

# Within-System and Cross-System Behavior-Based Biometric Authentication in Virtual Reality

Robert Miller

Natasha Kholgade Banerjee

Sean Banerjee\*

Clarkson University, Potsdam, NY

## ABSTRACT

In this paper, we perform the first analysis of behavior-based authentication within and across multiple virtual reality (VR) systems. The proliferation of VR in everyday applications such as healthcare, manufacturing, education, and remote teleoperation necessitates an understanding of operation over multiple VR systems as it is likely that a user will utilize several systems and requiring the user to re-train to a new system is time consuming. We collect the first multi-system dataset for VR biometrics consisting of 46 users performing a ball-throwing interaction using the Oculus Quest, HTC Vive, and HTC Vive Cosmos. Each user provides 10 training samples on the first day and 10 test samples on the second day for each of the three devices. We collect position and orientation data for devices such as the headset, left hand controller, and right hand controller, as well as trigger positions for the dominant hand controller. Our approach uses pairwise matches between trajectory features to represent high intra-user consistency and inter-user discriminative capacity. We analyze within-system and cross-system authentication accuracy for the 41 right-handed users in our 46-user dataset over varying combinations of matches for features such as point positions, device orientations, linear and angular velocities, and trigger grab or release for the right hand controller, left hand controller, and headset. Our approach provides maximum within-system authentication accuracy of 97%, 91%, and 91% when test trajectories are compared to training trajectories for the Vive, Quest, and Cosmos respectively. We provide maximum accuracies of 58% for testing with Cosmos and training with Quest, 70% for testing with Cosmos and training with Vive, and 85% for testing with Quest and training with Vive.

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

## 1 INTRODUCTION

The rapid incorporation of virtual reality (VR) in non-recreational applications propagating everyday environments, such as healthcare [2, 13], education [9], and virtual remote teleoperation and driving [6, 8, 12], necessitates the development of novel authentication techniques that do not use traditional PIN and 2D/3D pattern-based approaches [3, 18]. PIN or pattern-based approaches do not provide continual authentication, as once a malicious user gains access to the PIN or pattern they have full access to the system until the credentials are changed. Therefore, an authentication approach that uses innate characteristics of user-behavior that cannot be readily replicated by a malicious user are essential for VR authentication. Recently, a number of approaches have been dedicated to the use of human behavior in performing authentication of users in VR environments [1, 5, 10, 11, 14], with one approach having demonstrated how behavior-based tracking can be combined with regular

password-based access for continual authentication to prevent malicious session hijacking [10]. However, these approaches have focused on authentication where training and test behavior are obtained using a single VR system, e.g., an Android-based smartphone with Google VR [11] or the HTC Vive [1, 5, 10, 14]. To enable VR-based authentication to be rapidly propagated to VR systems in a wide variety of domains with minimal effort, two important challenges need to be addressed—(i) generalizability of authentication approaches to multiple VR systems, and (ii) interoperability of authentication, where training data is obtained from one system and testing is performed using a different system.

In this work, we provide the first evaluation of behavior-based authentication techniques within and across VR systems to assess generalizability and interoperability. As part of this work, we have collected a dataset of 46 subjects throwing a ball at a target in a VR environment using the devices, i.e., the right hand controller, the left hand controller, and the headset, of three VR systems—the Oculus Quest, the Vive, and the Cosmos. The ball-throwing interaction used in this work was first proposed in Kupin et al. [5], and has been subsequently used in Ajit et al. [1] and Miller et al. [10]. The interaction represents an activity where an impostor may attempt to perfectly mimic the action of a genuine user, necessitating fine-grained analysis of intra-user similarities and inter-user differences. The 3 VR systems demonstrate differences in the shapes and weights of the hand controllers and headsets, and method of tracking used—2 external IR lighthouses for the Vive, 4 onboard cameras for the Quest, and 6 onboard cameras for the Cosmos. Our dataset is the largest to date in terms of subject counts and number of VR systems.

Our work draws inspiration from approaches that use pairwise relationships between trajectories through fine-grained geometric analysis to discriminate whether trajectories belong to the same or to different users [1, 5, 10]. The work of Kupin et al. [5] uses matches for a single feature, i.e., position along a trajectory, to authenticate users. Ajit et al. [1] provide an improvement by combining matches for position and orientation features for the headset and hand controllers and analyze authentication performance for 21 feature combinations, and Miller et al. [10] provide a realtime implementation for the work of Ajit et al. [1] with discrimination of impostors internal and external to an organization. In order to provide higher assurance on accuracy for cross-device authentication analysis, our work provides the following novel contributions: (i) in addition to the position and orientation, we introduce matches between features corresponding to the linear and angular velocity over the trajectory to represent differences across users in rate of action performance, (ii) we incorporate control of the dominant hand controller trigger during release of the ball as a feature to model the differences in release timings across users, and (iii) we analyze 8,192 combinations of device-feature matches, where we test cross-matches not tested in Ajit et al. [1] such as one feature for one device combined with a separate feature for a different device, e.g., right hand velocity, left hand position, and headset orientation. We show authentication results for the 41 right-handed users in our 46-user dataset. For within-system authentication, our work provides a maximum authentication accuracy of 97% for the Vive, and 91% for the Quest and Cosmos. For cross-system authentication, we obtain a

\* e-mail: {romille,nbanerje,sbanerje}@clarkson.edu

maximum accuracy 58% for testing with Cosmos and training with Quest, 70% for testing with Cosmos and training with Vive, and 85% for testing with Quest and training with Vive.

