On Continuous User Authentication via Typing Behavior

Joseph Roth *Student Member, IEEE,* Xiaoming Liu, *Member, IEEE,* and Dimitris Metaxas, *Member, IEEE*

***Abstract*—We hypothesize that an individual computer user has a unique and consistent habitual pattern of hand movements, independent of the text, while typing on a keyboard. As a result, this paper proposes a novel biometric modality named "Typing Behavior (TB)" for continuous user authentication. Given a webcam pointing toward a keyboard, we develop real-time computer vision algorithms to automatically extract hand movement patterns from the video stream. Unlike the typical continuous biometrics such as keystroke dynamics (KD), TB provides reliable authentication with a short delay, while avoiding explicit key-logging. We collect a video database where 63 unique subjects type static text and free text for multiple sessions. For one typing video, the hands are segmented in each frame and a unique descriptor is extracted based on the shape and position of hands, as well as their temporal dynamics in the video sequence. We propose a novel approach, named bag of multi-dimensional phrases, to match the cross-feature and cross-temporal pattern between a gallery sequence and a probe sequence. The experimental results demonstrate superior performance of TB when compared to KD, which, together with our ultra-real-time demo system, warrant further investigation of this novel vision application and biometric modality.**

***Index Terms*—Continuous authentication, user authentication, biometrics, typing behavior, hand movements, bag of phrases, bag of multi-dimensional phrases, keystroke dynamics, keyboard.**

I. Introduction

IT is common to validate the identity of a user for any computer system. The standard password-based, *one-shot* user authentication may create an information system that is vulnerable immediately after login, since no mechanism exists to continuously verify the identity of the active user. This can be an especially severe problem for security-sensitive facilities, where compromised passwords or insufficient vigilance after initial login can leak confidential information or give unwanted privileges to the user. Hence, a method enabling *continuous* authentication for the active user is highly desired.

One popular alternative to password-based user authentication is to employ biometrics. Biometrics refers to the identification of humans by their physical characteristics (*e.g.*, face, fingerprint, iris) or behavioral traits (*e.g.*, gait, keystroke dynamics, mouse dynamics) [15]. Among these biometric modalities, face, fingerprint, keystroke dynamics (KD), and mouse dynamics have been used in continuous user authentication. Researchers develop this wide range of biometric modalities due to a number of reasons. One, users interact with a computer in many different ways. Some predominantly use the keyboard, others the mouse, others may consume information from the screen without interacting with peripherals, and other input methods (*e.g.*, gestures) may become more common in the future. Two, users may be present at the computer or may be logged in from a remote machine. Three, each modality has its own limitation. For example, the fingerprint sensor embedded on the mouse assumes user cooperation by pressing the thumb on a dedicated position of the mouse [32]. Face authentication requires continuously capturing and monitoring facial appearance, which may give up personal information *irrelevant* to identity, such as the emotional state. Also, the processing pipeline from face detection, landmark estimation, to authentication can be computationally intensive. Both keystroke dynamics and mouse dynamics require the length of probe sequences to be at least a few minutes for reliable authentication [38], [46], which indicates a long verification time - a higher risk of delayed detection of an imposter.

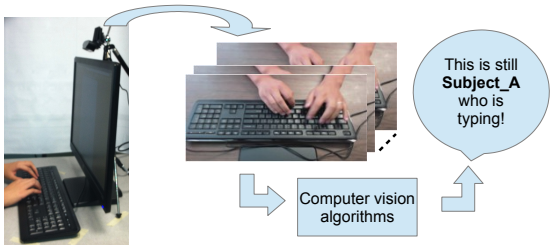


Fig. 1. Concept of typing behavior. With a webcam pointing at a keyboard, the video stream of a user's hand movements is continuously fed into the proposed computer vision algorithms, which verify the spatio-temporal patterns of the movement with those of the claimed subject, so as to achieve continuous user authentication.

Due to the aforementioned issues, much room remains from an academic point of view to explore novel biometric modalities for continuous computer user authentication. We aim to explore a biometric with *short* verification time, *ultra-real-time* efficiency, and *no interference* with normal computer operation, especially related to the standard input devices, *e.g.*, keyboard and mouse. In psychology literature, Ouellette and Wood state that frequent behavior forms a habit where actions from the past can predict actions in the future [25]. This claim of consistent user behavior leads us to study the potential of using the "frequent" keyboard typing behavior as a biometric modality. In particular, we hypothesize that every

Joseph Roth and Xiaoming Liu are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824.

Dimitris Metaxas is with the Computer Science Department, Rutgers University, Piscataway, NJ 08854.

Corresponding author: Xiaoming Liu, liuxm@cse.msu.edu

computer user has a unique and consistent habitual pattern of hand movements, independent of the text, while typing on a keyboard. Using a simple hardware setup demonstrated in Fig. 1, we can capture a video of the hands without any user cooperation and based on the video *alone* our system provides an authentication decision with a short verification time. We name this novel biometric modality as *Typing Behavior (TB)*.

Specifically, given one typing video frame, it segments the hands from the background and then separates the left and right hand regions. A shape context based feature is extracted from each hand region [7], which is concatenated with the centroid positions to form the descriptor for each frame. We use the *Bag of Words* [12] concept to convert the descriptor into a word, and represent a typing video as a histogram of words. Thus the matching of videos can be done between the histogram of a probe and that of the gallery. We also model the temporal dynamics of the words in a video via *Bag of Phrases* [45], where a phrase is a set of words with a certain temporal layout. Furthermore, we propose a novel extension of Bag of Phrases, *Bag of Multi-dimensional Phrases*, where a video frame is represented as a multi-dimensional word, and the phrase is composed of cross-feature and cross-temporal words. To study this novel biometrics, we conduct a two-phase data collection including 63 computer subjects type the static text or free text in multiple sessions. The extensive experiments demonstrate excellent and superior performance of TB. For example, on a 30-subject free text dataset collected weeks apart, TB achieves 4.9% Equal Error Rate (EER) with 20-second probes, while KD has 20.9% EER.

In summary, this paper has three main contributions:

◇ We propose a novel biometric modality that offers real-time continuous user authentication while typing. Through extensive experiments and an ultra-real-time demo system, we show that TB can provide excellent authentication performance with a short verification time.

◇ We collect a first-of-its-kind, multi-session video database of subjects typing on a keyboard. We will make the database and our experimental results publicly available so as to facilitate future research efforts on this topic.

◇ We propose a Bag of Multi-dimensional Phrases approach to match two multi-dimensional time series data. We experimentally demonstrate its strength over prior work.

