# MotionAuth: Motion-based Authentication for Wrist Worn Smart Devices

Junshuang Yang*, Yanyan Li*, Mengjun Xie
Department of Computer Science
University of Arkansas at Little Rock
Little Rock, Arkansas, USA
Email: {jxyang2,yxli5,mxxie}@ualr.edu

***Abstract*—Wrist worn smart devices such as smart watches become increasingly popular. As those devices collect sensitive personal information, appropriate user authentication is necessary to prevent illegitimate accesses to those devices. However, the small form and function-based usage of those wearable devices pose a big challenge to authentication. In this paper, we study the efficacy of motion based authentication for smart wearable devices. We propose MotionAuth, a behavioral biometric authentication method, which uses a wrist worn device to collect a user's behavioral biometrics and verify the identity of the person wearing the device. MotionAuth builds a user's profile based on motion data collected from motion sensors during the training phase and applies the profile in validating the alleged user during the verification phase. We implement MotionAuth using Android platform and test its effectiveness with real world data collected in a user study involving 30 users. We tested four different gestures including simple, natural gestures. Our experimental results show that MotionAuth can achieve high accuracy (as low as 2.6% EER value) and that even simple, natural gestures such as raising/lowering an arm can be used to verify a person with pretty good accuracy.**

## I. Introduction

Wearable smart devices, especially wrist worn devices such as smart watch and activity tracker wristband, become increasingly popular following the huge success of smartphone. All major smartphone vendors have released their smart watches including Apple Watch, Samsung Galaxy Gear, LG G Watch, etc. Those wrist worn devices usually carry multiple built-in sensors such as accelerometer and gyroscope sensors that can measure movements and biosensors that can measure heartbeat, skin temperature, respiratory rate, and other health related information. The functionality and convenience of wrist worn devices make it easy to collect sensitive personal data unobtrusively and continuously. Therefore, securing device accesses is necessary in order to protect data privacy.

On one hand, we need a mechanism to prevent unauthorized access to a wrist worn device. Due to the small form factor, those devices are prone to get stolen or lost. Current protection available on those devices is rather weak and inconvenient. For example, we can use PIN on smart watches. However, the small screen of smart watch makes PIN input difficult and error-prone. Therefore, no protection is applied in practice. However, given that those devices are usually paired with a smartphone through a wireless link (e.g., Bluetooth) for realizing richer functionality, weak or no authentication on wrist worn devices not only endangers the privacy of user's data on device but also poses a significant threat to the security of connected smartphone.

On the other hand, the plethora of sensors and transparent data collection also provide an opportunity of securing the device by the data it collects. As those collected data most often are pertinent to a specific person (i.e., the device owner), they can be used to build a profile of device owner, which can then be used for verification purpose in case data access is requested. For example, we can use kinetic data such as accelerometer and gyroscope readings to construct the owner's behavioral biometric profile. Recently, many research efforts have been devoted to user identification/verification based on behavioral biometrics for smartphones (e.g., [6], [24], [15]) and their results indicate that behavioral biometric based authentication is a viable means.

In this paper, we study the efficacy of motion based authentication for smart wearable devices. We propose MotionAuth, a behavioral biometric based authentication method, to collect a user's behavioral biometrics through a wrist worn device and verify the identity of the person wearing the device. MotionAuth builds a user's profile based on motion data collected from motion sensors during the training phase and applies the profile in validating the alleged user during the verification phase. We apply two different verification methods, a histogram based method (Histogram) and a dynamic time warping based method (DTW), to examine the accuracy of MotionAuth. We implement MotionAuth using an Android smart watch and test its effectiveness with real world data collected in a user study involving 30 users. Four different gestures are tested in which three are simple, natural gestures. Our experimental results show that MotionAuth can achieve high accuracy (as low as 2.6% EER value) and that the average EER value of the four gestures is lower than 5% with either of the two verification methods.

In summary, we have made the following contributions:

- We propose a behavioral biometric based authentication framework—MotionAuth—for wrist worn smart devices. MotionAuth exploits those devices' innate capability of continuous and transparent motion sensing to build a behavioral biometric profile for the device's owner. To our best knowledge, MotionAuth is the first authentication

* Both authors contributed equally to this work.

978-1-4799-8425-1/15/$31.00 ©2015 IEEE

system that utilizes behavioral biometrics for wrist worn smart devices.

- MotionAuth targets simple and natural gestures for authentication purpose. We have conducted preliminary evaluation in which 30 volunteers participated. We applied two different methods—Histogram and DTW–for verification and the experimental results are promising.