## 2 RELATED WORK

PIN and 2D/3D pattern-based approaches [3, 18] have been explored for providing user authentication in VR environments. None of these approaches provide seamless continual authentication in VR, as a user teleoperating a real world device in VR cannot pause their interactions to enter a PIN or 2D/3D password. Head movement patterns as users react to a musical stimuli [7], respond to questions [17], observe rapidly changing images [15], and follow virtual balls [11] have also been evaluated for performing VR-based authentication using the Google Glass [7, 15, 17] or Google Cardboard [11]. However, these approaches use a single device, i.e., the headset, and may not be directly extensible to multi-device systems where users perform interaction with hand controllers in addition to visualizing content in the headset. Prior approaches also fail to provide insights on how authentication performance changes when the user performs the same task using a different VR system.

The work of Pfeuffer et al. [14] uses more generalized VR actions, such as grabbing, typing, walking, and pointing, and the authors report an overall accuracy of 44.44% using a dataset of 22 subjects using random forest and support vector machine classifiers by computing higher-level features such as maximum, minimum, standard deviation, and mean from the hand controller and headset motion, position, orientation, and velocity. Higher-level features may lose the fine-grained trajectory level detail that can be used to authenticate users, and require larger amounts of data to prevent overfitting. Our approach uses 10 user trajectories for training, and represents the sparse data provided by a user in real-world situations.

Our work is most closely related to the work of Kupin et al. [5], Ajit et al. [1], and Miller et al. [10]. Kupin et al. [5] provide the first task-oriented approach for authenticating users in VR environments using the positional information obtained from the right hand controller. Their technique uses bounding box centering and a nearest neighbor matching approach to achieve an overall accuracy of 92.86% for 14 subjects performing a ball-throwing task. However, their work does not use orientation or trigger control data from the right controller or information from the left controller and headset, which provide an understanding of user attention and recessive hand behavior. Work by Ajit et al. [1] combine positional and orientation information obtained from the hand controllers and headset to perform authentication on a dataset of 33 subjects. The authors demonstrate that bounding box centering and nearest neighbor matching using orientation data from the headset and right controller provides the highest accuracy of 93.03%. Miller et al. [10] provide a real-time version of Ajit et al. [1] with threshold-based discrimination between impostors within and outside the training set.

While our approach uses the same task and experimental setup as Kupin et al. [5], Ajit et al. [1], and Miller et al. [10], our work makes the following novel contributions over the prior approaches: we introduce linear and angular velocity features to analyze the effect of differences in speed of performance across users in authentication accuracy, we capture trigger control information from the dominant controller, and we exhaustively analyze  $2^{13}$  or 8,192 subsets of 13 feature matches in contrast to Ajit et al. [1] who only analyze 21 feature subsets. We evaluate our approach on 2 additional VR systems not used in Kupin et al., Ajit et al., or Miller et al., i.e., the Quest and the Cosmos, and we also test cross-system authentication. Our work is the first to provide evaluation of task driven biometric authentication approaches using multiple VR systems, and the first to perform cross-system VR authentication. Our dataset is the current state-of-the-art VR biometrics dataset with 46 subjects (41 right-handed) using 3 VR systems across two distinct sessions, and is the largest in number of subjects and systems, in comparison to prior

work, with 14 users in Kupin et al. [5], 22 users in Pfeuffer et al. [14], 23 users in Mustafa et al. [11], and 33 users in Ajit et al. [1].

## 3 DATASET

Our dataset is collected using three off-the-shelf VR systems, namely the Oculus Quest, HTC Vive, and HTC Vive Cosmos. While each system comes with a headset, left hand controller, and right hand controller, the user tracking characteristics of each system are different. The Quest is a standalone system that uses 4 onboard cameras to track the user. The Vive is a tethered system that tracks the user using 2 external lighthouses. The Cosmos is also tethered system and enables tracking using either the 6 onboard cameras or the Vive external lighthouses. For this work, we track the user using the 6 onboard cameras for the Cosmos as external lighthouse-based tracking has not been released at the time of submission. We capture data at 45 fps for the Vive and Cosmos, and 75 fps for the Quest.

To enable comparisons to prior work in task-based VR authentication, we adapt the VR ball-throwing interaction developed by Kupin et al. [5] in Unity. Figure 1(a) shows the interaction performed by a subject and their view from within the headset. The subject stands on an X on the floor, lifts a ball of radius 0.125 m off a pedestal of height 0.6 m located 1.1 m away from the center of the X, and throws the ball at a target of radius 1.5 m distant 9.9 m from the X, with the target's bottom being 0.4 m off the ground. The mass, angular drag, and velocity multiplier on release for the ball object are set to 1.0, 0.05, and 1.4 units in Unity respectively. In addition to recording the position and orientation of the left hand controller, right hand controller, and headset, we also record the status of the trigger on the dominant hand controller. During the interaction, the user uses the trigger of their dominant hand to pick up and release the ball from the pedestal. Similar to Kupin et al. [5], Ajit et al. [1], and Miller et al. [10], we collect 10 training throws and 10 testing throws for each user on distinct days separated by a minimum of 24 hours. The user uses the Quest on the first 2 days, Vive on their next 2 days, and Cosmos on the last 2 days. The user is given 3 seconds to complete each throw. We capture 135 trajectory points using the Vive and Cosmos, and 225 points using the Quest.

