Secure Behavioral Biometric Authentication with Leap Motion

Grady Xiao[1], Mariofanna Milanova[2], and Mengjun Xie[2]
[1] *Johns Hopkins University, Email: gxiao2@jhu.edu*
[2] *University of Arkansas at Little Rock, Email: {mgmilanova, mxxie}@ualr.edu*

***Abstract*—In this paper we examine the effectiveness of user authentication using biometrics and behavioral motion data captured by the Leap Motion sensor. The biometrics data is derived from the user's hand and the behavioral motion data is generated when the user signs his or her signature using his or her hand in front of the sensor. We have developed a prototype system to collect experiment data from 10 participants and used the data to analyze the accuracy and effectiveness of our authentication method. The experimental results are measured by FAR, FRR, and EER. For the hand biometrics data involving 17 genuine hand samples and 162 attacking ones for each of the 10 users, the system has achieved an average EER of 34.80%. For the behavioral signature motion data involving 17 genuine samples and 262 attacking samples for each of the 10 users, the system has achieved an average EER of 3.75%. Our study indicates that behavioral biometrics with Leap Motion is a viable authentication approach.**

I. INTRODUCTION

The Leap Motion sensor is a camera-based sensing device that captures gestures and motion data from a user as input to a computing device [1]. Its capabilities allow users to play video games, control their mouse, enhance user interaction and so on. In this study, we extend the functionality of Leap Motion to use it for user authentication, which leverages users' hand biometrics and the 3-dimensional (3D) data generated by their motion when they simulate a signature above the sensor. Since the hand biometrics and the behavioral signature motion are difficult to mimic and less susceptible to theft, this new authentication method can potentially replace traditional authentication methods such as passwords, facial recognition, and fingerprints.

There are many user authentication methods. Among them, passwords and biometrics are widely used on computers and mobile platforms. Text-based passwords are dominant authentication method but have many well-known problems such as strong passwords being difficult to remember. For pattern-based passcode method where a user draws a preset pattern to gain access to the system, attackers can easily crack simple patterns and complex patterns can be difficult to draw and remember. A study by Andriotis *et al.* shows that the attackers were able to recover 54.54% of the patterns based on factors such as finger marks, oily residue, and smudges [4]. The accuracy of pattern-based password can achieve 77% accuracy with 19% false rejection rate (FRR) and 21% false acceptance rate (FAR) or around an Equal Error Rate (EER) of 23% [6].

Facial recognition and fingerprints become popular on mobile platforms. A user can gain access to a smartphone by presenting his or her face in front of the front-facing camera.

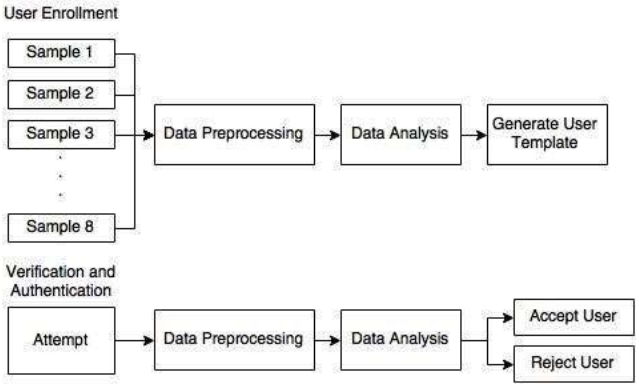

Figure 1. Process of user authentication in our study

Although this method can achieve an average FRR of 12% at 0.1% FAR [7] and an EER of 6% [8] in previous studies, attackers may be able to use a photo or video of the user to bypass the authentication method [5]. Fingerprint based methods can achieve an EER of 4.5% using cameras on a mobile phone [9] and 0.02% using high quality images [10]. However, fingerprints are vulnerable to theft. According to *Business Insider*, fingerprints can be easily reproduced from a photograph [11].

The Leap Motion sensor captures user's hand actions above the camera and translates them into 3D input using the two infrared cameras and the infrared LED within the sensor [2]. When activated, the sensor can constantly captures frames up to 200 frames per second, depending on the working environment. A user can interact within the 8 cubic feet interaction space directly above the sensor, with the most accurate range between 25 mm to 250 mm above the sensor [3], which allows a user to have close and seamless interaction with the user environment at an affordable price ($79.99 for one unit in early 2015).