The rest of this paper is organized as follows: Section II briefly describes the background of behavioral biometrics and related work. Sections III and IV present the design of MotionAuth and its prototype implementation. Section V details the evaluation including the method for data collection and the analysis of experimental results. Section VI briefly discusses the future work and concludes this paper.

## II. Background and Related Work

User authentication refers to the process in which a user submits her identity credential (often represented by paired username and password) to an information system and validates to the system that she is who she claims to be. In general, there are three types of authentication factors: something a user knows (e.g., a password), something a user has (e.g., a secure token), and something a user is (e.g., biometric characteristics). Passwords are the most common authentication mechanism. However, password-based authentication has many security issues [8], [11], [4] and is not suitable for wrist worn devices.

In general, a biometric authentication system verifies a person based on either his/her physiological traits (e.g., fingerprint, face, voice, iris, bioimpedance, etc) [12], [2], [5] or behavioral biometrics (e.g., finger or hand movements) [25], [6]. Thanks to rich sensing capabilities, both physiological and behavioral biometrics can be easily collected on today's smart mobile devices. While physiological traits can achieve high accuracy in user authentication, they are subjected to a variety of attacks [14], [20], [21] and also raise privacy concerns [19]. Moreover, accuracy of physiology-based mechanisms may be substantially degraded by environmental factors such as viewing angle, illumination, and background noise [13], [3]. In contrast, behavioral biometrics appear less sensitive to ambient light or noise.

The popularity of mobile devices especially smartphones and tablets have attracted a great deal of research efforts to investigate how to effectively apply behavioral biometrics to mobile device authentication. Researchers have studied behavioral biometric features extracted from regular touch operations [7], [23], unlock pattern and PIN operations [6], [24], and multitouch gestures [15], [17], [18].

Activity and gesture recognition has been extensively studied based on sensing data from accelerometer and some other sensors such as gyroscope on a mobile device. Kwapisz *et al.* used a single wrist-worn accelerometer to recognize and classify human activities such as sitting, walking and running and their method can achieve accuracy of more than 94.13% [9]. In [22] and [1], accelerometer based approaches for gesture classification are presented in which a 3-axis accelerometer was used to recognize hand movement gestures. Liu *et al.* proposed uWave, an accelerometer-based gesture recognition system [10]. They conducted a comprehensive analysis on accelerometer-based gesture recognition using Wii remote and studied both simple gestures and personalized gestures. uWave can also be applied to gesture-based authentication. Similar to our work, uWave uses 3D freestyle movement as a gesture for authentication. However, as reported in [10] while using complex personalized gestures can achieve high accuracy in authentication, gestures from simpler gesture group have lower accuracy. Our work focuses on simple, natural gestures that can be more practical in real-world deployment and shows that high verification accuracy can still be achieved.

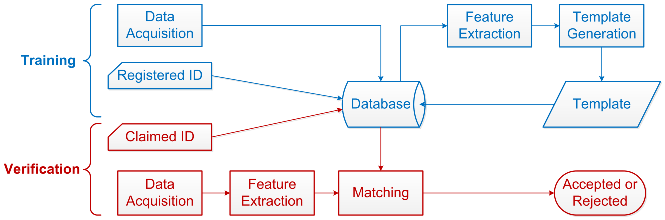


Fig. 1. Overview of MotionAuth Authentication Process.

## III. System Design

MotionAuth is designed as a behavioral biometric authentication framework for personal wrist worn smart devices (or simply "the device(s)" when context is clear). As monitoring human activities is one of primary applications for such devices, built-in motion sensors (i.e., accelerometer and gyroscope) in those devices continuously collect the device user's motion data. MotionAuth leverages the motion data collected during the personal gesture for verification being performed. It first builds a profile for the device owner and later verifies the device user with that profile.

MotionAuth imposes no constraint on the form of gesture, that is, simple gestures can be applied as well as complex ones although more complex, uncommon gestures generally can render higher discernibility. The design of MotionAuth is strongly motivated by a simple idea: verifying a user with simple, natural gestures that are often performed; therefore average people do not need to remember their verification gesture and can perform it effortlessly and consistently. To examine this idea, we selected three natural gestures plus a special one and asked volunteers to perform them in evaluating the prototype of MotionAuth, which is detailed in Section V.

