An Investigation on Touch Biometrics: Behavioral Factors on Screen Size, Physical Context and Application Context

Tao Feng, Xi Zhao, Nick DeSalvo, Tzu-Hua Liu, Zhimin Gao, Xi Wang and Weidong Shi
Computer Science Department, University of Houston
Email: tfeng3@cs.uh.edu

***Abstract*—With increasing privacy concerns and security demands present within mobile devices, behavioral biometric solutions, such as touch based user recognition, have been researched as of recent. However, several vital contextual behavior factors (*i.e.,* screen size, physical and application context) and how those effect user identification performance, remains unaddressed in previous studies. In this paper we first introduce a context-aware mobile user recognition method. Then a comparative experiment to evaluate the impacts of these factors in relation to user identification performance is presented. Experimental results have demonstrated that a user's touch screen usage behavior may be affected given different contextual behavior information. Furthermore, several interesting occurrences have been found in the results: 1) screen size of a smartphone device changes the way a user touches and holds the device. A larger screen size will provide more potential methods of interacting with the device and in effect, a higher user recognition accuracy as well; and 2) application context and physical activity context can aid in achieving higher accuracy for user recognition.**

I. Introduction

Based on market analysis found in [2], 640 million tablets and 1.5 billion smartphones will be in use globally by 2015. In response to the growth in popularity, doors have opened offering newer and enhanced capabilities in terms of the computing power within mobile devices. These advancements have allowed previously impractical applications able to be implemented which were once constrained by resources. A recent study has confirmed that placing calls is now only the fifth-most frequent use of smartphones having been replaced by other uses such as browsing and applications [1]. These new functions and uses require more effort spent on user identity authentication. In light of this touch screen gesture-based user recognition has recently gained popularity as a new "biometric" signature for user authentication [6], [7], [13], [16]. This is because: (1) touch data is indicative of two biometric features, *i.e.,* user hand geometry and muscle behavior. Such biometric characteristic variations have the potential to provide user discrimination; (2) touch data can be easily accessed with very low overhead on current mobile devices.

However, the majority of current work either requires users to perform pre-defined touch gesture patterns or touch screen data is collected under monitored laboratory environments. Thus, they only solve user identity recognition problems under controlled environments and the approaches do not aid in solving the problem of utilizing a user's natural touch screen usage of data. Moreover, the few recent papers that offer touch based user recognition in uncontrolled environments fail to address several important research questions that require further investigation, which are as follows:

1) **Will the touch usage based user recognition be affected by the screen size of the smartphone?** Fig. 1 depicts one subject's touch screen data on two different devices (iPhone 5s and Samsung Galaxy S4). We can clearly see that the touch screen usage has changed significantly between the two devices.
2) **Will the touch usage based user recognition be affected by the user's or phone's physical status?** The physical status of the user is mdefined as sitting, standing, walking, running, and etc. The physical status of the phone is defined as in left/right hand, transfer, on table, and etc. Both types of physical statuses may affect the way users interact with the screen. In the Fig. 2, we depict the different touch screen usages of the same user while he is standing and walking.
3) **Will the touch usage based user recognition be affected by the application context of the smartphone?** The touch screen usage under different applications of the same user is shown in Fig. 3. Amid explicit differences in touch inputs among different applications (as utilized by the same user) touch data should be processed independently, within its contextual boundaries.

To investigate these questions we have developed a context-aware mobile user recognition method and conducted comparative experiments. The touch usage data is first preprocessed to normalize the sensor readings on different platforms and models. Then biometric and behavior features ((i.e.,) Swipe Speed, Points Curvature, etc.) are extracted which further feed into the classification module. This module uses a novel Dynamic-Time-Warping based Sequential One-Nearest-Neighbor classifier (DTW-S1NN). To conduct the comparative experiments a touch dataset has been collected in an uncontrolled daily use condition. The motion and touch screen usage data is recorded for later procuring the contextual behavior information. The performance of the proposed mobile user recognition method is compared with and without the contextual behavior infor-

