

Combining Pairwise Feature Matches from Device Trajectories for Biometric Authentication in Virtual Reality Environments

Ashwin Ajit, Natasha Kholgade Banerjee, and Sean Banerjee
Clarkson University, Potsdam, NY
{ajita,nbanerje,sbanerje}@clarkson.edu

Abstract—In this paper we provide an approach to perform seamless continual biometric authentication of users in virtual reality (VR) environments by combining position and orientation features from the headset, right hand controller, and left hand controller of a VR system. The rapid growth of VR in mission critical applications in military training, flight simulation, therapy, manufacturing, and education necessitates authentication of users based on their actions within the VR space as opposed to traditional PIN and password based approaches. To mimic goal-oriented interactions as they may occur in VR environments, we capture a VR dataset of trajectories from 33 users throwing a ball at a virtual target with 10 samples per user captured on a training day, and 10 samples on a test day. Due to the sparseness in the number of training samples per user, typical of realistic interactions, we perform authentication by using pairwise relationships between trajectories. Our approach uses a perceptron classifier to learn weights on the matches between position and orientation features on two trajectories from the headset and the hand controllers, such that a low classifier score is obtained for trajectories belonging to the same user, and a high score is obtained otherwise. We also perform extensive evaluation on the choice of position and orientation features, combination of devices, and choice of match metrics and trajectory alignment method on the accuracy, and demonstrate a maximum accuracy of 93.03% for matching 10 test actions per user by using orientation from the right hand controller and headset.

Index Terms—virtual reality, VR, biometrics, seamless, continual authentication

I. INTRODUCTION

The behavior of a person has been used as a mechanism for biometric authentication in desktop environments using keyboard [17], [30], [31] and mouse movements [1], [10], in smart devices [21], [32], [26], [27], and more recently in virtual reality (VR) systems [11], [23], [18]. VR systems today are no longer solely used as recreational devices, and are rapidly gaining acceptance in applications such as military training [4], [22], [29], flight simulations [20], [6], therapy [14], [19], [7], manufacturing [2], [5], and education [8], [15]. Malicious users in mission critical applications can cause irreparable harm, and it is necessary to continually authenticate the user without the user needing to stop their activities. VR authentication techniques that use PIN and pattern matching methods [9], [34] are infeasible as once a PIN or password is compromised, a malicious user has complete access to the system. A periodic PIN or password entry system also intrudes on the usability of the VR system. For example, a pilot teleoperating a drone using a VR headset cannot pause the drone in the real world when the system needs to re-authenticate them. Behavior-based approaches for authenticating users in virtual

environments have used head motions and blink patterns [24], head movements to music [12], and bone conduction of sounds through the skull [25] using Google Glass. These approaches do not capture the broad range of actions that users may perform in virtual environments.

We provide an approach that enables seamless biometric authentication of users in virtual environments by combining position and orientation features obtained by tracking the trajectories of the VR headset, left hand controller, and right hand controller as a person performs VR interactions. The work of Mustafa et al. [18] performs authentication by using head motion data generated by a user following a ball using Google Cardboard without using hand controllers, which does not represent the full range of interactions that users are likely to perform in VR. Pfeuffer et al. [23] use classifiers such as random forests and SVMs with higher-level aggregated features such as the maximum, minimum, standard deviation, and mean obtained from low-level features such as device motion, orientation, velocity, distance in virtual space, and view direction. They provide authentication for motions such as pointing, grabbing, walking, and typing with the highest accuracy being 44.4%. In this work, we perform authentication while emulating a real-world VR scenario where a large quantity of annotated training data is unlikely to be obtainable for a range of users. We record the right hand controller, left hand controller, and headset for 33 subjects picking a ball in VR and throwing it at a target, where each subject provides a sparse set of 10 trajectories on a training day, and a set of 10 trajectories on a test day. We keep the days separate in order to prevent priming or muscle memory from repetitive throws from influencing authentication. Due to the sparseness of the training set, classification approaches that treat each sample independently, e.g., a supervised classifier that takes features from the sample as input and provides user identity as output as in Pfeuffer et al. [23], are unlikely to provide high accuracy, as low-level features such as position and orientation over time are likely to cause overfitting, while aggregated features may not capture enough fine-scale detail to represent inter-class uniqueness and intra-class consistency.

Rather than treat samples independently, our approach uses pairwise relationships between trajectories obtained using similarity matches that provide a low value when the trajectories belong to the same user and a high value when they are different. While the pairwise matches between trajectories for a single feature, i.e., the position of the right hand controller, have been used in Kupin et al. [11], our contribution is to enable the use of pairwise matches between multiple

features such as position and orientation of the right hand controller, left hand controller, and headset in authentication. Our approach estimates weights of a linear combination of the feature matches by training a perceptron classifier to learn that matches from two trajectories belonging to the same user should yield a low classifier score near 0, while matches from trajectories belonging to different users should yield a high score. We provide extensive analysis on the effect of using just position, just orientation, and both position and orientation, the influence of using various combinations of the left hand controller, right hand controller, and headset trajectories, and the contributions of various match metrics and trajectory alignment methods on the accuracy. In contrast to the approach of Kupin et al. [11], which provides an accuracy of 83.64% on our dataset of 33 subjects, we obtain a maximum accuracy of 93.03% for 10 test actions per user. This accuracy is attained when orientation alone for the right hand controller and headset are combined and matched using the distance between the nearest neighbors across both trajectories aligned by subtracting out their bounding box center.

II. RELATED WORK

Early approaches for authenticating users in virtual environments adopted approaches from mobile authentication by using PIN and 2D/3D pattern based methods [9], [34]. These approaches do not enable seamless continual user authentication as the user would need to stop their activity to enter their credentials using a PIN or password, nod to a question, observe changing images, or move to an external stimulus such as music. In the case of VR-based teleoperation [13], it is infeasible and can be catastrophic to pause the real world object being controlled to authenticate the user. To seamlessly authenticate the user such that the task of authentication does not take up the user's actions we must use the user's inherent behavior within the VR application to perform authentication.

