Online Signature Verification on Mobile Devices

Napa Sae-Bae and Nasir Memon, *Fellow, IEEE*

***Abstract*— This paper studies online signature verification on touch interface-based mobile devices. A simple and effective method for signature verification is developed. An online signature is represented with a discriminative feature vector derived from attributes of several histograms that can be computed in linear time. The resulting signature template is compact and requires constant space. The algorithm was first tested on the well-known MCYT-100 and SUSIG data sets. The results show that the performance of the proposed technique is comparable and often superior to state-of-the-art algorithms despite its simplicity and efficiency. In order to test the proposed method on finger drawn signatures on touch devices, a data set was collected from an uncontrolled environment and over multiple sessions. Experimental results on this data set confirm the effectiveness of the proposed algorithm in mobile settings. The results demonstrate the problem of within-user variation of signatures across multiple sessions and the effectiveness of cross session training strategies to alleviate these problems.**

***Index Terms*— Online signature, mobile device authentication, template aging, performance evaluation, behavioral biometric.**

I. Introduction

A HANDWRITTEN signature is a socially and legally accepted biometric trait for authenticating an individual. Typically, there are two types of handwritten signature verification systems: *off-line* and *online* systems. In an off-line system, just an image of the user's signature is acquired without additional attributes, whereas, in an online system, a sequence of x-y coordinates of the user's signature, along with associated attributes like pressure, time, etc., are also acquired. As a result, an online signature verification system usually achieves better accuracy than an off-line system [1].

The increasing number of personal computing devices that come equipped with a touch sensitive interface and the difficulty of entering a password on such devices [2] have led to an interest in developing alternative authentication mechanisms on them [3], [4]. In this context, an online signature is a plausible candidate given the familiarity users have with the concept of using a signature for the purpose of authentication.

There has been much work on online signature verification systems [5]–[10]. However, none of this has been directed to the context of authentication on mobile devices. Previous work has addressed online signatures acquired from traditional digitizers in a controlled environment. These are different from those acquired from mobile devices in dynamic environments. First, on mobile devices, a user performs his signatures in various contexts, i.e., sitting or standing, mobile or immobile, and holding a device at different angles and orientations. Secondly, availability of computational resources may differ from one signature instance to another and it could result in greater variation of input resolution when compared to that of stand-alone acquisition devices. Last, signatures on mobile devices are often drawn using a finger instead of a stylus resulting in less precise signals. An example of finger drawn signatures acquired from mobile devices is depicted in Figure 1.

Fig. 1. An example of finger drawn signatures on mobile devices.

Consequently, verification performance derived from traditional datasets, collected using stylus-based devices in a controlled environment, may not carry over to online signature verification on mobile device setting [5]–[8]. In addition, other characteristics of the system, i.e., a template aging [11] and effectiveness of cross-sessions training, may be different when signatures are obtained from a mobile device.

This paper proposes an online signature verification algorithm that is suitable to deploy on mobile devices. It is a computationally and space efficient algorithm for enrolling and verifying signatures. In addition, a signature template is stored in an irreversible form thereby providing privacy protection to an original online signature. The proposed method was evaluated on public datasets as well as new dataset collected in in uncontrolled setting from user owned mobile devices. The verification performance obtained is promising. The key contributions made by this paper are as follows:

1) A method to extract a model-free non-invertible feature set from an online signature is proposed. The feature set comprises of sets of histograms that capture

Manuscript received October 20, 2013; revised January 14, 2014; accepted April 1, 2014. Date of publication April 10, 2014; date of current version April 28, 2014. This work was supported by the National Science Foundation under Grant 1228842. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Patrizio Campisi.

The authors are with the Computer Science and Engineering Department, Polytechnic School of Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: nsaeb01@students.poly.edu; memon@poly.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2014.2316472

distributions of attributes generated from raw signature data sequences and their combinations. By evaluating the proposed method on public datasets, its verification performance is superior to several state of the art algorithms.

2) A new dataset was collected from 180 users in a mobile device verification environment. The signatures in this dataset were drawn with a fingertip, in an uncontrolled setting on user owned iOS devices and over six separate sessions with intervals ranging from 12 to 96 hours.
3) By applying the proposed method on the above dataset, the following aspects of online signature verification on mobile devices were investigated:
   - impact of template aging on online signatures,
   - effectiveness of using cross-session samples, or samples from multiple sessions, to train a classifier, and
   - security of the system against random forgery, or zero-effort attack, and its comparison to that of 4-digit PIN.

The rest of this paper is organized as follows. Section II presents a process of deriving a set of histograms from an online signature, gives details of the proposed online signature verification system, and analyzes its complexity. Section III provides experimental results on public datasets. In section IV, a method and apparatus for collecting a new dataset as well as experimental results and analysis on this dataset is presented. Lastly, in section V, conclusions and future work are discussed.

A. Previous Work

Typically, online signature verification techniques can be classified into two approaches, namely, function-based and feature-based [5]. The former refers to an approach where the matching process is done using, directly or indirectly, the original time series data points of a signature. The latter refers to an approach where the matching process is done using descriptive features of a signature. Examples of well-known function-based approaches include Dynamic Time Warping Algorithm (DTW) [6], [7], [12], and Hidden Markov Models (HMM) [8].

A function-based system typically yields better verification performance than a feature-based system [13]. However, during the matching process, a dynamic construction of the original signature is revealed resulting in a potential privacy problem if the matching has to be done remotely. Furthermore, the system is generally more complex and slower than feature-based systems [6]. Even worse, when a template protection approach is applied in order to provide biometric privacy and/or security, verification performance often deteriorates significantly. For instance, Maiorana et al [14] have proposed a convolution scheme to protect the original signature sequence of a user, that can be directly applied to any function based approach. The idea is to split the original input sequence into W subsequences. Each subsequence may have a different length based on random parameters. This technique has been applied with HMM and DTW based verification systems [14]–[16]. In each case, verification rates were lower when compared to using the original versions of the signatures.

With a feature-based system, an online signature is represented by a feature vector. Therefore, the original biometric sample need not be stored or transmitted. Further, many known algorithms [17], [18] can be used to derive cryptographic keys from feature vectors. However, the main challenge for a feature-based approach is to derive a descriptive set of features that can be used to effectively and efficiently verify an online signature [5], [6], [12]. In the literature, there are many proposals to derive a set of features from an online signature. In 2005, Fierrez-Aguilar et al [19] proposed a set of 100 features, such as total duration of the signature, number of pen ups, sign changes of (dx/dt) and (dy/dt), etc. to represent a signature and applied a feature selection method to rank the proposed features. Based on this 100 feature set, Nanni [9] proposed a multi-matcher method to verify an online signature. In addition, Guru and Prakash [10] derived a symbolic representation of an online signature and introduced the concept of writer independent threshold in order to improve verification accuracy. Recently, Argones et al [20] have proposed a set of HMM model features from a universal background model. The best reported verification performance obtained by their system is promising. However, the system extracts 4800 features from tuning 16 different HMM models, which is a computationally expensive task. Moreover, the universal background model is trained from a pool of 2500 genuine and forged signatures from 50 users on the same device specification, where a user-specific classifier is trained from 10 signatures. These make it less feasible to be employed for mobile device authentication, where the embedded sensors are different from model to model. In addition, the HMM-based method is not robust to the well known template-aging problem resulting in significant deterioration of verification performance when verifying samples and enrollment samples are from different sessions [11].

II. Online Signature Verification Algorithm

As illustrated in Figure 2, the proposed system comprises of three main components: a feature extractor, a template generator, and a matcher. First, an online signature is processed by the feature extractor in order to compute a set of histograms from which a feature vector is derived. Then, the template generator constructs a user-specific template using the feature sets derived from multiple enrolled signatures. This template is later used by the matcher to verify a test signature. The rest of this section describes these three components in detail and analyzes system complexity. (For more details, please refer to [21] for the earlier version of this work.)

A. Feature Extractor

In the proposed system, an online signature is represented by a set of histograms. These histogram features are designed to capture essential attributes of the signature as well as relationships between these attributes. It should be noted that histograms are widely used as a feature set to capture attribute statistics in many recognition tasks. For instance, in object

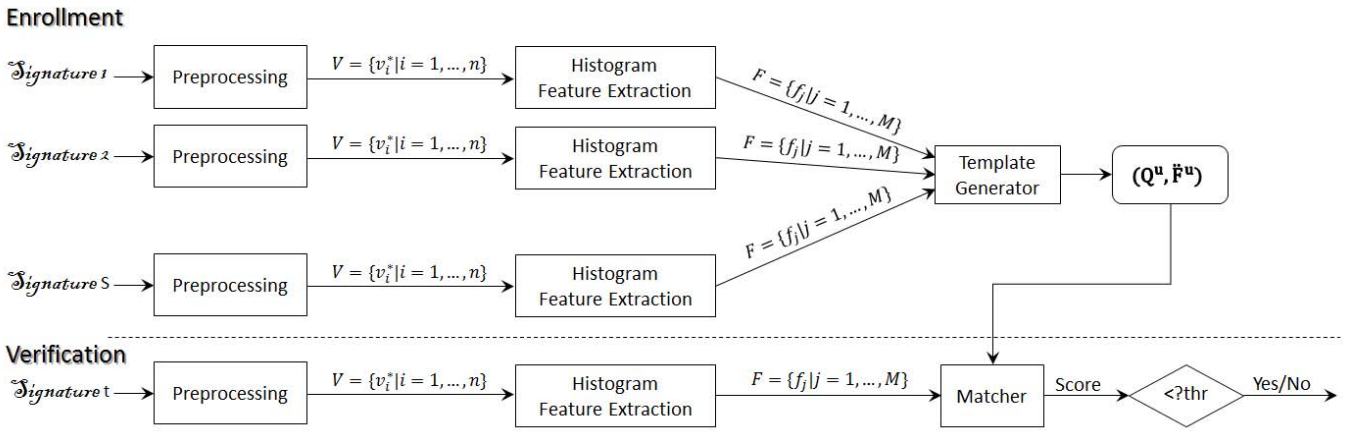


Fig. 2. The proposed online signature verification system.

recognition [22] and off-line signature verification [23]. Using histograms for online signature verification was first suggested by Nelson et al [24]. They have also been used as part of the feature set in [1] and [19]. However, in [1] and [19], the use of histograms is limited only to angles derived from vectors connecting two consecutive points in an online signature. In fact, as is shown below, much more information can be used to derive histograms useful in online signature verification. These include x-y trajectories, speed, angles, pressure, and their derivatives.