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

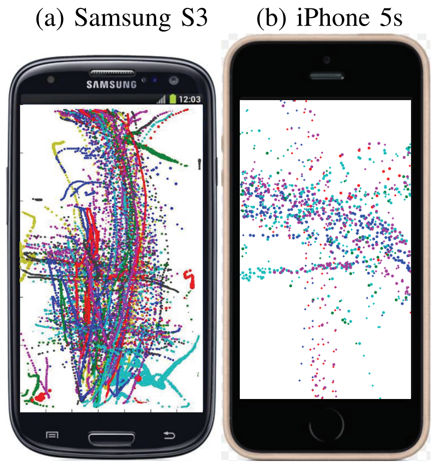


Fig. 1. Data on different phones

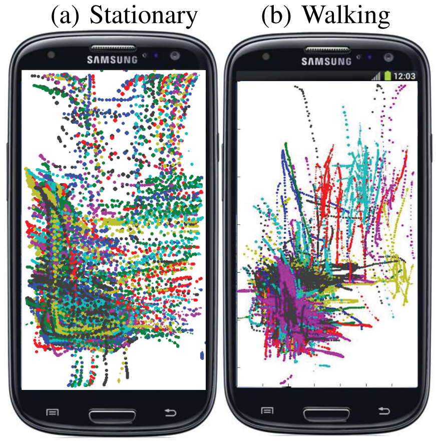


Fig. 2. Data in different physical context

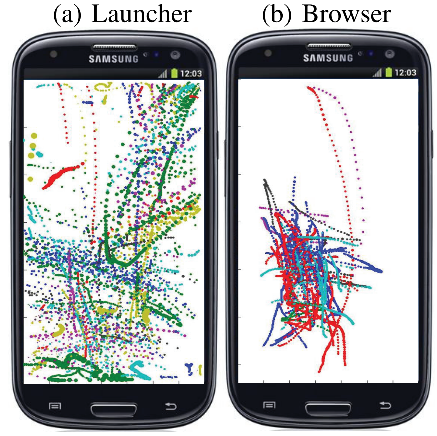


Fig. 3. Data in different application

mation. Thus, the three aforementioned unaddressed research questions are answered.

Our main contributions are:

- We study the characteristics of touch data in real-life uncontrolled environments and demonstrate how the contextual behavior affected the user's touchscreen usage.
- We proposed a context aware mobile user recognition method, which leverages a set of highly discriminant biometric and behavioral features as well as a sequential identification method.
- We implement background data collection apps which have collected a dataset of 41 subjects over four different types of phones in real-world uncontrolled conditions. Comparative experiments have been conducted to address the aforementioned research questions.

The rest of paper is organized as follows: First, we discuss the related work in section II. In section III, we define the contextual behavior information. The feature extraction of the touch gesture data is explained in section IV, and the classification methods are depicted in section V. Section VI presents the experiment setup and results. In section VII, we conclude our study and discuss the future work.

II. Related Works

Touch screen gestures, as a normal and widely used user-device interaction method, has been recently used as a biometric modality for user identity recognition and verification. In general, it can be further applied on Implicit and Continuous User Authentication, and Touch Gesture Based User Recognition under Controlled Environments problems.

Touch based user authentication has been conducted on mobile devices in [12], [15]. Feng *et al.* [6] extracted finger motion speed and acceleration of touch gestures as features. Luca *et al.* [9] directly computed the distance between gesture traces using the dynamic time warping algorithm. Sae-Bae *et al.* [13] designed 22 special touch gestures for authentication, most of which involve all five fingers simultaneously. They computed dynamic time warping distance and Frechet distance between multi-touch traces. Frank *et al.* [7] studied the correlation between 22 analytic features from touch traces and classified these features using k-nearest-neighbors and Support Vector Machines. Shahzad *et al.* [14] proposed to use touch screen gestures as a secure unlocking mechanism at the login screen. However, all prior works either require users to perform pre-defined touch gestures, or the data is collected under controlled experimental environments which might not be representative of natural user interactions. In this work, we explore implicit real-time user identification from data collected under more natural uncontrolled environments.

Meanwhile, several implicit identity sensing approaches have been proposed in the past that leverage the sensors on mobile devices such as accelerometer [10], GPS [11], and microphone [8]. Bo *et al.* [4] presented SilentSense, a framework to authenticate users silently and transparently by exploiting the dynamics mined from users' touch behavior biometrics and the micro-movement of the device caused by users' screen-touch actions. Although implemented on the Android platform as a background service, SilentSense does not explore the data variations in uncontrolled environments. Compared to other sensors for user authentication, touch screen modality, as one of the most used human mobile interface, has many advantages such as finger-related personalization, high signal-to-noise ratio, and low sampling cost in terms of collecting time and power. Furthermore, the proposed approach can also leverage the application context to improve performance. Although this work also investigates the touch based user recognition with uncontrolled data, several research questions still have not been touched, such as how phone model, screen size, and physical activity affect the user recognition performance.