Before performing data collection, we asked each subject to rate their prior experience in throwing sports, provide the type of throwing sport they play or have played, rate their prior familiarity with VR, and provide the types of VR devices used if not none. We also asked users to provide their current weight and height, and recorded the type of clothing being worn. Our dataset consists of 42 male and 4 female subjects. For this paper, we only use the 41 right-handed subjects in the dataset. Upon retaining only right-handed subjects our final dataset has 38 male and 3 female subjects. We exclude left-handed subjects due to the lower number of left-handed subjects in the dataset. We give all users 2 practice throws using each of the three VR systems to enable them to familiarize themselves with the process of lifting the ball from the pedestal and throwing it at the target. Users are asked to make their best effort to make contact with the target, however they are not penalized for missing the target.

All subjects for the study were recruited from the faculty, staff, and student body of a small, predominantly undergraduate academic institution in North America. Our dataset was captured in two phases. During the first phase 19 subjects provided data for the Quest and Vive as the Cosmos was not available in the market at the time. In the second phase, after the release of the Cosmos, data was collected on the Cosmos for the original 19 subjects and an additional 27 subjects were recruited to provide data for the Quest, Vive, and Cosmos. Our total dataset consists of 46 subjects, of whom 41 are right handed. Figure 1(b) shows trajectories for 3 users who used all 3 systems.

## 4 APPROACH TO PERFORM AUTHENTICATION

Given a VR system being matched to, i.e., the system with training trajectories, and a VR system that is being matched, i.e., the system

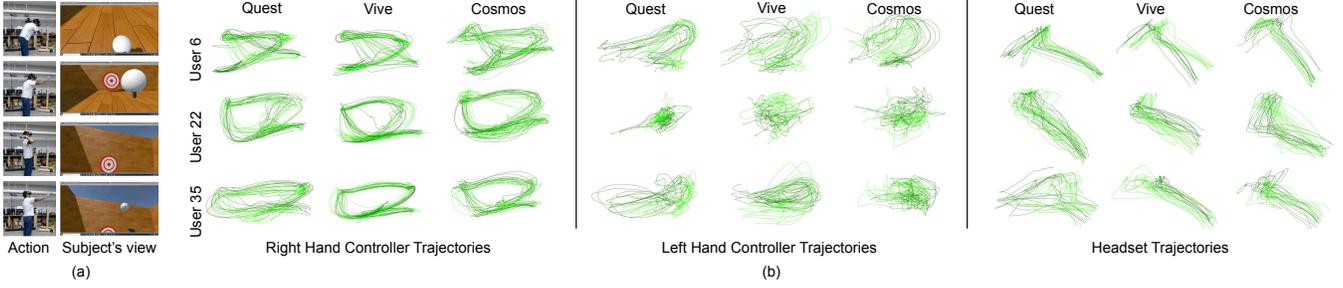


Figure 1: (a) System setup. (b) Examples of device trajectories for 3 users using the Quest, Vive, and Cosmos for the hand controllers and headset. Light green trajectories represent those from the first session, and dark green from the second session.

with test trajectories, our approach performs authentication for a user using the test VR system by extracting match values for features from the test trajectories of the user to the training trajectories for all users who have used the training VR system. In the results in Section 5, we show within-system authentication for each of the 3 VR systems, where we match users' test trajectories for the VR system captured on a test day against users' training trajectories for the same VR system captured on an alternate earlier training day. We also show cross-system results, where trajectories for users using a VR system at a later date are compared as test to those for users using a different VR system at an earlier date as training. To obtain an authentication result for an input user using a set of their device trajectories in the test VR system, our approach extracts matches between features over the device trajectories to features over the device trajectories in the training VR system for all users using the matching approach discussed in Subsection 4.1. Our approach combines the matches using a perceptron that outputs a confidence value as discussed in Subsection 4.2. The training user that provides the highest perceptron confidence is labeled as the authentication result for the input user. The perceptron weights are learned as discussed in Subsection 4.2 by using the match extraction discussed in Subsection 4.1 to compute matches amongst training trajectories alone for the training VR system.

