculus Quest, HTC Vive Cosmos	Position & orientation for both controllers, trigger, velocity, and angular velocity	91%-97%, 58%-85%*
Pfeuffer et al. [49]	Random Forests	22	Point, grab, walk, type	HTC Vive	Position, orientation, linear velocity, angular velocity for both controllers and headset	63.55%
Miller et al. [39]	Random Forests	511	360-degree videos	HTC Vive	Position & orientation for both controllers	95%
Olade et al. [47]	Nearest Neighbor	25	Grab, rotate, drop	HTC Vive	Position & orientation for both controllers and headset + eye position	98.6%
Mathis et al. [36]	FCNs	23	Point at 3D cube	HTC Vive	Position & orientation for both controllers	98.91%
Liebers et al. [31]	RNNs	16	Archery	Oculus Quest	Position & orientation for both controllers	90%
Miller et al. [42]	Siamese networks	41	Ball throwing	HTC Vive, Oculus Quest, HTC Vive Cosmos	Position & orientation for both controllers	98.04%-99.75%, 87.82%-98.53%*

Table 1: Summary of related work in VR biometrics (SVMs = support vector machines, FCNs = fully convolutional networks, RNN = recurrent neural networks). Enrollment and use-time data is provided on the HTC Vive. Unless stated, 'Success' refers to accuracy at identifying users from VR behavior. *Cross-system accuracy.

3 Dataset

Our experiments use the dataset of Miller et al. [41, 42] which consists of 41-subjects performing a ball-throwing action using three VR systems, namely the Oculus Quest, the HTC Vive, and the HTC Vive Cosmos. Each subject is asked to pick up a virtual ball and throw it at a target 10 times during two sessions. The sessions are separated by a minimum of 24 hours. Subjects are not penalized for missing the target and data is recorded for 3 seconds. Subjects are recruited from the faculty, staff, and student body after clearance from the university's Institutional Review Board (IRB). Each subject provides data for six sessions using the Oculus Quest, HTC Vive, and HTC Vive Cosmos in that order. The average time difference between sessions is 1.17 ± 0.80 days for the two Quest sessions, 2.27 $\pm$ 2.36 days between the second Quest and first Vive session, 3.00 $\pm$ 2.07 days for the two Vive sessions, 8.15 ± 8.61 days between the second Vive and first Cosmos session, and 2.05 ± 1.60 for the two Cosmos sessions.

We choose the Miller et al. dataset as it is the only known dataset with within- and cross-system user data. In the cross-system setting user trajectories for enrollment and input are captured using distinct tracking mechanisms, namely external infrared (IR) emitting lighthouses for the HTC Vive, 4 cameras on the Oculus Quest, and 6 cameras on the HTC Vive Cosmos. As shown in Figure 2, the tracking technology creates subtle differences in the user trajectory across different VR systems. The HTC Vive provides cleaner trajectories with fewer perturbations as the IR emitting lighthouses are less likely to lose tracking unless there is interference or improper setup. The Oculus Quest and HTC Vive Cosmos cameras lose tracking during fast movements or if the user's arm moves away from the camera's field of view. The HTC Vive Cosmos trajectories demonstrate perturbations attributable to imprecise tracking. Differences in tracking technology adversely affect performance in cross-system identification and authentication [41, 42] in contrast to comparisons within the same system.

4 Real-World Behavior Constraints

Our work evaluates the effect of modeling higher-level real-world constraints in device trajectories on the success of identification and authentication using matching networks such as those used in Miller et al. [42], and using N-class networks such as those used by Mathis et al. [36] and Liebers et al. [31]. In this section, we discuss our approach to include real-world constraints of spatial relationships and smoothing in the input data.

Spatial Relationships. We model inter-device spatial relationships by including displacement vectors between the positions of the headset and hand controllers as input features to neural networks. Given the position features $\mathbf{p}_r$, $\mathbf{p}_l$, and $\mathbf{p}_h$ for the right controller r, left controller l, and headset h, we compute normalized position features $\mathbf{p}'_r$, $\mathbf{p}'_l$, and $\mathbf{p}'_h$, using the approach used in Miller et al. [42] by first centering the points in each device to have zero mean and unit variance over all time instants t, i.e., as

$$\bar{\mathbf{p}}_\star[t] = (\mathbf{p}_\star[t] - \textstyle\sum_t \bar{\mathbf{p}}_\star[t]) \,/\, (\|\mathbf{p}_\star[t] - \textstyle\sum_t \bar{\mathbf{p}}_\star[t]\|)\,, \tag{1}$$

and re-aligning the points with respect to the center of the bounding box around the zero-mean and unit-variance trajectory as

$$\mathbf{p}'_\star[t] = \bar{\mathbf{p}}_\star[t] - 0.5\,(\max_t \bar{\mathbf{p}}_\star[t] + \min_t \bar{\mathbf{p}}_\star[t])\,, \tag{2}$$

to eliminate spatial imbalance in the center of the device's coordinate system due to high point accumulations in the start or end regions of the trajectory. In Equations (1) and (2), $\star$ may be either h, r, or l for the headset, right controller, or left controller respectively. Our work

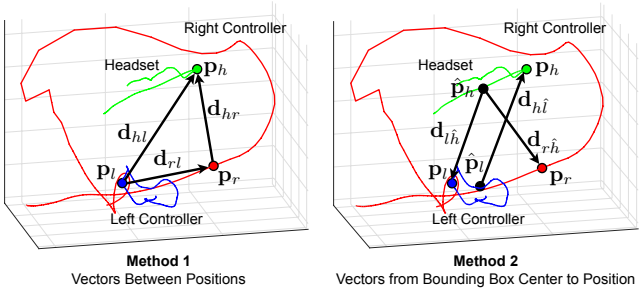

Figure 3: Calculation of displacement vectors. The curves in red, blue, and green show the trajectories for the right controller, left controller, and headset respectively. Colored points show positions at a particular time instant along the trajectory. Black points show the bounding box centers for each trajectory.

augments the normalized position features $\mathbf{p}'_r$, $\mathbf{p}'_l$, and $\mathbf{p}'_h$, and the original orientation features $\mathbf{q}_r$, $\mathbf{q}_l$, and $\mathbf{q}_h$ at each time instant with displacement vectors between pairs of devices. We evaluate two methods to obtain displacement vectors. In Method 1, we compute the displacement vectors from the right controller to the headset $\mathbf{d}_{hr}$, the left controller to the headset $\mathbf{d}_{hl}$, and from the left to the right controller $\mathbf{d}_{rl}$, as the difference between the positions of the corresponding devices at each time instant t as

$$\mathbf{d}_{hr}[t] = \mathbf{p}_h[t] - \mathbf{p}_r[t], \tag{3}$$
$$\mathbf{d}_{hl}[t] = \mathbf{p}_h[t] - \mathbf{p}_l[t], \text{ and} \tag{4}$$
$$\mathbf{d}_{rl}[t] = \mathbf{p}_r[t] - \mathbf{p}_l[t]. \tag{5}$$