We study the effectiveness of user authentication using hand biometrics and behavioral motion data captured by the Leap Motion sensor. The biometrics data is derived from the user's hand and the behavioral motion data is generated when the user signs his signature using his hand in front of the sensor. The general process of authentication in our study is shown in Figure 1, which includes sample collection, data preprocessing and analysis, user template generation, and template based user verification.

We have developed a prototype system and recruited 10 participants for data collection in our experiments. The experimental results are measured by metrics commonly used in authentication, i.e., FAR, FRR, and EER (in 95% confidence interval). For the hand biometric data that involves 17 genuine hand samples and 162 attacking ones for each of the 10 users, the system has achieved an average EER

ISBN: 978-1-4673-9865-7/16/$31.00 ©2016 IEEE

of 34.80%. For the behavioral signature motion data that involves 17 genuine samples and 262 attacking samples for each of the 10 users, the system has achieved an average EER of 3.75%. Our study has indicated that behavioral biometric authentication with Leap Motion is practical and effective.

The rest of this paper is organized as follows: Section II briefly describes background and related work. Sections III and IV present the methods for data acquisition and analysis respectively. Section V details the analysis of experimental results. Section VI concludes this paper.

II. Background and Related Work

Although the Leap Motion technology is relatively new, many studies have been conducted regarding the accuracy and shape recognition capability of Leap Motion sensors, and its application to user authentication using motion gestures.

The Leap Motion maker claims that the sensor can achieve the level of 0.1 mm accuracy. However, the research conducted by Weichert *et al.* shows that the accuracy of the Leap Motion sensor is actually around 0.4 mm [12]. This implies that the biometric data from the hand can be used to filter out users with significantly different hands but may not be sufficient to reliably authenticate users.

Lavy and Pham studied recognizing shapes generated by handwriting captured through a Leap Motion sensor [13]. Their study first transformed Leap Motion's 3D data into 2D data using the Least Square Fitting method, then extracted features of the shapes using Fourier Descriptor of Shape (FDS), and applied the KNN classification algorithm for recognition. Their study shows the feasibility of using the Fourier Descriptors for verification.

SignWave, a Leap Motion application, was released for authentication security shortly after the release of the sensor. The purpose of SignWave was to verify and authenticate users based on their motion gestures and their biometrics data [14]. Although SignWave can recognize motion gestures and differentiate motion gestures, it failed to distinguish a genuine user from imposters based on their biometric data [15]. It was eventually pulled from the Leap Motion App store due to its high false acceptance rate (FAR).

Nigam *et al.* have studied on verifying users using their signature captured by Leap Motion. Their study only used the 3D position (*X*/*Y*/*Z*) of the finger as the verification factor [16]. 3D Histogram of Oriented Trajectories (HOT) and Histogram of Oriented Optical Flow (HOOF) were used as the verification algorithms. The result of 91% accuracy rate was achieved after combining the Leap Motion sensory data with facial recognition, which adds an additional hardware requirement for verification. Although their study involves signature authentication, the data type, verification algorithm and components, and data collection differ significantly from our research. Their study shows that the 3D position of a finger can be accurately captured by the Leap Motion.

KinWrite is another work on verifying users using their handwriting [17]. However, the study is very different from our work in that it was conducted using the Microsoft Kinect sensor [18]. Unlike the built-in finger-tracking functions provided by the Leap Motion, the Kinect sensor (version 1) did not offer any fingertip tracking in its software development kit (SDK). Instead, KinWrite tracked the fingertip by recording the pixel with the minimum-depth value in each frame, which might not be the finger at all. At the time of this work, the Kinect (version 2) SDK for the newer Kinect sensor only provides tracking for thumb and hand instead of all fingertips [21][22]. Unlike the frame capture rate of up to 200 fps for Leap Motion, the Kinect sensor can only capture frames up to 30.00 fps [19]. Therefore, KinWrite had to implement an algorithm to interpolate the user's motion between gaps in the frames captured. The Leap Motion sensor is also 200 times more sensitive to the user's motion than the Kinect [20], requiring no smoothing algorithm to draw the signature and lowering the false estimation for the positions.

III. Data Acquisition

The Leap Motion SDK allows developers to easily obtain useful information from frames that are captured by the Leap Motion sensor [23]. When obtaining the hand biometrics data, we collect the finger lengths of each finger on the user's dominant hand. When obtaining the behavioral motion data, we collect the normalized fingertip position of the fingertip in 3D space, the magnitude of the velocity of the normalized fingertip, and the direction of the normalized fingertip (in pitch, roll, and yaw).