The remaining of this paper is organized as follows. Section II reviews previous work relevant to our study. The details of our algorithm in computing the similarity between two typing videos are provided in Section III, which is followed by applying our algorithm to the context of continuous user authentications in Section IV. In Section V, we describe the design, procedure, and outcome of collecting a keyboard typing database. The extensive experimental results, as well as the comparison with previous algorithms and keystroke dynamics, are presented in Section VI. We conclude this paper and provide future directions in Section VII.

II. Prior Work

To the best of our knowledge, there is no prior work in exploring the visual aspect of keyboard typing for user authentication. Hence, this section focuses on biometric modalities used in continuous authentication, computer vision tasks involving hands, and general continuous authentication research. To begin, we identify five different biometric modalities for continuous authentication: face, fingerprint, mouse dynamics, keystroke dynamics, and keystroke sound.

Face verification has been an active research topic for decades [17], [35]. In continuous computer authentication, users may routinely look away from the monitor (*i.e.*, camera) and researchers address this by integrating different modalities alongside faces. For example, Sim *et al.* integrate fingerprint [32], while Niinuma *et al.* use clothing information [24].

Two techniques are based on mouse interaction. *Fingerprint* authentication uses a sensor embedded on a specific area of the mouse [32]. *Mouse dynamics* has been used since users exhibit habitual patterns in moving the mouse while operating a computer [3], [26], [31]. Features are formed from angles, acceleration, and distance moved. However, more than 3 minutes of mouse interaction is required to make an authentication decision with 2.08% EER for free mouse movement, as indicated by a recent paper [46].

Keystroke dynamics, habitual time delays between key presses, is considered as a natural biometric modality for user authentication due to a few desirable properties, such as non-intrusiveness and no requirement for user cooperation [6], [30]. However, KD also has a number of drawbacks. First of all, given the limited 1-dimensional key-pressing signal, the distinctiveness of KD is less than desired, reflected by the fact that most prior work concentrate on static text [16], [47], *i.e.*, users type the same text. Only a few research efforts concern the free text, *i.e.*, users type arbitrary text, which is imperative for continuous authentication [14], [23]. Second, KD demands *long* probe sequences because its digraph features require a sufficient number of common pairs in both gallery and probe to make a reliable decision. For example, [38] needs a probe of at least 700 characters which is about 3 minutes long. This implies a long verification time or authentication delay for continuous authentication, which has the risk of delayed detection of an impostor. In real-world typing, after a user makes a spontaneous pause, during which an impostor could take control of the keyboard, KD will be unreliable during the authentication delay. The aforementioned limitations motivate us to explore the visual aspect of typing, with potentially improved performance due to higher-dimensional visual content.

Another recent technique based on keyboard interaction uses the sound from keystrokes to identify the user. Previous work use the discriminative abilities of sound to detect the key presses [5], [48]. Roth *et al.* demonstrate the potential of using *keystroke sound* alone for user authentication and suggest fusion with keystroke dynamics could result in improved performance [27]. While KD and keystroke sound explore the timing and acoustic aspects of keyboard typing respectively, our work, for the first time, studies the visual aspect of typing.

Hand is well studied in computer vision with extensive work on hand tracking [20], gesture recognition [22], American Sign Language recognition [37], [39], *etc*. There are prior work in using the handprint [40], finger-knuckle-print [42], [43], or hand shape [9] for person identification, which differ to our work in two aspects. One is that the hand is typically scanned

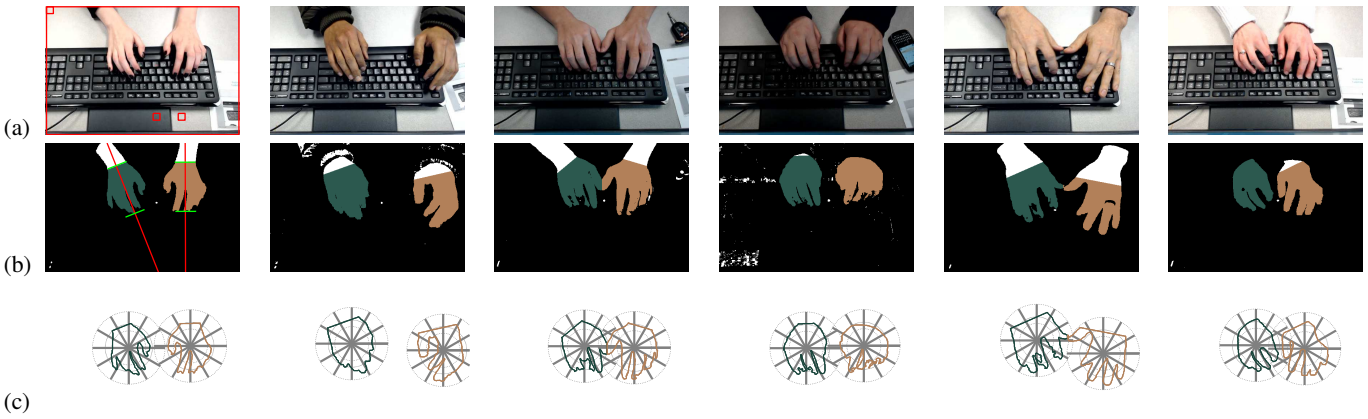


Fig. 2. Feature extraction steps: (a) original frames from multiple subjects, (b) foreground segmentation with hand separation, (c) shape context extraction. The top-left image shows four patches used in the linear regressor. The left-middle image shows the cutoff between hands and arms. Note the large appearance variations of typing video frames due to skin colors, lighting, cloth colors, and sleeve lengths.

in a fixed position while being fully open or clenched into a fist. The other is that little video or temporal information is utilized in these hand-based biometrics.

In terms of research on *continuous authentication*, a report prepared by the U.S. Military Academy [36] presents a set of guidelines for continuous authentication systems as required by the Department of Defense. They outline terminologies for reporting the performance of biometrics in the one-shot and continuous authentication scenarios. Serwadda *et al.* propose that the temporal distribution of false rejections is important in continuous authentication systems and can help determining when to update biometric templates [29]. In continuous authentication, the system has the opportunity to correct mistakes and operate in periods of uncertainty. Altinok and Turk discuss the temporal integration of biometrics in continuous authentication [4]. Certainty of an authentication decision decreases as time passes and an imposter has the opportunity to take over the computer. By incorporating face, voice, and fingerprint, their system maintains a level of confidence even when none of the biometrics produces measurements.

III. Video-to-Video Matching Algorithm

The core task in TB-based biometrics is to compute the similarity between a gallery and probe video sequence of hand movements during typing. In our work, a *gallery* is the typing video sequence collected during the enrollment of a known subject. A *probe* is the video sequence captured while a to-be-authenticated subject is typing. In this section, we first introduce the feature extraction of each video frame, and then present a series of algorithms to compute the similarity between two videos.