In the above equations, the notation $[t]$ for each quantity is used to indicate the value of that quantity at time instant t. We do not require the opposite vectors $\mathbf{d}_{rh}$, $\mathbf{d}_{lh}$, and $\mathbf{d}_{lr}$, as these are negations of the vectors $\mathbf{d}_{hr}$, $\mathbf{d}_{hl}$, and $\mathbf{d}_{rl}$ respectively. We evaluate various combinations of displacement vectors obtained using Method 1 as features, i.e., $\mathbf{d}_{hr}$, $\mathbf{d}_{hl}$, and $\mathbf{d}_{rl}$ alone, pairs of $\mathbf{d}_{hr}$ and $\mathbf{d}_{hl}$, $\mathbf{d}_{hr}$ and $\mathbf{d}_{rl}$, and $\mathbf{d}_{hl}$ and $\mathbf{d}_{rl}$, and the entire triplet of vectors.

Method 1 estimates displacement vectors with respect to a frame that moves with the reference device. Since point-to-point displacement vectors may encode noise due to inaccuracies in tracking, in Method 2, we represent the displacement vector for one device in terms of a fixed frame with respect to a second device acting as reference. We use the center of the bounding box around the trajectory of the reference device as an anchor to estimate the displacement vector for a different device. For each device, we have the choice of one of the other two devices as reference. We avoid using the right controller as reference, as tracking inaccuracies during fast motion, exhibited by the dominant hand during ball-throwing, cause it to be inaccurately tracked. As such, the positions $\mathbf{p}_l$ and $\mathbf{p}_h$ of the left controller and headset are anchored with respect to each other's bounding box centers, i.e., $\hat{\mathbf{p}}_h$ and $\hat{\mathbf{p}}_l$ respectively, yielding displacement vectors $\mathbf{d}_{l\hat{h}}$ and $\mathbf{d}_{h\hat{l}}$ given as

$$\mathbf{d}_{l\hat{h}}[t] = \mathbf{p}_l[t] - \hat{\mathbf{p}}_h[t] \text{ and } \mathbf{d}_{h\hat{l}}[t] = \mathbf{p}_h[t] - \hat{\mathbf{p}}_l[t], \tag{6}$$

where $\hat{\mathbf{p}}_h$ and $\hat{\mathbf{p}}_l$ are obtained as

$$\hat{\mathbf{p}}_h = 0.5\,(\max_t \mathbf{p}_h[t] + \min_t \mathbf{p}_h[t]) \text{ and} \tag{7}$$
$$\hat{\mathbf{p}}_l = 0.5\,(\max_t \mathbf{p}_l[t] + \min_t \mathbf{p}_l[t])\,. \tag{8}$$

For the right controller, we have the choice of anchoring it with respect to $\hat{\mathbf{p}}_h$ or $\hat{\mathbf{p}}_l$. We choose $\hat{\mathbf{p}}_h$ to model the joint hand-headset motion as a user's head and eye movements follow the dominant hand movement during their action. This yields displacement vector

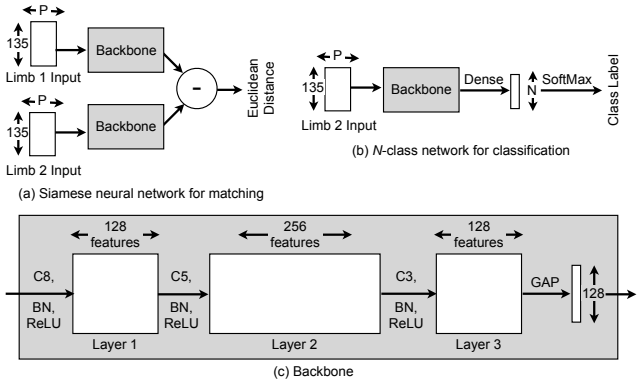

Figure 4: Architectures for (a) Siamese network and (b) *N*-class classification, with (c) backbone for both architectures. C*n*, BN, ReLU: Convolution with learned features of size *n*, batch normalization, and application of rectified linear unit activation, GAP: global average pooling, *P*: number of channels in input, and *N*: number of classes, i.e., users.

$\mathbf{d}_{r\hat{h}}$ for the right controller, whose value at time instant t can be expressed as

$$\mathbf{d}_{r\hat{h}}[t] = \mathbf{p}_r[t] - \hat{\mathbf{p}}_h[t]. \tag{9}$$

To represent all relationships, we evaluate Method 2 using the entire triplet $\mathbf{d}_{r\hat{h}}$, $\mathbf{d}_{l\hat{h}}$, and $\mathbf{d}_{h\hat{l}}$. Figure 3 shows displacement vectors using the two methods for trajectories of user 5 using the Vive.

We compare our proposed model against the ability of the original raw data to describe spatial relationships through a baseline model, i.e., Model A, where we use the raw data directly without preprocessing. Model A may malperform due to overall inter-user and intra-user offsets in position. We evaluate against baseline Model B to assess removal of translational offsets while retaining the spatial relationships represented by the raw data. We use two versions of Model B—Model Bc where the raw data is centered about the bounding box center of the entire data, rather than on a per-device basis, and Model Bn where the raw data is normalized to be zero-mean and unit variance as a whole. We also evaluate our work against the the original model of Miller et al. [42] which uses per-device normalized position features without inter-device spatial relationships. We refer to the Miller et al. model as Model C. In all cases, we use orientation features without normalization as they are expressed using Euler angles.

Smoothness Constraints. We represent the inherent smoothness of underlying real-world behavior by filtering the trajectory positions using a Gaussian filter g. At each time point t, the smoothed position $\mathbf{p}_{\star,s}$ for the device $\star$ is represented as

$$\mathbf{p}_{\star,s}[t] = \textstyle\sum_{v=-s}^{s} g[v]\mathbf{p}_\star[t+v]. \tag{10}$$

We perform device-centric normalization according to Equations (1) and (2) after smoothing. We compare the effect of introducing smoothing at a low scale of $s = 1$ time step and a higher scale of $s = 2$ time steps against using original non-smooth trajectories.

5 Neural Networks

5.1 Architectures

We evaluate two neural network architectures in this work.

1. The first architecture is a Siamese network similar to Miller et al. [42] who show that Siamese networks provide high accuracy for cross-system biometrics analyzed in our work. The network

takes position, orientation, and displacement vector features from the runtime input trajectories on the first limb and from enrollment trajectories in a library on the second limb, and returns a match distance between each enrollment-input pair as the output. We perform identification by returning the user for the enrollment trajectory with lowest distance as the label for the input trajectory, and demonstrate identification accuracies as results. We perform authentication by comparing match distances against a threshold, and obtaining the equal error rate (EER) as the value of false accept rate (FAR) where the FAR is identical to the false reject rate (FRR).