In Rogers et al. [24], infrared, gyroscope, and accelerometer data from a Google Glass was used to authenticate users based on blink and head movement patterns as a series of rapidly changing images were displayed to the subject. Head movement patterns as users are stimulated by external audio has also been investigated as a method for VR authentication [12]. User head movements as responses to specific questions have also been investigated as an authentication scheme for Google Glass [33]. Work by Mustafa et al. [18] uses head movements to identify Google Cardboard users by having subjects follow virtual balls as they appear on the screen. A dataset of 23 subjects was collected with each subject following 25 balls during each session with a total of 15 sessions per user. The authors report an overall equal error rate of 7% using logistic regression. These approaches do not incorporate data from hand controllers, which play an essential role in realistic VR interactions such as teleoperating a robot [13].

Work by Pfeuffer et al. [23] uses tasks such as pointing, grabbing, walking, and typing from the head and hand controllers of an HTC Vive along with eye tracking data from a Pupil Labs eye tracker to identify users. A dataset of 22 subjects was collected with 260 trials per session per

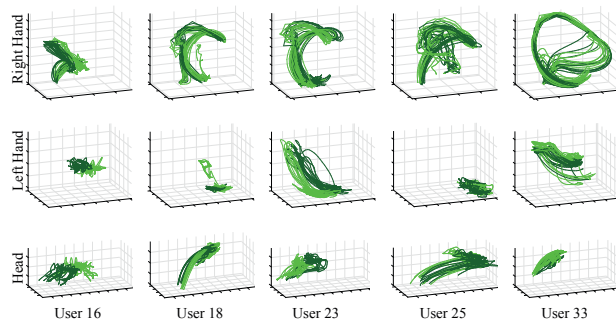


Fig. 1. Trajectories from the right hand controller, left hand controller, and headset on the first day (dark) and the second day (light) demonstrating variations between users, and consistency within user, for 5 of the 33 users in our dataset. Trajectories also show translational misalignments, which are best corrected in our work by subtracting the bounding box center.

subject for pointing, 260 for grabbing, 40 for walking, and 15 sentences with an average of 25.8 letters per sentence for typing. The authors exclude the 3 left handed subjects and report an overall accuracy of 44.44% by evaluating random forest and support vector machine classifiers using higher-level features of mean, maximum, minimum, and standard deviation computed from low-level features such as device motion, orientation, position, and velocity in the virtual space. To obtain higher accuracies from sparser training data of 10 samples per user in our work, we use pairwise matching techniques to express relationships between trajectories.

Work by Kupin et al. [11] uses a ball throwing task to authenticate users in virtual environments by using positional information of the dominant, i.e., right, hand controller and a nearest neighbor matching technique. They obtain an accuracy of 92.86% by comparing 10 throws for 14 subjects each on a test day to 10 throws on a training day. While we use the same task and experimental setup as Kupin et al. [11], we combine matches from position and orientation features from the headset and both hand controllers. The number of subjects in our dataset at 33 exceeds prior VR biometric datasets.

III. DATA COLLECTION

We utilize the ball throwing application developed by Kupin et al. [11] to collect data from 33 right-handed subjects using an HTC Vive system. We collect both 3D position and 3D orientation data for 135 time samples at 45 fps from the headset, left hand controller, and right hand controller. To prevent priming, we collect data in two sessions separated by a minimum of 24 hours with 10 throws collected during each session. Prior to the VR activity, we record the gender, prior experience in VR, and prior experience in throwing sports for each subject as part of our demographic data. As shown in Table I, we obtain an equal proportion of subjects with and without throwing sports experience and with no/limited and moderate/high VR experience. To reduce variability caused by subject motion, we asked each subject to stand on a white 'X' marked in the virtual space. We instructed them to lift a ball from a virtual pedestal and attempt to hit a virtual target marked on the wall in front of them. Figure 1 shows right

hand controller, left hand controller, and headset trajectories from 5 subjects in our dataset.

TABLE I
SUMMARY OF DEMOGRAPHICS FOR OUR DATASET. OUR DATASET CONTAINS AN EVEN SPLIT OF USERS WITH AND WITHOUT (NO/LIMITED) VR EXPERIENCE AS WELL AS EXPERIENCE IN THROWING SPORTS.

Number of subjects	33
Number of female subjects	7
Number of male subjects	26
Number of subjects with no VR experience	9
Number of subjects with limited VR experience	8
Number of subjects with moderate/high VR experience	16
Subjects with no throwing sports experience	17
Subjects with throwing sports experience	16

IV. TRANSLATIONAL ALIGNMENT OF TRAJECTORIES

While all trajectories start near a common spatial point, i.e., the location of the ball on the pedestal, differences in the extension of the arm of the user may induce translational offsets in the trajectories. We align each trajectory by subtracting out a center point computed from the trajectory. In Section VII, we compare the results of performing no subtraction, subtracting the mean of the points, i.e., the centroid, subtracting the center of the bounding box, and subtracting the centroid and bounding box center computed only for the x and z coordinates to preserve height differences in the y direction.

V. MATCH METRICS USED TO EVALUATE PROXIMITY BETWEEN TRAJECTORIES

Due to differences in wait times, corresponding points across the two trajectories do not yield a close match, as shown in Kupin et al. [11]. We use three types of distance metrics, discussed in Subsections V-A to V-C, to evaluate the proximity between two trajectories aligned to each other using one of the methods in Section IV. In Subsections V-A to V-C, $\mathcal{T}_*$ refers to a trajectory, $\mathbf{p}_*$ refers to a time series of 3D point positions for $\mathcal{T}_*$, while $\mathbf{q}_*$ refers to a time series of 3D point orientations for $\mathcal{T}_*$ expressed as unit quaternions. Here $*$ may be 1 for the first trajectory or 2 for the second trajectory. The i^{th} point on $\mathbf{p}_*$ or $\mathbf{q}_*$ is referred to using time series notation as $\mathbf{p}_*[i]$ or $\mathbf{q}_*[i]$ respectively, where $\mathbf{p}_*[i] \in \mathbb{R}^3$ and $\mathbf{q}_*[i] \in \mathbb{SO}^3$.