To use MotionAuth, the owner of a wrist worn device (we assume the owner is also the device user) first needs to take a training process, in which the person performs the gesture selected for verification a number of times with the arm wearing the device and his or her behavioral biometric profile is built from the readings of motion sensors collected while performing each gesture. Then, when user verification is invoked, the motion data of the gesture performance will be compared with the user profile to verify the user's authenticity. Fig. 1 depicts the overall process of authentication using MotionAuth.

As illustrated in Fig. 1, there are four important modules in the MotionAuth framework: data acquisition, feature extraction, template generation, and matching. In the data acquisition module, raw data are collected from motion sensors during gesture performance and stored into a database that is either local or remote. In the feature extraction module, a set of features are derived from raw sensor data and fed into the template generation module. In that module, each gesture sample is represented by a feature vector after applying certain transformation, and a template for each user is generated from feature vectors derived from the user's training samples. The matching module takes the user's template (selected based on the claimed identity) and features extracted from a test sample and applies certain matching algorithm to decide whether the test sample can be accepted as genuine. As can be seen, the overall design of MotionAuth is quite general and can be realized by different matching schemes for behavioral biometric verification.

There are two types of verification techniques that are commonly used in biometric based authentication. They are function based methods such as dynamic time warping (DTW) algorithm and hidden Markov models (HMM) and feature based methods that use descriptive features of the biometric. We consider both types of techniques for MotionAuth; We choose DTW as the representative function based method and a histogram method, which is shown effective for online signature verification in [16], as the representative feature based method. DTW has been widely used in various studies on behavior biometric authentication [6], [10]. As gestures can be treated as a freestyle 3D signature written with arm, it would be interesting to apply the histogram method (or simply Histogram) to gesture-based verification.

Both Histogram and DTW use the same data acquisition process. Data collection from accelerometer and gyroscope (we currently only use these two sensors due to their ubiquity in smart wearable devices) begins when a user starts the gesture and ends when the user finishes it. Given three dimensional data for both accelerometer and gyroscope, total six raw time series are generated when performing each gesture. In the rest of the section, we first present the threat model for MotionAuth, then briefly describe the Histogram and DTW methods, and finally discuss issues of deploying the MotionAuth framework.

### A. Threat Model

In the threat model for MotionAuth, we assume that an adversary attempts to gain illegitimate access to protect applications or data on the device by launching mimicry attacks, that is, the adversary already knows the gesture for authentication and pretends to be the device's owner by performing that gesture. We assume the adversary does not seek to gain access to the data stored on the device through network-based attacks or other means. We also assume that the attacker can only make up to a specific number of attempts for authentication. If all attempts fail, the predefined protection mechanism, e.g., encrypting all the data on device, will be invoked.

TABLE I
PART OF THE HISTOGRAMS USED IN MOTIONAUTH

| Histogram | Min | Max | Bin # | Description |
|---|---|---|---|---|
| $M$ | $\mu - 3\sigma$ | $\mu + 3\sigma$ | 16 | Magnitude of acceleration. $M = \sqrt{AX^2 + AY^2 + AZ^2}$ |
| $\theta_x$ | $-\Pi/2$ | $\Pi/2$ | 16 | Angle between magnitude and $X$-axis. $\theta_x = \arccos(AX/M)$ |
| $\theta_y$ | $-\Pi/2$ | $\Pi/2$ | 16 | Angle between magnitude and $Y$-axis. $\theta_y = \arccos(AY/M)$ |
| $\theta_z$ | $-\Pi/2$ | $\Pi/2$ | 16 | Angle between magnitude and $Z$-axis. $\theta_z = \arccos(AZ/M)$ |

### B. Histogram Method

*1) Feature Extraction:* First, from six raw readings of 3-axis acceleration (from accelerometer) and angular velocity (from gyroscope), we derive 30 features in total: three accelerations and three angular velocities (in $X$, $Y$, $Z$) and their corresponding first and second derivatives (total 18 features), the magnitude of acceleration $M$, the three angles between $M$ and $X/Y/Z$ (denoted as $\theta_x$, $\theta_y$, and $\theta_z$) and their corresponding first and second derivatives (total 12 features). Table I gives more details about $M$ and $\theta_x$, $\theta_y$, and $\theta_z$. Each feature is represented by a vector denoted as $V = \{v_i | i = 1, 2, ..., n\}$. We use $V'$ and $V''$ to denote the first and second derivatives of $V$ respectively, where $V' = \{v'_i | v'_i = v_{i+1} - v_i, i = 1, 2, ..., n-1\}, V'' = \{v''_i | v''_i = v'_{i+1} - v'_i, i = 1, 2, ..., n-2\}$. To simplify, accelerations in $X$, $Y$, $Z$ are denoted as $AX$, $AY$, $AZ$ respectively, and their derivatives as $AX'$, $AY'$, $AZ'$, $AX''$, $AY''$, and $AZ''$. These derivatives reflect the change rate and change acceleration of specific attributes.