2. The second architecture is an N-class classification network similar to Mathis et al. [36] and Liebers et al. [31] that takes in position, orientation, and displacement vector features from a user as input, and returns the identity (ID) of the user as output for identification. We perform authentication by comparing the network probability to varying thresholds, and obtaining EER as the value of FAR where FAR and FRR are identical.

As shown in Figures 4(a) and 4(b), we adapt the Siamese network of Miller et al. [42] and fully convolutional networks (FCNs) shown to provide highest accuracy in Mathis et al. [36] to take in displacement vectors as additional input. Both networks use the same backbone as shown in Figure 4(c). We use an input of size $135\times P$ consisting of 135 time samples and P channels, where the channels consist of 3 coordinates X, Y, and Z for the position of each of the 3 devices yielding 9 position features, 3 Euler angles for the orientation of the 3 devices yielding 9 orientation features, and the coordinates of the displacement vectors. Displacement vector features may be 3 for single vectors, i.e., $\mathbf{d}_{hr}$, $\mathbf{d}_{hl}$, and $\mathbf{d}_{rl}$ alone, 6 for pairs, i.e., $\mathbf{d}_{hr}$ and $\mathbf{d}_{hl}$, $\mathbf{d}_{hr}$ and $\mathbf{d}_{rl}$, and $\mathbf{d}_{hl}$ and $\mathbf{d}_{rl}$, or 9 for triplets $\mathbf{d}_{hr}$, $\mathbf{d}_{hl}$ and $\mathbf{d}_{rl}$ (termed 'All'), and $\mathbf{d}_{r\hat{h}}$, $\mathbf{d}_{l\hat{h}}$ and $\mathbf{d}_{h\hat{l}}$ (termed 'Bbc'). For Figure 4(b), the identity of the user is returned as the one with the highest probability over N user classes.

5.2 Training and Testing Method

We use the Miller et al. [41,42] 41-user dataset to assess the within-system and cross-system identification accuracy and authentication EER of our approach against baselines without displacement vectors, with and without smoothing. We perform an n-fold evaluation in order to test our proposed contributions for varying sizes of test user groups and enrollment libraries. We evaluate performance for small test user groups via a 10-fold cross-validation, where 9 folds have 4 test users per fold and the 10th has 5 test users. To assess scalability with increase in the number of test users, we evaluate performance using a 5-fold cross-validation, where 4 folds have 8 test users per fold, and the 5th has 9 test users. For each fold, we train a Siamese network by using the input and enrollment pairs from users left out of the fold as training. The users included in the fold form the test users. During training, we use the Adam optimizer with a batch size of 128, and a cyclic learning rate varying as a triangle wave between 10^{-6} to 10^{-3} and a cycle of 5 epochs.

For each cross-validation, we evaluate two versions of enrollment library sizes, one where the enrollment data comes from the test users only, termed '10TS' and '5TS' for '10-fold Test using Siamese' and '5-fold Test using Siamese' respectively. This scenario represents the typical situation where a biometric algorithm trained to perform matching using enrollment and input data from a training set of users would be deployed within an organization. In the second version of the results, we use a larger enrollment library consisting of the enrollment data from the test and training users, i.e., all 41 users in the dataset. We term this set of results '10AS' and '5AS' for '10-fold All using Siamese' and '5-fold All using Siamese' respectively. This set of experiments allows us to determine scalability to a larger enrollment library. The input data still comes solely from the test users to prevent contamination of input from the training set.

To assess performance of N-class networks over varying user groups, we perform three analysis—one where a single network is trained on the enrollment data of all users termed 'AN' for 'All using N-class', and one where neural networks are trained per fold using the enrollment data within a fold for 10-fold and 5-fold evaluation, termed '10N' and '5N' for '10-fold using N-class' and '5-fold using N-class' respectively. Similar to the Siamese networks, we use the Adam optimizer with a batch size of 128 and a cyclic learning rate varying between 10^{-6} to 10^{-3} as a triangle wave with cycle of 5 epochs. For 'AN', testing is performed using input from all users, and for '10N' and '5N', testing is performed using input from users within the fold.

6 Results

We assess the highest accuracy and lowest EER across all three smoothing levels from Tables 2 to 5 for 10-fold and overall analyses, and Tables 1 to 3 in the supplementary for 5-fold analyses. In each table, we report the rank-1 and rank-2 accuracies and EERs, together with the corresponding best performing model for three smoothing levels—'S0' for no smoothing, 'S1' for smoothing using 1 time step, and 'S2' for smoothing using 2 time steps. For within-system identification and authentication, we assess system pairs where enrollment and input data comes from the same system, i.e., Q1/Q2 for the Quest, V1/V2 for the Vive, and C1/C2 for the Cosmos, where '1' and '2' refer to the first and second session respectively. The first session is used for enrollment and the second session as input. For cross-system assessment, we use data from the system used earlier as enrollment and compare it against data from the system used as input. With two Quest, two Vive, and two Cosmos sessions with data provided in that order, we have 12 system pairings—Q1/V1, Q1/V2, Q2/V1, and Q2/V2 for the Quest and Vive, Q1/C1, Q1/C2, Q2/C1, and Q2/C2 for the Quest and Cosmos, and V1/C1, V1/C2, V2/C1, and V2/C2 for the Vive and Cosmos. Since performance across both sessions of the devices in a pair are similar, we report average metrics Q/V, Q/C, and V/C over the 4 session combinations in each device pairing, where Q/V provides averages over Q1/V1, Q1/V2, Q2/V1, and Q2/V2, Q/C over Q1/C1, Q1/C2, Q2/C1, and Q2/C2, and V/C over V1/C1, V1/C2, V2/C1, and V2/C2.

6.1 10-Fold Analysis for Siamese Networks

As shown in Table 2, we obtain the highest accuracy when using '10TS'. For the rank-1 accuracy, we observe that our approach yields the highest accuracy across the 3 smoothing levels in 4 out of 6 system combinations. Our approach with *hr* and *rl* features achieves the highest accuracy at 98.05% for Q1/Q2 and 98.83% for V1/V2. For C1/C2, our approach provides the highest accuracy of 93.17% using *hr* and *hl* as features. In cross-system combinations, Model Bn achieves highest accuracy for V/C, while our approaches with *hl* features achieves the highest accuracy for Q/V at 87.81%. Our approach with all features provides the highest accuracy at 80.12% for Q/C. The HTC Vive (V) achieves the highest accuracy in within-system comparisons as it tracks user behavior with external lighthouses and is less likely to stop tracking unlike camera-based systems, such as the Oculus Quest (Q) and HTC Vive Cosmos (C). The Cosmos has the lowest accuracy due to the sensitivity of the cameras to changes in ambient lighting [15]. As shown in Figure 2, the Vive and Quest generate fewer tracking errors than the Cosmos. In cross-system combinations, the Quest/Vive (Q/V) pairing has the highest accuracy due to the quality of the tracking mechanism as shown in Figure 2. Cross-system matching does not perform as well as within-system matching due to the limited data. With larger sample sizes it is expected that the neural network will be able to more accurately learn the systematic differences. In the 4 rank-1 cases where our approach provides highest accuracy, displacement vectors *hr*, *rl*, and *hl* contribute to the highest accuracy 3, 3, and 2 times respectively.