A. Data Filtering

In order to avoid capturing unnecessary frames, we use functions provided by the Leap Motion SDK to create criteria that can filter out unnecessary or inaccurate frames. Such criteria include:

- Confidence [24]: During the data collection process, the user may move his or her hand slightly out of the confident detection zone of the sensor. These inaccurate frames can be eliminated by filtering out the frames where the confidence of the frame is less than 0.20 on a scale from 0 to 1 by using the following condition in an IF statement:

  ```
  hand.confidence() >= 0.20
  ```

- Hand detection [25]: Assuming the user will be using only one hand to verify hand biometrics and simulate a signature, we discard unnecessary data by filtering the frames based on hand detection. Since the Leap Motion sensor constantly captures frames, we can filter out those frames where no hand is detected or multiple hands are detected with the following condition:

  ```
  frame.hands().count() == 1
  ```

- Finger number detection for biometrics [26]: Assuming the user will be stretching all five fingers for the hand biometric collection, we can filter out the frames with 4 or less detected fingers with the following condition:

  ```
  hand.fingers().extended().count() == 5
  ```

- Finger number detection for signature [26]: Assuming the user will be only extending index finger and maybe thumb finger for signature, we can filter out the frames with 3 or more detected fingers for signature collection and make sure the front-most extended finger is the index finger by using the following conditions:

```
(hand.fingers().extended().count() <= 2) &&
(frontmostFinger.type() ==
frontmostFinger.type().TYPE_INDEX)
```

B. Data Preprocessing

Due to the size of the interaction space above the sensor, the user may simulate his or her signature at different positions each time. In order to accommodate this, we offset the signature by its starting position to minimize the differences. By storing the starting position and offsetting the signature position by its starting position, we can capture the signature's position relative to its starting position by using the following code:

```
startX = frame.interactionBox().normalizePoint(
        frontmostFinger.tipPosition(), false).getX();

startY = frame.interactionBox().normalizePoint(
        frontmostFinger.tipPosition(), false).getY();

startZ = frame.interactionBox().normalizePoint(
        frontmostFinger.tipPosition(), false).getZ();
```

C. Procedure

For the hand biometric data acquisition, each user is asked to extend all five fingers above the Leap Motion sensor until the sensor captures 50 frames of biometrics information about all five fingers. For the behavioral motion data acquisition, each user is asked to simulate his or her signature above the Leap Motion sensor at his or her natural speed, allowing the sensor to capture the necessary information about the fingertip motion.

Both the hand biometrics data and the behavioral motion data are collected in each trial. A total of 18 genuine trials is performed by each user over a span of 5 days. Attackers perform an additional 10 insider attack trials for each user.

D. Graphical User Interface (GUI)

The GUI of data collection for this research is designed using the Java Swing library [27], which makes it easier for a user to capture hand biometrics and signature data. The GUI does not include online authentication. User verification is conducted offline for extensive experimentation and analysis in this study. The GUI has the following four pages.

Terms and Conditions. It is the homepage with the Terms and Conditions for the users, which has been approved by the Institutional Review Board at the University of Arkansas at Little Rock.

Participant Configuration. It is the page for participant ID input and configurations. A user can configure the program to draw the signature when the program is collecting the behavioral signature motion data. Since the drawing can decrease the natural speed and accuracy of the data, the default setting is not to draw user signature. The user can also configure the program to display the raw data being captured. By default, the raw data is hidden.

Figure 2. Home page of the GUI

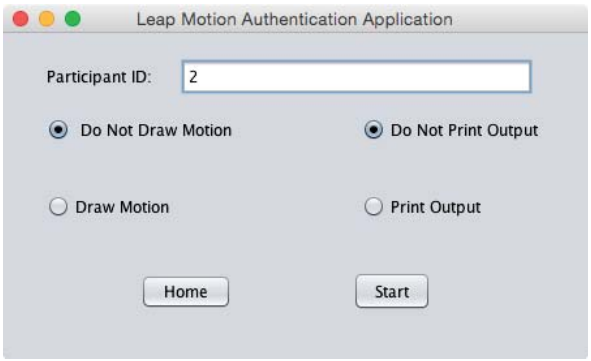

Figure 3. Configuration page of the GUI

Hand Biometrics Collection. This page displays the collection progress for the hand biometrics data. The page turns red when the confidence is less than 0.2, green when the confidence is equal to or greater than 0.2, or blue when the biometric data collection process is complete.