A. Match Between Nearest Trajectory Points

For the i^{th} point on the first trajectory $\mathcal{T}_1$, we estimate the nearest point on the second trajectory $\mathcal{T}_2$ by computing the sum-squared Euclidean distance between the i^{th} point position $\mathbf{p}_1[i]$ on $\mathcal{T}_1$ and the j^{th} point position $\mathbf{p}_2[j]$ on $\mathcal{T}_2$ as

$$d(\mathbf{p}_1[i], \mathbf{p}_2[j]) = \|\mathbf{p}_1[i] - \mathbf{p}_2[j]\|^2, \quad (1)$$

and obtaining the index of the nearest point j_i on $\mathcal{T}_2$ as

$$j_i = \arg \min_j d(\mathbf{p}_1[i], \mathbf{p}_2[j]). \quad (2)$$

To maintain symmetry, we similarly compute the index of the nearest point l_k on $\mathcal{T}_1$ to the k^{th} point on $\mathcal{T}_2$ as

$$l_k = \arg \min_l d(\mathbf{p}_2[k], \mathbf{p}_1[l]), \text{ where} \quad (3)$$

$$d(\mathbf{p}_2[k], \mathbf{p}_1[l]) = \|\mathbf{p}_2[k] - \mathbf{p}_1[l]\|^2.$$

The nearest-neighbor distance $d_p^{\text{nn}}(\mathbf{p}_1, \mathbf{p}_2)$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ in terms of point positions is obtained by adding individual nearest-neighbor distances for both trajectories over N points per trajectory as

$$d_p^{\text{nn}}(\mathbf{p}_1, \mathbf{p}_2) = \sum_{i=1}^N d(\mathbf{p}_1[i], \mathbf{p}_2[j_i]) + \sum_{k=1}^N d(\mathbf{p}_2[k], \mathbf{p}_1[l_k]). \quad (4)$$

Equation (4) is the same metric used for trajectory matching in Kupin et al. [11]. We obtain the nearest-neighbor match $d_o^{\text{nn}}(\mathbf{p}_1, \mathbf{p}_2)$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ in terms of orientations as

$$d_o^{\text{nn}}(\mathbf{p}_1, \mathbf{p}_2) = \sum_{i=1}^N (1 - \langle \mathbf{q}_1[i], \mathbf{q}_2[j_i] \rangle^2) + \sum_{k=1}^N (1 - \langle \mathbf{q}_2[k], \mathbf{q}_1[l_k] \rangle^2). \quad (5)$$

In Equation (5), the notation $\langle \cdot, \cdot \rangle$ refers to the cosine of the angle between the quaternions, which for close quaternions is high, i.e., near 1. The notation $1 - \langle \cdot, \cdot \rangle^2$ enables a lower value to represent higher proximity.

B. Match Between Trajectory Points and Nearest Projections

While the distance between the nearest neighbors on both trajectories provides a good approximation of distance between the trajectories, the closest position on $\mathcal{T}_2$ to a point on $\mathcal{T}_1$ corresponds to a projection onto a line segment joining two points, rather than a particular point. For the i^{th} point on $\mathcal{T}_1$ with position $\mathbf{p}_1[i]$, we obtain the position of its projection $\mathbf{p}_{ij}$ onto the line segment between point positions $\mathbf{p}_2[j]$ and $\mathbf{p}_2[j+1]$ on $\mathcal{T}_2$ as

$$\mathbf{p}_{ij} = \alpha_{ij} \mathbf{p}_2[j] + (1 - \alpha_{ij}) \mathbf{p}_2[j+1]. \quad (6)$$

The value of the interpolation coefficient α_{ij} is given as

$$\alpha_{ij} = \text{clamp} \left(\frac{(\mathbf{p}_2[j+1] - \mathbf{p}_1[i])^T (\mathbf{p}_2[j+1] - \mathbf{p}_2[j])}{\|\mathbf{p}_2[j+1] - \mathbf{p}_2[j]\|^2}, 0, 1 \right), \quad (7)$$

where the ‘clamp’ function truncates the value of α_{ij} to 0 if $\alpha_{ij} \leq 0$ and to 1 if $\alpha_{ij} \geq 1$ so as to restrict the projection from lying outside the line segment joining $\mathbf{p}_2[j]$ and $\mathbf{p}_2[j+1]$. The index j_i^* of the point on $\mathcal{T}_2$ that starts the line segment with the nearest projection to $\mathbf{p}_1[i]$ is given as

$$j_i^* = \arg \min_j \|\mathbf{p}_1[i] - \mathbf{p}_{ij}\|^2, \quad (8)$$

where the term under the minimization represents the distance between $\mathbf{p}_1[i]$ and its projection $\mathbf{p}_{ij}$. The best interpolation coefficient α_{ij}^* and nearest projection $\mathbf{p}_{ij}^*$ are obtained by replacing j_i with j_i^* in Equations (6) to (7). To maintain symmetry, we obtain the best coefficient α_{kl}^* and best projection $\mathbf{p}_{kl}^*$ on $\mathcal{T}_1$ for the k^{th} point position $\mathbf{p}_2[k]$ on $\mathcal{T}_2$ by interchanging 1 and 2, and substituting k for i , and l for j in Equations (6) to (8). We obtain the nearest-projection distance in point positions $d_p^{\text{np}}(\mathbf{p}_1, \mathbf{p}_2)$ between the trajectories by summing individual Euclidean distances from the position of each point on one trajectory to the nearest projection on the other trajectory as

$$d_p^{\text{np}}(\mathbf{p}_1, \mathbf{p}_2) = \sum_{i=1}^N \|\mathbf{p}_1[i] - \mathbf{p}_{ij}^*\|^2 + \sum_{k=1}^N \|\mathbf{p}_2[k] - \mathbf{p}_{kl}^*\|^2. \quad (9)$$

Data: Matrix $\mathbf{D}$ of running distances.

Result: Set $\mathcal{I}$ of 2-tuples, where the first element in each 2-tuple is an index to a matched point in $\mathbf{p}_1$ and the second element is an index to a matched point in $\mathbf{p}_2$.

```

 $\mathcal{I} \leftarrow \{N, N\}, i \leftarrow N, j \leftarrow N.$ 
while  $i > 0$  &  $j > 0$  do
   $(i^+, j^+) \leftarrow \arg \min_{(\tilde{i}, \tilde{j})} \mathbf{D}[\tilde{i}, \tilde{j}]$ , where
   $(\tilde{i}, \tilde{j}) \in \{(i-1, j), (i, j-1), (i-1, j-1)\}.$ 
  if  $i > 0$  &  $j > 0$  then
     $\mathcal{I} \leftarrow \mathcal{I} \cup \{(i^+, j^+)\}, i \leftarrow i^+, j \leftarrow j^+.$ 
  end
end

```

Algorithm 1: Algorithm to generate indices for matched pairs using dynamic time warping.