As shown in Table 3, in '10AS' our approach has the highest

	Q1/Q2			V1/V2			C1/C2			Q/V			Q/C			V/C		
	S0	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**
1-Acc.	97.32	98.05	96.83	96.59	96.83	96.83	93.17	94.15	93.17	87.81	87.31	87.01	78.72	80.12	78.11	83.29	82.26	82.38
1-EER	7.81	9.09	8.49	8.57	9.40	9.44	10.08	11.88	11.31	15.63	15.91	14.82	20.92	20.2	21.63	19.71	20.46	20.47
1-Model	A	**Ours**	A	**Ours**	**Ours**	C	**Ours**	Bn	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	Bc	Bn	**Ours**	Bn
		hr, *rl*		*rl*	*hr*, *rl*		*hr*, *hl*		All	*hl*	All	All	*hl*	All			*rl*	
2-Acc.	96.59	97.07	96.1	96.34	96.59	95.85	92.93	92.44	92.93	87.56	87.2	86.89	78.47	78.35	77.44	82.92	82.01	80.42
2-EER	9.48	8.97	10.20	9.56	9.00	9.57	11.47	11.71	11.43	15.79	16.45	15.13	21.33	21.48	22.89	19.45	20.42	20.19
2-Model	Ours	A	Ours	C	Ours	Ours	Ours	Ours	A	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Bn	Ours
	hr, *rl*		*hl*, *rl*		*hl*	*hr*	*hr*	*hl*, *rl*		*hr*, *rl*	Bbc	*hl*	*hl*, *rl*	*hr*, *hl*	*hr*, *rl*	*hr*, *rl*		All

Table 2: Maximum and second maximum accuracies per system combination and smoothing level when using Siamese networks with 10-fold analysis, and with input and enrollment from test users ('10TS'). Smoothing levels used are 'S0' for no smoothing, 'S1' for smoothing with 1 time step, and 'S2' for smoothing with 2 time steps. Q1/Q2, V1/V2, and C1/C2 are comparisons of Quest (Q), Vive (V), and Cosmos trajectories between sessions 1 and 2. Q/V, Q/C, and V/C are averages of cross-system performance for Quest to Vive, Quest to Cosmos, and Vive to Cosmos when enrollment data from the earlier system is compared against input data from the later system. 1-EER/1-Model and 2-EER/2-Model represent the EER or model at the first and second maximum accuracies respectively. 'Ours' represents the maximum over approaches that incorporate distance vectors. 'All' represents the triplet hr, hl, rl, and 'Bbc' stands for the triplet that uses distance vectors computed with respect to bounding box centers, i.e., $r\hat{h}, l\hat{h}, h\hat{l}$. We perform best for 11 combinations.

	Q1/Q2			V1/V2			C1/C2			Q/V			Q/C			V/C		
	S0	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**
1-Acc.	96.34	97.32	94.63	94.63	95.12	94.63	90.49	91.22	92.93	85.18	83.66	85.00	73.78	76.03	72.56	76.77	76.70	76.10
1-EER	1.75	2.20	2.53	2.72	2.66	2.65	3.36	3.41	3.38	3.69	4.02	3.66	5.13	4.65	4.96	5.25	5.24	5.56
1-Model	A	**Ours**	**Ours**	C	**Ours**	**Ours**	A	A	A	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**
		hr, *rl*	*hr*		*rl*	*hr*				*hl*, *rl*	Bbc	All	*hl*, *rl*	All	*hr*, *rl*	*hr*, *hl*	*hr*, *rl*	All
2-Acc.	95.61	96.34	94.63	94.39	94.63	94.39	89.27	90.49	88.29	84.39	83.35	84.64	73.41	73.17	71.77	76.64	76.1	75.85
2-EER	2.31	2.18	2.43	2.61	2.64	2.45	2.96	2.72	3.49	3.74	3.61	3.38	5.76	5.43	5.18	5.25	5.64	5.39
2-Model	Ours	Ours	Ours	Ours	Ours	Ours	Bn	Bn	Ours	Ours	Ours	Ours	Bc	Ours	Ours	Ours	Ours	Ours
	hr, *rl*	All	*hl*, *rl*	*rl*	*hr*	*hl*			*hr*	*hr*, *hl*	All	*hl*		*hr*, *hl*	*hl*	*hr*, *rl*	*rl*	*hr*, *rl*

Table 3: Maximum and second maximum accuracies per system combination and smoothing level when using Siamese networks with 10-fold analysis, and with input from test users and enrollment from all users ('10AS'). Similar conventions used as in Table 2. We perform best for 13 combinations.

accuracy at 97.32% using *hr* and *rl* features for Q1/Q2. Model A has the highest accuracy at 92.93% for the C1/C2 combination. Our approach with *rl* features gives the highest accuracy at 95.12% for V1/V2. In cross-system combinations our approach outperforms the baseline and provides highest accuracy at 85.18% with *hl* and *rl* features for Q/V, 76.03% with all features for Q/C and 76.77% with *hr* and *hl* for V/C. In the 5 instances where we achieve highest rank-1 accuracy over the 3 smoothing level, *hr*, *rl*, and *hl* contribute 3, 4, and 3 times toward highest accuracies. In all cases we observe that smoothing has limited effect on the accuracy.

6.2 5-Fold Analysis for Siamese Networks

As shown in Table 1 in the supplementary, our approach provides highest accuracy in 5 out of 6 system pairings when assessed over all smoothing levels. The displacement vectors *hr*, *rl*, and *hl* contribute to improved accuracy in 4, 4, and 5 cases. With '5TS' we observe that for the Q1/Q2 combination, our approach using *hr* and *hl* as features provides the highest accuracy of 93.17%. When analyzing within-system performance, We achieve the highest accuracy of 93.90% with the V1/V2 combination using our approach and *hl* and *rl* as features. For cross-system combinations, the Model Bn provides the highest accuracy of 74.58% for the V/C combination. Our approach with all features provides the highest accuracy of 78.41% for the Q/V pairing and 69.27% for the Q/C pairing.