Then, each vector is converted to a probability distribution histogram through binning. We create a given number of equidistant histogram bins with the given min and max values of the histogram and put each element of the vector into those bins. We then calculate the frequency of each bin by dividing the number of elements falling into that bin by the total number of elements. Let $b_i$ be a frequency value for bin $i$. A feature vector $B_j$ ($1 \le j \le 30$) is represented by concatenating the bin frequency values for feature $j$, i.e., $B_j = \{b^j_1 || b^j_2 || ... || b^j_{jm}\}$, where $jm$ is the number of bins. Finally, a gesture sample is represented by concatenating all the feature vectors $B_j$ into a single feature vector $F$, i.e., $F = \{B_1 || B_2 || ... || B_{30}\} = \{b^1_1 || b^1_2 || ... || b^1_{1m} || ... || b^{30}_1 || b^{30}_2 || ... || b^{30}_{30m}\}$.

*2) Template Generation:* A user's template for a given gesture is generated from the feature set derived from training samples of that gesture. Since each gesture sample is represented by a feature vector $F_i$, we have a sequence of vectors $F_1, F_2, ...F_k$ from $k$ training samples. For each $F_i, 1 \le i \le k$, $F_i = \{f^i_1 || f^i_2 || ... || f^i_n\}$, where $f^i_j$ ($1 \le j \le n, n = 1m + 2m + ... + 30m$) is a frequency value. The template $F_t$ is defined as $F_t = \{f^t_1 || f^t_2 || ... || f^t_n\}$, where

$$f^t_j = \frac{avg(f^1_j, f^2_j, ..., f^k_j)}{std(f^1_j, f^2_j, ..., f^k_j) + \epsilon}, 1 \le j \le n,$$

and $\epsilon$ is a small value 0.002 to prevent division by zero. The feature vector $F_t$ and standard deviation vector $Q = \{std(f^1_1, f^2_1, ..., f^k_1) || std(f^1_2, f^2_2, ..., f^k_2) || ... || std(f^1_n, f^2_n, ..., f^k_n)\}$ are stored as the user's profile for verification purpose.

*3) Matching:* To verify the claimed user, given a testing gesture sample $F_s$, the similarity distance score $D_{sim}$ is calculated using Manhattan Distance between $F_t$ and $F_s$. $D_{sim} = \sum_{i=1}^{n} |f_i^t - f_i^s/Q_i|$. If the score is less than a predefined threshold the sample is accepted and the person who performed the gesture passes the verification. Otherwise, the sample is rejected.

### *C. DTW Method*

*1) Feature Extraction:* The DTW method extracts the same set of features as the Histogram method does, but its data representation of samples is different. All samples in DTW are represented using original time series. Assume $n$ features ($n$ is 30 in our case) are extracted from a sample with length (i.e., the number of time points) $m$, the sample is represented as a $n \times m$ matrix.

*2) Template Generation:* Let $S$ and $Train$ be a training sample and the set of training samples, respectively. We use $min(d(S, Train - \{S\}))$ to denote the minimum DTW distance between $S$ and all the other training samples. First, we calculate DTW distances between every pair of training samples to derive the average of minimum DTW distances $avg(D_{min})$, where $D_{min} = \{min(d(S, Train - \{S\})) : S \in Train\}$. Then, we identify a training sample $T$ that has the minimal sum of the DTW distances to the other training samples. This sample is used as the user's gesture template. The minimum DTW distance between $T$ and the rest of training samples, $min(d(T, Train - \{T\}))$, is saved along with $avg(D_{min})$ as the user profile.

*3) Matching:* Assume sample $S'$ is collected for the user to be verified. We compute the similarity score $D_{sim}$ between sample $S'$ and template $T$.

$$D_{sim} = \left| \frac{min(d(S', Train)) - min(d(T, Train - \{T\}))}{avg(D_{min})} \right|$$

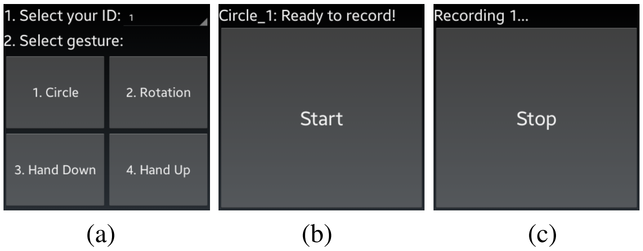
where $d(S', Train)$ returns the set of DTW distances between $S'$ and each training sample. The testing sample $S'$ will be accepted if $D_{sim}$ is lower than the predefined threshold; otherwise it will be rejected.