The feature extraction process of the proposed system begins by converting the time-series data of a signature in to a sequence of cartesian vectors and attributes, as well as their derivatives. Then, each cartesian vector is also converted to a vector in the polar coordinate system. Finally, histograms from these vector sequences are derived. Details of the feature extraction process are as follows.

Let $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_n\}$, and $P = \{p_1, p_2, \ldots, p_n\}$ be the x and y co-ordinates and pressure attribute, respectively, of a signature with length n sampled at times $T = \{t_1, t_2, \ldots, t_n\}$. For datasets used in this first experiment, all signatures were sampled at a constant rate. Hence the time information is implicit and is ignored. Note that if time intervals are not a constant, a normalization process using information from T can be applied to the sequences X, Y, and P prior to being processed by the system.

To begin with, the vectors X^1, Y^1, and P^1 including their derivatives are computed as follows,

$$X^1 = \{x_i^1 | x_i^1 = x_{i+1} - x_i\}, \tag{1a}$$
$$Y^1 = \{u_i^1 | y_i^1 = y_{i+1} - y_i\}, \tag{1b}$$
$$P^1 = \{p_i\}, \tag{1c}$$

and

$$X^k = \{x_i^k | x_i^k = x_{i+1}^{k-1} - x_i^{k-1}\}, \tag{1d}$$
$$Y^k = \{y_i^k | y_i^k = y_{i+1}^{k-1} - y_i^{k-1}\}, \tag{1e}$$
$$P^k = \{p_i^k | p_i^k = p_{i+1}^{k-1} - p_i^{k-1}\}, \tag{1f}$$

where $k > 1$ and $i = 1, 2, \ldots, n - k$.

Note that, by computing differences between each pair of successive points as above, the vectors X^1 and Y^1 capture positional invariant features of the signature. And by repeating this process of taking differences k times yields the k^{th} order derivative, X^k and Y^k, of the original X and Y sequences respectively.

Then, a sequence of vectors $V = \{v_i^* | i = 1, 2, \ldots, n\}$, is constructed where each vector element, $v_i^* = \{v_i^1 \| \ldots \| v_i^j\}$ is the concatenation of v_i^k which is a five-tuple consisting of the k^{th} order derivative of the cartesian and polar coordinates and pressure attributes as follows:

$$v_i^k = \langle x_i^k, y_i^k, r_i^k, \theta_i^k, p_i^k \rangle, \tag{2}$$

where $\theta_i^k = tan^{-1}\left(y_i^k / x_i^k\right)$, $r_i^k = \sqrt{\left(x_i^k\right)^2 + \left(y_i^k\right)^2}$, and $i = 1, 2, \ldots, n - k$ (see Figure 3 for illustration of the process of deriving θ distribution from an online signature).

A set of histograms is then derived from distributions of attributes of vectors v_i^* in V. The detailed descriptions of these uniform width histograms are given in Table I. Specifically, there are two types of histograms:

1) One dimensional histograms – these capture distributions of individual attributes. For example, the histogram Φ^1 captures the angle distribution of an online signature which reflects the similarity between two signature shapes. Similarly, Φ^2 captures the distribution of the angles of the first derivative since it provides information about how these vectors are aligned, an aspect that is completely ignored in the histogram Φ^1. R^1 captures the speed distribution of an online signature which is one of the distinctive features that is unique among users and especially useful in combating skilled forgeries.
2) Two dimensional histograms – these capture distributions of relationship between pairs of attributes. For example, $< \Phi^1, R^1 >_{(1)}$ and $< \Phi^1, R^1 >_{(2)}$ captures the distribution of dependency between speed and angle of the first and the second halves of an online signature. Similarly, $< \Phi^1, \Phi^1_{d(1,2)} >$ captures the distribution of the relationship between three consecutive angular coordinates of an online signature sequence while providing warping flexibility when comparing two different signatures from the same user.

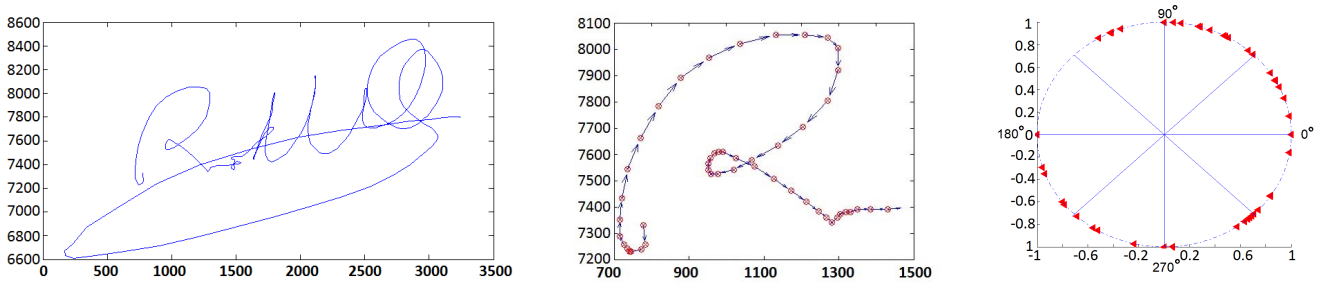


Fig. 3. Process of deriving histogram of θ: (a) an original signature, (b) the sequences of the vector from the first 60 points of the signature, and (c) the distribution of θ derived from the vectors in (b).

TABLE I

DESCRIPTIONS OF HISTOGRAMS THAT ARE USED IN THE PROPOSED TECHNIQUE

No.	Histogram	Input Attributes	Min	Max	Number of bins	Output Attributes
1	Φ^1	$\{\theta_1^1,\ldots,\theta_n^1\}$	$-\pi$	π	16	Relative frequency
2	Φ^2	$\{\theta_1^2,\ldots,\theta_n^2\}$	$-\pi$	π	24	Relative frequency
3	$<\Phi^1,\Phi^1_{d(1,2)}>$	$\{\theta_1^1,\ldots,\theta_{n-1}^1,\theta_1^1,\ldots,\theta_{n-2}^1\}$,	$-\pi$	π	8	Absolute frequency
		$\{\theta_2^1,\ldots,\theta_n^1,\theta_3^1,\ldots,\theta_n^1\}$	$-\pi$	π	8	
4	R^1	$\{r_1^1,\ldots,r_n^1\}$	0	$\mu+3\sigma$	16	Absolute frequency
5	R^2	$\{r_1^2,\ldots,r_n^2\}$	0	$\mu+3\sigma$	16	Absolute frequency
6	X^1	$\{x_1^1,\ldots,x_n^1\}$	$\mu-3\sigma$	$\mu+3\sigma$	8	Relative frequency
7	Y^1	$\{y_1^1,\ldots,y_n^1\}$	$\mu-3\sigma$	$\mu+3\sigma$	8	Relative frequency
8	X^2	$\{x_1^2,\ldots,x_n^2\}$	$\mu-3\sigma$	$\mu+3\sigma$	8	Relative frequency
9	Y^2	$\{y_1^2,\ldots,y_n^2\}$	$\mu-3\sigma$	$\mu+3\sigma$	8	Relative frequency
10	$<X^1,X^2>$	$\{x_1^1,\ldots,x_n^1\}$	$\mu-3\sigma$	$\mu+3\sigma$	6	Relative frequency
		$\{x_1^2,\ldots,x_n^2\}$	$\mu-3\sigma$	$\mu+3\sigma$	4	
11	$<Y^1,Y^2>$	$\{y_1^1,\ldots,y_n^1\}$	$\mu-3\sigma$	$\mu+3\sigma$	6	Relative frequency
		$\{y_1^2,\ldots,y_n^2\}$	$\mu-3\sigma$	$\mu+3\sigma$	4	
12	$<\Phi^1,R^1>_{(1)}$	$\{\theta_1^1,\ldots,\theta_{\lfloor n/2\rfloor}^1\}$,	$-\pi$	π	8	Relative frequency
		$\{r_1^1,\ldots,r_{\lfloor n/2\rfloor}^1\}$	0	$\mu+3\sigma$	4	
	$<\Phi^1,R^1>_{(2)}$	$\{\theta_{\lfloor n/2\rfloor}^1,\ldots,\theta_n^1\}$,	$-\pi$	π	8	Relative frequency
		$\{r_{\lfloor n/2\rfloor}^1,\ldots,r_n^1\}$	0	$\mu+3\sigma$	4	
13	$<\Phi^2,R^2>_{(1)}$	$\{\theta_1^2,\ldots,\theta_{\lfloor n/2\rfloor}^2\}$,	$-\pi$	π	8	Relative frequency
		$\{r_1^2,\ldots,r_{\lfloor n/2\rfloor}^2\}$	0	$\mu+3\sigma$	4	
	$<\Phi^2,R^2>_{(2)}$	$\{\theta_{\lfloor n/2\rfloor}^2,\ldots,\theta_n^2\}$,	$-\pi$	π	8	Relative frequency
		$\{r_{\lfloor n/2\rfloor}^2,\ldots,r_n^2\}$	0	$\mu+3\sigma$	4	
14	$<\Phi^1,R^2>_{(1)}$	$\{\theta_1^1,\ldots,\theta_{\lfloor n/2\rfloor}^1\}$,	$-\pi$	π	8	Relative frequency
		$\{r_1^2,\ldots,r_{\lfloor n/2\rfloor}^2\}$	0	$\mu+3\sigma$	4	
	$<\Phi^1,R^2>_{(2)}$	$\{\theta_{\lfloor n/2\rfloor}^1,\ldots,\theta_n^1\}$,	$-\pi$	π	8	Relative frequency
		$\{r_{\lfloor n/2\rfloor}^2,\ldots,r_n^2\}$	0	$\mu+3\sigma$	4	
15	$P^1_{(1)}$	$\{p_1^1,\ldots,p_{\lfloor n/2\rfloor}^1\}$	0	$\mu+3\sigma$	8	Absolute frequency
	$P^1_{(2)}$	$\{p_{\lfloor n/2\rfloor}^1,\ldots,p_n^1\}$	0	$\mu+3\sigma$	8	Absolute frequency
16	$P^2_{(1)}$	$\{p_1^2,\ldots,p_{\lfloor n/2\rfloor}^2\}$	$\mu-3\sigma$	$\mu+3\sigma$	8	Relative frequency
	$P^2_{(2)}$	$\{p_{\lfloor n/2\rfloor}^2,\ldots,p_n^2\}$	$\mu-3\sigma$	$\mu+3\sigma$	8	Relative frequency