Behavioral Signature Motion Collection. This page displays the collection progress of behavioral signature motion data. It turns red when the confidence of the hand falls below 0.2, when more than 2 fingers are detected, or when the front-most finger is not the index finger. Otherwise, the screen turns green. If configured, the page can display the position of signature in 2D in real time.

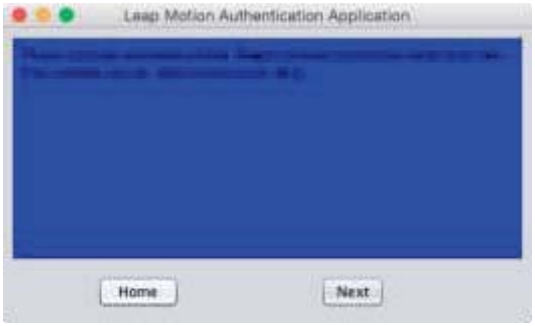

Figure 4. Hand biometrics collection page with color change

Figure 5. Signature motion collection page with color change

IV. Methods For Data Analysis

After a user's hand biometrics and behavioral signature data are captured, several algorithms can be taken into consideration for data analysis including the Fourier Descriptors, the Seven Invariant Moments, and the Dynamic Time Warping (DTW). After comparison, the DTW algorithm is chosen for data analysis in this study.

A. Fourier Descriptors

The Fourier Descriptors method uses Discrete Fourier Transform (DFT) [28] to extract features about a shape based on its positions in *X*/*Y* axis and return descriptor coefficients that describe the shape. The Inverse DFT can be applied to the coefficients to view the represented shape (shown in Figures 6 and 7). From the algorithm [29], a set of feature descriptors coefficients are returned. With these descriptor coefficients, the features represented by the coefficients can become scale invariant, rotational invariant, and translational invariant [13][30].

- Scale invariant – For sets of shapes in different scales, the shapes are considered the same since the scale of the shapes does not matter. In order to achieve this, all of the Fourier descriptors coefficients must be divided by the second coefficient.
- Rotational invariant – For sets of shapes in different rotations, the shapes are considered the same since the rotation of the shapes does not matter. In order to achieve this, the first coefficient is set to 0.
- Translational invariant – For sets of shapes at different positions, the shapes are considered the same since the translation of the shapes does not matter.

Although the Fourier Descriptors are useful to create coefficients of shapes that are rotational invariant, scale invariant, and translational invariant, the Fourier Descriptors can only take in the positions as variables for the descriptors. The results of the Fourier Descriptors may also differ significantly even if the signatures vary in a minor manner, requiring each trial to capture constant number of frames and only a small amount of variation. However, since user's signature may vary in time, position, and direction from trial to trial, the Fourier Descriptors is not adopted.

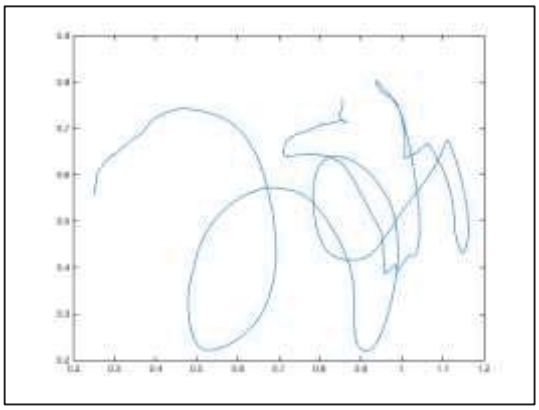

Figure 6. Sample signature before DFT

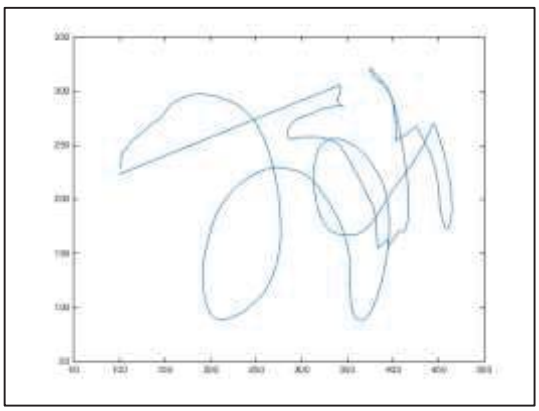

Figure 7. Reconstructed signature after DFT

B. Seven Invariant Moments (Hu's Invariant)

The Invariant Moments algorithm [31] performs in a similar manner as the Fourier Descriptors by returning coefficients that can be used to describe the images. However, Hu's Invariant fails to accommodate slight differences in rotation and scale for similar signatures [32]. Since a user's signature may vary slightly in scale and rotation, the Hu's Invariant is not applied in this study either.