To obtain the nearest-projection distance $d_o^{\text{np}}(\mathbf{q}_1, \mathbf{q}_2)$ between the orientations $\mathbf{q}_1$ and $\mathbf{q}_2$ of the trajectories $\mathcal{T}_1$ and $\mathcal{T}_2$, we first obtain the orientation $\mathbf{q}_{ij}^*$ at the nearest projection on $\mathcal{T}_2$ for the i^{th} point on $\mathcal{T}_1$ as

$$\mathbf{q}_{ij}^* = \mathbf{q}_2[j_i^*] (\mathbf{q}_2[j_i^*]^{-1} \mathbf{q}_2[j_i^* + 1])^{\alpha_{ij}^*}, \quad (10)$$

and the the orientation $\mathbf{q}_{kl}^*$ at the nearest projection on $\mathcal{T}_1$ for the k^{th} point on $\mathcal{T}_2$ as

$$\mathbf{q}_{kl}^* = \mathbf{q}_1[l_k^*] (\mathbf{q}_1[l_k^*]^{-1} \mathbf{q}_1[l_k^* + 1])^{\alpha_{kl}^*}. \quad (11)$$

In Equations (10) and (11), the right hand side represents a spherical linear interpolation [28] between the best adjacent orientations $\mathbf{q}_2[j_i^*]$ and $\mathbf{q}_2[j_i^* + 1]$ in $\mathbf{q}_2$, and $\mathbf{q}_1[l_k^*]$ and $\mathbf{q}_1[l_k^* + 1]$ in $\mathbf{q}_1$. We then express $d_o^{\text{np}}(\mathbf{q}_1, \mathbf{q}_2)$ as

$$d_o^{\text{np}}(\mathbf{q}_1, \mathbf{q}_2) = \sum_{i=1}^N (1 - \langle \mathbf{q}_1[i], \mathbf{q}_{ij}^* \rangle^2) + \sum_{k=1}^N (1 - \langle \mathbf{q}_2[k], \mathbf{q}_{kl}^* \rangle^2). \quad (12)$$

C. Match Using Dynamic Time Warping (DTW)

DTW [3] uses dynamic programming to match trajectories with varying speeds while preserving index order. DTW maintains a matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$, where the element $\mathbf{D}[i, j]$ of the matrix represents the running sum of the distance between the trajectories up to the position of the i^{th} point $\mathbf{p}_1[i]$ in $\mathcal{T}_1$ and the j^{th} point $\mathbf{p}_2[j]$ in $\mathcal{T}_2$, and is given by

$$\mathbf{D}[i, j] = \min_{(\tilde{i}, \tilde{j})} \mathbf{D}[\tilde{i}, \tilde{j}], \text{ where } (\tilde{i}, \tilde{j}) \in \{(i-1, j), (i, j-1), (i-1, j-1)\}. \quad (13)$$

The matrix is initialized to 0 prior to start, and filled by iterating over $i \in [2, N]$ and $j \in [2, N]$. The DTW distance $d_p^{\text{dtw}}(\mathbf{p}_1, \mathbf{p}_2)$ between positions of the trajectories is given by the last element in the matrix, i.e., $d_p^{\text{dtw}}(\mathbf{p}_1, \mathbf{p}_2) = \mathbf{D}[N, N]$. To obtain the DTW match $d_o^{\text{dtw}}(\mathbf{q}_1, \mathbf{q}_2)$ between orientations of the trajectories, we obtain a set $\mathcal{I}$ of matched index pairs between $\mathbf{p}_1$ and $\mathbf{p}_2$, generated according to Algorithm 1 [3]. We express $d_o^{\text{dtw}}(\mathbf{q}_1, \mathbf{q}_2)$ as the sum of the matches between orientations related through the matched pairs in $\mathcal{I}$, i.e., as

$$d_o^{\text{dtw}}(\mathbf{q}_1, \mathbf{q}_2) = \sum_{(i,j) \in \mathcal{I}} (1 - \langle \mathbf{q}_1[i], \mathbf{q}_2[j] \rangle^2). \quad (14)$$

VI. PERCEPTRON-BASED LEARNING OF OPTIMAL WEIGHTS ON MATCHES

To match an action for a user comprised of trajectories for the right hand, left hand, and head to actions in a library, we learn a set of weights on combining the position and orientation feature distances from trajectory matches for the three body parts. We perform learning using a training set of trajectories on the first day for every user, and we perform weighted matching and user authentication using the a test set of trajectories from the second day.

During the training phase, given n training right hand, left hand, and head trajectories per user for $P = 33$ users, we align each trajectory using one of the five alignment methods discussed in Section IV, and we match each body part trajectory to every trajectory for the same body part using one of the three matching approaches discussed in Section V. The matching provides a set of $(Pn)^2$ matches for right hand position d_{p-r}^* , right hand orientation d_{o-r}^* , left hand position d_{p-l}^* , left hand orientation d_{o-l}^* , head position d_{p-h}^* , and head orientation d_{o-h}^* , where $*$ may be ‘nn’ for nearest neighbors, ‘np’ for nearest projection, or ‘dtw’ for DTW. We learn weights w_f and bias b for a perceptron classifier,

$$y = \sigma \left(\sum_{f \in \mathcal{F}} w_f d_f^* + b \right), \quad (15)$$

where the set $\mathcal{F}$ represents a combination of the position and orientation feature matches from the three body parts, and $\sigma(\cdot)$ represents the perceptron activation function. In this work, we analyze the following combinations for $\mathcal{F}$: (1) position (p) and orientation (o) of all body parts, i.e., right (r), left (l), and head (h), where $\mathcal{F} = \{p-r, o-r, p-l, o-l, p-h, o-h\}$, (2) position only of all body parts, i.e., $\mathcal{F} = \{p-r, p-l, p-h\}$, (3) orientation only of all body parts, i.e., $\mathcal{F} = \{o-r, o-l, o-h\}$, (4) position and orientation of sets of two body parts, i.e., $\mathcal{F} = \{p-r, o-r, p-l, o-l\}$, $\{p-r, o-r, p-h, o-h\}$, or $\{p-l, o-l, p-h, o-h\}$, (5) position only of sets of two body parts, i.e., $\mathcal{F} = \{p-r, p-l\}$, $\{p-r, p-h\}$, or $\{p-l, p-h\}$, (6) orientation only of sets of two body parts, i.e., $\mathcal{F} = \{o-r, o-l\}$, $\{o-r, o-h\}$, or $\{o-l, o-h\}$, and (7) position and orientation of one body part, i.e., $\mathcal{F} = \{p-r, o-r\}$, $\{p-l, o-l\}$, or $\{p-h, o-h\}$. Summing together all choices for $\mathcal{F}$ from (1) to (6) gives us 15 feature sets. We use the hyperbolic tangent sigmoid as the activation function $\sigma(\cdot)$ in this work. The target for the output y is 0 when the trajectories matched are from the same user, and 1 when the trajectories are from different users. We use scaled conjugate gradients implemented in MATLAB to train the perceptron.