The histograms above are computed by splitting the range of attribute values (specified by Min and Max columns in Table I), into a number of equal width bin intervals (also given in Table I), and counting the number of elements that fall into each particular bin. For an angle attribute and its derivative, the range of its histogram is defined as $[-\pi,\pi]$. For an attribute that has no explicit boundary, an outlier process with cutoff at three standard deviations from its mean is applied prior to computing the mean and standard deviation of the attribute in order to derive its implicit range described in Table I. For example, the histogram Φ^1 is derived from a sequence $\{\theta_i^i; i=1,\ldots,n\}$ by forming a 24 bin histogram with equal width bin intervals beginning from $-\pi$ to π and counting the number of elements, $\{\theta_i^1\}$, that fall into each of the 24 bins. It then results in a vector of 24 bin frequencies. As can be noted from Table I histograms comprise of two types of frequencies: 1) absolute frequency, an actual count of elements that fall into a particular bin, and 2) relative frequency, the absolute frequency normalized by the total number of elements in the histogram, or in other words the

length n of a signature. Using the absolute frequency results in more implicit importance given to the length of the signature whereas using the relative frequency ignores the length. Out of the 21 histograms listed in Table I, only 5 are described by absolute frequency. These are the speed and its first derivative histogram, the pressure histograms of the first and second half of a signature, and the $< \Phi^1, \Phi^1_{d(1,2)} >$ histogram. These five histograms were empirically chosen as they derive from the lowest order derivative of online signature attributes as well as they provide higher recognition rate when describe with absolute frequency.

After all these histogram are computed, they are concatenated and used as a signature's feature vector as follows. Let B_i be a vector of bin frequencies of i^{th} histogram. A feature vector F is defined as $F = \{B_1 \| B_2 \| \cdots \| B_j\}$, where j is the total number of histograms, and $\|$ is the concatenation operator. Once the feature vector F is constructed, each of the elements is used independently as a feature component of an online signature. For the rest of the paper, F is treated as a feature vector without distinguishing which histogram each each individual feature element belongs to. In other words, an online signature is represented by a feature vector $F = \{f_i; i = 1, \ldots, M\}$ where M is the total number of histogram bins from all j histograms.

B. User Template Generator

A user template is generated during the enrollment process where multiple signatures are acquired from a user and a feature set is computed from each of the samples. Then, the variance of each feature component is computed and is used to construct a user-specific uniform quantizer for each feature element resulting in a quantization step size vector Q^u that is used to quantize each of the feature vectors derived from the enrollment samples. Finally, the average of these quantized feature vectors is used as the template $\ddot{F}^u$ for that user. This feature vector along with the quantization step size vector are stored corresponding to the identity of the user.

During verification, a user claiming an identity u is asked to produce one instance of an online signature which is again represented by the set of features. Then the quantization step size vector, Q^u corresponding to that identity is used to derive a quantized feature vector from the signature input. Next, the system compares this quantized feature vector against the stored feature vector template, $\ddot{F}^u$. The signature is accepted if the Manhattan distance between these two vectors is less than a predefined threshold, otherwise it is rejected. The details on how to derive the quantization step size vector Q^u and the template feature vector are given below.

Let S be the total number of enrolled samples and M be the total number of features for each sample. And let $F^{s_j} = \{f_i^{s_j} | i = 1, \ldots, M\}$ be a feature vector of the enrolled sample s_j of the user u where $j = 1, \ldots, S$. The quantization step size vector of the user u, $Q^u = \{q_i^u | i = 1, \ldots, M\}$, is obtained by computing the standard deviations over all the enrolled samples for each feature and using a multiple of this as the quantization step size. That is,

$$q_i^u = \beta \sqrt{\frac{1}{S} \sum_{j=1}^{S} \left(f_i^{s_j} - \mu_{f_i^{(u)}} \right)^2}, \quad i = 1, \ldots, M \tag{3}$$

where $\mu_{f_i^{(u)}} = \frac{1}{S} \sum_{j=1}^{S} f_i^{s_j}$, β is experimentally fixed at 1.5.

Then, the quantized feature vector of each enrolled sample s of the user u, $\hat{F}^{(s_j|u)} = \{\hat{f}_i^{s_j} | i = 1, \ldots, M\}$ is derived from the quantization step sizes q_i^u in Q^u (adding a small ϵ to prevent division by zero) as follows,

$$\hat{f}_i^{(s_j|u)} = \left\lfloor \frac{f_i^{s_j}}{q_i^u + \epsilon} \right\rfloor, \quad i = 1, \ldots, M \tag{4}$$

where ϵ is at 0.002 and 0.8 for histograms with absolute and relative frequencies, respectively.

Lastly, the user-specific feature vector template, $\ddot{F}^u = \{\ddot{f}_i^u | i = 1, \ldots, M\}$, is derived by averaging the quantized feature vectors of all the enrolled online signature samples from the user u.

$$\ddot{f}_i^u = \left\lfloor \frac{\sum_{j=1}^{S} \hat{f}_i^{(s_j|u)}}{S} \right\rfloor, \quad i = 1, \ldots, M \tag{5}$$

A pair $\left(Q^u, \ddot{F}^u\right)$ comprising of the quantization step size vector and its associated feature vector template is then stored and later used to verify a claimed signature of the user u.

It should be noted that while the feature vector in this experiment was extracted from attributes of multiple histograms, it is possible to use other proposed methods, e.g., the one in [25], that can extract a fixed-size feature vector from an online signature with variable length. However, the method proposed in this paper to derive a quantization step size vector and a user-specific feature vector template is not robust to a feature with an extreme value, i.e., an outlier. This could possibly happen in certain features spaces, for example DCT coefficients in [25], but is rarely the case for histogram features. Such an occurrence could result in verification performance deterioration.

C. Matcher

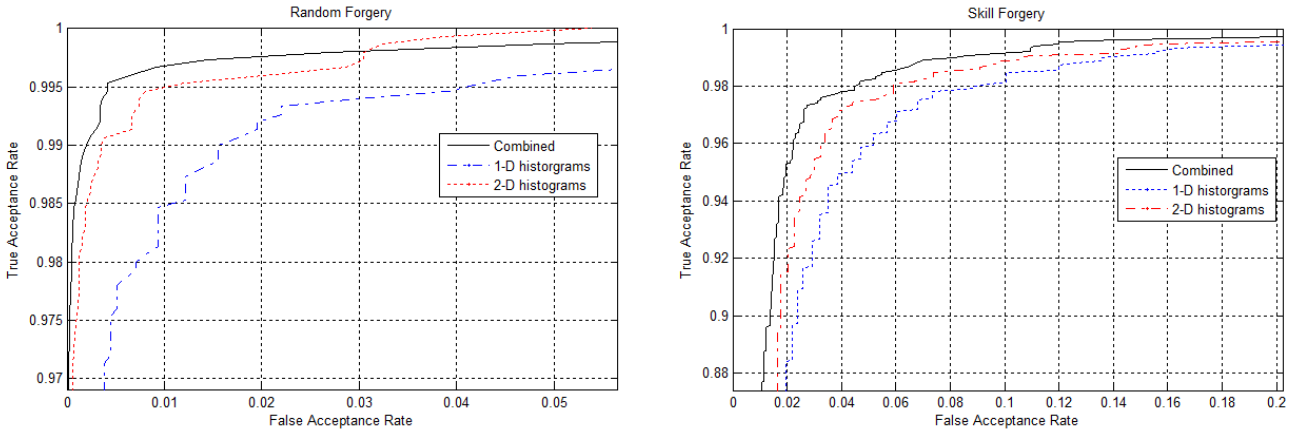
During verification, given that t is claimed to be an online signature sample from user u, $\hat{F}^{(t|u)}$ is calculated using Q^u. Then the system derives a dissimilarity score using manhattan distance between $\ddot{F}^u$ and $\hat{F}^{(t|u)}$ as,

$$Score = \sum_{i=1}^{M} |\hat{f}_i^{(t|u)} - \ddot{f}_i^u| \tag{6}$$

The system then accepts the sample t if the dissimilarity score is less than a predefined threshold, otherwise it rejects.