C. Dynamic Time Warping (DTW)

The DTW algorithm [33] calculates the difference (or distance) between two datasets: $dtw\_c(a,b)$ where a represents the first dataset and b represents the second dataset. The advantages of using DTW over Fourier Descriptors or Invariant Moments are two-fold: (1) Any type of numerical data can be applied to DTW, allowing the input not to be limited to *X/Y/Z* positions but to include more such as magnitude of the velocity and the direction's pitch, roll, and yaw; (2) The number of frames collected in each trial can vary instead of being constant, tolerating the slight variation in the number of frames captured in each trial.

The DTW method is chosen in that it is simple, versatile and has shown to be effective in many authentication studies. To use DTW in user verification, a threshold is needed to distinguish a true user (distances smaller than the threshold) from imposters. The threshold should accommodate slight changes in scale, rotation, speed, direction, and position between genuine signatures. In our study, we select one sample from training samples to derive a user's template.

For data analysis of this study, MATLAB [34] is employed due to easy implementation of DTW and data processing in MATLAB. For each participant, 18 genuine samples for hand biometrics data and for behavioral signature motion data were collected. The first 8 samples are used to derive the user template. For each of the 8 samples, we apply DTW between it and each of the rest and calculate the summation of the distances. We select the sample with the smallest total DTW distance away from all others to be the user template.

After deriving the user template, an array of DTW distances for genuine samples is generated by calculating the DTW distance between the user template and the other 17 genuine samples. An array of DTW distances for attacking samples is also generated by calculating the DTW distances between the user template and the insider attacking samples and between the template and all the samples from other users.

The arrays of genuine and attacking DTW distances are then used to generate a histogram showing the distribution of DTW distances for genuine and attacking samples. This can be used to determine the optimal threshold for the user. Given a threshold, a testing sample will be accepted if its DTW distance is below the threshold or rejected otherwise.

An ROC curve is also generated in data analysis, showing the relationship between the false acceptance rate and false rejection rate at different thresholds. The results of the collection process are measured in terms of false acceptance rate (FAR), false rejection rate (FRR), and equal error rate (EER) [35].

- Receiver Operating Characteristics (ROC) curve – The ROC curve shows the relationship between FAR and FRR, and can be used to determine the EER.
- False Acceptance Rate (FAR)
$$FAR = \frac{\#\ of\ false\ positives}{\#\ of\ false\ positives + \#\ of\ true\ positives}$$
where a true positive is a genuine sample being correctly accepted and a false positive is an attacking sample being wrongly accepted.
- False Rejection Rate (FRR)
$$FRR = \frac{\#\ of\ false\ negatives}{\#\ of\ false\ negatives + \#\ of\ true\ negatives}$$
where a true negative is an attacking sample being correctly rejected and a false negative is a genuine sample being wrongly rejected.
- Equal Error Rate (ERR): The value when FAR equals the FRR.

V. Experimental Results

During the data collection process, both genuine hand biometrics and signature motion samples and insider attack signatures were collected for each of the 10 participants. For each participant, the genuine user's DTW distance array for hand biometrics data and the one for behavioral signature data contain 17 values (excluding the sample used as the template). The attacking DTW distance array for hand biometrics data contains 162 values (9*18, excluding the user's genuine samples). The attacking DTW distance array for behavioral signature motion data contains 262 values (9 users with 18 samples for each user plus 10 users with 10 insider attack samples for each). Using the genuine and attacking DTW distance arrays, a histogram and ROC curves for both hand biometrics data and signature motion data are generated for each participant.

The histogram of each user shows the distribution of the DTW distances for genuine samples and attacking ones. This histogram is then used to determine the threshold. For example, in Figure 8, the optimal threshold for participant 2 is around 300. An attempt with DTW distance of 300 or below from the user's template will be accepted while an attempt with DTW distance of 301 or above from the user's template will be rejected.