III. Contextual Behavior Awareness

In practice, when users handle a smartphone device, a substancial amount of contextual behaviors can impacts the performance of touch based user recognition, including: Context Screen Size (C_S), Context Application (C_A), and Context Physical Activity (C_P). As stated in aforementioned section,

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

touch gestures under different contextual behaviors should be processed independently. Given a touch gesture $\mathfrak{G}$, its contextual behavior $C = C_S \cap C_A \cap C_P$. Thus, when analyzing the identity I of a touch gesture $\mathfrak{G}$, we have to first detect its contextual behavior C.

Context Screen Size C_S**.** An important contextual distinction is the screen size of the smartphone device. The screen size of the smartphone may affect the user's usage pattern in two aspects: the way user holds the smartphone device and the way user's fingers interact with the device. For instance, users can hold and use a smartphone device with either one hand (If the screen size of the smartphone is small) or two hands (if the screen size is large). Furthermore, a larger screen size will have more potential interaction patterns in contrast to small screen devices. The value of the screen size is fixed for each specific smartphone model which can be read from its model information. We utilize four different models of smartphones in this paper, their screen size from small to large are iPhone 5s, LG Nexus 4, Samsung Galaxy S3, and Samsung Galaxy S4, so $C_S \in \{iP, LG, S3, S4\}$.

Context Application C_A**.** As stated in the previous section, the running application context is extremely important for identity recognition. User's touch gestures in the launcher are significantly different from the same user's touch gestures in other applications. In this paper, we employ four application to evaluate the user recognition performance under different applications, which are the Launcher (L), Email (E), Browser (B), and Map (M), where $C_A \in \{L, E, B, M\}$. The value of C_A is recorded by a context application change detector running as a background service.

Context Physical Activity C_P**.** Touch patterns of smartphone users may vary when he/she uses the device in different physical motion statuses. When a user is sitting, the way he/she interacts with the device would be a bit different than when standing or walking. In addition, if the user places the device on the table and interacts with it, the touch pattern would contrast from that of when he/she holds the device in his/her hand or hands. We define two kinds of Context Physical Activity, one of which is the user's physical body movements, including Walking (W) and Stationary (S) and another presents mobile's physical status, such as In-Left-Hand (LH), In-Right-Hand (RH), In-Both-Hand (BH), and On-Table (OT), where $C_P \in \{\{W \cap \{LH, RH, BH\}\}, \{S \cap \{LH, RH, BH, OT\}\}\}$. The C_P can be analyzed from motion sensor and touch screen readings.

IV. Context Specific Touch Features

After preprocessing the touch gestures into correct subcategory of contextual behavior C, we can move on to process the underlying biometric and behavioral features inside each single touch gesture $\mathfrak{G}$. Due to the discrete capacitive sensors deployed on smartphone touch screen and the delay of the touch sensing mechanism, the raw touch data T collected from touch screen sensors is a series of point vectors ($T = \{P_1, P_2, ...P_n\}$), where each point P_i consists of a x-y coordinates values, finger contact size readings, time stamp, $P_i = (x_i, y_i, s_i, t_i), i \in \{1, 2, ..., n\}$. From the raw sensor data T, we extract a set of features to represent the touch gesture $\mathfrak{G}$, including biometric features (BI): Swipe Speed (SS_i), Points Curvature (PC_i) and Contact Size (CS_i) at each touch point P_i, and behavior features: Touch Location (TL), Swipe Length (SL), and Swipe Curvature (SC) for each touch gesture $\mathfrak{G}$. For a touch gesture with m touch points, it can be represented in the following:

$$\begin{aligned} \mathfrak{G} &= (\hat{BI}_n, TL, SL, SC) \\ \hat{BI}_n &= (SS_i, PC_i, CS_i), i \in 1, 2, ..., m \end{aligned} \tag{1}$$

A. Biometric Features

Previous studies have shown that user's touch screen data may have specific characteristics in terms of biometric features such as swipe speed and contact size.