D. Complexity

Given n as the length of an online signature, X^k, Y^k, R^k, Φ^k, and P^k can be computed in time $O(n)$. These vectors are then used to derive h histograms which requires $O(h * n)$ or $O(n)$ time complexity for deriving a feature vector as h is a constant. For the classification process, a feature vector



(a) (b)

Fig. 4. ROC of skill and random forgery obtained from MCYT-100 dataset when 1-D histogram, 2-D histogram, and both sets are applied. (a) ROC of random forgery. (b) ROC of skill forgery.

is first computed and quantized in $O(n)$ time and then used to construct or to compare against the feature vector template. Since the number of features is a constant, the time complexity is $O(n)$. The space required to store a template is clearly a constant as it consists of two fixed-length vectors. As a result, the proposed method requires constant space to store a user's template and linear time computational complexity for enrolling and verifying a signature.

III. Experiments on Public Datasets

This section presents experimental results with the proposed technique and compares it to others published in the literature on common datasets. Experiments were performed on the two well known datasets: 1) MCYT [8], which consists of signatures from 100 individuals with 25 genuine samples and 25 skilled forgery samples, and 2) SUSIG [7], which consists of signatures from 94 individuals with 20 genuine samples from two separate sessions and 10 skilled forgery samples.

In terms of training samples, there have been two approaches taken in the literature. Some papers [10], [26] randomly select k samples as the training set and then they average over multiple such random selections to derive the final performance result. The reason is that such a training method captures within-user variation better. Other papers [7], [8], [14], [15], and [20] choose the first k samples, according to the original order in which the data was acquired, as the training set. For online signature verification, most studies have chosen this second approach since it mimicks what a verification application will actually do. In other words, the actual verification application will always acquire enrollment samples before verifying samples. The results from such an experimental protocol also captures the effect of the template aging problem [13], [26], [27], which is, in fact, one of the major causes for degradation of verification performance.

In this experiment, the second protocol was followed. That is, the first k samples of the set from a specific user were used to enroll a template, and the rest were used to evaluate the False Rejection Rate (FRR) at different threshold levels. In the random forgery scenario or zero effort attack, i.e., an attacker simply attacks the system using his own signature, all samples from all other individuals were used to evaluate the False Acceptance Rate, namely FAR-RF. On the other hand, for the skilled forgery scenario, 25 skilled forgery samples from MCYT dataset and 10 skilled forgery samples from SUSIG dataset for each user were used to evaluate the False Acceptance Rate, namely FAR-SF. The Equal Error Rate (EER), the rate at which FAR and FRR are equal, was also used to compare the verification performance of different approaches.

Finally in this experiment, only the first and second derivative sequences of vectors $v^* = [v_i^1, v_i^2]$, as described in Section 2.1, were used to compute the features of an online signature (i.e. $k = 2$). This resulted in a set of 440 features by concatenating the histograms described in Table I (See Appendix A for experiments where parameters in the table are varied.)

A. Effectiveness of 1-D Versus 2-D Histogram Features

As mentioned in section II-A, the proposed histogram features are derived from two types of histograms namely one dimensional histograms and two dimensional histograms. The plot of the receiver operator characteristic (ROC) curve obtained from MCYT-100 dataset when each of these histograms is used as well as when they are combined, using 10 enrollment samples, is depicted in Figure 4.

The results show that 2-D histograms are indeed a more effective feature set in terms of discrimination power against both skilled and random forgeries as compared to 1-D histograms provided that they are employed with larger bin widths. They also appear to work well with less information since pressure information was not included in the 2-D histograms we computed. The results also demonstrate that 1-D histogram features provide complementary information since the best result is observed when the two sets are combined.

TABLE II
EER OF THE PROPOSED SYSTEM DERIVED FROM MCYT-100 DATASET WHEN DIFFERENT NUMBER OF SAMPLES ARE USED FOR TRAINING

No. of Training Samples	3	5	7	10	20
EER-SF	5.74	4.02	3.43	2.72	2.72
EER-RF	1.43	1.15	0.87	0.44	0.35

B. Verification Performance of the Proposed System

The performance of the proposed verification system derived from MCYT-100 dataset using different number of training samples per user is reported in Table II. The results demonstrate that the proposed system can effectively verify a user's online signature even when only three samples are supplied during enrollment. However, verification performance at every operating point slightly improves as the number of training samples grows.

C. Comparison With Previous Work

This subsection provides a comparison of performance between the systems that are considered as the state of the art for feature and function based approaches and the proposed approach. Results reported on the proposed system as well as the other systems are derived from the same datasets. The function based approach considered here includes Dynamic Time Warping technique (DTW), Hidden Markov Model (HMM), and their template protection approach. The feature based approach considered include one utilizing Fourier descriptor features, and a 100-feature system in conjunction with three different classifiers.

Table III lists the verification performance of these different techniques on the MCYT-100 dataset. As seen, the proposed system outperforms other systems especially when a few training samples are supplied. These results emphasize the competitiveness of the proposed system despite its computational and space efficiency.

Table IV reports verification performance for the previous techniques listed above on the SUSIG dataset. As mentioned in [7] and [26], they choose to more heavily weigh the signing duration feature as they observed that a skilled forgery signature typically takes twice as long as a genuine one on the average. Hence their FAR-SF is lower than FAR-RF. However, as reported in [7] and [26], the EER for the skilled forgery case in this dataset is greatly influenced by the significance that a classifier gives to the length disparity between two given signatures. In the proposed system, less weightage is given to this length disparity since most of histogram features in the proposed set are attributed by their relative frequency in which the actual length of the signature is ignored. Only 112 histogram features or 25% in the set are attributed by absolute frequency, where the actual length of the signature gets reflected in the feature values. This results in lower FAR-RF but higher FAR-SF than the system in [7] and [26]. However, when more weight is given to the histograms with frequency attributes, the EER of skill forgery is reduced from 5.86 to 4.59 whereas that of random forgery remains unchanged. This implies that the verification performance could potentially improve if there are more histograms created using absolute frequency and not relative frequency. It also demonstrates that the forger's skill in these two datasets are very different.

We acknowledge that reported verification rate is not always the best justification for a system's effectiveness, since each system might have been trained and tested differently as well as the difference in employing skilled forgery model. In addition, the system might apply different set of features to employ their classifiers. Nevertheless, the results show that the proposed system, at the very least, is comparable to others in terms of the verification performance.

IV. EXPERIMENTS ON MOBILE DEVICE DATASET

Performance comparison of online signature recognition algorithms has been traditionally done using public datasets. The two well-known public datasets, the SUSIG and MCYT-100 datasets, have been collected using a static stylus-tablet in a controlled and supervised setting. The previous section reported results of the proposed algorithm obtained on these two datasets. However, as mentioned in the introduction, online signature verification for user authentication on mobile devices has unique aspects that are not reflected in these datasets. It has been reported that performance obtained with algorithms on traditional digitizers in a static environment suffers significant deterioration when used on a mobile platform where users were holding the device instead of it being in a fixed position [29]. The performance may further degrade when online signatures are acquired without supervision as opposed to those in public datasets which were typically supervised. This section describes an experimental study to evaluate the performance of the proposed system while using a dataset comprising of signatures collected from user owned devices taken over a sufficient time window in an unsupervised, uncontrolled setting.

A. Data Collection Procedure

The process for data collection began by recruiting users from a departmental mailing list by sending a brief description of the experiment and offering a $10 dollar gift card to those who participate. Users who volunteered, were asked to create an online account with an email address, username and password on a webpage. The system then immediately took them to an introduction page that briefly described the purpose of the experiment and the procedure they would be expected to follow. An example signature was provided but no other instruction was given on the type of signature they should create. However, they were motivated to provide quality signatures by rewarding the top ten users with the most consistent samples over the entire experiment with an additional $40 gift card. Then the user was asked to create a signature and draw it five times on the screen. Visual feedback was provided so the user could see the signature they drew. All signatures were performed on the users' personal devices.

An experimental protocol was designed to capture time variation effects in user signature input over the course of approximately seven days. After the first session, multiple

TABLE III
EER OF DIFFERENT VERIFICATION APPROACHES ON MCYT-100 DATASET WHERE n IS THE NUMBER OF TRAINING SAMPLES

Matching Types	Approaches	$n = 5$		$n = 10$		Remarks
		EER-SF	EER-RF	EER-SF	EER-RF	
Function-based	DTW [15]	5.53	-	3.93	-	
	DTW [28]	9.81	-	-	-	
	DTW-Protected [15]	8.13	-	5.22	-	
	HMM [14]	10.29	-	6.33	-	
	HMM-Protected [14]	13.30	-	7.95	-	
Feature-based	Fourier Descriptors [26]	14.53	-	-	-	5 training samples are randomly selected.
	100 global features in [19]	6.89	2.2	-	-	
	100 global features in [9]	7.1	1.6	-	-	
	100 global features in [10]	6.12	2.05	-	-	5 training samples are randomly selected.
	UBM-HMM [20]	-	-	2.785	-	UBM are trained from 2500 genuine and skill forgery signatures of 50 users. PCA model are trained from 1000 genuine signatures of 100 users.
	The proposed method	**4.02**	**1.15**	**2.72**	0.44	

TABLE IV
EER OF DIFFERENT VERIFICATION APPROACHES ON SUSIG DATASET WHERE THE NUMBER OF TRAINING SAMPLES IS 5

Matching Types	Approaches	EER-SF	EER-RF	Remarks
Function-based	DTW [7], [26], [28]	**3.30**	4.08	
Feature-based	Fourier Descriptors in [26]	6.20	-	
	The proposed method	6.08	**2.94**	
	The proposed method*	4.37	**2.91**	* when the weight of R and $\Phi - \Phi^{d(1,2)}$ histogram attributes is given 3 times of others

sessions of data collection were performed where a user entered his signature 5 times for each session. At the end of each session, the screen was locked and the user was instructed to wait for a time period after which a reminder to perform the next session would be sent to his email address. The minimum time interval between each of the sessions and its immediate successor was twelve hours for the first and second, the second and third, and the third and fourth sessions. The minimum time intervals imposed between the fourth and the fifth, and between the fifth and the last session were 96 hours and 24 hours respectively. The time intervals between sessions were chosen to introduce variation in times of the day when the signatures were performed and hence the context they would be performed under.