### *D. Discussion*

Given the variation of wrist worn smart devices in terms of their form, user interface, and usage, the actual implementation of MotionAuth may vary. For example, for those smart watches, the entire authentication process can be performed on the smart watch; or the computation can be executed on the smartphone if the watch is used as a companion of the phone. The protection of MotionAuth can present as a screen lock that is unlocked by a user specified gesture. For activity tracking bands that do not have a visual interaction interface, MotionAuth may use vibration or sound as the means for indication of verification result. For those devices without any user interaction interface, MotionAuth could attach the verification result as auxiliary information to sensor data so that the result can be viewed later when the data are synchronized to another device for review.

## IV. Implementation

We implemented a prototype of MotionAuth on Android platform. For ease of evaluation, the Android application of the current prototype is used as front end to collect data and all the computations are performed on a back end PC. The application was developed for smart watches that run on Android 4.x or upper version platforms and carry built-in accelerometer and gyroscope sensors. Figure 2 shows the user interface of the application.



(a) (b) (c)

Fig. 2. Android app for data collection in the prototype

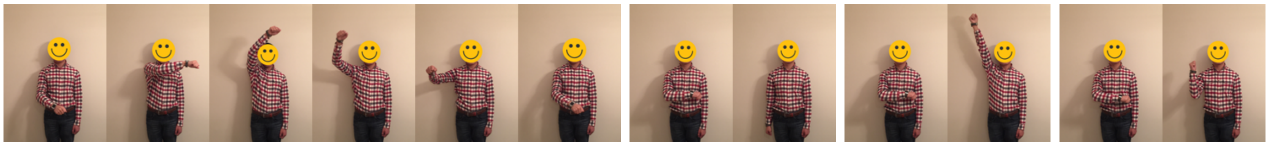
The main user interface (UI), shown in Fig. 2 (a), has a user identity (ID) selector and lists four gestures each requiring a user to perform 10 times. The detail of the gestures is given in Section V. When using the app to collect data, a user first selects his or her assigned ID and then clicks each gesture button in turn to perform those gestures. Once a gesture trial (10 actions) is completed, the button for that gesture is disabled. To make data collection more accurately, a user needs to explicitly press the "start" button (Fig. 2 (b)) and "stop" button (Fig. 2 (c)) to indicate the beginning and end of the gesture, respectively. The app uses Android sensor framework, specifically the `SensorManager` class, to access accelerometer and gyroscope readings, register and unregister sensor event listeners, and acquire sensor data at a specific delay acquisition rate. In the app we set the delay rate as `SENSOR_DELAY_FASTEST`, which may vary on different hardware. The rate is 100 Hz (±3 Hz) on our testing smart watch (Samsung Galaxy Gear), i.e., sensor readings are recorded every 10 milliseconds. Readings of 3-axis accelerometer and gyroscope as well as corresponding timestamps are recorded during gesture being performed and saved to specified files at the end of each gesture.

To facilitate analysis and evaluation, we developed a data processing suite in Matlab that implements all the modules of the MotionAuth framework. Extending the prototype to a comprehensive Android application that realizes the entire MotionAuth framework including verification will be our future work.

## V. Evaluation

### *A. Data Collection*

We conducted a user study to evaluate the viability of MotionAuth. We recruited 30 volunteers (24 males and 6 females) to participate in the study that spanned from June to Sept. in 2014. Among them, 7 are high school students, 20 are undergraduate or graduate students, 3 are college faculty.



(a) Circle (I)

(b) Down (II)

(c) Up (III)

(d) Rotation (IV)

Fig. 3. Illustration of the four gestures.