For '5AS', as shown in Table 2 in the supplementary, in the 5 out of 6 system pairings where using displacement vectors shows highest accuracy, the features *hr*, *rl*, and *hl* contribute in 4, 2, and 5 cases. The higher contribution of *hl* for larger test user pools may point to the advantage of including a feature that maps largely static components, i.e., the head and left hand. As expected, the accuracy for all system combinations is lower when compared to enrollment and input data coming from test users only. Our approach with *hr* and *hl* as features provides the highest accuracy of 92.20% for the Q1/Q2 combination. Our approach with all features provides the highest accuracy of 93.17% for the V1/V2 combination. For the C1/C2 combination our approach provides the highest accuracy of 86.64% using *hl* as the feature. For the Q/V combination, Model Bn gives the highest accuracy of 73.96%, while for the remaining two cross-system combinations our approach provides the highest accuracy of 62.07% for Q/C with *hr* and *hl* as features and 67.44% for V/C with all features. Similar to '10TS' and '10AS', smoothing has minimal impact on accuracy.

6.3 *N*-class Classification Analysis

In Table 4, we provide results for N-class classification using 'AN'. For the within- and cross-system combinations of Q1/Q2, V1/V2, and C1/C2 our approach provides highest accuracy of 91.46% with all features for Q1/Q2 and 91.71% and 85.65% with *hr* as features for V1/V2 and C1/C2 respectively. For cross-system combinations, our approach provides highest accuracy with 54.33% with *hr* and *hl* as features for Q/C and 67.86% and 60.85% with all features for Q/V and V/C respectively. Our approach provides the highest accuracy in all pairing, with feature contributions appearing 6 times for *hr*, 3 times for *rl*, and 4 times for *hl*.

As shown in Table 5, for '10N' our approach provides the highest accuracy for all cross-system pairings. Our approach with all

	Q1/Q2			V1/V2			C1/C2			Q/V			Q/C			V/C		
	S0	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**
1-Acc.	91.46	91.22	91.71	91.71	90.24	90.73	83.66	84.39	85.85	67.86	67.56	67.56	54.15	54.02	54.33	59.69	59.94	60.85
1-EER	2.90	2.60	2.70	2.40	2.90	3.50	5.30	5.50	4.70	11.00	10.45	9.68	13.82	14.35	13.57	15	15.45	14.25
1-Model	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	Bn	Bn	**Ours**	**Ours**	**Ours**	**Ours**
	All	*hr*,*hl*	*hl*,*rl*	*hr*	*hr*,*rl*	*hr*	*hr*,*hl*	All	*hr*	All	*hr*,*rl*	*hl*,*rl*			*hr*,*hl*	*hr*,*rl*	*hl*,*rl*	All
2-Acc.	90.73	90.98	91.46	91.46	89.76	88.29	83.17	83.66	84.88	67.13	67.20	67.44	53.84	53.54	54.27	59.63	59.33	60
2-EER	2.50	3.60	2.60	3.40	2.70	2.90	5.60	5.50	4.90	11.12	10.75	10.15	13.70	13.60	13.55	15.12	15.00	15.60
2-Model	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours	Ours
	hr	All	All	*hr*,*hl*	*rl*	*hr*,*rl*	*hr*	*hr*,*rl*	*hl*,*rl*	*hl*,*rl*	All	*hl*	Bbc	All	All	All	*hr*,*rl*	Bbc

Table 4: Maximum and second maximum accuracies per system combination and smoothing level when using *N*-class networks, with training using enrollment from all users and testing with input from all users ('AN'). Similar conventions used as in Table 2. We perform best for 16 combinations.

	Q1/Q2			V1/V2			C1/C2			Q/V			Q/C			V/C		
	S0	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**	**S0**	**S1**	**S2**
1-Acc.	95.12	94.39	94.15	96.10	97.56	95.61	90.73	90.24	91.22	80.73	81.65	82.02	76.40	75.80	77.07	75.24	74.75	75.55
1-EER	2.80	3.90	3.50	2.40	1.30	2.50	5.80	6.80	5.80	12.78	11.95	12.05	15.05	15.55	13.93	16.35	16.30	15.58
1-Model	Bn	**Ours**	Bn	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	**Ours**	Bn	**Ours**	**Ours**	**Ours**	**Ours**
		hr,*rl*		*hr*,*rl*	All	*hr*,*rl*	All	*rl*	*hl*	All	Bbc	All	All		*hl*,*rl*	*hr*,*rl*	All	*hl*,*rl*
2-Acc.	94.39	94.15	94.15	95.37	95.37	95.37	90.00	90.00	90.73	80.30	81.34	81.03	76.04	75.42	76.83	74.94	74.45	75.37
2-EER	3.80	3.30	3.60	2.60	3.10	2.90	7.20	5.60	5.50	12.40	11.80	11.82	14.95	15.47	14.60	16.55	16.18	15.85
2-Model	Ours	Bn	***Ours***	Ours	Ours	Ours	C	Ours	Ours	Ours	Ours	Ours	Bn	Ours	Ours	Ours	Ours	Ours
	rl		*hr*,*hl*	*hr*	*hl*	All		*hl*,*rl*	*hl*,*rl*	Bbc	*hr*,*hl*	Bbc		*hl*,*rl*	Bbc	Bbc	Bbc	All

Table 5: Maximum and second maximum accuracies per system combination and smoothing level when using *N*-class networks, with training and testing using enrollment and input from users restricted to each fold of a 10-fold split ('10N'). Similar conventions used as in Table 2. We perform best for 15 combinations and are tied for 1 combination.

features provides the highest accuracy of 82.02% for Q/V. For the Q/C pairing our approach with *hl* and *rl* as features provides the highest accuracy of 77.07%. For the V/C using *hl* and *rl* as features we achieve the highest accuracy of 75.55%. Among within-system pairings, Model Bn achieves the highest accuracy of 95.12% for Q1/Q2, while our approach with all features achieves best accuracy of 97.56% for V1/V2 and 91.22% for C1/C2 with *hl* as feature. Table 3 in the supplementary provides results using '5N'. For within-system combinations our approach yields highest accuracy. In the Q1/Q2 combination we achieve accuracy of 94.39% with *hr* and *hl* as features. For V1/V2 *hr* as features provides accuracy of 94.63%. Our approach with all features gives an accuracy of 90.24% for C1/C2. For cross-system pairs, Model Bn shows highest accuracy 71.22% for the V/C pair while our approach provides the highest accuracy of 79.94% with vectors computed using the bounding box center as features for Q/V, and 72.44% for Q/C with all non-bounding-box displacement vectors. For the 5 pairs where we outperform prior models, feature vectors *hr*, *rl*, and *hl* contribute in 2, 4, and 5 cases for '10N' and in 4, 3, and 2 cases for '5N'.

6.4 Best Performing Models and Vectors

Overall, across 42 cases consisting of the 6 system pairings and 7 experiments (4 for Siamese and 3 for *N*-class), using displacement vectors shows higher accuracy than baseline models in 36 cases, indicating that displacement vectors have the potential to improve performance of deep learning algorithms for VR biometrics. Overall, the displacement vectors *hl*, *hr*, and *rl* appear 26, 26, and 23 times respectively. The results suggest that the relationships of the controllers to the headset play a higher role in improving performance compared to the relationships between controllers. Using the bounding box of the trajectories appears to have reduced impact in comparison to using point-to-point displacements. The point-to-point displacements also offer the advantage that they can be computed in real time, unlike the trajectory's bounding box center which requires complete knowledge of the trajectory.