During the test phase, given m right hand, left hand, and head trajectories per user for the P users, we align each trajectory using the same alignment method used during training, and we match each body part trajectory in the test set to all Pn trajectories for the same body part in the training set using the training match metric to obtain $P^2 mn$ matches for the right hand position, right hand orientation, left hand position, left hand orientation, head position, and head orientation. For each test user action, we apply the perceptron classifier trained for a feature set from the list above to the match values of all Pn training user actions matched to the test action. We obtain the best matching user as the user for the training action with

the lowest value of the output in Equation (15), since the likelihood of a user for a test action being the same as that of a training action is high for a match close to 0.

VII. RESULTS

To understand how soon authentication can be performed during a user action, we obtain identification results by varying the number of trajectory points N . To evaluate the quantity of training data needed to reliably authenticate a user, we perform our accuracy computations by varying the number of training actions per user n between 1 and 10, while keeping the number of test actions m at 10 actions per user. We also evaluate how many actions are needed to reliably authenticate a user at test time by varying the number of test actions per user m from 1 to 10 while keeping the number of training actions, n at 5. We compute accuracies, i.e., percentage of total Pm user actions that are correctly identified, for authenticating users from their second day session actions using the 5 alignment methods from Section IV, the 3 matching approaches from Section V, and the 15 feature sets used to train the perceptron in Section VI, together with the choices of N , n , and m . We augment our feature sets with matching using position only, and orientation only for a single body part, where a weighted combination of features is not necessary, and the closest user is represented by the user corresponding to an action with the lowest match value. This provides a total of 21 feature sets.

We show the best results of our accuracy computations for each of the 5 alignment methods and 3 matching approaches as subplots in Figures 2 and 3. Figure 2 provides match results when n is varied between 1 and 10 along the horizontal axis of each bar chart, and m is 10, while Figure 3 provides match results when n is kept at 5, while m is varied between 1 and 10 along the horizontal axis. The text in each bar represents the best feature choice, i.e., position (p), orientation (o), or both (po), the best combination of right hand controller (r), left hand controller (l) and headset (h) trajectories, the best choice of number of trajectory points N , and the best value of the accuracy as a percentage. Figure 2 demonstrates that using the bounding box center provides highest performance for all three matching methods. As demonstrated in Figure 1, without alignment, the trajectories are too far apart for a match. The mean in the second column tends to yield lower results as the mean may be weighted near higher density point concentrations, e.g., the start of the trajectory, and differences in wait times influence the displacement of the mean. Alignment in the XZ plane alone lowers accuracy, indicating that subjects may show intra-user variations in the Y direction, which are corrected by a full bounding box center alignment. Among the matching metrics, the nearest neighbor match performs best, while the nearest projection follows closely. DTW performs worst indicating that misalignments during match detection in DTW may propagate throughout the trajectories.

For all methods, the highest accuracies are generally obtained when the right hand is combined with the head, likely since the left hand may be involved in non-goal-oriented motions during a ball-throwing activity. Additionally, highest accuracies are obtained largely using orientation alone, followed by using position and orientation together, indicating

that orientation is a stronger user signature than position during ball-throwing, especially for the right hand and head. The highest accuracy of 93.03% is obtained by using 9 training trajectories with nearest neighbors, bounding box center alignment, 100 trajectory points, and orientation alone for the right hand and head. We notice that even with 6 training trajectories and higher we obtain accuracies of 89.70% and above, and with 1 and 2 trajectories we show 61.82% and 72.12% accuracy where chance is 3.03%. Figure 4 shows an extended analysis with nearest neighbors and bounding box center for all 21 feature sets, where N , the number of trajectory points is varied along the horizontal axis of each graph, and each plot line corresponds to one value of n . While the graphs for the right hand and head demonstrate high accuracies for using orientation or both position and orientation, we notice that high accuracies may also be obtained if position alone of the right hand is combined with the left hand, indicating that users may swing their left hand in a unique fashion, however they may shake it around in a random fashion which influences orientation. We can authenticate users with around 80% accuracy and above starting at 60 trajectory points (i.e., within 1.33 seconds) and 6 or more training trajectories using right + left position, right + head orientation, or right + head and both position and orientation, all (right, left, and head) orientation, or all and both position and orientation. The top-left graph represents the approach of Kupin et al. [11], and only attains a maximum accuracy of 83.64%.

Figure 6(a) shows the confusion matrix at the 93.03% accuracy point discussed earlier. For all but 5 users, accuracies are 90% and above. Of the 5 users, users 13 and 23 show accuracies of 80%, user 8 shows 70%, user 14 60%, and user 27 shows the worst accuracy of 30%. As shown in Figure 6(d), the head for user 27 on the second day does not follow their head on the first day. As such, as shown in Figures 6(b) and 6(d), user 27 tends to match with user 5 whose right hand and head (red day 1 trajectory) are close to the day 2 trajectories of user 27. Figure 3 demonstrates similar trends as Figure 2 when the number of test trajectories is varied, with the exception that as more test trajectories are added, the accuracy drops, with the drop generally being highest from 1 to 2 test trajectories. This may be attributed to users on the second day developing a habituation to the VR system earlier than on the first day. In future, we are interested in analyzing the effect of habituation in VR systems, similar to the work of Syed et al. [30] in smart devices. We observe higher accuracies when the nearest projection is used, however, the performance is comparable to nearest neighbors. The nearest projection and bounding box center accuracy of 96.97% by using just the first test day trajectory indicates that our approach can be used to immediately authenticate users using a VR system for the first time after providing an initial set of 5 training data samples. Figure 5 reflects the trends in feature sets of Figure 4, with orientation again showing dominance in distinguishing users.