A. Feature Extraction

Given a typing video, the first step is to extract certain feature representations from each frame, the collection of which over time constitutes a rich signature for the video. There are many potential features for representing human hands, such as shape, color, texture, *etc*. In this work, we choose to use the hand shape as the feature for three considerations. First of all, considering the large amount of motion in typing, there is more *dynamic* information in shape to be explored than appearance, which is the most unique perspective distinguishing our TB biometrics from the prior hand-based biometrics. Second, hand shape is potentially more discriminative than color or texture, as demonstrated in handprint biometrics [40]. Third, shape features can be very efficient to extract and match due to their lower dimensionality, which is a critical property for computational-sensitive continuous authentication applications. However, color and texture can still be useful for the hand-based biometrics, as demonstrated in the finger-knuckle-print [42]. Thus, we will incorporate them to constitute a richer feature representation, as part of the future work.

To extract shape features, we follow the procedure of foreground segmentation, hand detection and separation, and feature computation, as shown in Fig. 2. On one hand, the background in our application is rather favorable to vision algorithms because of the popularity of black color keyboards and neutral-colored desks. On the other hand, our algorithm needs to handle large appearance variations due to skin colors, lighting, cloth colors, and sleeve lengths. Also, the continuous authentication application demands highly efficient visual processing, in order to avoid interfering with the normal computer operation. These observations lead to a carefully designed low-level image processing module, as described below.

Foreground segmentation: An efficient foreground (skin) segmentation converts each RGB pixel in image $\mathbf{I}$ to a scalar via a vector $\mathbf{w}$, followed by applying a threshold θ to the scalar,

$$\mathbf{M} = \mathbf{I}\mathbf{w} > \theta. \tag{1}$$

We determine the optimal vector $\mathbf{w}$ and threshold via a set of (27 in our experiments) representative RGB images, each denoted as an $E \times 3$ matrix $\mathbf{I}_i$ where E is the number of pixels, along with their ground-truth masks $\mathbf{M}_i$. Linear Discriminative Analysis [21] determines the optimal $\mathbf{w}$ that best separates the hand regions from the background. Due to the lighting diversity across videos, an image-dependent threshold may lead to superior segmentation than a global threshold. The threshold with least-square error for each image θ_i is determined by minimizing the Frobenius norm between

the thresholded image and the ground-truth mask, *i.e.*,

$$\theta_i = \arg\min_x \|\delta(\mathbf{I}_i\mathbf{w} - x) - \mathbf{M}_i\|_2, \quad (2)$$

where $\delta()$ is the indicator function. Using the set of $\mathbf{I}_i$ and θ_i, we learn a linear regressor that can predict θ_i from the mean intensities of four pre-defined patches in $\mathbf{I}_i$,

$$\hat{\theta_i} : \mathbf{I}_i \mapsto r_0 + \sum_{j=1}^{4} r_j\, m(\mathbf{I}_i\mathbf{w}, \mathbf{b}_j), \quad (3)$$

where $m()$ defines the mean intensity of a patch $\mathbf{b}_j$ and r_j is the coefficients of the learned regressor. The top-left image of Fig. 2 displays the four patches used in our experiments. Thus $\mathbf{w}$ and the regressor can be used to adaptively estimate a segmentation threshold for an arbitrary typing video frame.

Hand detection and separation: We detect the existence of hands when a sufficient number of foreground pixels are present. If detected, we examine the largest two connected components as candidate hand regions. Infrequently one region is substantially larger than the other, an indication of merged hands, and we must perform hand separation by finding the trough in the curve which spans the width of image and sums the number of foreground pixels in the vertical direction. Finally people wear sleeves with different lengths, which results in various vertical lengths of hand regions. To mitigate their effect on the shape feature, we compute the direction of maximum variance for a hand region and perform a constant cutoff along this direction according to the average hand length, as shown in the far-left image of Fig. 2 (b). Although this solution is less ideal than a more sophisticated hand fitting method based on 2D or 3D hand models, it is more efficient and we observe experimentally that it is adequate to handle the variability in our collected video data.

Shape context representation: Given the rich literature on shape representations, we choose to use the shape context as the shape feature mainly due to its efficient computation and proven effectiveness in describing the shape of objects [7]. Also, being a histogram, the shape context is robust to intra-class shape variations, which is favorable for our application since small finger movement or segmentation error will likely have little effect on the resultant shape context. Specifically, after finding the boundary pixels of the hand region, for each hand we use a shape context of three rings and 12 angular sections, centered at the centroid of the hand region, as shown in Fig. 2 (c). The radiuses of three rings are constant for all subjects and determined such that each covers 33%, 66%, and 100% of all hand sizes respectively. By binning the boundary pixels into sections, we obtain two 36-dimensional shape context histograms, normalized w.r.t. the total number of boundary pixels in each hand. We denote the histograms as $\mathbf{s}_l$, $\mathbf{s}_r$ for the left and right hand respectively. Similarly, the two centroid locations normalized between $[0, 1]$ w.r.t. the image size are denoted as $\mathbf{x}_l$, $\mathbf{x}_r$. For each frame, all features are concatenated to form a 76-dimensional descriptor encoding the hand shapes and locations, denoted as $\mathbf{f} = [\mathbf{x}_l^\top\ \mathbf{x}_r^\top\ \mathbf{s}_l^\top\ \mathbf{s}_r^\top]^\top$.

B. Bag of Words

Given the descriptors of each video $\{\mathbf{f}_i\}$, we use a number of methods to compute the similarity between two videos. The first is the popular Bag of Words (BoW) approach [8], [12], [33], which is known to be efficient and robust to intra-class variations. To have a word representation for a video, we learn a codebook of D codewords via K-means clustering. To take advantage of redundant information in a high-frame-rate video, we treat the concatenated descriptors from consecutive L frames, $\mathbf{g}_i^{(L)} = [\mathbf{f}_{i-L+1}^\top \dots \mathbf{f}_i^\top]^\top$, as an input sample for clustering. The input samples are collected from video frames of the gallery set, $\{\mathbf{g}_i^{(L)}\}$, where $i = L, \frac{3L}{2}, 2L, \dots$ if $L \geq 2$, or else $i \in \mathbb{N}^+$. As shown in Fig. 3 (a), the consecutive samples have half overlap between their L-frame segments. A larger segment length L will lead to less samples per second and hence more efficient computation. The learned D codewords describe the dominant combinations of shape and locations between two hands, within L frames.