#### 4.1 Feature Match Extraction for Trajectory Proximity

Following the findings from the approach of Ajit et al. [1], we use nearest neighbor point position matching to identify point correspondences across two device trajectories  $\mathcal{T}_1$  and  $\mathcal{T}_2$ , between which a match is to be computed. Once correspondences are estimated, our approach estimates matches between position, orientation, velocity, angular velocity, and trigger features (where applicable) for each device, i.e., the right hand controller, left hand controller, and headset. Small translational mis-alignments may exist between trajectories of the same user as they may for instance unintentionally move their right hand or head slightly between throw trials. To correct the mis-alignments, we align the trajectories prior to computing the correspondences by subtracting out the center of the bounding box, determined by Ajit et al. to yield highest accuracies. We use nomenclature similar to Ajit et al., where  $\mathbf{p}_\diamond[i] \in \mathbb{R}^3$ ,  $\mathbf{q}_\diamond[i] \in \mathbb{SO}(3)$ ,  $\mathbf{v}_\diamond[i] \in \mathbb{R}^3$ ,  $\omega_\diamond[i] \in \mathbb{SO}(3)$ , and  $c_\diamond[i] \in \mathbb{R}$  respectively represent the  $i^{\text{th}}$  values for aligned 3D point positions, 3D point orientations, 3D linear velocities, 3D angular velocities, and trigger control in corresponding time series  $\mathbf{p}_\diamond$ ,  $\mathbf{q}_\diamond$ ,  $\mathbf{v}_\diamond$ ,  $\omega_\diamond$ , and  $c_\diamond$  for the trajectory  $\mathcal{T}_\diamond$ , where  $\diamond$  is either '1' for the first trajectory or '2' for the second trajectory, and  $i \in \{1, 2, \dots, N_\diamond\}$ . The 3D point positions and the orientation quaternions are directly obtained from each VR system during the data collection in Section 3.

*Orientation Correction.* If the test and training trajectories come from different VR systems, device orientations, especially of the hand controllers, may differ across the two VR systems. In this

work, we perform a simple correction where rather than working with absolute orientations, we compute the match between relative orientations expressed with respect to the first orientation in the series. Each orientation time series  $\mathbf{q}_\diamond$  is re-expressed as

$$\mathbf{q}_\diamond[i] \leftarrow \mathbf{q}_\diamond[1]^{-1} \mathbf{q}_\diamond[i], \forall i \in \{1, 2, \dots, N_\diamond\} \quad (1)$$

where  $\mathbf{q}_\diamond[1]$  is the first element in the time series. The products are performed using quaternion multiplication.

*Correspondence Computation.* For the  $i^{\text{th}}$  and  $j^{\text{th}}$  points on  $\mathcal{T}_1$  and  $\mathcal{T}_2$  respectively, the indices of their nearest neighbors on the other trajectory, i.e.,  $j_i$  on  $\mathcal{T}_2$  and  $l_k$  on  $\mathcal{T}_1$  are given as

$$j_i = \arg \min_{j \in \{1, 2, \dots, N_2\}} \|\mathbf{p}_1[i] - \mathbf{p}_2[j]\|^2, \text{ and} \\ l_k = \arg \min_{l \in \{1, 2, \dots, N_1\}} \|\mathbf{p}_2[k] - \mathbf{p}_1[l]\|^2. \quad (2)$$

*Position Matches.* We obtain match between 3D point positions  $d_{\mathbf{p}}^*(\mathbf{p}_1, \mathbf{p}_2)$  across the trajectories  $\mathcal{T}_1$  and  $\mathcal{T}_2$  by summing distances between each point on one trajectory and the nearest point on the second trajectory, i.e., as

$$d_{\mathbf{p}}^*(\mathbf{p}_1, \mathbf{p}_2) = \sum_{i=1}^{N_1} \|\mathbf{p}_1[i] - \mathbf{p}_2[j_i]\|^2 + \sum_{k=1}^{N_2} \|\mathbf{p}_2[k] - \mathbf{p}_1[l_k]\|^2. \quad (3)$$

In Equation (3),  $\star$  is a placeholder for a device, and may be replaced with 'r' for the right hand controller, 'l' for the left hand controller, and 'h' for the headset. For symmetry, we obtain distances from  $\mathcal{T}_1$  to  $\mathcal{T}_2$ , as well as from  $\mathcal{T}_2$  to  $\mathcal{T}_1$ .

*Orientation Matches.* We obtain the match  $d_{\mathbf{q}}^*(\mathbf{q}_1, \mathbf{q}_2)$  between two orientation quaternion time series  $\mathbf{q}_1$  and  $\mathbf{q}_2$  using the inner product [4] adjusted to serve as a distance, i.e., as

$$d_{\mathbf{q}}^*(\mathbf{q}_1, \mathbf{q}_2) = \sum_{i=1}^{N_1} (1 - \langle \mathbf{q}_1[i], \mathbf{q}_2[j_i] \rangle)^2 \\ + \sum_{k=1}^{N_2} (1 - \langle \mathbf{q}_2[k], \mathbf{q}_1[l_k] \rangle)^2. \quad (4)$$

*Linear Velocity Matches.* Given the position time series  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , we obtain linear velocity time series  $\mathbf{v}_1$  and  $\mathbf{v}_2$  by first setting the  $i^{\text{th}}$  value in each trajectory  $\mathbf{v}_\diamond$  to be

$$\mathbf{v}_\diamond[i] = (\mathbf{p}_\diamond[i+1] - \mathbf{p}_\diamond[i]) f_\diamond, \forall i \in \{1, 2, \dots, N_\diamond - 1\}, \quad (5)$$

where  $f_\star$  refers to the frame rates of the respective VR system, and then obtaining the matches between the linear velocity time series  $d_{\mathbf{v}}(\mathbf{v}_1, \mathbf{v}_2)$  using the Euclidean distance as

$$d_{\mathbf{v}}^*(\mathbf{v}_1, \mathbf{v}_2) = \sum_{i=1}^{N_1-1} \|\mathbf{v}_1[i] - \mathbf{v}_2[j_i]\|^2 \\ + \sum_{k=1}^{N_2-1} \|\mathbf{v}_2[k] - \mathbf{v}_1[l_k]\|^2. \quad (6)$$