TABLE II
EER (%) OF THE HISTOGRAM (H) AND DTW (D) METHODS (LEAVE-ONE-OUT CROSS VALIDATION)

| Gesture | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ | $U_{11}$ | $U_{12}$ | $U_{13}$ | $U_{14}$ | $U_{15}$ | $U_{16}$ | $U_{17}$ | $U_{18}$ | $U_{19}$ | $U_{20}$ | $U_{21}$ | $U_{22}$ | $U_{23}$ | $U_{24}$ | $U_{25}$ | $U_{26}$ | $\mu \pm \sigma$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I (H) | 2.6 | 0.6 | 5.3 | 11.3 | 2.4 | 2.5 | 7.0 | 2.4 | 5.7 | 5.9 | 2.8 | 2.3 | 0.6 | 0.1 | 6.7 | 0.0 | 2.5 | 0.4 | 0.1 | 0.2 | 0.0 | 1.9 | 2.4 | 2.4 | 0.0 | 0.0 | 2.6 ± 2.8 |
| I (D) | 2.5 | 2.8 | 2.5 | 5.7 | 2.4 | 7.4 | 7.3 | 0.1 | 10.8 | 9.8 | 2.6 | 7.2 | 8.6 | 0.5 | 7.7 | 0.0 | 5.8 | 0.0 | 2.3 | 3.8 | 1.3 | 2.1 | 2.0 | 3.6 | 1.7 | 0.0 | 3.8 ± 3.2 |
| II (H) | 0.0 | 7.1 | 1.9 | 5.0 | 2.7 | 1.6 | 5.1 | 2.3 | 8.3 | 10.5 | 0.3 | 5.3 | 2.0 | 2.2 | 8.7 | 0.1 | 7.1 | 0.0 | 2.4 | 0.3 | 7.1 | 1.0 | 0.0 | 0.4 | 0.4 | 0.0 | 3.1 ± 3.2 |
| II (D) | 0.0 | 4.8 | 1.5 | 0.0 | 2.1 | 3.0 | 6.8 | 0.8 | 7.9 | 9.9 | 0.0 | 5.1 | 6.9 | 4.8 | 10.4 | 2.1 | 10.3 | 0.8 | 2.1 | 4.7 | 4.3 | 9.2 | 0.1 | 1.8 | 4.9 | 0.0 | 4.0 ± 3.5 |
| III (H) | 2.2 | 2.8 | 2.7 | 5.9 | 0.0 | 0.1 | 7.9 | 3.1 | 1.9 | 2.2 | 1.4 | 4.5 | 2.1 | 3.4 | 9.2 | 0.1 | 9.3 | 0.1 | 7.2 | 2.6 | 2.4 | 4.9 | 0.0 | 0.1 | 0.0 | 0.0 | 2.9 ± 2.9 |
| III (D) | 3.2 | 7.0 | 3.6 | 0.0 | 0.0 | 9.7 | 10.6 | 4.9 | 2.9 | 5.1 | 5.6 | 3.1 | 17.0 | 2.4 | 13.2 | 3.0 | 11.3 | 0.6 | 2.2 | 0.0 | 9.9 | 4.8 | 0.0 | 2.9 | 0.0 | 0.5 | 4.7 ± 4.6 |
| IV (H) | 0.0 | 2.7 | 0.0 | 14.4 | 5.5 | 2.5 | 5.0 | 2.9 | 11.1 | 0.2 | 5.0 | 18.2 | 2.8 | 1.8 | 22.7 | 2.6 | 5.8 | 0.6 | 0.2 | 0.1 | 7.4 | 4.9 | 0.6 | 4.3 | 0.2 | 0.0 | 4.7 ± 5.8 |
| IV (D) | 0.0 | 2.6 | 0.0 | 2.9 | 2.3 | 17.9 | 15.2 | 2.1 | 18.9 | 10.7 | 4.8 | 7.7 | 19.8 | 4.5 | 23.0 | 3.0 | 10.7 | 9.9 | 4.3 | 1.9 | 21.2 | 14.4 | 0.2 | 3.0 | 0.4 | 0.4 | 7.7 ± 7.5 |

The ages of participants range from 19 to 48. The study was reviewed and approved by the IRB of the authors' institution.

In the study each participant was asked to wear a Samsung Galaxy Gear smart watch and use the arm wearing the watch to perform the same set of 4 designated gestures each 10 times in one trial of experiment. Each participant was required to repeat the entire experiment 3 times, each in a different day. Actual intervals between two consecutive experiments span 3 to 7 days.

We assigned a unique integer to each participant as their ID to protect participants' privacy. At the beginning of data collection, we illustrated the testing gestures to each participant. Note that we do not ask participants to follow the demo gestures exactly. Instead, they were encouraged to perform those gestures in their own natural and comfortable manner. For example, different participants could make a "circle" gesture in different speed and shape.