Swipe Speed, SS_i**.** reflects how fast a user performs a swipe touch gesture. Although this feature might be affected by user's current emotional state or environment, it is usually determined by the user's finger and hand muscles. Assume the speed of the first point $SS_1 = 0$, this feature can be calculated by the following equations, the θ_s is the screen size adjust metric:

$$\begin{aligned} SS_i &= \frac{\sqrt{\Delta x_i^2 + \Delta y_i^2}}{(t_i - t_{i-1}) * \theta_s}, i \in 2, 3, ..., m \\ \Delta x_i^2 &= (x_i - x_{i-1}) \\ \Delta y_i^2 &= (y_i - y_{i-1}) \\ \theta_s &= \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2} \end{aligned} \tag{2}$$

Points Curvature, PC_i**.** represent the curvature between two consecutive touch points. Unlike the Swipe Curvature SC, which is most impacted by human behavior factors, this feature is most affected by user's hand and finger geometry. For each touch gesture $\mathfrak{G}$, the initial $PC_1 = 0$, and the rest are as show in Eq. 3.

$$PC_i = arctan(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}), i \in 2, 3, ..., m \tag{3}$$

Contact Size, CS_i is the contact surface area between user's finger and the touch screen surface. The contact size value can be affected by how hard the user touches the screen, therefore sometimes it is also used as an approximation of touch pressure. Different mobile models employs different system readings to represent the contact size information, for instance, Samsung Galaxy S3 and S4 use TOUCH MAJOR, TOUCH MINOR, and WIDTH MAJOR, and Nexus 4 employs PRESSURE and TOUCH MAJOR. The contact size feature CS_i can be calculated from these readings.

$$CS_i = \frac{s_i}{s_{max} - s_{min}}, i \in 1, 2, ..., m \tag{4}$$

B. Behavioral Features

Besides the aforementioned biometric features, specific behavioral features, such as Touch Location, Swipe Length, and Swipe Curvature are also good indicators of users' behavioral patterns of interaction with mobile devices. We confirm this later in our experimental evaluation. These behavioral features are determined not only by users' touch behaviors, but also by the manner in which users hold the mobile device.(*e.g.*, left-hand or right-hand holding, one hand vs. both hands). We will explain these behavioral features respectively.

Touch Location, TL indicates the swipe location preference. For instance, when performing a vertical swipe gesture, some users like to do it on the left part of the touch screen, while some others may prefer the right part of the touch screen. We fractionalize the touch screen into 16 areas, and assign values to each area of the touch screen, the value matrix VM is shown in the following matrix.

$$VM = \begin{bmatrix} (0,0) & (1,0) & (2,0) & (3,0) \\ (0,1) & (1,1) & (2,1) & (3,1) \\ (0,2) & (1,2) & (2,2) & (3,2) \\ (0,3) & (1,3) & (2,3) & (3,3) \end{bmatrix}$$

,

By locating all the touch points P_i of a touch gesture $\mathfrak{G}$ in these areas, we can analyzing the touch location of $\mathfrak{G}$.

Swipe Length, SL represents the length of swipe gestures. This feature is application dependent. For example, during a left-to-right screen scroll operation in the launcher application, some users may swipe all the way on the touch screen while others may only swipe a short distance. The SL can be calculated by Eq. 5.

$$SL = \frac{\sqrt{(x_{end} - x_{start})^2 + (y_{end} - y_{start})^2}}{\sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}} \quad (5)$$

$$SC = arctan(\frac{y_{end} - y_{start}}{x_{end} - x_{start}}) \quad (6)$$

Swipe Curvature, SC is another useful feature which represents the slope of a user's swipe gestures. The consistency of this feature can be seen from the swipe gestures shown in Fig. 1 and the value of SC is calculated by Eq. 6.

V. CLASSIFICATION

In this section, we describe the details of our classification method that has been employed to solve the touch based user recognition problem. When a new touch gesture $\mathfrak{G}$ is input in the mobile device, we first acquire its context, extract features. Then we select the matching templates based on the contextual behavior information and the behavioral features. To perform user identity recognition, we propose a Dynamic-Time-Warping based Sequential One-Nearest-Neighbor classifier (DTW-S1NN). The detailed classification process of the touch based user recognition is depicted in Alg. 1.