During the experiment, if a user forgot his signature, he was provided with two options. First, he could reset the account and redo the process from the beginning. Second, he could click the "forgotten" button and the system would show his previous signature on the screen. In this case, he would be prompted to clear the screen before he could enter his new signature. This was to prevent a user from tracing his own signature thereby increasing consistency artificially.

All of the signatures were collected via the HTML5 platform. The purpose of using the HTML5 framework was to allow the user to enter online signatures anytime and anywhere without supervision. As a result, intra-user variations caused by several factors that could happen while signing in a mobile context, i.e., signing/device-holding pose, environmental condition, could potentially be captured. In addition, the HTML5 platform allowed users to participate without requiring them to install any application. While the application was running, the browser collected the touch input information and sent it back to the server and provided visual feedback to the user in the form of piecewise linear curve between the pair of drawing point in real time. For each signature, the data consisted of time-series of x-y coordinates and time-stamps.

B. Signature Characteristics

This subsection provides some characteristics of signatures that were collected. These characteristics included the length (number of points) and the number of strokes in the signature. A stroke in this study is defined as a sequence of touch points beginning from touch-down event to the next touch-up event. Figure 5 shows the distribution of signature lengths, the distribution of signature strokes and the distribution of the difference in the number of signature strokes within the same user. It is seen from Figure 5 that signature length varies significantly from user to user whereas the average length of a signature was 142.11 with a standard deviation of 85.67. Different signatures were also comprised of various numbers of strokes. Generally, the number of strokes can be influenced by the language and writing styles of users. In addition, it can also be caused by discontinuity of interaction between

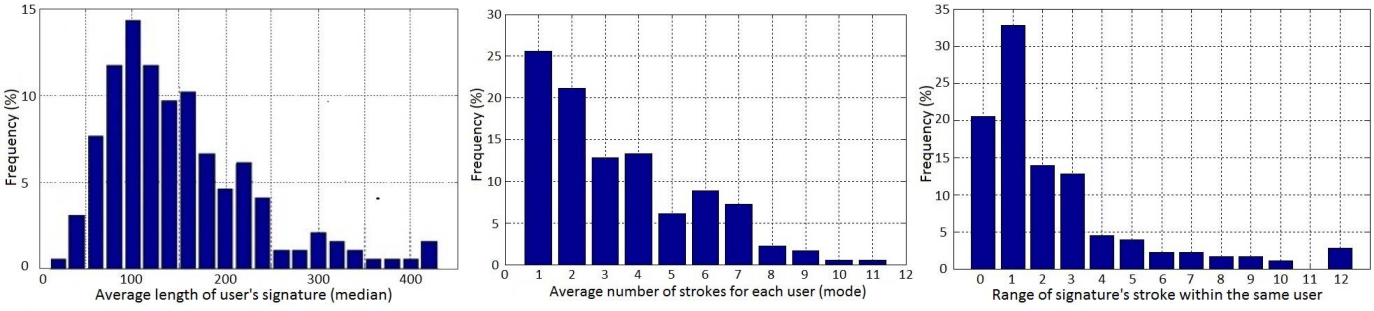


Fig. 5. Characteristics of online signatures in the dataset. (a) The distribution of signature lengths. (b) The distribution of signature strokes. (c) The distribution of the difference in the number of signature strokes within the same user.

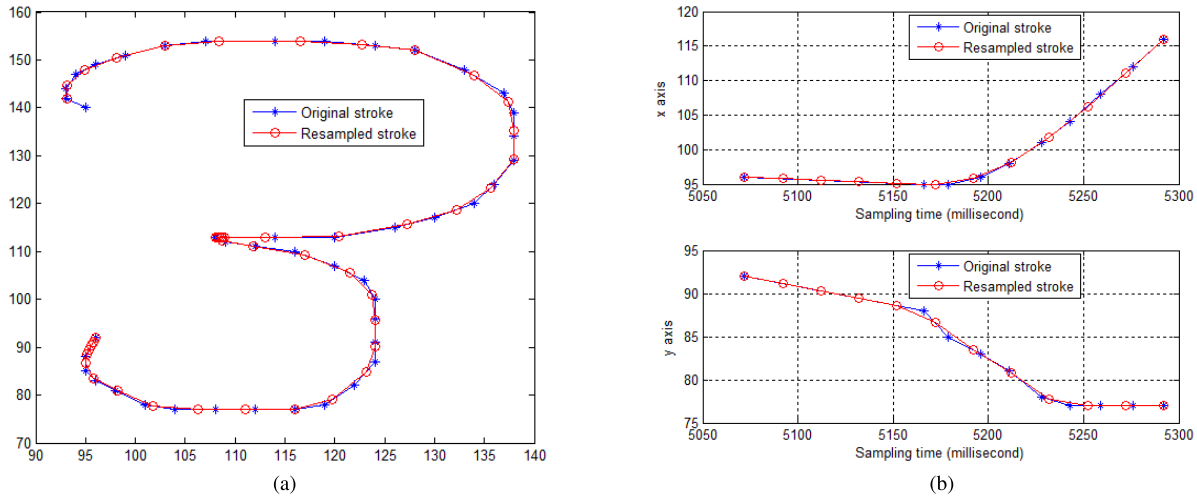


Fig. 6. A part of an online signature example before and after time normalization. (a) x-y coordinates. (b) x-t and y-t coordinates.

the hardware sensor and the input device (fingertips in this case.) In this experiment, the users were recruited from the mailing list of a university where English is used as the first language. Nevertheless, the variation in the number of strokes from user to user is noticeable. On average, a signature had 3.38 strokes with a standard deviation of 2.28. In addition, we observed that the number of strokes were often not consistent even within the same user. On average, the number of strokes varied by 3.5 over the entire set of 30 samples taken from the same user. Only 20% of the users had a consistent number of strokes across all samples. Therefore it is important that an online signature verification system for mobile devices should be designed such that it is robust to signatures with varying number of strokes.

C. Signature Preprocessing

The variation in the number of strokes per signature and sampling rate introduced in the dataset can affect verification performance. Hence, all signatures were pre-processed by time normalization and stroke concatenation before extracting histogram features. Details of the time normalization and stroke concatenation steps used are as follows.

1) Time Normalization: When a signature is acquired from typical touch sensitive computing devices (iOS device in this case), it is typically sampled with non-uniform rate. The rate depends on the availability of computational resources at a given time as well as the latency of network connection. Therefore, time normalization was used in order to derive a uniformly sampled signature. This helped to minimize the variation of signatures due to different sampling rates. The process used was as follows. Let $S = \{v_1, v_2, \ldots, v_N\}$ be an online signature with a sequence of N strokes where each stroke $v_§i = \{(x_1^i, y_1^i), \ldots, (x_M^i, y_M^i)\}$ is a sequence of touch points (x_j, y_j) sampling at time $T = \{t_1^i, \ldots, t_M^i\}$. The normalized stroke s_i was computed by interpolating the stroke v_i at $T = \{t_1^i, t_1^i + R, t_1^i + 2R, \ldots, t1i + \frac{\lfloor t_M^i - t_1^i \rfloor}{R} \times R\}$. An example of a normalized stroke is depicted in Figure 6. After the process, all time-normalized signatures have a fixed sampling rate of 50 times per second, or 20 milliseconds apart.

2) Signature Stroke Concatenation: As shown in Figure 7b, most of the signatures in this dataset have multiple strokes. Signatures with multiple strokes may pose a challenge to verification algorithms by introducing positional variation for each of the strokes. This variation could become larger when the signatures are signed on touch devices using a fingertip since each touch point may not coincide with user's intention [30]. In order to cope with this variation, signature strokes were concatenated before verification as follows.

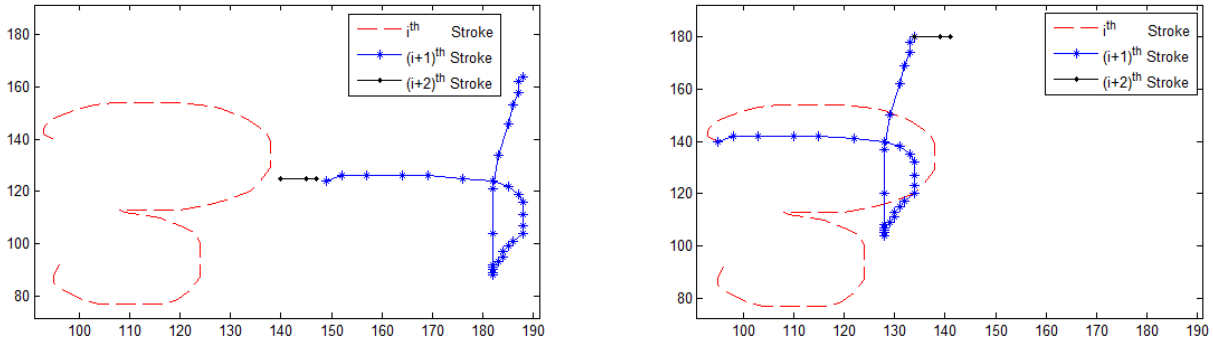


Fig. 7. Illustration of signature concatenation process. (a) A part of a signature with multiple strokes. (b) The same part after stroke concatenation.