*Angular Velocity Matches.* We obtain the angular velocity time series  $\omega$  for the first trajectory  $\mathcal{T}_1$  by performing finite differencing

over the quaternion time series  $\mathbf{q}_1$ . The  $i^{\text{th}}$  value in the angular velocity time series  $\omega_1[i]$  is represented as

$$\omega_1[i] = 2 \ln(\mathbf{q}_1[i+1] \mathbf{q}_1[i]^*), \quad (7)$$

where the symbol  $*$  represents the quaternion conjugate. Given that  $\mathbf{q}_1$  and  $\mathbf{q}_2$  may come from VR systems with different frame rates, we match the frame rate of the second VR system to that of the first by estimating an interpolated quaternion  $\hat{\mathbf{q}}_2[i]$  between  $\mathbf{q}_2[i]$  and  $\mathbf{q}_2[i+1]$  spaced at the same time interval from  $\mathbf{q}_2[i]$  as  $\mathbf{q}_1[i+1]$  is from  $\mathbf{q}_1[i]$ . To ensure interpolation rather than extrapolation, we assume that the frame rate  $f_2$  of  $\mathcal{T}_2$  is lower than that of  $\mathcal{T}_1$ , i.e., than  $f_1$ . In case  $f_2 > f_1$ , we swap  $\mathcal{T}_1$  and  $\mathcal{T}_2$  prior to the start of match computation. Given the frame rates  $f_1$  and  $f_2$  of  $\mathcal{T}_1$  and  $\mathcal{T}_2$ , we calculate the interpolating coefficient  $\alpha$  as  $f_2/f_1$ . We obtain the value of  $\hat{\mathbf{q}}_2[i]$  through spherical linear interpolation as

$$\hat{\mathbf{q}}_2[i] = \mathbf{q}_2[i] \left( \mathbf{q}_2[i]^{-1} \mathbf{q}_2[i+1] \right)^\alpha. \quad (8)$$

Then the  $i^{\text{th}}$  value in the second angular velocity time series  $\omega_2[i]$  can be represented as

$$\omega_2[i] = 2 \ln(\hat{\mathbf{q}}_2[i] \mathbf{q}_2[i]^*). \quad (9)$$

We express the match between the angular velocity time series  $d_\omega(\omega_1, \omega_2)$  using the sum of adjusted dot products between angular velocities, similar to the quaternions in Equation (4), i.e., as

$$d_\omega^*(\omega_1, \omega_2) = \sum_{i=1}^{N_1-1} (1 - \langle \omega_1[i], \omega_2[j_i] \rangle^2) + \sum_{k=1}^{N_2-1} (1 - \langle \omega_2[k], \omega_1[l_k] \rangle^2). \quad (10)$$

*Trigger Control Matches.* Each VR system provides trigger control values where 0 represents a full release of the trigger below the hand controller, 1 represents a full grab, and a value between 0 and 1 represents the trigger at an intermediate state of grab. We use sum-squared distance to measure the match  $d_c(c_1, c_2)$  between trigger control values between corresponding points across the trigger control time series  $c_1$  and  $c_2$  of the two trajectories as

$$d_c^*(c_1, c_2) = \sum_{i=1}^{N_1} (c_1[i] - c_2[j_i])^2 + \sum_{k=1}^{N_2} (c_2[k] - c_1[l_k])^2. \quad (11)$$

For the trigger control match  $\star$  can only be ‘r’, given that trigger values is only available for the right hand controller.

## 4.2 Match Combination

Using the matching approach discussed in Section 4.1, we obtain a set of five matches for the right hand controller, i.e.,  $d_p^r$ ,  $d_q^r$ ,  $d_v^r$ ,  $d_\omega^r$ , and  $d_c^r$ , a set of four matches for the left hand controller, i.e.,  $d_p^l$ ,  $d_q^l$ ,  $d_v^l$ , and  $d_\omega^l$ , and a set of four matches for the headset, i.e.,  $d_p^h$ ,  $d_q^h$ ,  $d_v^h$ , and  $d_\omega^h$ , providing 13 matches in total for two sets of trajectories. We use the weights of a perceptron [16] as done in Ajit et al. [1] to combine the matches into a single output.

*Training Phase.* We use a training phase to learn the weights of the perceptron and a bias value by extracting 13 matches per pair of training trajectories belonging the training VR system, and optimizing the perceptron’s loss function so that a pair of trajectories belonging to the same user is encouraged to provide a high confidence, while a pair where the trajectories belonging to different users provides a low confidence. Ground truth confidences are set to 1 for pairs coming from the same user, and 0 for pairs where each trajectory belongs to different users. Given  $m$  trajectories per user in the training set for  $P$  users, we perform training with a total of  $(Pm)^2$  trajectory pairs, with  $Pm^2$  trajectories belonging to the same user and  $P(P-1)m^2$  trajectories belonging to different users.