Algorithm 1 Touch Based User Recognition Process

Input: Touch Gesture Input $\mathfrak{G}$
Output: User Identity $\mathfrak{I}$

1: Read and acquire contextual behavior C of $\mathfrak{G}$(Sec. III).
2: Extract biometric and behavioral features, from raw data T(Sec. IV).
3: Select the Matching Template Set $\mathfrak{S}_M$ in all the saved templates(Sec. V-A).
4: Calculate the DTW distance of the $\mathfrak{G}$ with the templates in $\mathfrak{S}_M$(Sec. V-B).
5: Aggregate the normalized DTW distance of a sequence of $\mathfrak{G}$(Sec. V-B).
6: Identify user's identity $\mathfrak{I}$ using one nearest neighbor by the sequence result in step 5(Sec. V-B).

A. Matching Template Selection

Since matching inputs with the templates under different contexts is a waste of resources, it is unnecessary to compare an incoming gesture with all the available template gestures in the gallery. We employ a Matching Template Selection (MTS) technique to reduce the computational complexity while maintaining suitable performance.

The algorithm of the MTS is shown in Algorithm 2. The α and β respectively are the swipe length and swipe curvature modification. For instance, for a touch gesture $\mathfrak{G}$ with a SL of 400 and a SC of 30, if α and β are set to be 0.5 and 20, only templates have SL' in (200,600) and a SC' in (10,50) will be matched with $\mathfrak{G}$.

Algorithm 2 Matching Template Selection

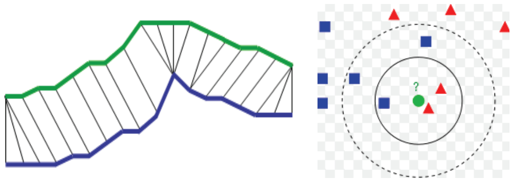
Input: Touch Gesture Input $\mathfrak{G}$
Output: Matching Template Set $\mathfrak{S}_M$

1: Use the contextual behavior C of $\mathfrak{G}$ to locate a subset, $\mathfrak{S}_C$, of all the templates shares the same C with $\mathfrak{G}$.
2: Use TL of $\mathfrak{G}$ to further get a subset, $\mathfrak{S}_T$, of $\mathfrak{S}_C$ which share the same touch location with $\mathfrak{G}$.
3: Use SL, SC of $\mathfrak{G}$ to acquire the final Matching Template Set, $\mathfrak{S}_M$, which is a subset of $\mathfrak{S}_T$, and have a SL' in $((1-\alpha)*SL, (1+\alpha)*SL)$ and a SC' in $(SC-\beta, SC+\beta)$.

B. DTW-S1NN Classifier

Dynamic Time Warping [3] is considered an efficient way to measure the similarity between two temporal series which may vary in time and speed. It works by computing the distance between any two input sequences of feature vectors and finds the optimal sequence alignment using dynamic programming. In this paper, we use Euclidean distance as the function to calculate the distance between two touch gestures. A simple illustration of the alignment process is shown in Fig. 4(a). The blue and green lines are two sequences. The black lines between them are the distance value, and by summing the

978-1-4799-1737-2/15/$31.00 ©2015 IEEE



(a) DTW (b) 1NN

Fig. 4. Simple illustration to DTW and 1NN

shortest distance of each point on the two sequence, we can acquire the DTW distance of the two sequence.

One way to recognize the user's identity is to always use the single newest incoming gesture and compare it with the ones in the gesture template library as described above. However, this approach would not capture the temporal correlation of consecutive gesture inputs under natural uncontrolled environments. We perform sequential user identity recognition by first observing X number of consecutive gesture examples and accumulating their individual DTW distances (resulting from each pair of gesture comparison). We call X the *authentication length* and use it as a metric to define the number of most recent gestures used before providing an identity recognition result. Gestures within the authentication length will be normalized and aggregated. Then the One Nearest Neighbor classifier is employed using the aggregated value to recognize the identity of the new touch input sequence. We then apply One Nearest Neighbor Classifier [5] on the sequential gestures. Specifically, when applying it, we calculate the DTW distance between an incoming touch input gesture (*e.g.*, the green circle in Fig. 4(b)) and all candidate gesture templates in the library. The label assigned to the incoming gesture is that of the closest gesture template in the library according to the DTW distance (*e.g.*, the red triangle in Fig. 4(b)).