Algorithm 1 Signature Stroke Concatenation

$s* := s_1$
for i:= 2:N **do**
$si : (x_j^i, y_j^i) \leftarrow (x_j^i + x_M * - x_1^i, y_j^i + y_M * - y_1^i)$
$s* := \{(x_1*, y_1*), \ldots, (x_M*, y_M*), (x_1^i, y_1^i), \ldots, (x_M^i, y_M^i)\}$
end for
return $S = \{s*\}$

Let $S = \{s_1, s_2, \ldots, s_N\}$ be an online signature with a sequence of normalized strokes where each stroke $s_i = \{(x_1^i, y_1^i), \ldots, (x_M^i, y_M^i)\}$ is a sequence of touch points (x_j, y_j) with length M. The immediately succeeding stroke is concatenated to the main stroke by translating the origin of the latter stroke to the end of the former stroke. The algorithm is as follows (see Figure 7 for illustration of the process),

D. Experimental Results

In this subsection we present experimental results using the data collection procedure and the pre-processing techniques described above. The algorithm used is the same as the one described in section 2, except two key differences. First, the histograms related to pressure information were discarded as they were not available in this dataset. Second, the additional histograms from Table V are empirically added as they provide higher recognition accuracy when incorporated.

1) Impact of Signature Aging and Effectiveness of Cross-Session Training Strategy: The objective of collecting the dataset over six separate sessions was to study template aging issues in online signatures, as well as to study whether training samples from multiple sessions can better represent within-user variation thereby enhancing verification performance. In this context, first the performance of online signature verification when training and test samples are drawn from the same session is reported in Table VI. Note that the results are the average performance from leave-one-out cross validation test. That is, the classifier is trained from all the samples from the same users but one, and each and every sample is used as a positive sample exactly once. The result shows that verification performance of the system when the training and test samples are from the same sessions is just slightly lower than that of public datasets given in section 3.

TABLE V
DESCRIPTIONS OF ADDITIONAL HISTOGRAMS THAT ARE USED IN THE DATASET COLLECTED FROM MOBILE DEVICE

Histogram	Input Attributes	Min	Max	# bins
1. Absolute Frequency Output				
Φ^1	$\{\theta_1^1, \ldots, \theta_n^1\}$	$-\pi$	π	16
R^3	$\{r_1^3, \ldots, r_n^3\}$	0	$\mu + 3\sigma$	16
$< \Phi^3, R^3 >_{(1)}$	$\{\theta_1^3, \ldots, \theta_{\lfloor n/2 \rfloor}^3\}$,	$-\pi$	π	8
	$\{r_1^3, \ldots, r_{\lfloor n/2 \rfloor}^3\}$	0	$\mu + 3\sigma$	4
$< \Phi^3, R^3 >_{(2)}$	$\{\theta_{\lfloor n/2 \rfloor}^3, \ldots, \theta_n^3$	$-\pi$	π	8
	$\{r_{\lfloor n/2 \rfloor}^3, \ldots, r_n^3$	0	$\mu + 3\sigma$	4
2. Relative Frequency Output				
$< X^1 - Y^2 >$	$\{x_1^1, \ldots, x_n^1\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	6
	$\{y_1^2, \ldots, y_n^2\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	4
$< X^2 - Y^1 >$	$\{x_1^2, \ldots, x_n^2\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	6
	$\{y_1^1, \ldots, y_n^1\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	4
Φ^3	$\{\theta_1^3, \ldots, \theta_3^3\}$	$-\pi$	π	24

TABLE VI
THE VERIFICATION PERFORMANCE (EER) OF INTRA-SESSION EXPERIMENT USING LEAVE-ONE-OUT CROSS VALIDATION

Session	1	2	3	4	5	6
EER(%)	3.64	3.18	2.67	2.76	3.16	3.12

This is an encouraging result given that all signatures in this dataset were collected in an uncontrolled, unsupervised setting, without pressure information, and they were drawn using a fingertip, not a stylus.

However, biometric performance can be affected severely by the template aging problem as reported in many biometric modalities, namely, iris [31], [32], pericular [33], ECG signal [34], and speech [35]. To understand the aging problem of online signatures, the performance results, when a signature template is trained using samples from one session was used to verify samples from a different session, are reported in Table VII. It is seen that the performance when training and test samples are from different sessions noticeably deteriorates as compared to when they are from the same session. Specifically, with training samples from session 1 and test samples from session 2, the error rate in terms of EER

TABLE VII
THE VERIFICATION PERFORMANCE (EER) WHEN THE TRAINING AND TEST SAMPLES ARE DRAWN FROM DIFFERENT SESSIONS

Test Session	EER(%) when training are from session k				
	k = 1	k = 2	k = 3	k = 4	k = 5
2	7.04	-	-	-	-
3	6.11	5.84	-	-	-
4	7.24	6.19	5.54	-	-
5	7.38	7.38	5.94	5.04	-
6	7.74	8.14	5.82	5.90	5.34

TABLE VIII
THE VERIFICATION PERFORMANCE (EER) WHEN THE TRAINING SAMPLES ARE DRAWN FROM n IMMEDIATELY PRECEDING SESSIONS

Test Session	EER(%) at				
	n = 1	n = 2	n = 3	n = 4	n = 5
2	7.04	-	-	-	-
3	5.84	2.56	-	-	-
4	5.54	3.96	3.27	-	-
5	5.04	3.16	3.51	3.07	-
6	5.34	3.35	2.93	2.96	2.89

increases from 3.18% to 7.04% as compared to when training are from session 2.

It is also observed that the best performance in inter-session verification is obtained when the classifier is trained using samples from the preceding session. For example, the best verification performance of samples from session 6 is achieved when the training is from session 5. This suggests that a template updating mechanism should be in place in order to retain system reliability. Interestingly, for short-term effect, i.e., approximately one week, the time interval between the training and the test sessions seemed to have little influence on the performance. Specifically, the verification performance of the fifth session, which was at least four days apart from the previous session, did not deteriorate as compared to other sessions.

In terms of training strategies, previous work on ECG [34] and speaker verification [35] has shown that cross-session samples can be used to improve biometric performance. That is because they can accommodate within-user variation better than those from a single session. In order to evaluate the effectiveness of this training strategy on an online signature modality, a user's online signature template was trained using samples from multiple sessions in stead of those from a single session. The results for multiple sessions training are given in Table VIII. In general, there is a significant improvement of verification performance when training samples are drawn from multiple sessions instead of a single session. Specifically, training a user's signature template with samples from two sessions improves the EER to 3.27%, on an average, as compared to 5.76% when training samples are drawn from one session. Note that, while collecting enrolled samples from a user over multiple sessions is not a user-friendly option, one could possibly devise a procedure to collect samples with more realistic intra-user variation from a single session. One scenario involves interrupting users with a short interactive activity.

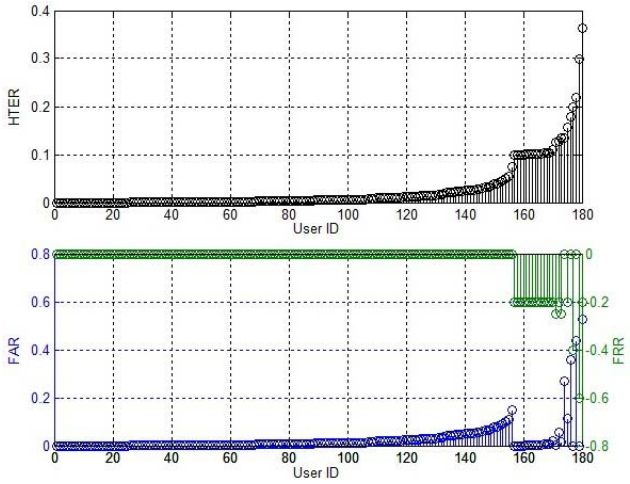


Fig. 8. Distribution of HTER (Half Total Error Rate), FAR, and FRR when training samples are drawn from sessions 1 and 2.

TABLE IX
THE AVERAGE FALSE REJECTION RATE (FRR) FOR THE TOP n% USERS WHEN FALSE ACCEPTANCE RATE (FAR) IS FIXED FOR EACH USER AT DIFFERENT LEVEL AND TRAINING SAMPLES ARE DRAWN FROM THE PRECEDING SESSION

FAR(%)	FRR(%) of top m % of Users			
	m = 100	m = 95	m = 90	m = 85
0	34.71	31.28	27.46	23.38
0.1	22.52	18.45	14.26	10.77
0.5	13.27	8.82	5.71	3.58
1	9.07	4.83	2.57	1.25

TABLE X
THE AVERAGE FALSE REJECTION RATE (FRR) FOR THE TOP n% USERS WHEN FALSE ACCEPTANCE RATE (FAR) IS FIXED FOR EACH USER AT DIFFERENT LEVEL AND TRAINING SAMPLES ARE DRAWN FROM THE TWO PRECEDING SESSIONS

FAR(%)	FRR(%) of top m % of Users			
	m = 100	m = 95	m = 90	m = 85
0	28.19	24.41	20.24	16.56
0.1	18.37	14.07	10.17	7.32
0.5	9.16	4.85	2.46	1.31
1	5.66	2.06	0.84	0.16

Further effort in designing such enrollment procedure that better captures intra-user variation is left for future work.

2) Adaptive Thresholding: It is known that the score distribution of biometric samples differs from user to user. Consequently, the resulting FAR of different users with respect to the same threshold is different. This can be seen from Figure 8 which depicts FAR, FRR and HTER (Half Total Error Rate between FAR and FRR) of each individual when a global threshold that corresponds to the corresponding EER is applied. However, as far as security is concerned, FAR should be kept very low for each and every user. The performance in this case can be theoretically realized by adjusting the threshold for each user individually according to the desirable FRR. Tables IX and X report the system performance with

TABLE XI
SUCCESSFUL ZERO-EFFORT IMPERSONATION RATE OF THE TOP THREE USERS WITH RESPECT TO DIFFERENT FAR LEVELS WHERE TRAINING SAMPLES ARE DRAWN FROM SESSIONS 1 AND 2

FAR(%)	Impersonation success rate of the user at rank		
	1^{st}	2^{nd}	3^{rd}
0	0%	0%	0%
0.1	0.48%	0.32%	0.28%
0.5	2.38%	1.47%	1.34%
1	4.66%	2.63%	2.46%

TABLE XII
THE TOP THREE MOST POPULAR PINS AND THEIR FREQUENCIES [36]

Ranking	1^{st}	2^{nd}	3^{rd}
PIN	'1234'	'1111'	'0000'
Frequency	10.71%	6.02%	1.88%

such a threshold selection strategy in terms of the average FAR derived from the top $n\%$ of users with respect to different FRRs. The training samples in these two tables are from one and two preceding sessions respectively. Note that, in practical applications, this empirical decision threshold can be estimated by using the pool of signatures in the database where each signature is represented by a feature vector.