Given the D codewords, a video can be represented as a collection of visual words, and denoted as $\mathbf{V} = \{w_i, t_i\}$, where w_i is the closest codeword for the feature $\mathbf{g}_i^{(L)}$, and $t_i = i$ is the time index of the corresponding word w_i. Note that each word carries the local dynamic information of hands within a small window of L frames when $L \geq 2$. In Section VI, we will compare the authentication performances under various values of L. With this representation, BoW computes a D-dimensional histogram $\mathbf{h} = [h_1, h_2, \dots, h_D]^\top$ of video $\mathbf{V}$ by $h_d = \frac{\sum_i \delta(w_i = d)}{|\mathbf{V}|}$, where the denominator is the total number of words in $\mathbf{V}$. The similarity between two videos, $\mathbf{V}$ and $\mathbf{V}'$, is the inner-product of two histogram vectors, $K_w(\mathbf{V}, \mathbf{V}') = \mathbf{h}^\top\mathbf{h}'$, which is a typical choice for BoW approaches [33].

C. Bag of Phrases

Although BoW works well in many applications, it discards all structural, spatial or temporal, information among the words. Hence, many prior works aim to extend BoW by incorporating spatio-temporal relationship of the words [28], [34], [45]. We leverage the recently proposed Bag of Phrases (BoP) approach [45], due to its higher efficiency in capturing high-order correlation among words when compared to other BoW extensions.

An order-k phrase refers to a collection of k words arranged in a certain order and relative position. A histogram of phrases models the distribution of these collections of words. However, the number of possible phrases, *i.e.*, the length of the histogram of phrases, is extremely huge because of all possible combinations of words and their relative positions. Hence, it is not only computationally infeasible to calculate the histograms of phrases for two videos and their inner product, but also inefficient due to the high sparsity of the histograms, *i.e.*, most of the histogram bins are zero. In other words, we are only interested in how the non-zeros bins of two histograms are similar to each other.

It has been shown that the similarity of two histograms of phrases can be efficiently computed via the Co-Occurring Phrases (COPs) between two videos [45]. A COP is a set of k words appearing with the same temporal layout in both videos. We can use the concept of the offset space to efficiently identify the COPs. As shown in Fig. 3 (b), both videos

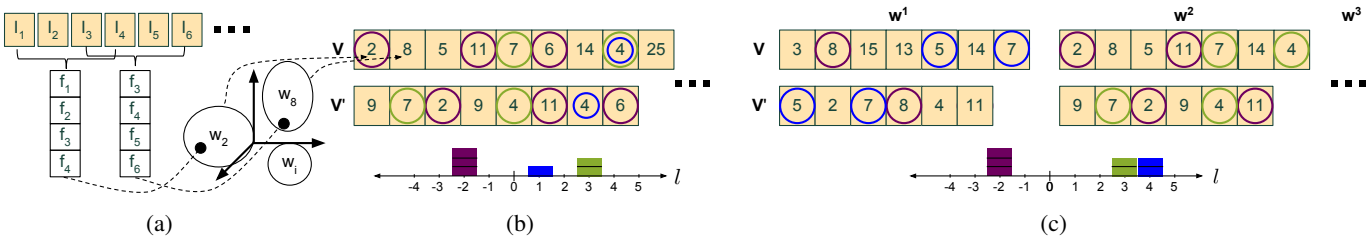


Fig. 3. (a) The process of converting a L-frame segment into a word where $L = 4$; (b) COP calculation via an offset space in BoP; (c) Co-occurring multi-dimensional phrase calculation via a single offset space in BoMP. Circles or blocks of the same color constitute a COP, *e.g.*, the words 2, 11, and 6 in (b) form an order-3 COP and contribute 3 votes to $l = -2$ (best viewed in color).

are represented as sequences of words, $\mathbf{V} = \{w_i, t_i\}$ and $\mathbf{V}' = \{w'_j, t'_j\}$. If two words are the same, $w_i = w'_j$, we add one vote at the location of $l = t_i - t'_j$ in the offset space. For example, the same word 7 in two videos contributes one vote to $l = 3$ since their time indexes differ by 3.

Now we need to calculate the number of order-k phrases for all possible k. Given the resultant offset space, the number of votes n_l at the location l indicates that n_l words exist in both videos with a temporal offset of l. If $n_l \geq k$, it contributes to the similarity by $\binom{n_l}{k}$, since an order-n_l phrase may be viewed as $\binom{n_l}{k}$ number of order-k phrases. For example, there are six order-1 phrases, four order-2 phrases, and one order-3 phrase in Fig. 3 (b). With that, we compute the number of order-k COPs as $K_k(\mathbf{V}, \mathbf{V}') = \sum_l \binom{n_l}{k}$.

Furthermore, the similarity between two videos is the total number of COPs at various k, as follows:

$$K_p(\mathbf{V}, \mathbf{V}') = \sum_{k=1}^{\infty} \frac{K_k(\mathbf{V}, \mathbf{V}')}{\sqrt{K_k(\mathbf{V}, \mathbf{V}) K_k(\mathbf{V}', \mathbf{V}')}}. \quad (4)$$

The prior work [44] shows that the total number of all COPs, $\sum_{k=1}^{\infty} K_k(\mathbf{V}, \mathbf{V}')$, equals the inner product of two sparse and high-dimensional histograms of phrases. The normalization in Eqn. 4 is to make the similarities comparable when two videos with different lengths are matched.

D. Bag of Multi-dimensional Phrases

Bag of Phrases assumes the features of a video frame can be sufficiently and efficiently represented by one word, which converts a video to a 1-dimensional time series data. However, this assumption may not be satisfied for the following reasons. First, from one frame, we can extract multiple features with different feature types, such as the centroid of hands or the shape context. It might not be best to concatenate multiple feature types into one vector and wish it is well clustered in the feature space. Second, given the fact that certain feature pairs within the concatenated vector may have relatively low correlation, it wastes the representational power of words by modeling their joint distribution via clustering. For example, two frames of the same subject may differ only in the position of the left hand and yet be mapped to different codewords despite being otherwise identical, which is therefore not effective to model intra-subject variations.

To address this issue, we propose a novel extension of BoP, *Bag of Multi-dimensional Phrases (BoMP)*. It allows the use of multiple words, rather than one word, to describe the features, and a phrase will be learned across both words and other domains, *e.g.*, the temporal domain. Specifically, given multiple feature representations for video frames in the gallery, we learn the codebook for each individual feature representation. For example, in our application, the K-means clustering is conducted four times, once each for the shape context and the centroid of the two hands respectively. With four codebooks, a typing video is represented as a 5-dimensional time series data $\mathbf{V} = \{w_i^1, w_i^2, w_i^3, w_i^4, t_i\}$. As a result, while matching two time series data, the COP will be a set of words with the same temporal and cross-feature layout between both data.