VI. Experiments and Evaluation

A. Data Acquisition

We implement an Android background service and an IPhone touch hook function that implicitly collects touch screen data continuously on both Android and iOS platforms. In the mean time motion sensor data and application context data are collected by another background service to provide contextual behavior information. All the touch inputs, motion sensor data, and application contexts are collected within the system level. No information is provided from the application side so there is no need to modify each application to acquire touch data, and the data acquisition is completely transparent to the phone user.

We recruited 41 subjects for our study, of which 38 were right-handed, 29 male and 12 females. The services were installed by the subjects on 4 different model of smartphone devices (Samsung Galaxy S-III, Galaxy S-IV, iPhone 5S, and LG Nexus 4) and the C_S information was logged. The subjects can use the device naturally as before and there was no further operations that could disrupt them. The service collected user's touch data, C_A and C_P information. Recognition Rate is used most frequently as the evaluation criterion for comparative recognition experiments.

TABLE I. Touch points overlap statistics.

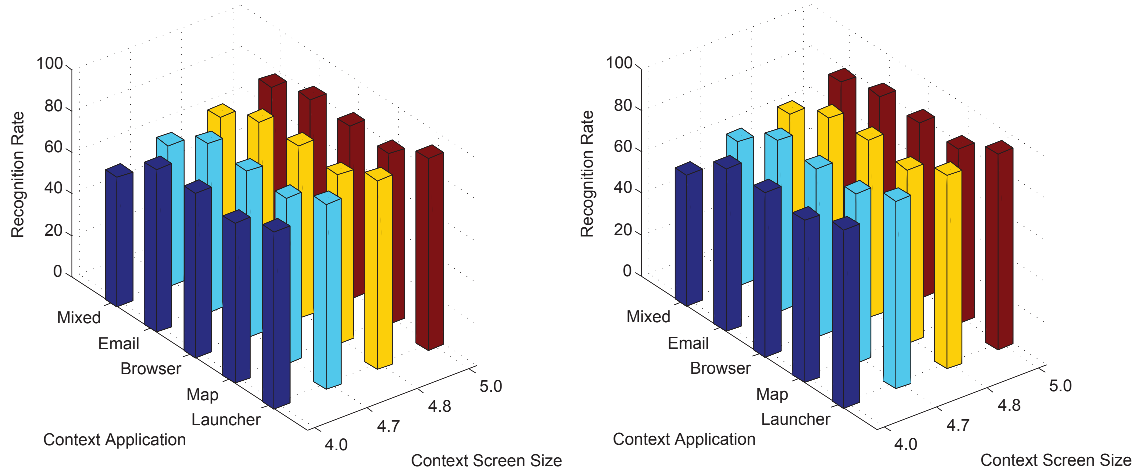
Overlaps	iPhone 5s	Nexus 4	Galaxy S3	Galaxy S4
Less than 5	7.24%	10.93%	13.17%	14.32%
5 to 10	17.76%	29.37%	29.12%	31.21%
11 to 15	40.52%	33.60%	33.79%	32.14%
Above 15	34..48%	26.09%	23.92%	22.32%

B. Experiment Results

In this paper, we evaluated the context aware mobile user recognition method on the aforementioned four smartphone devices with different screen sizes, which are 4, 4.7, 4.8, and 5 inches for the IPhone 5S, LG Nexus 4, Samsung Galaxy S3, and Samsung Galaxy S4 respectively. In addition we evaluated the proposed method on five different application context settings, which are mixed, email, browser, maps, and launcher. Fig. 5(a) depicts the identity recognition performance on the four smartphone device platforms with these screen size contexts and application context settings.

We can clearly see from Fig. 5(a), as the screen size increases, the user recognition performance tends to be more accurate, while if the screen size of the touchscreen is too small, *(*i.e.,) IPhone 5S, it will be hard to acquire an equal user recognition performance in comparison to those smartphones with large touchscreens. To further look into this phenomenon, we analyzed the touch points overlap statistics on the collected data and the result are shown in Tab. VI-B. The overlaps value in the Tab. VI-B means the percentage of the touch points. For instance, for a touch point on the touchscreen. if it has been touched for 6 times, we label these 6 touches with a overlap value of 6, and these 6 touch points should located at "5 to 10". We can see that the iPhone 5s users have much more overlapped touch points. And from our observation, we found that iPhone 5s users are trend to hold and use the smartphone device in similar way. That's because the smartphone is long and narrow, users can easy hold the device and use it with one hand, while for other device such as Samsung Galaxy S4, not all the users can comfortably use it with one hand as iPhone 5s.