VIII. DISCUSSION

We provide an approach to combine position and orientation features from VR headsets and hand controllers to perform

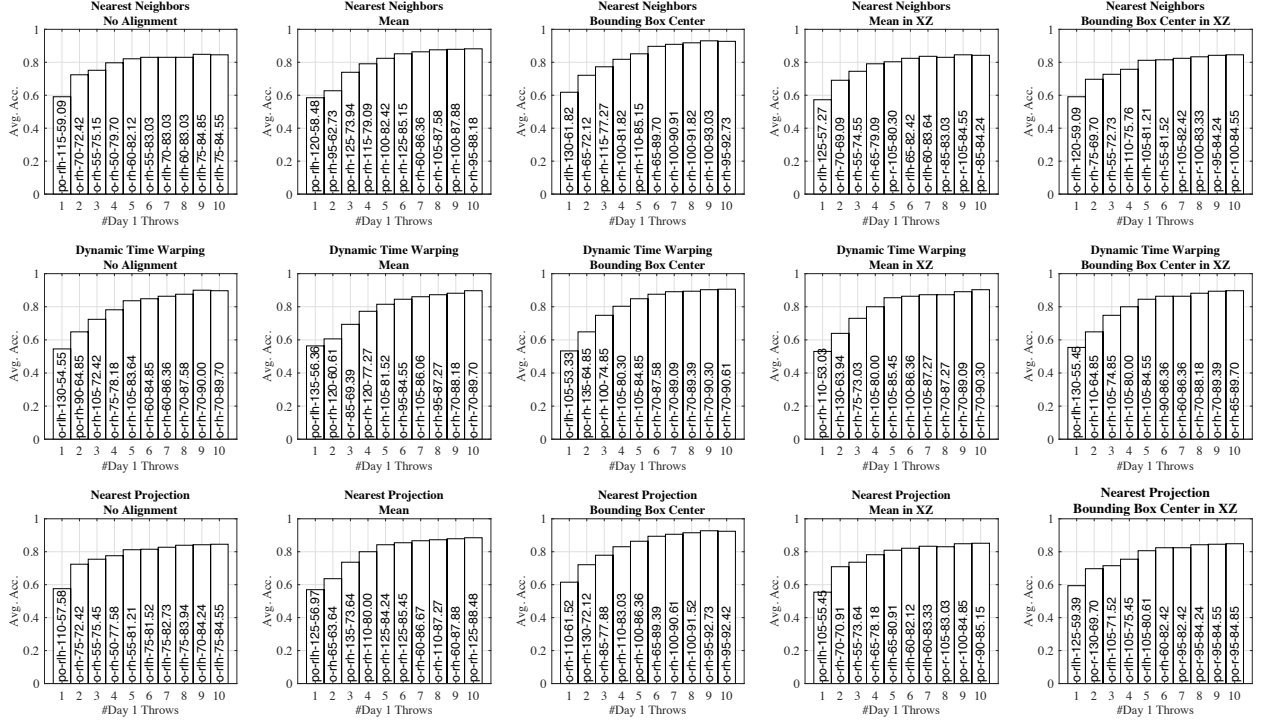


Fig. 2. Highest accuracies for the 5 alignment methods and 3 match metrics used in this work, when 10 test day trajectories per user are matched to 1 through 10 training day trajectories per user. Notation in each bar is as follows: choice of position (p)/orientation (o)/both (po) - device, i.e., right (r), left (l), and head (h) combinations - number of trajectory points - accuracy as a percentage.

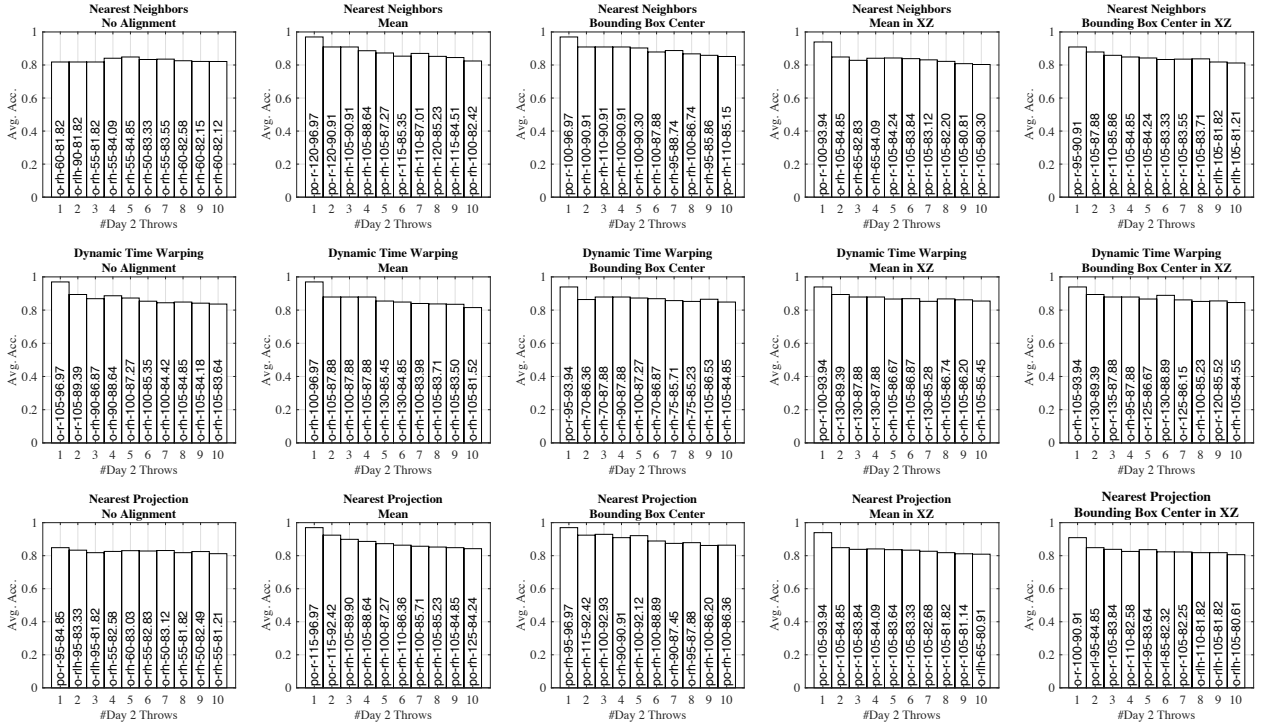


Fig. 3. Highest accuracies for the 5 alignment methods and 3 match metrics used in this work, when 1 through 10 test day trajectories per user are matched to 5 training day trajectories per user. Notation in each bar is as follows: choice of position (p)/orientation (o)/both (po) - device, i.e., right (r), left (l), and head (h) combinations - number of trajectory points - accuracy as a percentage.