Even though BoMP seems more complicated than BoP, we can still *efficiently* compute the COPs with a change to the offset space voting. As shown in Fig. 3 (c), we define a single offset space for the words in all dimensions. The process of finding the same words is conducted between the word series in the same dimension. Once a pair with the same word is found, *e.g.*, $w_i^2 = w_j'^2$, one vote is added at the location of $l = t_i - t'_j$ in the offset space, where a high vote indicates the co-occurrence of a set of words at different time intervals and different feature dimensions. With the resultant offset space, we use the same method to calculate $K_k(\mathbf{V}, \mathbf{V}')$ and the normalized $K_p(\mathbf{V}, \mathbf{V}')$.

In BoP and BoMP, we use the online (or incremental) offset space calculation [45] to operate in a continuous manner, where a small shift in the probe window only requires subtracting and adding to the offset space from the boundary frames without fully recomputing the offset space.

In essence, a hand typing video is a series of varying postures of two hands. When one subject types the *same* content twice, there is no guarantee that the postures of one video will exactly match with those of the other video, *i.e.*, share the same word at every time instance, because of factors such as subject's emotion, health status, *etc*. When one subject types *free text* in two videos, this statement is even more true. Hence, a good matching algorithm should be able to discard the occasional noisy hand postures that might deviate from the normal movement pattern of a subject, and focus on the essential common postures between two videos. The BoP and BoMP algorithms are exactly suitable for this purpose. As shown in Figs. 3 (b,c), the words without circles are not used in the offset space and hence make no contribution to the similarity computation. In contrast, the words with circles in BoMP, as part of COPs, indicate the subset of features as well as the time instances where feature matching happens between two videos. This powerful matching scheme is the

main reason that our system goes beyond the BoW approach and achieves superior performances, as will be demonstrated in Section VI. We believe that matching two multi-dimensional time series data while ignoring noisy samples is essential to general activity or behavior recognition problems, and hence our proposed BoMP algorithm is applicable to these general problems as well. For example, the early version of BoMP, BoP, has demonstrated the state-of-the-art performances on a wide variety of activity recognition benchmark databases [45]. Note that, while other work [11], [41] present many improvements to BoW, *e.g.*, keypoint selection and histogram density mapping, they are not along the same direction as BoP and proposed BoMP, which focus on higher order interactions among the local words.

IV. TB-based Continuous Authentication

In this section, we discuss a number of aspects related to implementing the video-to-video matching algorithm in a practical system, so as to achieve typing behavior based continuous user authentication. These include the two stages of the system, gallery and probe schemes, continuous vs. periodic authentication, multiple gallery sessions, and the computational cost.

There are two main stages in our authentication system. In the *enrollment* stage, the system captures a user typing a given paragraph or given task, extracts the full 76-dimensional feature vector from each frame of the video, translates the features to the words using the previously trained codebook, and stores the indexes of each word to a lookup table for efficient matching, similar to the inverse indexing idea of the retrieval community [49]. The word lookup table comprises the *biometric template* for the user. In the *authentication* stage, after the user claims an identity, the system uses the real-time video feed to identify the words for each video frame and compute a similarity score against the biometric template. If the score is above a certain threshold, the user is allowed access to the system, otherwise deemed an impostor and rejected access.

Figure 4 depicts the two schemes designed for evaluating our algorithms. In Scheme 1, the probe length l_p remains constant and shifts throughout the video to obtain a large number of probes, which allows us to study the relation between the probe length and the authentication performance. In Scheme 2, the probe length starts small and expands to include the most recent video frames without removing any of the past. In Section VI, we conduct experiments with both schemes to determine how much typing data is necessary to make a reliable decision and to compare algorithms.

During authentication, the user may remove his or her hands from the keyboard to use the mouse or adjust papers on the desk, or the user may consistently type for a prolonged period of time. When the hands are removed from the camera's field of view, the similarity computation must be restarted since a different user could now be in control. A practical system will incorporate as much information as it can confidently assign to the active computer user by using either Scheme 2 or another technique for accumulating match scores from

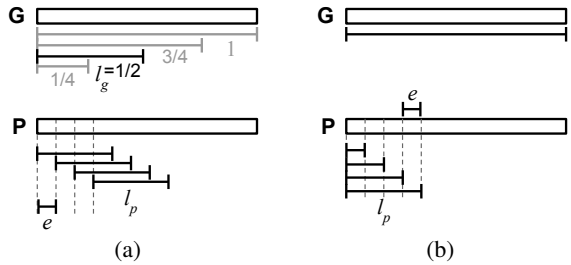


Fig. 4. Visualization of both gallery and probe schemes. In Scheme 1 (a), we choose a gallery length, l_g, as a fraction of the stream and the probe is of length l_p with a sampling interval of e both in seconds. In Scheme 2 (b), we use the full gallery stream with probes of varying length l_p all originating from the start of the stream.

previous probes. Thus, how to integrate all prior decisions and arrive at a new decision for the current time instance can be part of our future work.

Strictly speaking, continuous authentication monitors the identity of the user throughout their entire duration of device use. Practically speaking, computers can only implement periodic authentication where it ascertains the user's identity at discrete intervals. As the interval length becomes shorter than the time required for an impostor to gain access to the system, the system achieves a working version of continuous authentication. As the interval decreases in length, it becomes paramount that the system operates in a passive manner without placing any requirements on the user. We conduct most experiments with an interval length of $e = 1$ second.

As with any behavioral biometric, a single gallery session may not incorporate all intra-subject variations due to emotions, stress, tiredness, or other subject conditions. Using multiple gallery sessions and averaging the similarity score of the probe compared against all galleries can better encompass the intra-subject variations. This technique is used with keystroke dynamics as well [13].

Given a gallery session of length m frames and a probe sequence of length n frames, the computational cost for the system is as follows. To process the raw video into the 76-dimensional feature vectors is $O(n)$. To assign codewords is $O(Dn)$. To perform the probe normalization and update the similarity score is $O(n^2)$ and $O(nm)$ in the worst case respectively, but this only occurs when all frames in both videos are assigned to a single codeword. In practice, both steps are $O(n)$ on average since we use the lookup table. The entire process then is $O(Dn)$ and is theoretically efficient. In Section VI we present the empirical speed of the system broken down for the various tasks.