6.5 Impact of Smoothing

For all approaches smoothing has limited impact on improving accuracy or lowering EER. While the dataset examined contains non-smooth trajectories such as those generated using the Cosmos, the lack of smoothness is systematic and occurs due to small perturbations. Where high accuracies are generated, the networks may be capable of learning the nature of small systematic perturbations, as a result of which smoothing may have limited effect. Where low accuracies are generated, e.g., for matching between the Vive and Cosmos, the low matches may likely be due to coarse overall differences in trajectory appearance as shown in Figure 2, rather than due to variation in smoothness across systems or users.

7 Discussion

We demonstrate that using a combination of device-centric trajectory specification and inter-device displacement vectors as features improves the performance in authentication and identification using behavioral biometrics in VR over baseline approaches that use raw trajectories or that perform global normalization without treating each device's trajectory independently. The limited impact of smoothing may be attributed to the VR systems used and the data collection protocol. The HTC Vive uses external lighthouses to track the hand controllers and the headset. Barring any changes to set up, improper placement of the lighthouses, or external interference sources it is expected that the lighthouses will track the hand controllers and headset with limited variability across captures. The lighthouses were permanently secured during the collection of the dataset of Miller et al. [41, 42] for the duration of the study. The Oculus Quest and HTC Vive Cosmos use external cameras to perform tracking of the hand controllers and headset that are more

susceptible to changes in lighting and environment conditions. The dataset provided by Miller et al. [41, 42] was collected in a single research lab with no external windows and non-adjustable lighting, thus the influence of ambient lighting across sessions is limited. Large-scale datasets that vary physical capture space set up, lighting, and environment conditions are needed to understand the true effect of smoothing in improving the performance of VR biometrics.

For non-smooth trajectories in the dataset, the lack of smoothness is largely systematic. Negative impact of non-smooth trajectories is likely to occur if large infrequent perturbations occur, e.g., the motions involved in conducting the task are such that some users may occasionally track outside the range of the tracking mechanisms, so that the neural networks lack sufficient data to learn these anomalous trajectory patterns. To handle these anomalies, in future work we are interested in investigating intelligent smoothing approaches by using a robotic arm programmed with a known motion pattern to probe the limits of the tracked space, relating the known pattern to the motion picked up by the tracker, and creating approaches to automatically synthesize the ground truth motion from a novel tracked motion. Additionally, this work only performs smoothing on the position vectors, and does not study effect of tracking noise on the orientation quaternions and the use of smoothing to eliminate noise in the orientations. For an activity such as ball-throwing with largely in-plane motion for the dominant hand and head, orientation vectors may demonstrate limited noise in tracking, however, activities involving higher degrees of rotational freedom such as flying a drone, driving, moving and handling objects, or writing and drawing in VR may be more susceptible to noise in the orientation vectors. As part of future work, we are interested in expanding the space of activities to include tasks with high rotational freedom occurring at various distances from the tracking mechanism.

While algorithmic pre-processing approaches may address some challenges related to collection at scale and the need for large datasets to train deep learning methods, we recognize that to capture the diversity of behavior patterns and actions in the real-world, scaling up VR data collection is essential to create robust security algorithms in future work. Work in behavior-based biometrics for keystroke, mouse, and smartphones benefits from large-scale datasets containing multiple devices due to the ubiquitous nature of laptops and smartphones and the ease of capturing data through web-based or downloadable applications. Currently, the largest VR biometrics dataset is that of Miller et al. [39] with 511 subjects. While the dataset has a large number of subjects, its ethological validity remains a concern, since users exhibited limited movement as the study focused on users watching a series of videos and answering questions. Investigation of behavioral biometrics using multiple VR systems has been limited to a maximum of 3 systems by Miller et al. [41, 42]. Data collection will be accelerated only when VR systems are placed into the hands of large groups of consumers. This in turn requires the development of VR devices that are affordable. As more standalone VR systems, such as the Oculus Quest and HTC Vive Focus, and low-cost options, such as the sub $300 128GB Oculus Quest 2, enter the market, one may expect an uptick in the number of consumers purchasing VR systems. While the Google Cardboard attempted to make VR accessible to a broader group of users, it lacked the ability to track full-range motions in VR. To encourage the average consumer to rapidly embrace VR applications, it is necessary to develop VR systems that leverage modern smartphones that contain multiple front and rear facing cameras along with depth sensors to perform inside out-tracking. The average consumer is more likely to procure a low-cost system similar to Google Cardboard that can be integrated with their existing smartphone and provides capabilities similar to modern standalone VR systems.

It is also essential that VR applications everyday activities, e.g., banking, social networking, physical fitness, and office and educational productivity, are such that users automatically find VR to be a more seamless environment for those activities in comparison to traditional devices such as laptops, smartphones, and tablets. So far, discussions of security versus usability in the VR space have remained confined to application access [48]. However, to lay the foundation for at-scale collection of user behavior, future research must include exploration of usability from the standpoint of the physical characteristics of the device such as weight and bulkiness, the seamlessness of the interaction mechanisms, e.g., latency, comfort level of long-term handheld device usage, and miss rate of hand tracking algorithms for hands-free interaction, and user preferences on use and security for VR versus traditional devices for everyday applications. While some studies do require human subjects data collection, a multimodal approach may be leveraged, by conducting questionnaires on user preferences, or analyzing small focus groups for device usage comfort. Currently, most datasets for behavior-based VR security focus on singular repetitive actions such as throwing, swinging, picking, and pointing. While developing VR applications and their security mechanisms, focus should be laid on providing and analyzing multi-step activities, e.g., visiting a bank, that involve multiple unit actions such as talking to a teller, filling a deposit slip, and depositing a check, with varying choices and permutations of actions in different instances of the activity.

If VR systems are to replace smartphones and laptops as the de-facto instrument for work, education, entertainment, and critical systems then a symbiotic relationship needs to develop between research teams in VR biometrics, manufacturers of VR systems, and creators of VR applications to enable collection of behavior-based data in realistic environments as opposed to using lab designed applications. The VR community at large plays a critical role in garnering security and ethical concerns regarding widespread VR usage and encouraging the participation of average consumers in providing high-assurance behavior-based security measures for VR. Until VR devices include sensors to collect traditional biometrics, such as fingerprint, face, and iris, behavior-based mechanisms will be necessary to provide behavior- and system-independent continuous authentication. It is also critical to perform translational work to propagate VR by visiting companies, schools, colleges, and older adult care facilities to demonstrate how VR applications for office productivity, education, and health can transform user experience and quality of life.