One important question is what gestures should be tested for this study? As our ultimate goal is to deploy MotionAuth in real world where people often forget their authentication credentials, we set the following criteria for our gesture selection: natural and simple. We selected 3 simple gestures: arm down (marked as Down), arm up (Up), forearm rotation about 90 degree clockwise (Rotation). We also considered a more complex gesture commonly studied in gesture recognition, Drawing-A-Circle (Circle), as a complement to these simple gestures. The 4 gestures are illustrated in Fig. 3.

### *B. Data Analysis*

We use false acceptance rate (FAR), false rejection rate (FRR), and equal error rate (EER) as the metrics for the evaluation. FAR measures the likelihood of an unauthorized user being incorrectly accepted while FRR measures the likelihood of an authorized user being incorrectly rejected. EER is the rate when FAR and FRR are equal at a certain threshold value. In general, the lowest EER indicates the most accurate authentication decision. We evaluate authentication accuracy of MotionAuth in terms of EER for all the 4 gestures. As 4 out of 30 participants did not complete all the required experiments, their gesture data are not included in the evaluation. We collected 40 samples from each of the remaining 26 participants (i.e., the "users") for each gesture and in total 4,160 gesture samples are used in our evaluation. To measure a gesture's FRR for each user, we use leave-one-out cross validation to test that user's samples of the gesture. To measure a gesture's FAR for each user, all other users' samples of that gesture are treated as impostors' gesture samples and are tested against the genuine user's template. All testing results are represented by similarity scores from which we calculate FAR and FRR and derive EER.

Table II shows the EER value (in %) for each user and each gesture obtained by using the Histogram and DTW methods. We can see that MotionAuth achieves very high accuracy (close to 0 EER) in verification for some users (e.g., $U_{26}$). For all four gestures, a gesture's average EER value given by the Histogram method is smaller than that attained by the DTW method although two methods' accuracy varies per individual user. Among the four gestures, Circle achieves the lowest EER (2.6% mean for Histogram and 3.8% for DTW) while Rotation gives the highest EER (4.7% mean for Histogram and 7.7% for DTW). Surprisingly, even simple Down and Up gestures can achieve pretty good accuracy (no more than 5% on average) and the accuracy of Down is closely comparable to that of Circle. This may be explained by the fact that both Down and Up use the whole arm while Rotation only uses the forearm, which makes it less random and unique. Some users such as $U_{15}$ have much higher EER values across all four gestures, which suggests that some users may have difficulty in performing certain gestures consistently assuming no bias or error introduced in data collection. Overall, small EERs achieved by two different methods make it promising to apply MotionAuth in practice and motivate us to explore better verification techniques.

Figure 4 shows the EER distribution of each gesture for

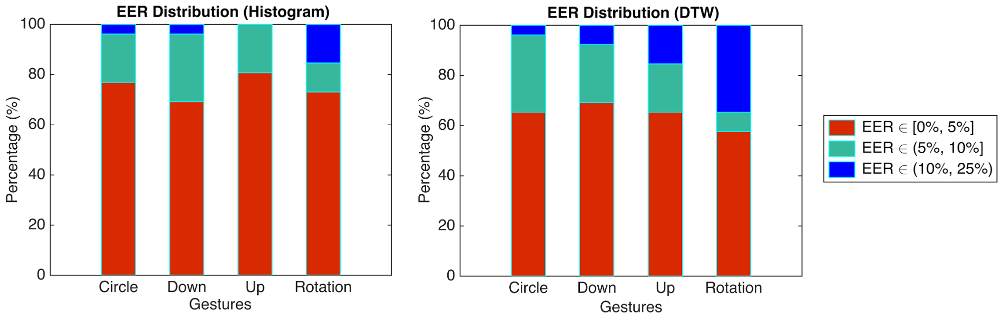


Fig. 4. User Distribution of EER for both Histogram and DTW

both the Histogram and DTW methods. It is very clear that the higher the proportion of users with large EER values (more than 10%), the worse the gesture's accuracy is. In general, the majority of user's EER values are lower than 5% for all the tested gestures.

## VI. Conclusion

In this paper, we presented a motion based user authentication scheme called MotionAuth for effectively verifying a person wearing a wrist worn smart device. Compared to conventional PIN or password based authentication, We applied two distinct algorithms—Histogram and DTW—for verification and implemented MotionAuth based on Android platform. We conducted a user study in which 30 people were involved. Our experimental results show that simple, natural gestures can achieve EER values lower than 5%. As an ongoing study, we will continue the investigation of gesture testing, algorithm design, and mobile application development with a large-scale system assessment in future.