V. Database Collection

A good database is critical for computer vision research. In particular, the design of a data capture protocol can greatly influence the outcome of the biometric research. Since TB is a novel biometric modality with no prior public database, we aim to develop a capture protocol that will ensure the database is not only useful for the current research problem, but also beneficial to other typing-based research. All collected data used in this paper is publicly available at

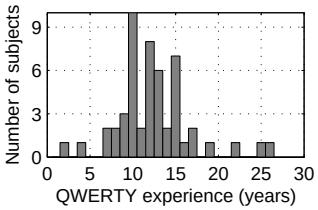


Fig. 5. Typing experience distribution of Phase 1 subjects.

http://www.cse.msu.edu/~liuxm/typing, in order to facilitate future research and performance comparison on this topic. In this section we present an overview of the database including the technical setup and the motivation behind the protocols.

Our database collection consists of two phases. In Phase 1, 51 subjects type a static text and a free text session. This allows us to explore the distinctiveness, collectability, and performance [15] of TB in a constrained environment. In Phase 2, 30 subjects type multiple fully unconstrained sessions each on a different day across the span of a month. This allows us to explore the permanence and performance under changing environmental factors. There are 18 overlapping subjects between two phases for a total of 63 unique subjects.

Equipment and environment: For both phases, we use the same US standard QWERTY keyboard, HP computer, and monitor. Although there are a variety of options for video collection, we decide to utilize an inexpensive webcam with an embedded microphone. We fix the webcam with a tripod centered behind the monitor and point it down at the keyboard. This setup uses commodity equipment and allows for the capture of audio data along with the visual data, which may be useful for a multi-modal typing system. It captures 30 FPS videos at a frame size of 1280×720 pixels.

For Phase 1, the system is setup at a spare desk in our lab at the Michigan State University (MSU). Under the monitor of the researcher working on the project, subjects perform the typing with the same chair, fixed keyboard position, lighting, and as similar computing environment as possible. We collect Phase 1 during October and November of 2012. For Phase 2, the system is moved to a shared access lab where participants are able to complete their typing at their time of choice, *without* the monitor of the researcher. Subjects come multiple times from March through July of 2013. Different chairs in the lab are used based on the subject's preference. There is also a large window in the room that causes different lighting scenarios based on the time and weather of the day.

Phase 1: For Phase 1, each subject performs two typing sessions. In the first session, a subject continuously types the first paragraph of "A Tale of Two Cities" by Charles Dickens displayed on the monitor for four times, with a 3–4 second break between consecutive times. The subject is asked to remove their hands from the keyboard during the break so they get a fresh hand position. In the second session, the subject types a half-page letter with any content to his or her family. As observed from the data in this session, most subjects make spontaneous pauses during the typing, which mimics well the real-world typing scenario. These two sessions correspond to the case of static text and free text respectively. Normally one

TABLE I
AGE DISTRIBUTION OF PHASE 1 SUBJECTS.

Age	10–19	20–29	30–39
Number of subjects	11	36	4

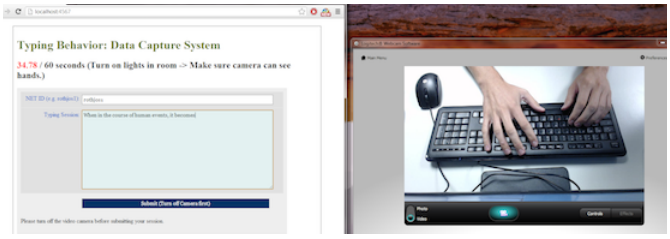


Fig. 6. Screen capture of Phase 2 data collection.

subject spends 5-8 minutes on each session depending on the typing speed, with a 30-second break between the two sessions to reiterate the instructions.

To study the various factors that may affect TB's distinctiveness, each subject is asked to finish a survey with four questions, *viz.*, the age group, years of experience in using keyboard, the major type of keyboard, and years of experience in using QWERTY. Figure 5 presents the typing experience distribution, and Table I demonstrates the subject age. Most subjects are from the CSE department of MSU, either students or faculty members, but some are friends of participants from other departments or outside the university.

Phase 2: In Phase 2, we collect multiple 60-second sessions from each subject with a simple HTML form to enter content, as shown in Fig. 6. Subjects are given no instructions on what to type. Examples of content entered during the collection are transcribing a news article, writing a diary entry, composing an e-mail, or complaining about schools. A subject can come to the lab for a typing session *any time*, while at most two sessions per week and one session per day. A 60-second timer is displayed above the form that counts while the subject types. Unlike Phase 1, keystroke press and release timings are captured through Javascript to allow for direct comparison between TB and KD. This setup is similar in nature to prior KD work [13], [38], whose Javascript-based keystroke timing has achieved satisfying KD performance. Since Phase 2 collection is free text, entirely *unmonitored* by our researchers, there are many typical real-world variations due to lighting, chair, sleeves, background environment, and distinct typing contexts that are out of our control. Greater efforts have been made to lure participants back and so far we have 140 sessions, *i.e.*, 5 sessions per subject for the majority of 30 subjects.

In Phase 1, we denote the static text session as $S1$ and the free text session as $S2$. Furthermore, the video for each session is split into 4 sequences, which are denoted as $S11$, $S12$, $S13$, $S14$ for each typing of the paragraph in the static text session, and $S21$, $S22$, $S23$, $S24$ for equal time length divisions of the free text session. The letter S indicates the *single* day collection of these sessions. In Phase 2, we denote each session Mi where i is the index of the chronological ordering of sessions for a given subject. The letter M indicates

TABLE II
SUMMARY OF ∼9 HOURS KEYBOARD TYPING VIDEOS.

Phase 1			Phase 2		
# sub	video length		# sub	video length	
51	$S1$	∼400 sec.	30	Mi	∼5 × 60 sec.
	$S2$	∼254 sec.			

the *multiple* days used during the collection of a subject. We use the first five 60-second typing sessions, denoted as $M1$, $M2$, $M3$, $M4$, and $M5$. Table II gives a summary of the collected ∼9-hour video database.

VI. EXPERIMENTAL RESULTS

In this section, we present the experimental results on databases of both phases. We begin by comparing and analyzing the algorithms, tuning the internal and external parameters of BoMP, and demonstrating the distinctiveness and performance of TB with the Phase 1 dataset. We then take a detailed look at the effectiveness of TB for continuous authentication by comparing with KD, understanding the effects of multiple gallery sessions and probe intervals, and looking at the computational efficiency with the Phase 2 dataset.