8 Conclusion

In this work, we address the concern of performing VR identification and authentication using deep learning with small datasets by evaluating spatial and smoothing methods to pre-process input data. Our work assesses deep learning methods to perform classification using convolutional neural networks and matching using Siamese neural networks. We demonstrate that the inclusion of displacement vectors in the input data improves the performance of Siamese neural networks and N-class networks when compared to baseline approaches. On average, we observe a 0.63% to 0.73% improvement for 10-fold, 0.55% to 0.85% for 5-fold, and 0.93% to 2.08% for N-class classification when comparing our work to existing approaches. While limited impact is obtained using smoothing, the higher identification success observed for 36 out of 42 combinations of user sets and VR system pairings indicates the promise of methods that explicitly represent spatial relationships in the input. As part of future research, we are interested in creating VR applications that represent virtual versions of everyday multi-step activities performed by users, e.g., working in an office, attending a meeting, visiting the bank, going to class, and visiting the doctor's office. Our work will enable investigation of the usability, acceptability, and security of VR applications for proliferation of VR to educational, business, consumer, and mission critical spaces.

REFERENCES

[1] A. Ajit, N. K. Banerjee, and S. Banerjee. Combining pairwise feature matches from device trajectories for biometric authentication in virtual reality environments. In *Proc. AIVR*. IEEE, New York, USA, 2019.

[2] F. A. Alsulaiman and A. El Saddik. A novel 3d graphical password schema. In *Proc. VECIMS*. IEEE, New York, USA, 2006.

[3] F. A. Alsulaiman and A. El Saddik. Three-dimensional password for more secure authentication. *IEEE Transactions on Instrumentation and Measurement*, 57(9):1929–1938, Sep 2008.

[4] F. Boutros, N. Damer, K. Raja, R. Ramachandra, F. Kirchbuchner, and A. Kuijper. Fusing iris and periocular region for user verification in head mounted displays. In *Proc. FUSION*. IEEE, New York, USA, 2020.

[5] F. Boutros, N. Damer, K. Raja, R. Ramachandra, F. Kirchbuchner, and A. Kuijper. On benchmarking iris recognition within a head-mounted display for ar/vr applications. In *Proc. IJCB*. IEEE, New York, USA, 2020.

[6] F. Boutros, N. Damer, K. Raja, R. Ramachandra, F. Kirchbuchner, and A. Kuijper. Periocular biometrics in head-mounted displays: A sample selection approach for better recognition. In *Proc. IWBF*. IEEE, New York, USA, 2020.

[7] M.-S. Bracq, E. Michinov, and P. Jannin. Virtual reality simulation in nontechnical skills training for healthcare professionals: A systematic review. *Simulation in Healthcare*, 14(3):188–194, Jun 2019.

[8] A. G. Campbell, T. Holz, J. Cosgrove, M. Harlick, and T. O'Sullivan. Uses of virtual reality for communication in financial services: A case study on comparing different telepresence interfaces: Virtual reality compared to video conferencing. In *Proc. FCC*. Springer, Berlin, Germany, 2019.

[9] B. J. Concannon, S. Esmail, and M. Roduta Roberts. Head-mounted display virtual reality in post-secondary education and skill training: A systematic review. In *Proc. FIE*. Frontiers, Switzerland, 2019.

[10] E. Czerniak, A. Caspi, M. Litvin, R. Amiaz, Y. Bahat, H. Baransi, H. Sharon, S. Noy, and M. Plotnik. A novel treatment of fear of flying using a large virtual reality system. *Aerospace medicine and human performance*, 87(4):411–416, Apr 2016.

[11] Z. Fang, A. Czajka, and K. W. Bowyer. Robust iris presentation attack detection fusing 2d and 3d information. *IEEE Transactions on Information Forensics and Security*, 16(8):510–520, Aug 2020.

[12] H. Feng, C. Li, J. Liu, L. Wang, J. Ma, G. Li, L. Gan, X. Shang, and Z. Wu. Virtual reality rehabilitation versus conventional physical therapy for improving balance and gait in parkinson's disease patients: A randomized controlled trial. *Medical science monitor: international medical journal of experimental and clinical research*, 25(5):4186–4192, Jun 2019.

[13] G. Fiumara, P. Flanagan, J. Grantham, K. Ko, K. Marshall, M. Schwarz, E. Tabassi, B. Woodgate, and C. Boehnen. National Institute of Standards and Technology Special Database 302: Nail to Nail Fingerprint Challenge. Technical Note 2007, National Institute of Standards and Technology, Aug. 2018. doi: 10.6028/NIST.TN.2007

[14] M. Funk, K. Marky, I. Mizutani, M. Kritzler, S. Mayer, and F. Michahelles. Lookunlock: Using spatial-targets for user-authentication on hmds. In *Proc. CHI Extended Abstracts*. ACM, New York, USA, 2019.

[15] D. Gajsek. HTC Vive vs Oculus: An In-Depth Guide On Which Headset Is Better For Business and Personal Use. https://circuitstream.com/blog/htc-vs-oculus/.

[16] A. Gangwar and A. Joshi. Deepirisnet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition. In *Proc. ICIP*. IEEE, New York, USA, 2016.

[17] S. J. Garbin, Y. Shen, I. Schuetz, R. Cavin, G. Hughes, and S. S. Talathi. Openeds: Open eye dataset. *arXiv preprint arXiv:1905.03702*, 2019.

[18] C. George, D. Buschek, A. Ngao, and M. Khamis. Gazeroomlock: Using gaze and head-pose to improve the usability and observation resistance of 3d passwords in virtual reality. In *Proc. AVR*. Springer, Berlin, Germany, 2020.

[19] 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 *Proc. VR*. IEEE, New York, USA, 2019.

[20] C. George, M. Khamis, E. von Zezschwitz, M. Burger, H. Schmidt, F. Alt, and H. Hussmann. Seamless and secure vr: Adapting and evaluating established authentication systems for virtual reality. In *Proc. NDSS*, 2017.

[21] J. Gurary, Y. Zhu, and H. Fu. Leveraging 3d benefits for authentication. *International Journal of Communications, Network and System Sciences*, 10(08):324–338, Aug 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*. IEEE, New York, USA, 2016.

[23] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. Searching for mobilenetv3. In *Proc. ICCV*. IEEE, New York, USA, 2019.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proc. CVPR*. IEEE, New York, USA, 2017.

[25] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*. PMLR, 2015.

[26] L. Jensen and F. Konradsen. A review of the use of virtual reality head-mounted displays in education and training. *Education and Information Technologies*, 23(4):1515–1529, Jul 2018.

[27] A. Kupin, B. Moeller, Y. Jiang, N. K. Banerjee, and S. Banerjee. Task-Driven Biometric Authentication of Users in Virtual Reality (VR) Environments. In *Proc. MMM*. Springer, Berlin, Germany, 2019.