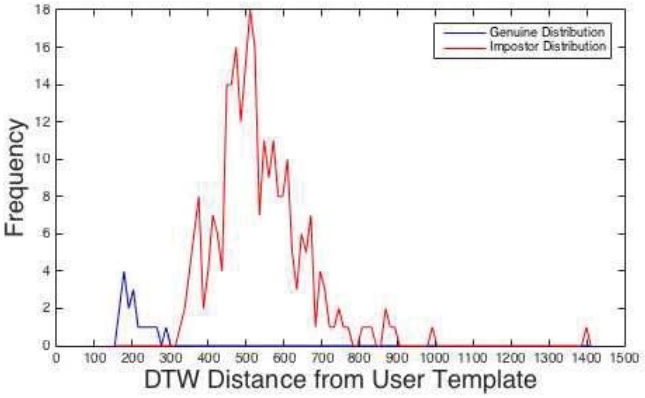

Figure 8. Histogram of DTW distances for signature data of participant 2

The ROC curves for hand biometrics data from all the participants are displayed in Figure 9. From the ROC curves, we can determine the relationship of FAR and FRR. As shown in Table I, the average EER (%) of the hand biometrics data is 34.8%, meaning that on average, 65.2% of the attacking attempts are rejected and 65.2% of the true user

attempts are accepted. This EER may be attributed to that the Leap Motion appears to snap the raw hand data to some preexisting built-in hand models. After capturing the first frame, the data provided by the Leap Motion sensor seems to be the finger lengths of the hand model instead of the actual raw data that the sensor is reading in. When only a few hand models are used in Leap Motion SDK, similar hands will be represented by the same model even if there are slight differences in the hands, rendering the accuracy to be insufficient for identifying imposters. Due to this higher than expected EER value, the hand biometrics data cannot be relied on alone in authentication, but should be combined with other verification methods.

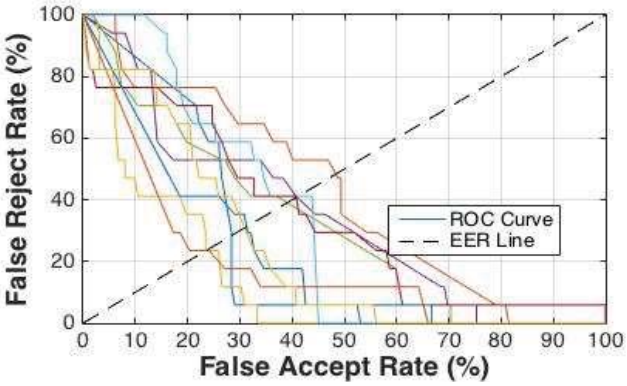

Figure 9. ROC curves of hand biometrics data (each colored line representing a participant)

TABLE I. EER VALUES FOR HAND BIOMETRICS (FIGURE 9) AND FOR BEHAVIORAL SIGNATURE MOTION DATA (FIGURE 10)

Participants	EER (%, hand biometrics)	EER (%, behavioral)
1	28	6.5
2	48	0
3	24	0
4	41	16
5	39	4
6	41	0
7	41	6
8	31	1
9	24	2
10	31	2
Average	**34.80**	**3.75**
Stdev	**8.27**	**4.93**
95% CI	**5.13**	**3.05**
Max	**48**	**16**
Min	**24**	**0**
# of Attempts	**1.53**	**1.04**

The ROC curves for behavioral signature motion data from all the 10 participants are displayed in Figure 10. As shown in Table I, the EER of the signature data is 3.80%, that is, on average 96.20% of the true user attempts are accepted and 96.20% of the attacking attempts are rejected. On average, a user only needs to perform 1.039 attempts to be authenticated. This low EER indicates that the behavioral signature motion authentication is effective.

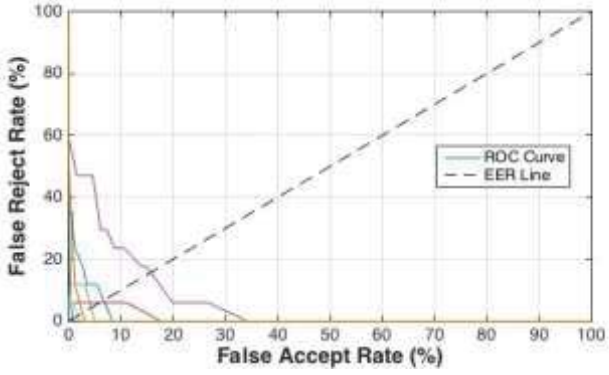

Figure 10. ROC curves of behavioral signature motion data with each colored line representing a participant (three overlap at EER of 0)

In order to further examine the accuracy of the behavioral biometric signature data, we also derive the FRR values of the system when none of the attacking attempts is accepted, i.e., when FAR is 0. We find that the system achieves the FRR of 24.71 ± 14.05% when FAR is 0, meaning that on average 75.29% of the true user attempts are accepted when all of the imposters are rejected. On average, the system can identify a genuine user with 1.328 attempts. This result might be further lowered if more genuine samples were collected and used in training.