Gallery and probe: Static text in both gallery and probe is an easier problem and initial KD works only examine static text. Since we find excellent results using the more challenging free text comparison, we ignore using $S1$ as probes and only use it as a gallery to compare different typing contexts, transcription, and letter writing. Under Scheme 1 (Fig. 4 (a)), we use fractions of $S11$ or $S21$ as the gallery sequence, *e.g.*, $\frac{1}{2}$ $S11$ gallery is the first half of the first typing of "A Tale of Two Cities". We generate probe sequences by moving a fixed length (5-20 seconds) window along the last three quarters of the second session ($S22$, $S23$, $S24$) where the consecutive windows differ by 1 second. We choose a short length ($\leq$ 20 seconds) probe to minimize the initial verification time whenever a subject starts to use a keyboard.

For Phase 2 dataset, we use the first 1, 2, 3, or 4 sessions as gallery and the remaining sessions as probes. The total number of genuine and imposter matches with 20-second probes in Phase 1 and Phase 2 is ∼$594,000$ and ∼$35,730$ respectively, for each of the 5 galleries.

Evaluation metrics: We use the ROC curve as the performance measurement, which has two axes, False Positive Rate (FPR) and True Positive Rate (TPR). FPR measures the fraction of impostors incorrectly granted access to the system, while TPR measures the fraction of genuine matches correctly identified. We also use "Area Under the Curve" (AUC) and Equal Error Rate (EER) to summarize the ROC. Furthermore, for user authentication applications, there are typically preferred operation points on the ROC. For example, it is very important to reduce the FPR since the cost of wrongly authenticating an impostor is extremely high. On the other hand, it is also preferred to maximize the TPR since wrongly rejecting a genuine user is not convenient to the user. Actually, the European standard for access-control systems specifies a TPR of more than 99%, with an FPR of less than 0.001% [10]. Therefore, we use two additional overall metrics, TPR when FPR is 0.001% (denoted as TPR_0) and FPR when TPR is 99% (denoted as FPR_0), for performance comparison. Finally, we measure verification time, since it is vital to make a decision with the minimal delay to prevent unauthorized access.

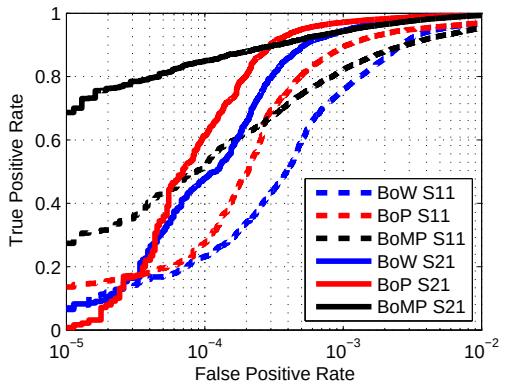


Fig. 7. ROC curves of three algorithms.

A. Algorithm Comparison with Phase 1

Comparing approaches: Using Scheme 1, we compare the performance of the three algorithms (BoW, BoP, BoMP) with $l_p = 5$ and $e = 1$ for the probes and $l_g = \frac{1}{4}S11$ or $\frac{1}{4}S21$ for the gallery. In total we have $12,623$ genuine matches and $618,527$ imposter matches. The first method is the conventional BoW approach, where we are interested in the performance of a collection of frames in a probe sequence. The second method, BoP, explores the temporal pattern in a sequence by representing each frame as one word. The third method, BoMP, investigates the cross-feature and cross-time pattern by representing a sequence as a multi-dimensional time series data. In all methods, we use $L = 1$ and $D = 200$ in learning a codebook by clustering the features of all gallery data, except that BoMP has four codebooks.

Figure 7 illustrates the comparison of the three methods, and Table III lists three metrics of ROC. Higher values of AUC and TPR_0 are desired while a lower value for FPR_0 signifies better performance. We can make a number of observations here. First, the overall performance of $S21$ gallery is better than that of $S11$ gallery. This is expected because $S21$ has context relevant to $S22$–$S24$ probe as both regard letter writing, while $S11$ is in a different context of transcription. Given the fact that in real-world applications, the context being typed in the gallery and probe will likely be very different, we should pay more attention to the tests of $S11$ gallery since it mimics the application scenario better. Second, comparing among the three methods, the overall performance improves from BoW to BoP to BoMP, indicated by increasing AUC values. Specifically, the improvement of BoMP over other methods concentrates on the primary operation points of ROC, especially for TPR_0 in the scenario of $S11$ gallery, which is very important as this is where the system will be operating in practice. For example, BoMP at least doubles TPR_0, and FPR_0 reduces to a quarter for $S11$ gallery, and BoMP has more than 10 times better TPR_0 for $S21$ gallery.

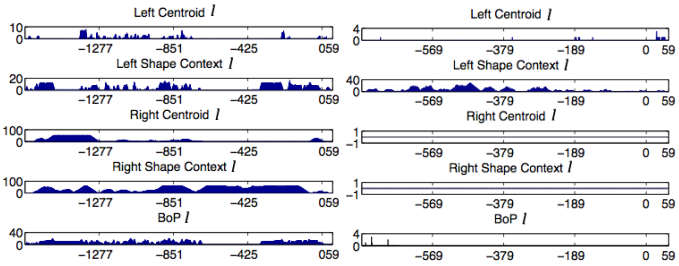
TABLE III
METHOD COMPARISON BY AUC, TPR_0 (%), AND FPR_0 (%).

	AUC	TPR_0	FPR_0
BoW	0.9934	6.9	21.5
BoP	0.9937	13.6	21.7
BoMP	0.9975	27.4	4.3

(a) $S11$ gallery

	AUC	TPR_0	FPR_0
BoW	0.9993	6.5	0.40
BoP	0.9993	0.8	0.42
BoMP	0.9996	68.7	0.73

(b) $S21$ gallery



(a) A typical genuine match (b) A challenging genuine match

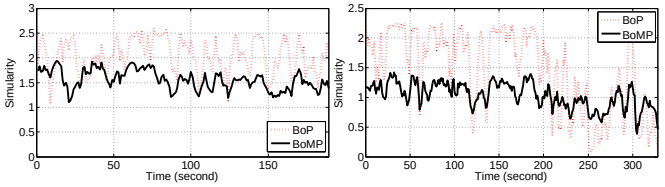
Fig. 8. Offset spaces of genuine matches with BoMP (top four rows) and BoP (the bottom row). The horizontal and vertical axis are the offset l and the number of votes n_l respectively. Note the different *scales* of the vertical axes.

One may question whether temporal information contributes to the performance. With the same setup as Table III (b), we employ a simple nearest neighbor classifier using only a single frame's 76-dimensional feature vector $\mathbf{f}_i$ to make an authentication decision. By using these static features only, we find 92%, far less than the $> 99\%$ accuracy obtained by incorporating the dynamic information with BoW, BoP, or BoMP. Hence, the dynamic information from hand movements allows for significant performance improvement over the simple static feature of hand shapes and locations.