*Test Phase.* At test time, given the 3 device trajectories for an input user performing a single action using the test VR system, we

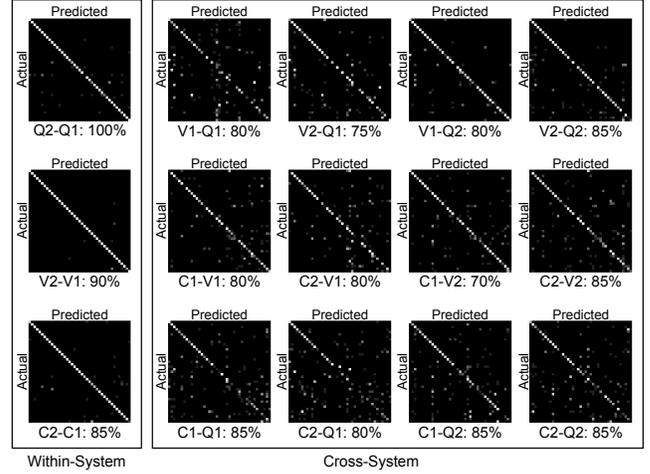


Figure 2: Confusion matrices for within-system and cross-system authentication for the highest accuracies per trajectory percentage.

compute the 13 feature matches for the user against all trajectories for every user in the dataset for the training VR system. We pass the matches through the perceptron, and we use the user corresponding to the training set of device trajectories that provide highest confidence to label the input user’s action as the authentication result. Results in Section 5 are reported by averaging over the accuracy of authentication reported for all 10 actions for each of the 41 users.

*Feature match subsets.* As discussed in Ajit et al. [1], not all feature matches may contribute to high accuracy. In particular, they demonstrate that removing certain features improves accuracy, likely attributed to the reduction in goal-orientedness by the retention of poor performing features. We determine the effectiveness of the feature matches by exhaustively analyzing all  $2^{13}$  or 8,192 subsets of feature match combinations from the 13 matches.

## 5 RESULTS

For all results, we test with data that has been provided later in time than the training data to maintain temporal continuity. Tables 1 and 2 provide the highest accuracies for the best feature set chosen over 8,192 feature combinations for within-system and cross-system training. Accuracies are reported for using the entire trajectory, i.e., 100% of the points, and using smaller portions of the trajectory to determine the effect of removal of later behavior. In the ‘Type’ column, the letter represents the name of the system (Q=Quest, V=Vive, and C=Cosmos), the numeric value represents the session number, and the nomenclature is test-train. The highest within-system accuracies are obtained using 90% of the points, while highest cross-system accuracies are observed using 80% of the points.

Highest within-system accuracies are at 0.91 for the Quest and Cosmos, and reach 0.97 for the Vive, with minimal change when the trajectory sizes are reduced. Cross-system accuracies are lower. Accuracies for comparing Vive as test against Quest as training peak at 0.85. While accuracies drop when the Cosmos is tested against the Vive, the drop may occur due to the Cosmos being a camera-based system where camera inaccuracies contribute to significant trajectory variability, and the Vive being a lighthouse-based system where trajectories show a limited range of variability. Accuracies for comparing the Cosmos to the Vive peak at 0.70. The maximum accuracy for comparing the Cosmos to the Quest is 0.58. The reason for low Cosmos-to-Quest accuracy may be due to matching between two camera-based systems which show high variability, where the nature of variability varies from system to system, and due to loss of information when the controllers move out of the field of view of

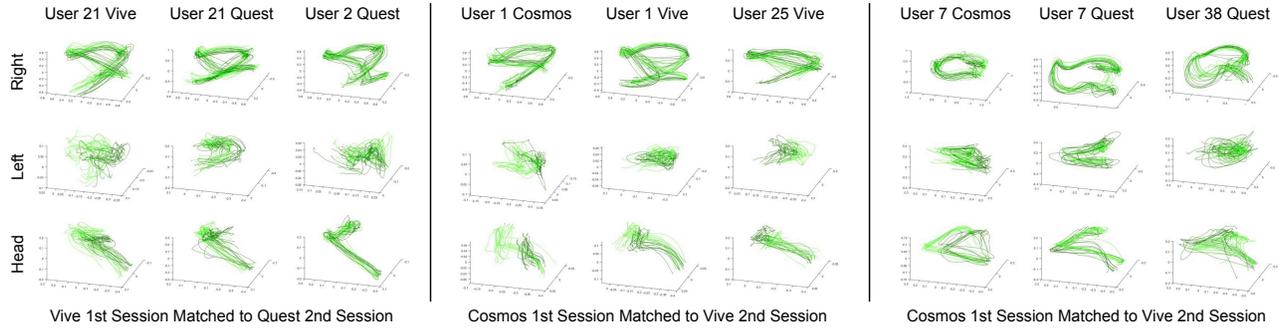


Figure 3: Users that misclassify (dark green trajectories of left columns of each panel) for a given VR system, and test user (light green trajectories of center columns) and confounding user (light green trajectories of right columns) for training VR system.

the cameras or when they move at too high tracking velocities.