[28] B. M. Kyaw, N. Saxena, P. Posadzki, J. Vseteckova, C. K. Nikolaou, P. P. George, U. Divakar, I. Masiello, A. A. Kononowicz, N. Zary, et al. Virtual reality for health professions education: systematic review and meta-analysis by the digital health education collaboration. *Journal of medical Internet research*, 21(1), Jan 2019.

[29] M. Lager and E. A. Topp. Remote supervision of an autonomous surface vehicle using virtual reality. *IFAC-PapersOnLine*, 52(8):387–392, Jul 2019.

[30] M. Li, S. Ganni, J. Ponten, A. Albayrak, A.-F. Rutkowski, and J. Jakimowicz. Analysing usability and presence of a virtual reality operating room (vor) simulator during laparoscopic surgery training. In *Proc. VR*. IEEE, New York, USA, 2020.

[31] J. Liebers, M. Abdelaziz, L. Mecke, A. Saad, J. Auda, U. Grünefeld, F. Alt, and S. Schneegass. Understanding user identification in virtual reality through behavioral biometrics and the effect of body normalization. In *Proc. CHI*. ACM, New York, USA, 2021.

[32] K. R. Lohse, C. G. Hilderman, K. L. Cheung, S. Tatla, and H. M. Van der Loos. Virtual reality therapy for adults post-stroke: a systematic review and meta-analysis exploring virtual environments and commercial games in therapy. *PloS one*, 9(3):e93318, Mar 2014.

[33] A. Luca, R. Giorgino, L. Gesualdo, G. M. Peretti, A. Belkhou, G. Banfi, and G. Grasso. Innovative educational pathways in spine surgery: Advanced virtual reality–based training. *World Neurosurgery*, 140(8):674–680, Aug 2020.

[34] M. Maciaś, A. Dabrowski, J. Fraś, M. Karczewski, S. Puchalski, S. Tabaka, and P. Jaroszek. Measuring performance in robotic teleoperation tasks with virtual reality headgear. In *Proc. AUTOMATION*. Springer, Berlin, Germany, 2019.

[35] M. G. Maggio, G. Maresca, R. De Luca, M. C. Stagnitti, B. Porcari, M. C. Ferrera, F. Galletti, C. Casella, A. Manuli, and R. S. Calabrò. The growing use of virtual reality in cognitive rehabilitation: fact, fake or vision? a scoping review. *Journal of the National Medical Association*, 111(4):457–463, Aug 2019.

[36] F. Mathis, H. I. Fawaz, and M. Khamis. Knowledge-driven biometric authentication in virtual reality. In *Proc. CHI Extended Abstracts*. ACM, New York, USA, 2020.

[37] F. Mathis, J. Williamson, K. Vaniea, and M. Khamis. Rubikauth: Fast and secure authentication in virtual reality. In *Proc. CHI Extended Abstracts*. ACM, New York, USA, 2020.

[38] 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*, 28(1):1–44, Jan 2021.

[39] 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*, 10(1):1–10, Oct 2020.

[40] R. Miller, A. Ajit, N. K. Banerjee, and S. Banerjee. Realtime behavior-

based continual authentication of users in virtual reality environments. In *AIVR*. IEEE, New York, USA, 2019.

[41] R. Miller, N. K. Banerjee, and S. Banerjee. Within-system and cross-system behavior-based biometric authentication in virtual reality. In *Proc. VRW*. IEEE, New York, USA, 2020.

[42] R. Miller, N. K. Banerjee, and S. Banerjee. Using siamese neural networks to perform cross-system behavioral authentication in virtual reality. In *Proc. VR*. IEEE, New York, USA, 2021.

[43] T. Mustafa, R. Matovu, A. Serwadda, and N. Muirhead. Unsure how to authenticate on your vr headset? come on, use your head! In *Proc. IWSPA*. ACM, New York, USA, 2018.

[44] S. Neumeier, N. Gay, C. Dannheim, and C. Facchi. On the way to autonomous vehicles teleoperated driving. In *Proc. AmE*. VDE, 2018.

[45] S. Neumeier, P. Wintersberger, A.-K. Frison, A. Becher, C. Facchi, and A. Riener. Teleoperation: The holy grail to solve problems of automated driving? sure, but latency matters. In *Proc. AutomotiveUI*. ACM, New York, USA, 2019.

[46] M. M. North, S. M. North, and J. R. Coble. Virtual reality therapy: an effective treatment for the fear of public speaking. *International Journal of Virtual Reality*, 3(3):1–6, Jan 1998.

[47] I. Olade, C. Fleming, and H.-N. Liang. Biomove: Biometric user identification from human kinesiological movements for virtual reality systems. *Sensors*, 20(10):2944, May 2020.

[48] I. Olade, H.-N. Liang, C. Fleming, and C. Champion. Exploring the vulnerabilities and advantages of swipe or pattern authentication in virtual reality (vr). In *Proc. ICVARS*. ACM, New York, USA, 2020.

[49] 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 *Proc. CHI*. ACM, New York, USA, 2019.

[50] G. Pizzi, D. Scarpi, M. Pichierri, and V. Vannucci. Virtual reality, real reactions?: Comparing consumers' perceptions and shopping orientation across physical and virtual-reality retail stores. *Computers in Human Behavior*, 96(0):1–12, Jul 2019.

[51] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147(0), Apr 2020.

[52] X. Shen, Z. J. Chong, S. Pendleton, G. M. J. Fu, B. Qin, E. Frazzoli, and M. H. Ang. Teleoperation of on-road vehicles via immersive telepresence using off-the-shelf components. In *Intelligent Autonomous Systems*, pp. 1419–1433. Springer, Berlin, Germany, 2016.

[53] A. J. Snoswell and C. L. Snoswell. Immersive virtual reality in health care: Systematic review of technology and disease states. *JMIR Biomedical Engineering*, 4(1), Jan-Dec 2019.

[54] R. Tilhou, V. Taylor, and H. Crompton. 3d virtual reality in k-12 education: A thematic systematic review. In *Emerging Technologies and Pedagogies in the Curriculum*, pp. 169–184. Springer, Berlin, Germany, 2020.

[55] P. Wang, P. Wu, J. Wang, H.-L. Chi, and X. Wang. A critical review of the use of virtual reality in construction engineering education and training. *International journal of environmental research and public health*, 15(6):1204, Jun 2018.

[56] S. Weise and A. Mshar. Virtual reality and the banking experience. *Journal of Digital Banking*, 1(2):146–152, Sep 2016.

[57] L. Xue, C. J. Parker, and H. McCormick. A virtual reality and retailing literature review: Current focus, underlying themes and future directions. In *Augmented Reality and Virtual Reality*, pp. 27–41. Springer, Berlin, Germany, 2019.

[58] Z. Yu, H.-N. Liang, C. Fleming, and K. L. Man. An exploration of usable authentication mechanisms for virtual reality systems. In *Proc. APCCAS*. IEEE, New York, USA, 2016.