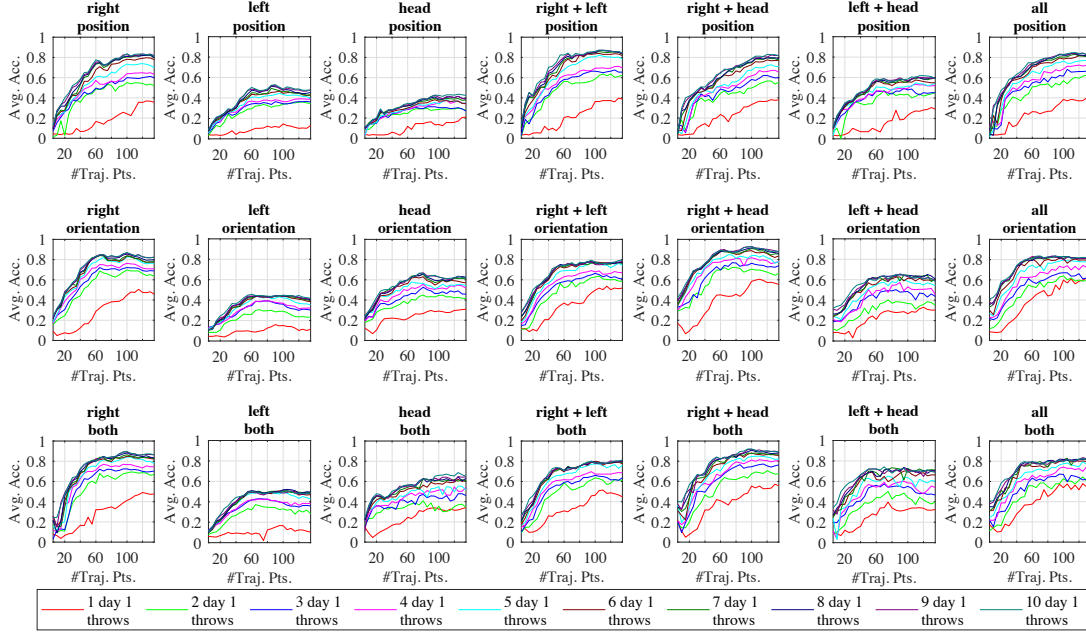


Fig. 4. Results for using position, orientation, and both from various combinations of body parts using nearest neighbor matching and bounding box center alignment when 10 test day trajectories are matched to 1 through 10 training day trajectories.

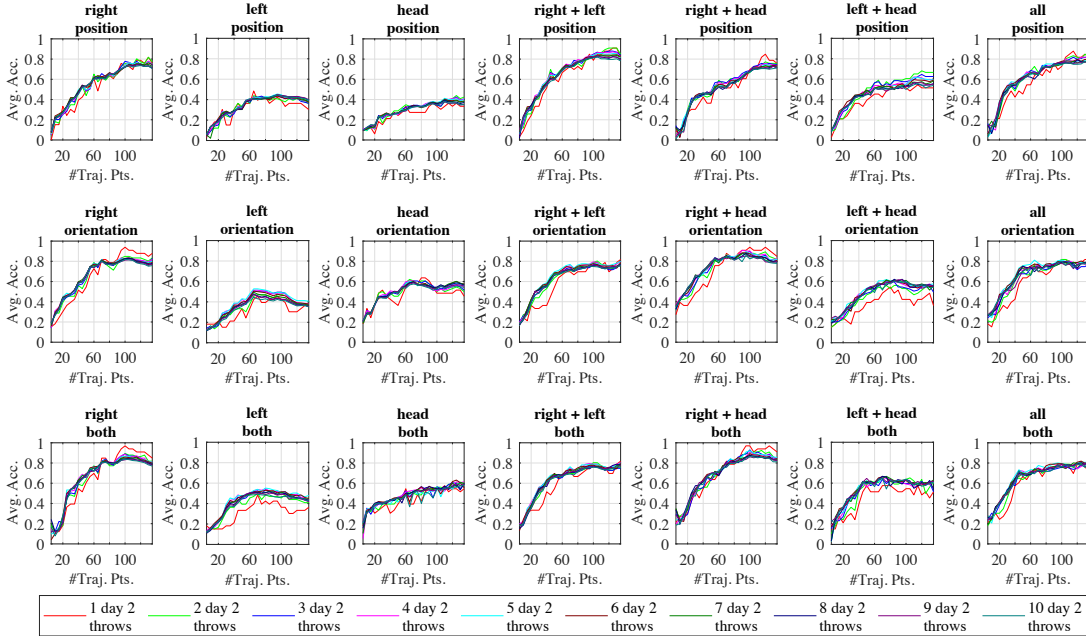


Fig. 5. Results for using position, orientation, and both from various combinations of body parts using nearest projection matching and bounding box center alignment when 1 through 10 test day trajectories are matched to 5 training day trajectories.

task based seamless continual authentication in virtual environments. Our approach demonstrates a maximum accuracy of 93.03% for matching 10 test actions from 33 users by using the orientation information from the the headset and right hand controller, subtracting the bounding box center to align the trajectories and using nearest neighbor matching. While we demonstrate results using the HTC Vive, we expect similar

results from an Oculus Rift, Samsung Gear VR, or other VR devices consisting of both a headset and hand controllers. We provide results for both position and orientation and demonstrate that using orientation by itself provides the best results. However, there may be occasions when obtaining orientation information may be impossible, for instance, if a VR system does not contain a gyroscope. In this case, we