As shown in Table I and also observed in the experiments where FAR is 0, the behavioral signature motion system achieved an EER of 0 for 3 of the 10 participants, showing that the system is capable of differentiating genuine user attempts from attacking attempts all the time for the 3 participants. After analyzing and comparing the data from users with high EERs to those with low EERs, two major factors are identified that appear to affect the results: (1) consistency of the collected signatures and (2) complexity of a signature, with the former affecting most. As the consistency of the genuine attempts increases, the system in general is able to distinguish the attacking attempts from genuine ones more accurately, and likewise, a more accurate user template can be generated. Although the complexity of signature does not affect results substantially, a small correlation between the complexity of signature and the EER result was observed in the experiments.

VI. CONCLUSION

This paper has presented the design and evaluation of a behavioral biometric authentication system using the Leap Motion sensor. As shown in the experimental results, the system designed in our study can reach an average EER of 3.8%. For FAR being 0, the system can still reach a FRR of 0 for some users, while the average is 24.71%. This result may improve further if more user samples were collected and used to generate the user template. Compared to about 6% EER of facial recognition (vulnerable to attacks using

photos and videos) and 0.02-4.5% EER of fingerprint recognition (vulnerable to attacks using reproduced fingerprints), the 3% EER of the behavioral signature verification using Leap Motion shows a high potential of an effective secure authentication method. Some future work includes collecting more samples, user template protection, and development of an application that authenticates users in real time.

ACKNOWLEDGMENT

This work is supported in part by the Research Experiences for Undergraduates (REU) Program of the National Science Foundation under Award Number 1359323.

REFERENCES

[1] "Leap Motion." [Online]. Available: https://www.leapmotion.com/product/desktop. [Accessed: 15-Jun-2015].

[2] "Leap Motion Developers." [Online]. Available: https://developer.leapmotion.com/articles/intro-to-motion-control. [Accessed: 14-Jun-2015].

[3] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik, "An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking," *Sensors*, vol. 14, no. 2, pp. 3702–3720, Feb. 2014.

[4] T. T. Panagiotis Andriotis, "A pilot study on the security of pattern screen-lock methods and soft side channel attacks," pp. 1–6, 2013.

[5] "A Survey on User-Device Authentication on Emerging HCI Interfaces," *J. Latex Cl. Files*, vol. 6, no. 1, Jan. 2007.

[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 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 987–996.

[7] "IEEE Xplore Abstract - FRVT 2006 and ICE 2006 Large-Scale Experimental Results." [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4803846. [Accessed: 31-Jul-2015].

[8] A. Pabbaraju and S. Puchakayala, "Face Recognition in Mobile Devices," *Univ. Mich. Ann Arbor*.

[9] M. O. Derawi, B. Yang, and C. Busch, "Fingerprint Recognition with Embedded Cameras on Mobile Phones," in *Security and Privacy in Mobile Information and Communication Systems*, R. Prasad, K. Farkas, A. U. Schmidt, A. Lioy, G. Russello, and F. L. Luccio, Eds. Springer Berlin Heidelberg, 2012, pp. 136–147.

[10] D. Gafurov, P. Bours, B. Yang, and C. Busch, "GUC100 Multi-scanner Fingerprint Database for In-House (Semi-public) Performance and Interoperability Evaluation," in *2010 International Conference on Computational Science and Its Applications (ICCSA)*, 2010, pp. 303–306.

[11] "Hackers Say They Can Copy Your Fingerprint From Just a Photograph," *Gizmodo*. [Online]. Available: http://gizmodo.com/chaos-computer-club-says-they-can-hack-your-fingerprint-1675845311. [Accessed: 31-Jul-2015].

[12] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the Accuracy and Robustness of the Leap Motion Controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, May 2013.

[13] D. Lavy and D. Pham, "Virtual Shape Recognition using Leap Motion," *Boston Univ. Boston MA*, May 2015.

[14] "Battelle SignWave(TM) Unlock App for Leap Motion Lets You Wave Goodbye to Passwords," *Marketwire*. [Online]. Available: http://www.marketwired.com/press-release/battelle-signwave-unlock-app-for-leap-motion-lets-you-wave-goodbye-to-passwords-1814268.htm. [Accessed: 15-Jun-2015].