3) Security Against Zero-Effort Impersonations: This experiment was conducted in order to analyze the security of signatures against zero-effort impersonations and compare it against that of 4-digit PINs. For 4-digit PINs, if all users choose their 4-digit PINs at random, guessing a PIN of an unknown user with each and every PIN number would result in the same success rate at 0.01%. However, in practice, some PINs are chosen much more frequently than the others. In Table XII, the three most popular four-digit PIN and their frequencies are presented. An attacker could maximize the success rate of guessing a PIN of a random user by choosing the most popular one, i.e., '1234'.

Similarly, given that users can choose their signatures freely, some users might choose similar signatures. That is, their signatures may form a cluster in a specific feature space. Consequently, an attacker could maximize the chance of successful impersonation by mimicking a signature of the user that belongs to a cluster with a large number of users. To evaluate clustering characteristic of online signatures, we derive a success rate of zero-effort impersonation for each user when his signatures are used as impersonation samples against all other users.

Table XI presents the top three users with the highest success rate for zero-effort impersonation in different FAR levels. While the success rate of guessing user's 4-digit PINs with '1234' is at 10.71%; the success rate of impersonating other users by signatures from the user with the highest potential is lower at 4.66% for 1% FAR. The rate is lower at 2.38% and 0.48% for 0.5% and 0.1% FAR, respectively. That is, online signatures are distributed more evenly than four-digit PINs. As a result, they can better protect users against guessing attack as compared to four-digit PINs when there is no password blacklist policy enforced against common pins.

TABLE XIII
ATTEMPT RATE (AR) AND REJECTION RATE (RR) OF THE PROPOSED ONLINE SIGNATURE VERIFICATION SYSTEM WHEN TRAINING SAMPLES ARE DRAWN FROM SESSIONS 1 AND 2 WHILE TEST SAMPLES ARE FROM SESSION 3

FAR(%)	of top 100% users		of top 95% users	
	AR	RR	AR	RR
0	1.2	6.34%	1.19	4.04%
0.1	1.12	3.23%	1.12	1.19%
0.5	1.05	1.07%	1.04	0%
1	1.02	0.93%	1.01	0%

4) Authentication Effort and Performance: Verification performance is one of the key factors that influence the usability of an authentication system. In particular, false rejection can lead to either an increase in the number of authentication attempts or user rejection and temporary lockout. While both cause usability issues, the recovery effort of the latter is more time-consuming than the former. Assuming that three failed attempts are allowed before rejecting and temporarily locking out users, rejections of genuine signatures are classified into two categories. These are failed attempts and user rejections. The first type leads to an increase in average number of attempts for successful authentication while the second one leads to an increase in user rejection rate. That is, the attempt rate is defined as,

$$\text{Attempt Rate (AR)} = \frac{\text{\# Total Attempts}}{\text{\# Successful logins}} \tag{7}$$

and the user rejection rate is defined as,

$$\text{Rejection Rate (RR)} = \frac{\text{\# Authen. Failures}}{\text{\# Authen. Failures} + \text{\# Successful logins}} \tag{8}$$

Table XIII reports an average number of attempts as well as user rejection rates at different FAR levels. Note that, in this experiment, the signatures are verified in a circular and timely order. If a signature is rejected for the first time, at most two consecutive failures can be made before the attempt is counted as a user rejection. Also, if the last signature is rejected, two more chances are given to the next two samples in a circular order. On the other hand, if a signature is accepted, the attempt is counted as successful and the number of trials before succeeding is recorded. The results show that the rejection rate, or authentication failure, is greatly reduced from the FRR reported earlier in Table X. That is because, in many cases, the users can recover from failed attempts within the next two attempts. However, similarly to any other traditional authentication system, authentication failures are inevitable and the system should always have a secure and proper backup authentication mechanism in place.

V. CONCLUSION AND FUTURE WORK

This paper proposes a simple and effective online signature verification system that is suitable for user authentication on a mobile device. The benefits of the proposed algorithm are as follows. First, a histogram based feature set for representing

TABLE XIV
EER FOR DIFFERENCE HISTOGRAM FEATURE SPACES DERIVED FROM MCYT-100 DATASET WHEN 10 SAMPLES ARE USED FOR TRAINING

Iteration No.	Histogram	Experiment 1		Experiment 2					
				$F = 1$		$F = 0.5$		$F = 1.5$	
		EER-SF	EER-RF	EER-SF	EER-RF	EER-SF	EER-RF	EER-SF	EER-RF
0	Baseline (All histograms)	2.73	0.44	2.73	0.44	4.80	1.26	2.89	0.41
1	X^2	2.67	0.48	37.13	28.20	33.35	41.97	37.93	27.55
2	Y^1	2.75	0.47	24.83	24.40	38.99	36.21	25.54	21.47
3	Y^2	2.80	0.45	36.55	31.66	33.87	35.01	36.15	30.86
4	R^2	2.89	0.48	16.57	13.26	17.95	11.97	18.35	13.66
5	Φ^2	2.93	0.49	26.45	16.00	23.95	16.06	30.20	19.21
6	$P^1_{(1,2)}$	2.93	0.51	12.51	10.18	14.00	10.54	12.11	8.47
7	Φ^1	3.02	0.53	17.85	9.89	18.75	10.88	17.13	9.60
8	X^1	3.12	0.55	27.53	21.27	39.33	29.06	26.00	18.87
9	R^1	3.18	0.62	17.35	12.49	15.20	11.47	18.60	12.21
10	$< Y^1, Y^2 >$	3.45	0.64	27.15	20.55	36.47	33.12	24.93	17.33
11	$< X^1, X^2 >$	3.76	0.61	25.53	17.46	32.07	24.92	25.41	14.34
12	$P^2_{(1,2)}$	4.07	0.62	29.35	25.90	31.75	26.35	29.79	20.81
13	$< \Phi^1, R^2 >_{(1,2)}$	4.75	0.79	11.05	4.41	19.55	17.93	11.27	3.54
14	$< \Phi^2, R^2 >_{(1,2)}$	6.00	1.08	14.92	7.13	28.55	15.66	14.73	6.41
15	$< \Phi^1, \Phi^1_{d(1,2)} >$	8.55	2.26	12.66	3.51	11.73	7.09	13.62	5.21
-	$\{ < \Phi^1, R^1 >_{(1,2)}$(left)$\}$	-	-	8.55	2.26	19.55	17.93	7.85	1.47

Experiment 1 : An experiment where attributes of each histogram are iteratively removing from the feature set
Experiment 2 : An experiment where attributes of each individual histogram are used as a feature set and # histogram bins $= F\times$ the ones in Table I

an online signature can be derived in linear time and the system requires a small and fixed-size space to store the signature template. In addition, since the feature set represents only statistics about distribution of original online signature attributes, the transformation is non-invertible. As a result, the privacy of the original biometric data is well-protected. Second, a user-specific classifier comprising of a user-specific quantization step size vector and its associated quantized feature vector can be trained using only enrollment samples from that user without requiring a training set from a large number of users. Several experiments performed on MCYT and SUSIG datasets demonstrate effectiveness of the proposed method in terms of verification performance as compared to existing algorithms.

A new dataset was collected for evaluating system performance of user authentication on a mobile device. Signatures in this dataset were drawn using a fingertip in an uncontrolled setting on user owned iOS devices over six separate sessions. Experimental results on this dataset provide similar performance as older dataset on stylus based devices and controlled environments. This shows that finger drawn signatures are plausible candidates for user authentication on mobile devices. The experiments also indicated that the well know template aging problem of online signatures is very relevant to the context of user authentication. That is, it was observed that the older the training samples as compared to the test samples, the lower the verification performance of the online signature classifier. This suggests that template updating mechanism is needed in order to stabilize the performance. Also, a significant inter-session verification performance improvement when a classifier is trained using cross-session samples, as compared to the one trained from a single-session samples, implied that samples from multiple session can reflect a more accurate intra-user variation. Last but not least, security analysis of online signature verification system as compared to that of 4-digits PIN, and two usability metrics is also presented.

One interesting area for future work is the design of an enrollment protocol that can capture a intra-user variation effectively within a single session. In addition, it is currently possible to match different signature templates generated from the same online signature samples and thereby learn that two leaked biometric templates belong to the same user. Further investigation includes the use of other biometric key binding approaches, like fuzzy commitment, in order to strengthen security of the system, even when stored templates, helper data etc., are compromised, while preserving verification performance. Lastly, it is possible to derive a fusion approach by combining the proposed method with other existing approaches, e.g., DTW, HMM-based, etc., in order to improve verification performance, especially for applications where privacy of the signature traits is less critical.

APPENDIX
DEPENDENCY OF VERIFICATION PERFORMANCE ON HISTOGRAM PARAMETERS

The histogram feature configuration in Table I is parameterized by several key factors and their modification could impact the verification performance of the system. These include the number of bins for each histogram and the inclusion of each histogram in the feature set. In order to elaborate the effect of these parameters towards verification performance, the performance result when modifying the parameters is presented in Table XIV. First, to analyze the contribution of each histogram

towards verification performance, an experiment to iteratively remove the histogram with the least contribution is performed and the results are reported. In this case, the initial set of features consists of all attributes from all histograms described in Table I, and the performance in this case is reported as a baseline. In addition, the performance when each of the histogram is solely applied as a feature set is also reported. These histograms are derived at three different resolutions: 1) the one described in Table I, 2) 50% decrease from 1), and 3) 50% increase from 1).