## References

[1] A. Akl, C. Feng, and S. Valaee. A novel accelerometer-based gesture recognition system. *IEEE Transactions on Signal Processing*, 59(12):6197–6205, 2011.

[2] E. T. Anzaku, H. Sohn, and Y. M. Ro. Privacy preserving facial and fingerprint multi-biometric authentication. In *Proc. the 9th Intl. Conf. on Digital watermarking*, IWDW'10, pages 239–250, 2011.

[3] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP J. Appl. Signal Process.*, 2004:430–451, Jan. 2004.

[4] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proc. the 2012 IEEE Symposium on Security and Privacy*, pages 553–567, 2012.

[5] C. Cornelius, R. Peterson, J. Skinner, R. Halter, and D. Kotz. A wearable system that knows who wears it. In *Proc. MobiSys '14*, pages 55–67, 2014.

[6] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: Implicit authentication based on touch screen patterns. In *Proc. CHI '12*, pages 987–996, 2012.

[7] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, 1 2013.

[8] D. V. Klein. Foiling the cracker: A survey of, and improvements to, password security. In *Proc. the 2nd USENIX Security Workshop*, pages 5–14, 1990.

[9] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, Mar. 2011.

[10] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive Mob. Comput.*, 5(6):657–675, Dec. 2009.

[11] A. Narayanan and V. Shmatikov. Fast dictionary attacks on passwords using time-space tradeoff. In *Proc. the 12th ACM Conf. on Computer and Communications Security*, pages 364–372, 2005.

[12] H. S. Own, W. Al-Mayyan, and H. Zedan. Biometric-based authentication system using rough set theory. In *Proc. the 7th Intl. Conf. on Rough sets and current trends in computing*, RSCTC'10, pages 560–569, 2010.

[13] P. Phillips, J. Beveridge, B. Draper, G. Givens, A. O'Toole, J. Bolme, D.; Dunlop, Y. M. Lui, H. Sahibzada, and S. Weimer. An introduction to the good, the bad, & the ugly face recognition challenge problem. In *Proc. the IEEE Intl. Conf. on Automatic Face Gesture Recognition*, pages 346–353, 2011.

[14] N. Ratha, J. Connell, and R. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[15] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: A novel approach to authentication on multi-touch devices. In *Proc. CHI '12*, pages 977–986, 2012.

[16] N. Sae-Bae and N. Memon. Online signature verification on mobile devices. *IEEE Transactions on Information Forensics and Security*, 9(6):933–947, June 2014.

[17] M. Shahzad, A. X. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it. In *Proc. the 19th Annual Intl. Conf. on Mobile Computing and Networking*, MobiCom '13, pages 39–50, 2013.

[18] M. Sherman, G. Clark, Y. Yang, S. Sugrim, A. Modig, J. Lindqvist, A. Oulasvirta, and T. Roos. User-generated free-form gestures for authentication: Security and memorability. In *Proc. MobiSys '14*, pages 176–189, 2014.

[19] Q. Tang, J. Bringer, H. Chabanne, and D. Pointcheval. A formal study of the privacy concerns in biometric-based remote authentication schemes. In *Proc. the 4th Intl. Conf. on Information security practice and experience*, ISPEC'08, pages 56–70, 2008.

[20] U. Uludag and A. K. Jain. Attacks on biometric systems: a case study in fingerprints. In *Proc. SPIE, Volumn 5306, Security, Steganography, and Watermarking of Multimedia Contents VI*, pages 622–633, 2004.

[21] L. Whitney. Hacker video shows how to thwart apple's touch id. http://www.cnet.com/news/hacker-video-shows-how-to-thwart-apples-touch-id/, September 2014.

[22] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li. Gesture recognition with a 3-d accelerometer. In *Proc. the 6th Intl. Conf. on ubiquitous intelligence and computing (UIC)*, pages 25–38, 2009.

[23] H. Xu, Y. Zhou, and M. R. Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 187–198, July 2014.

[24] N. Zheng, K. Bai, H. Huang, and H. Wang. You are how you touch: User verification on smartphones via tapping behaviors. In *Proc. ICNP'14*, 2014.

[25] N. Zheng, A. Paloski, and H. Wang. An efficient user verification system via mouse movements. In *Proc. the 18th ACM Conf. on Computer and communications security*, CCS '11, pages 139–150, 2011.