Another obvious observation from Fig. 5 is that the touch gesture based user recognition performance can be greatly improved by take the context application into consideration. The user recognition performance under the Mixed context application setting all share a low recognition rate (mostly below 80%). This result proves that user's touch gestures under different context application do have different biometric and behavior features. And we can find that the touch gestures under Launcher application has the highest recognition rate. We consider this is because the Launcher application gives user most freedom to perform touch gestures. Although the touch gestures under Launcher are meaningful as touch gestures in other applications, the context contents are much more simple: In Launcher, all the swipes from left to right moves the system to the left nearest screen. While in other applications, such as Email or Browser, users need to consider the swipe length to control the scroll length of the contents.



(a) Without context physical activity (b) With context physical activity

Fig. 5. A depiction on the results of user recognition performance for different context application, context screen size and context physical activity: The height of bars is the Recognition Rate, the X, Y axes respectively are context screen size value and context application. Note that here Mixed means we do not take context application information into consideration when performing user recognition.

Fig. 5(b) presents the user recognition performance with context physical activity information. With the help of the context physical activity information, we can see most of the user recognition performance under different context application and context screen size settings will have an average 1.30 percentage improvement. Although the promotion is not as large as the benefits brought by context application, it still helped reducing a number of miss classified touch gesture inputs. And since the context physical activity is not fine-grained, potential improvements still exists for further research.

VII. Conclusion

This paper introduces a novel concept, Contextual Behavior, in touch based user recognition. In addition, we employed a novel context aware user recognition method on uncontrolled touch data for comparative experiments. We have evaluated the performance of the user identity performance with and without the new introduced concept. As supported by experiment results, we find the screen size greatly impacts how users interact with the device and the application context and physical activity context can aid in achieving higher accuracy in touch based user recognition. In terms of future work, we will further evaluate the performance on a more broad range of devices, including tablets, to see the impact of even larger screen sizes on user touch behavior. More application contexts and physical activity contexts will also be considered and evaluated.

References

[1] Making calls fifth most popular use for smartphones, says report. *http://www.whathifi.com/news*.

[2] Tablet market and smartphones market: Global database & forecast (2010 - 2015). http://www.researchandmarkets.com/reports/2076988/tablet_market_and_smartphones_market_global.

[3] Dynamic time warping. In *Information Retrieval for Music and Motion*, pages 69–84. Springer Berlin Heidelberg, 2007.

[4] C. Bo, L. Zhang, and X. Li. Silentsense: Silent user identification via dynamics of touch and movement behavioral biometrics. In *MobiCom*, 2013.

[5] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, January 1967.

[6] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, pages 451–456, 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. *Information Forensics and Security, IEEE Transactions on*, 8(1):136–148, 2013.

[8] H. Lu, A. Bernheim Brush, B. Priyantha, A. Karlson, and J. Liu. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In *IEEE Pervasive Computing*. 2011.

[9] A. D. 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 *ACM CHI*, 2012.

[10] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. marja Makela, and H. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[11] P. Marcus, M. Kessel, and C. Linnhoff-Popien. Securing mobile device-based machine interactions with user location histories. In *Security and Privacy in Mobile Information and Communication Systems*. Springer Berlin Heidelberg, 2012.

[12] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In *ACM CHI*, 2012.

[13] N. Sae-Bae, N. Memon, and K. Isbister. Investigating multi-touch gestures as a novel biometric modality. In *IEEE BTAS*, 2012.

[14] M. Shahzad, A. 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 *MobiCom*, 2013.

[15] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong. Senguard: Passive user identification on smartphones using multiple sensors. In *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 141–148, 2011.

[16] X. Zhao, T. Feng, and W. Shi. Continuous mobile authentication using a novel graphic touch gesture feature. In *IEEE BTAS*, 2013.

978-1-4799-1737-2/15/$31.00 ©2015 IEEE