[15] "Hacking Leap Motion apps: Security researchers spoof biometric auto-login system | VentureBeat | Security | by John Koetsier." [Online]. Available: http://venturebeat.com/2013/08/13/hacking-leap-motion-security-researchers-spoof-biometric-auto-login-airspace-app/. [Accessed: 20-Jun-2015].

[16] I. Nigam, M. Vatsa, and R. Singh, "Leap signature recognition using HOOF and HOT features," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 5012–5016.

[17] J. Tian, C. Qu, W. Xu, and S. Wang, "KinWrite: Handwriting-Based Authentication Using Kinect," in *Proceedings of the 20th Annual Network & Distributed System Security Symposium*, 2013.

[18] "Kinect for Windows." [Online]. Available: https://www.microsoft.com/en-us/kinectforwindows/default.aspx. [Accessed: 20-Jul-2015].

[19] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, Feb. 2012.

[20] "Leap Motion launches 'world's most accurate 3-D motion' gesture controller today," *VentureBeat*. [Online]. Available: http://venturebeat.com/2013/07/22/leap-motion-finally-launches-worlds-most-accurate-3-d-motion-gesture-controller/. [Accessed: 14-Jun-2015].

[21] "Kinect for Windows features." [Online]. Available: https://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx. [Accessed: 20-Jul-2015].

[22] "finger tracking not working." [Online]. Available: https://social.msdn.microsoft.com/Forums/en-US/7011ee48-b913-4e32-8209-7987771a4d54/finger-tracking-not-working?forum=kinectv2sdk. [Accessed: 15-Jul-2015].

[23] "Leap Motion Developers." [Online]. Available: https://developer.leapmotion.com/downloads. [Accessed: 14-Jun-2015].

[24] "Data Confidence | Leap Motion Developers." [Online]. Available: https://developer.leapmotion.com/gallery/data-confidence. [Accessed: 12-Jun-2015].

[25] "Frame — Leap Motion Java SDK v2.2 documentation." [Online]. Available: https://developer.leapmotion.com/documentation/java/api/Leap.Frame.html#javaclasscom_1_1leapmotion_1_1leap_1_1_frame_1a06e1aae4587d103bdffc62ae2f92d0b1. [Accessed: 10-Jun-2015].

[26] "Finger — Leap Motion Java SDK v2.2 documentation." [Online]. Available: https://developer.leapmotion.com/documentation/java/api/Leap.Finger.html#id30. [Accessed: 13-Jun-2015].

[27] "Lesson: Learning Swing with the NetBeans IDE (The Java™ Tutorials > Creating a GUI With JFC/Swing)." [Online]. Available: http://docs.oracle.com/javase/tutorial/uiswing/learn/index.html. [Accessed: 15-Jun-2015].

[28] "How to implement the discrete Fourier transform." [Online]. Available: http://www.nayuki.io/page/how-to-implement-the-discrete-fourier-transform. [Accessed: 15-Jun-2015].

[29] V. Saxena, "Fourier descriptors under rotation, scaling, translation and various distortion for hand drawn planar curves," *J. Exp. Sci.*, vol. 3, no. 1, pp. 5–7, 2012.

[30] "Wolfram Demonstrations Project." [Online]. Available: http://demonstrations.wolfram.com/FourierDescriptors/. [Accessed: 27-Jun-2015].

[31] "The Seven Invariant Moments - File Exchange - MATLAB Central." [Online]. Available: http://in.mathworks.com/matlabcentral/fileexchange/33975-the-seven-invariant-moments. [Accessed: 27-Jun-2015].

[32] Z. Huang and J. Leng, "Analysis of Hu's moment invariants on image scaling and rotation," in *2010 2nd International Conference on Computer Engineering and Technology (ICCET)*, 2010, vol. 7, pp. V7–476–V7–480.

[33] "Dynamic Time Warping (DTW) - File Exchange - MATLAB Central." [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/43156-dynamic-time-warping--dtw-. [Accessed: 20-Jun-2015].

[34] "MATLAB - The Language of Technical Computing." [Online]. Available: http://www.mathworks.com/products/matlab/. [Accessed: 30-Jun-2015].

[35] N. Sae-Bae and N. Memon, "Online Signature Verification on Mobile Devices," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 6, pp. 933–947, Jun. 2014.