According to the result, verification performance of skill and random forgery gradually degrades when attributes of each histogram are iteratively removed from the feature set. It implies that these histograms, when applied with the proposed verification approach, provide complimentary information that are useful for online signature verification. In addition, the degradation rate at the higher performance is noticeably lower than the one at the lower performance. This implicates the challenge of improving the system with very high performance. With respect to histogram resolution, i.e., numbers of histogram bins, the result shows that, when all the histograms are applied, verification performance of the system with the current histogram resolutions and the one with the resolutions at 50% higher are comparable. In addition, verification performance of individual histogram at the current configuration is slightly lower than the one when the resolution increases by 50%, where it is higher than the one when the resolution decreases by 50%. These infer the robustness of the proposed approach under different configurations. However, adjusting those parameters can sometimes cause disagreement between performance of random forgery and skill forgery. For example, when increasing the number of bins of X^2, EER of skill forgery increases but that of random forgery decreases.

Future work to examine a method that can select those parameters holistically and optimally for both skill forgery and random forgery cases is warranted. Noting that, in this work, these parameters are defined experimentally on MCYT dataset and are used as an example of working configurations to demonstrate effectiveness of histogram features for online signature verification application against other datasets.

REFERENCES

[1] A. Fallah, M. Jamaati, and A. Soleamani, "A new online signature verification system based on combining Mellin transform, MFCC and neural network," *Digital Signal Process.*, vol. 21, no. 2, pp. 404–416, 2011.

[2] L. Findlater, J. O. Wobbrock, and D. Wigdor, "Typing on flat glass: Examining ten-finger expert typing patterns on touch surfaces," in *Proc. Annu. Conf. Human Factors Comput. Syst.*, New York, NY, USA, 2011, pp. 2453–2462.

[3] 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*, 2012, pp. 977–986.

[4] N. Sae-Bae, N. Memon, K. Isbister, and K. Ahmed, "Multitouch gesture-based authentication," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 4, pp. 568–582, Apr. 2014.

[5] L. G. Plamondon and R. Plamondon, "Automatic signature verification and writer identification—The state of the art," *Pattern Recognit.*, vol. 22, no. 2, pp. 107–131, 1989.

[6] H. Feng and C. C. Wah, "Online signature verification using a new extreme points warping technique," *Pattern Recognit. Lett.*, vol. 24, no. 16, pp. 2943–2951, 2003.

[7] A. Kholmatov and B. Yanikoglu, "SUSIG: An on-line signature database, associated protocols and benchmark results," *Pattern Anal. Appl.*, vol. 12, no. 3, pp. 227–236, 2008.

[8] J. Ortega-Garcia *et al.*, "MCYT baseline corpus: A bimodal biometric database," *IEE Proc. Vis. Image Signal Process.*, vol. 150, no. 6, pp. 395–401, Dec. 2003.

[9] L. Nanni, "An advanced multi-matcher method for on-line signature verification featuring global features and tokenised random numbers," *Neurocomputing*, vol. 69, nos. 16–18, pp. 2402–2406, 2006.

[10] D. Guru and H. Prakash, "Online signature verification and recognition: An approach based on symbolic representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1059–1073, Jun. 2009.

[11] J. Galbally, M. Martinez-Diaz, and J. Fierrez, "Aging in biometrics: An experimental analysis on on-line signature," *PLOS ONE*, vol. 8, no. 7, p. e69897, 2013.

[12] M. Faundez-Zanuy, "On-line signature recognition based on VQ-DTW," *Pattern Recognit.*, vol. 40, no. 3, pp. 981–992, 2007.

[13] J. Fierrez, J. Ortega-Garcia, D. Ramos, and J. Gonzalez-Rodriguez, "HMM-based on-line signature verification: Feature extraction and signature modeling," *Pattern Recognit. Lett.*, vol. 28, pp. 2325–2334, Dec. 2007.

[14] E. Maiorana, P. Campisi, J. Fierrez, J. Ortega-Garcia, and A. Neri, "Cancelable templates for sequence-based biometrics with application to on-line signature recognition," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 3, pp. 525–538, May 2010.

[15] E. Maiorana, P. Campisi, and A. Neri, "Template protection for dynamic time warping based biometric signature authentication," in *Proc. 16th Int. Conf. Digital Signal Process.*, Jul. 2009, pp. 1–6.

[16] L. Nanni, E. Maiorana, A. Lumini, and P. Campisi, "Combining local, regional and global matchers for a template protected on-line signature verification system," *Expert Syst. Appl.*, vol. 37, pp. 3676–3684, May 2010.

[17] M.-H. Lim, A. B. J. Teoh, and K.-A. Toh, "An efficient dynamic reliability-dependent bit allocation for biometric discretization," *Pattern Recognit.*, vol. 45, no. 5, pp. 1960–1971, 2012.

[18] Y.-J. Chang, W. Zhang, and T. Chen, "Biometrics-based cryptographic key generation," in *Proc. IEEE ICME*, vol. 3. Jun. 2004, pp. 2203–2206.

[19] J. Fierrez-Aguilar, L. Nanni, J. Lopez-Peñalba, J. Ortega-Garcia, and D. Maltoni, "An on-line signature verification system based on fusion of local and global information," in *Audio- and Video-Based Biometric Person Authentication* (Lecture Notes in Computer Science), vol. 3546. Berlin, Germany: Springer-Verlag, 2005, pp. 627–656.

[20] E. Argones Rua, E. Maiorana, J. Alba Castro, and P. Campisi, "Biometric template protection using universal background models: An application to online signature," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 269–282, Feb. 2012.

[21] N. Sae-Bae and N. Memon, "A simple and effective method for online signature verification," in *Proc. Int. Conf. BIOSIG*, 2013, pp. 1–12.

[22] B. Schiele and J. Crowley, "Object recognition using multidimensional receptive field histograms," in *Proc. ECCV*, 1996, pp. 610–619.

[23] Y. Qiao, J. Liu, and X. Tang, "Offline signature verification using online handwriting registration," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[24] W. Nelson, W. Turin, and T. Hastie, "Statistical methods for on-line signature verification," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 8, no. 3, pp. 749–770, 1994.

[25] L. Nanni and A. Lumini, "A novel local on-line signature verification system," *Pattern Recognit. Lett.*, vol. 29, no. 5, pp. 559–568, 2008.

[26] B. Yanikoglu and A. Kholmatov, "Online signature verification using Fourier descriptors," *EURASIP J. Adv. Signal Process.*, vol. 1, p. 260516, Jan. 2009.

[27] U. Uludag, A. Ross, and A. Jain, "Biometric template selection and update: A case study in fingerprints," *Pattern Recognit.*, vol. 37, no. 7, pp. 1533–1542, 2004.

[28] A. Kholmatov and B. Yanikoglu, "Identity authentication using improved online signature verification method," *Pattern Recognit. Lett.*, vol. 26, pp. 2400–2408, Nov. 2005.

[29] N. Houmani, S. Garcia-Salicetti, B. Dorizzi, and M. El-Yacoubi, "On-line signature verification on a mobile platform," in *Mobile Computing, Applications, and Services* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 76, M. Gris and G. Yang, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 396–400.

[30] P. Song, W. B. Goh, C.-W. Fu, Q. Meng, and P.-A. Heng, "WYSIWYF: Exploring and annotating volume data with a tangible handheld device," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2011, pp. 1333–1342.

[31] P. Tome-Gonzalez, F. Alonso-Fernandez, and J. Ortega-Garcia, "On the effects of time variability in IRIS recognition," in *Proc. 2nd IEEE Int. Conf. BTAS*, Oct. 2008, pp. 1–6.

[32] D. Rankin, B. W. Scotney, P. J. Morrow, and B. Pierscionek, "Iris recognition failure over time: The effects of texture," *Pattern Recognit.*, vol. 45, no. 1, pp. 145–150, 2012.

[33] U. Park, R. R. Jillela, A. Ross, and A. K. Jain, "Periocular biometrics in the visible spectrum," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 96–106, Mar. 2011.

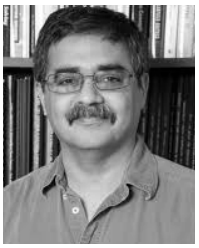
[34] I. Odinaka, P. Lai, A. Kaplan, J. O'Sullivan, E. Sirevaag, and J. Rohrbaugh, "ECG biometric recognition: A comparative analysis," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1812–1824, Dec. 2012.

[35] R. Vogt and S. Sridharan, "Explicit modelling of session variability for speaker verification," *Comput. Speech Lang.*, vol. 22, no. 1, pp. 17–38, 2008.

[36] DataGenetics. (2012, Aug. 14). *Pin Analysis* [Online]. Available: http://www.datagenetics.com/blog/september32012/

Napa Sae-Bae received the B.E. degree in telecommunication engineering and the M.E. degree in information system engineering from the King Mongkut's Institute of Technology, Ladkrabang, Thailand.

She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Polytechnic School of Engineering, New York University. Her research interests lie in the area of biometric, authentication, signal processing, pattern recognition, and consumer security.



Nasir Memon (F'11) is a Professor with the Department of Computer Science and Engineering. His research interests include digital forensics, data compression, and multimedia computing and security. He received the M.S. and Ph.D. degrees in computer science from the University of Nebraska in 1989 and 1992, respectively.

He has authored more than 250 articles in journals and conference proceedings. He holds more than a dozen patents in image compression and security. He was a recipient of several awards, including Best Paper Awards and Educator Awards. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION SECURITY AND FORENSICS. He has served on the editorial board of many journals and is currently on the editorial board of the *IEEE Security and Privacy Magazine*. He was a Distinguished Lecturer of the IEEE Signal Processing Society from 2011 to 2012.