Tables 1 and 2 demonstrate that features that have the highest contributions toward accuracy include head orientation, left position, right orientation, and right position, occurring 58, 55, 45, and 36 times respectively in the tables. Ajit et al. [1] demonstrate that head and right controller orientation play a significant role, however, we demonstrate that when device positions are included, the positions of both hand controllers also contribute to high accuracy. The next highest contributions are from the right controller angular velocity, right controller trigger, linear velocity of the headset, left controller angular velocity, and left controller orientation at counts of 30, 28, 26, 25, and 23 respectively. The right angular velocity’s contribution matches that of the right orientation, while headset velocity may indicate that motion patterns of the head have a contribution. It is likely that users preserved a preferred twist in the wrist of their non-dominant hands, which may explain the contribution of the left orientation features. The remaining features, especially linear velocities for hand controllers, have limited contribution, indicating that users may have variations in controller velocities over the trajectory.

Figure 2 shows the confusion matrices for best performing percentages of trajectory points from Tables 1 and 2 per system-to-system match. Figure 3 shows the trajectories of the users which demonstrate misclassifications in cross-system authentication, together with the users they misclassify most with. For User 21, the Vive right trajectory demonstrates an inward curve in the lift phase, while their Quest trajectory shows an outward curve, causing user 21 to match to user 2 instead. User 1’s Cosmos trajectories demonstrate a similar issue. For User 7’s Cosmos trajectory, the ring shape is more concise, while their Quest trajectory has the ring spread out. These differences may be attributed to the effect of variation in controller mass, handle length, and center of gravity, on the user’s action performance. The headset motion for User 7 in the Cosmos captures a curve not represented by the Quest. While orientation features tend to be prominent, the point positions also play a role in correspondence computation, due to which there may exist regions of the curve for which meaningful correspondences cannot be obtained.

## 6 DISCUSSION

In this work, we provide an evaluation of within-system and cross-system accuracy for performing authentication in VR environments. Within-system analyses have been performed by testing second session data against first session data for the corresponding system, while cross-system analyses have been performed by comparing data collected using VR systems used later against data collected using VR systems used earlier. We show that user characteristics are represented by combining matches from the right controller orientation, left controller position, and headset orientation. We show high within-system accuracy, around 0.90 and higher, and while cross-system accuracies are lower, we observe that higher cross-system

Table 1: Maximum accuracy (‘Acc’) for within-system and cross-system authentication, together with best features, for 100% and 90% points (‘%Pts’). Under ‘Type’, systems are listed as test-train, Q = Quest, V = Vive, C = Cosmos, 1 = 1st session, 2 = 2nd session.

%Pts	Type	Acc	Right			Left			Head					
			p	q	ω	v	p	q	ω	v	p	q	ω	v
100	<b>Q2-Q1</b>	0.91	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V2-V1</b>	0.97	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C2-C1</b>	0.90	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V1-Q1</b>	0.68	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V2-Q1</b>	0.60	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C1-Q1</b>	0.42	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C2-Q1</b>	0.42	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V1-Q2</b>	0.83	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V2-Q2</b>	0.71	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C1-Q2</b>	0.51	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C2-Q2</b>	0.45	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C1-V1</b>	0.66	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C2-V1</b>	0.62	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C1-V2</b>	0.63	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>C2-V2</b>	0.59	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
90	<b>Q2-Q1</b>	0.91	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V2-V1</b>	0.97	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C2-C1</b>	0.91	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V1-Q1</b>	0.69	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V2-Q1</b>	0.61	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C1-Q1</b>	0.47	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C2-Q1</b>	0.42	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V1-Q2</b>	0.84	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>V2-Q2</b>	0.73	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C1-Q2</b>	0.57	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C2-Q2</b>	0.52	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C1-V1</b>	0.68	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C2-V1</b>	0.63	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	<b>C1-V2</b>	0.65	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>C2-V2</b>	0.59	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

accuracies are obtained when a lighthouse-based system such as the Vive is used in testing, with accuracies reaching 0.85. Improvement over prior work [1, 5] is attributed to incorporation of velocity and trigger features, and exploration of 8,192 feature sets.

As part of future work, we will develop approaches to model the quantity and nature of the variability in VR systems that use lighthouse-based and camera-based tracking, as well as the effect of device geometry and mass distribution, so as to improve the accuracy

Table 2: Maximum accuracy ('Acc') for within-system and cross-system authentication, together with best features, for 85% and 80% points ('%Pts'). Under 'Type', systems are listed as test-train, Q = Quest, V = Vive, C = Cosmos, 1 = 1st session, 2 = 2nd session.