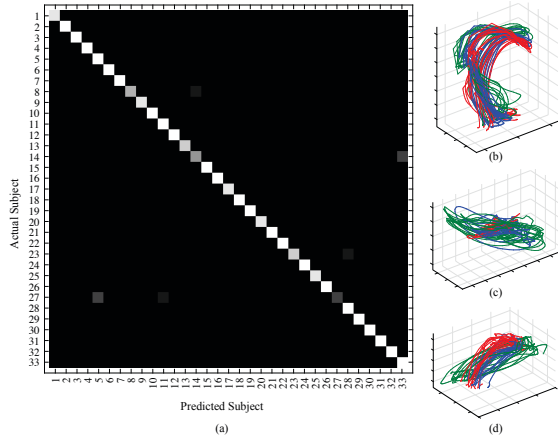


Fig. 6. (a) Confusion matrix for highest accuracy when 10 test day trajectories are compared to 9 training day trajectories using nearest neighbors. (b) Right hand, (c), left hand, and (d) head trajectories for user 27 (blue day 2 and green day 1) and their confounding user 5 (red day 1).

show that using position alone also provides acceptable results, enabling our approach to be generalized to any VR device that provides either position or orientation. Our work specifically looks at the case where an impostor is likely to precisely mimic the motions of a genuine user. In future research, we will explore how attempts to make precise mimicry in more complex actions, such as virtual drone flying or driving, which may have higher intra-user variability will affect accuracy. Intra-user variability may reduce accuracy when larger number of users are added. Future work will investigate the role of goal-orientatedness in authentication accuracy, and examine approaches to match trajectory segments to improve accuracy for larger user sets. While we do not show it in this work, the output of the neural network can be compared against a threshold to detect impostors external to the training set as shown in our companion demonstration paper [16].

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation (NSF) grant #1730183.

REFERENCES

- [1] A. A. E. Ahmed and I. Traore. A new biometric technology based on mouse dynamics. *TDSC*, 2007.
- [2] L. P. Berg and J. M. Vance. Industry use of virtual reality in product design and manufacturing: a survey. *Virtual reality*, 2017.
- [3] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDDW*, 1994.
- [4] K. K. Bhagat, W.-K. Liou, and C.-Y. Chang. A cost-effective interactive 3d virtual reality system applied to military live firing training. *Virtual Reality*, 2016.
- [5] S. Choi, K. Jung, and S. D. Noh. Virtual reality applications in manufacturing industries: Past research, present findings, and future directions. *Concurrent Engineering*, 2015.
- [6] R. M. Clifford, H. Khan, S. Hoermann, M. Billinghurst, and R. W. Lindeman. Development of a multi-sensory virtual reality training simulator for airborne firefighters supervising aerial wildfire suppression. In *VAR4Good*, 2018.
- [7] 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*, 2016.

- [8] L. Freina and M. Ott. A literature review on immersive virtual reality in education: state of the art and perspectives. In *eLSE*, 2015.
- [9] 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 *NDSS*, 2017.
- [10] Z. Jorgensen and T. Yu. On mouse dynamics as a behavioral biometric for authentication. In *ASIACCS*, 2011.
- [11] A. Kupin, B. Moeller, Y. Jiang, N. K. Banerjee, and S. Banerjee. Task-Driven Biometric Authentication of Users in Virtual Reality (VR) Environments. In *MMM*, 2019.
- [12] 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 *PerCom*, 2016.
- [13] J. I. Lipton, A. J. Fay, and D. Rus. Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *Robotics and Automation Letters*, 2017.
- [14] 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*, 2014.
- [15] 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*, 2014.
- [16] R. Miller, A. Ajit, N. K. Banerjee, and S. Banerjee. Realtime behavior-based continual authentication of users in virtual reality environments. In *AIVR*, 2019.
- [17] F. Monrose and A. D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Generation computer systems*, 2000.
- [18] T. Mustafa, R. Matovu, A. Serwadda, and N. Muirhead. Unsure how to authenticate on your vr headset?: Come on, use your head! In *IWSPA*, 2018.
- [19] M. M. North, S. M. North, and J. R. Coble. Virtual reality therapy: an effective treatment for the fear of public speaking. *IJVR*, 2015.
- [20] M. Oberhauser and D. Dreyer. A virtual reality flight simulator for human factors engineering. *Cognition, Technology & Work*, 2017.
- [21] N. Palaskar, Z. Syed, S. Banerjee, and C. Tang. Empirical techniques to detect and mitigate the effects of irrevocably evolving user profiles in touch-based authentication systems. In *HASE*, 2016.
- [22] F. Pallavicini, N. Toniazzi, L. Argenton, L. Aceti, and F. Mantovani. Developing effective virtual reality training for military forces and emergency operators: from technology to human factors. In *MAS*, 2015.
- [23] 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 *CHI*, 2019.
- [24] C. E. Rogers, A. W. Witt, A. D. Solomon, and K. K. Venkatasubramanian. An approach for user identification for head-mounted displays. In *ISWC*, 2015.
- [25] S. Schneegass, Y. Oualil, and A. Bulling. Skullconduct: Biometric user identification on eyewear computers using bone conduction through the skull. In *CHI*, 2016.
- [26] A. Serwadda, V. V. Phoha, and Z. Wang. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *BTAS*, 2013.
- [27] C. Shen, Y. Zhang, X. Guan, and R. A. Maxion. Performance analysis of touch-interaction behavior for active smartphone authentication. *TIFS*, 2015.
- [28] K. Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH*, 1985.
- [29] K.-C. Siu, B. J. Best, J. W. Kim, D. Oleynikov, and F. E. Ritter. Adaptive virtual reality training to optimize military medical skills acquisition and retention. *Military medicine*, 2016.
- [30] Z. Syed, S. Banerjee, Q. Cheng, and B. Cukic. Effects of user habituation in keystroke dynamics on password security policy. In *HASE*, 2011.
- [31] Z. Syed, S. Banerjee, and B. Cukic. Leveraging variations in event sequences in keystroke-dynamics authentication systems. In *HASE*, 2014.
- [32] Z. Syed, J. Helmick, S. Banerjee, and B. Cukic. Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability. *JSS*, 2019.
- [33] S. Yi, Z. Qin, E. Novak, Y. Yin, and Q. Li. Glassgesture: Exploring head gesture interface of smart glasses. In *INFOCOM*, 2016.
- [34] Z. Yu, H.-N. Liang, C. Fleming, and K. L. Man. An exploration of usable authentication mechanisms for virtual reality systems. In *APCCAS*, 2016.