%Pts	Type	Acc	Right				Left				Head					
			p	q	$\omega$	v	p	q	$\omega$	v	p	q	$\omega$	v		
85	Q2-Q1	0.90	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓
	V2-V1	0.97	✓		✓			✓	✓					✓	✓	✓
	C2-C1	0.91	✓	✓			✓	✓						✓	✓	✓
	V1-Q1	0.69	✓		✓	✓	✓							✓	✓	✓
	V2-Q1	0.62		✓	✓		✓							✓	✓	✓
	C1-Q1	0.49	✓			✓	✓							✓	✓	✓
	C2-Q1	0.45	✓	✓			✓							✓	✓	✓
	V1-Q2	0.83	✓	✓			✓	✓						✓	✓	✓
	V2-Q2	0.75	✓	✓			✓		✓					✓	✓	✓
	C1-Q2	0.58	✓		✓		✓	✓						✓	✓	✓
	C2-Q2	0.52	✓		✓		✓	✓	✓					✓	✓	✓
	C1-V1	0.69	✓	✓	✓		✓	✓	✓					✓	✓	✓
	C2-V1	0.56	✓	✓	✓		✓							✓	✓	✓
	C1-V2	0.65	✓	✓	✓		✓	✓	✓					✓	✓	✓
	C2-V2	0.61	✓	✓	✓		✓	✓						✓	✓	✓
80	Q2-Q1	0.90	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓
	V2-V1	0.97	✓		✓		✓	✓	✓					✓	✓	✓
	C2-C1	0.90	✓	✓	✓		✓	✓						✓	✓	✓
	V1-Q1	0.69	✓	✓			✓							✓	✓	✓
	V2-Q1	0.63	✓	✓			✓	✓						✓	✓	✓
	C1-Q1	0.48	✓		✓		✓							✓	✓	✓
	C2-Q1	0.46	✓				✓							✓	✓	✓
	V1-Q2	0.85	✓	✓			✓		✓					✓	✓	✓
	V2-Q2	0.74	✓	✓			✓		✓					✓	✓	✓
	C1-Q2	0.57	✓		✓		✓	✓						✓	✓	✓
	C2-Q2	0.55	✓				✓							✓	✓	✓
	C1-V1	0.70	✓	✓	✓		✓	✓	✓					✓	✓	✓
	C2-V1	0.64	✓	✓			✓							✓	✓	✓
	C1-V2	0.66	✓	✓	✓		✓	✓	✓					✓	✓	✓
	C2-V2	0.60	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓

of cross-system authentication. A potential approach is to analyze user behavior for a subset of users performing a standardized tasks in all systems, and extrapolating behavior to other users and to tasks with similar trajectories to the standardized tasks. Our analysis will incorporate determining when the controllers move outside the tracking region of the cameras for camera-based systems, and to weight down points in high-velocity regions. We have shown results for the 41 right-handed users in our 46 user set, however, we are interested in developing techniques to provide high authentication accuracy when left-handed users are included in the system. One method is to build front-end classifiers that distinguish users as left and right-handed based on quantity of controller motion, and then use handedness-specific authentication to identify users. Our work analyzes authentication accuracy for user identification, however, as future work, we will explore comparison metrics that enable the high-assurance classification of impostors outside the training set.

Our work performs authentication analysis for a ball-throwing action, which while simple, represents a task with high potential for mimicry from an impostor, thereby necessitating an understanding of feature sets that can protect against intrusion. As part of future work, we are interested in determining the extensibility of the matching approach discussed in this work, and the features used to other actions, for use in a wide range of applications where protection from intrusion is essential. Actions such as pointing, drone flying, or driving may involve limited positional change, however, may involve extensive change in wrist and headset orientation as part

of navigation. For actions involving exploring a virtual room to, e.g., gain access to privileged information, high accuracy may be obtained by tracking features representing the bobbing motions of the head during walking such as position of linear velocity, or those representing rummaging using the dominant hand, such as dominant hand velocity, position, and orientation.

## ACKNOWLEDGMENTS

This work was partially supported by NSF grant #1730183. We thank Ashwin Ajit and Damon Gwinn for assistance with data collection.

## 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, NY, USA, 2019.
- [2] 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), Apr. 2016.
- [3] 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*. The Internet Society, Reston, USA, 2017.
- [4] D. Q. Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2), Oct. 2009.
- [5] 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.
- [6] M. Lager and E. A. Topp. Remote supervision of an autonomous surface vehicle using virtual reality. *IFAC*, 52(8), July 2019.
- [7] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser. Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns. In *Proc. PerCom*. IEEE, NY, USA, 2016.
- [8] M. Maciaś, A. Dąbrowski, J. Fraś, M. Karczewski, S. Puchalski, S. Tabaka, and P. Jaroszek. Measuring performance in robotic teleoperation tasks with virtual reality headgear. In *Proc. Conf. Autom.* Springer, Berlin, Germany, 2019.
- [9] Z. Merchant, E. T. Goetz, L. Cifuentes, W. Keeney-Kennicutt, and T. J. Davis. Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis. *Computers & Education*, 70(1), Jan. 2014.
- [10] R. Miller, A. Ajit, N. K. Banerjee, and S. Banerjee. Realtime behavior-based continual authentication of users in virtual reality environments. In *Proc. AIVR*. IEEE, NY, USA, 2019.
- [11] 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, NY, USA, 2018.
- [12] 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, NY, USA, 2019.
- [13] 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), Jan. 2015.
- [14] 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, NY, USA, 2019.
- [15] C. E. Rogers, A. W. Witt, A. D. Solomon, and K. K. Venkatasubramanian. An approach for user identification for head-mounted displays. In *Proc. ISWC*. ACM, NY, USA, 2015.
- [16] F. Rosenblatt. *The perceptron, a perceiving and recognizing automaton*. Cornell Aeronautical Laboratory, 1957.
- [17] S. Yi, Z. Qin, E. Novak, Y. Yin, and Q. Li. Glassgesture: Exploring head gesture interface of smart glasses. In *Proc. INFOCOM*. IEEE, NY, USA, 2016.
- [18] 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, NY, USA, 2016.