Algorithm analysis: Here we explore the benefits of the novel BoMP over BoP. We look at the behavior for a single time step, across an entire typing session, and the overall distribution to see why BoMP corrects cases that BoP fails.

Offset space: We show the COP offset spaces of BoMP and BoP in Fig. 8. Note that the four separate offset spaces of BoMP are only for visualization purposes, and in practice only one offset space, the summation of corresponding votes in these four, is used. For a typical genuine match in Fig. 8 (a), although the left hand shape and centroid have few votes, *i.e.*, less COPs between gallery and probe, the right hand features have many more votes. Also, the actual offsets of the higher-order phrases in the right hand shape and centroid are quite different. Both differences, the difference in matched feature types and offset, can jointly explain why the votes in BoP is low for this subject since the same word pair becomes much less when using a concatenated feature. For a challenging genuine match in Fig. 8 (b), BoP does not find any COPs. However, when matching the shape context of the left hand, BoMP finds some COPs, which allows it to make a correct decision. In practical applications, we see more cases where only partial features may match, and hence BoMP is a more favorable choice for the similarity calculation.

Performance over time: Figure 9 shows the similarity scores over an entire typing session. In Fig. 9 (a), BoMP tends to have



(a) An easy-to-authenticate user (b) A hard-to-authenticate user

Fig. 9. Genuine match similarity scores (Eqn. 4) of BoP and BoMP from all probes of two subjects.

TABLE IV
TPR_0 (%) FOR DIFFERENT PROBE LENGTHS (SECONDS) AND GALLERY LENGTHS (FRACTION).

l_g \ l_p	2	5	10	20
$\frac{1}{8}$	39.8	61.6	71.4	78.8
$\frac{1}{4}$	40.0	58.4	72.3	78.8
$\frac{1}{2}$	45.4	63.7	78.2	86.3
1	50.2	66.5	80.9	87.4

(a) S11 gallery

l_g \ l_p	2	5	10	20
$\frac{1}{8}$	45.3	59.7	84.9	92.0
$\frac{1}{4}$	66.6	85.6	92.3	95.5
$\frac{1}{2}$	76.4	91.9	95.1	98.3
1	82.3	94.3	98.2	**99.5**

(b) S21 gallery

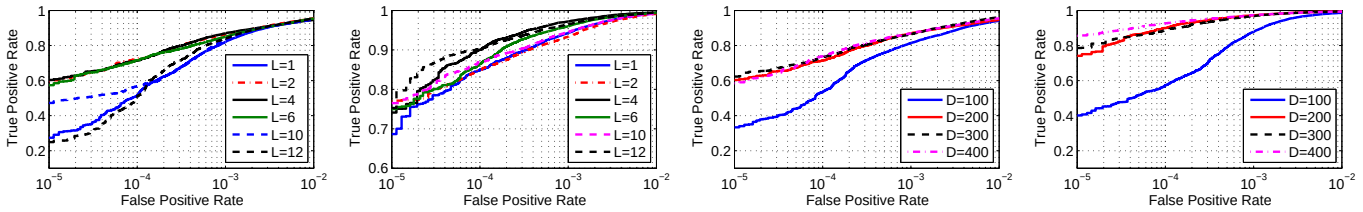
lower scores in areas of high similarity compared to BoP, but both algorithms correctly classify this subject as a genuine. In Fig. 9 (b), BoMP always identifies the correct match, but BoP fails on the hard-to-classify parts of the sequence.

Similarity distribution: Figures 15 (a) and (b) show the similarity distributions of both genuine and impostor match scores for BoP and BoMP on the Phase 1 dataset. While BoP has a greater mean for genuine match scores, the overall distribution is flatter and has more genuine scores with extreme low values. While hard to see in the figure, the impostor distribution of BoP has a longer right tail with more false matches due to high impostor scores. BoMP has a tighter distribution, which allows it to handle larger intra-subject variations. This is a very important property since real-world applications might have even more variations than our dataset due to changes in camera angles, keyboard types, the nature of the typing task, the emotion of the user, *etc.*

BoMP parameter optimization: As BoMP demonstrates the best performance, we now seek to identify good values for its two internal parameters, the segment length L and the codebook size D. We keep the same gallery and probes that we have used in all experiments so far, $l_g = \frac{1}{4}S11$ or $\frac{1}{4}S21$, $l_p = 5$, and $e = 1$. Figures 10 (a,b) show the results of using L consecutive frames, while fixing $D = 200$. We see a clear advantage of L being 2, 4, or 6 on $S11$. For both galleries, $L = 4$ performs at or near the best in terms of AUC and TPR_0. Hence, we chose to further work with $L = 4$.

Figures 10 (c,d) show the results of using different numbers of codewords D, with the same setup as above and $L = 4$. As D increases, we see improvement to TPR_0 with saturation occurring by 400. The K-means algorithm also sometimes produces empty clusters at higher values of D, which indicates that we have achieved good representation of the feature space using 400 codewords.

Gallery, and probe length: The performance is also dependent on gallery and probe length. Using the learned parameters ($L = 4$ and $D = 400$), the accuracy of BoMP



(a) $S11$ gallery with varying L (b) $S21$ gallery with varying L (c) $S11$ gallery with varying D (d) $S21$ gallery with varying D

Fig. 10. ROC curves for parameter tuning of L and D with the $S11$ or $S21$ gallery.

TABLE V
EER OF GP WITH DIFFERENT COMBINATIONS OF R-MEASURE AND A-MEASURE USING THE FULL 60-SECOND KD PROBES.

Similarity	R_2	R_2A_2	R_2R_3	$R_2A_2R_3$	$R_2A_2R_3A_3$
EER	19.53	17.08	32.73	21.82	27.22

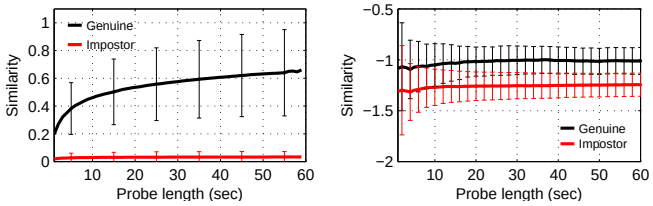
improves as both longer gallery and probe sequences are used, as shown in Table IV. From an efficiency standpoint, all combination run in real time at 31-33 FPS. Little processing time is spent finding COPs, which allows us to further increase the robustness through multiple galleries or a longer gallery session. Increasing the probe length, on the other hand, presents a practical problem, as there becomes a longer delay in detecting impostors at the keyboard. Hence, it is desirable to achieve good accuracy with a shorter probe. Fortunately with probes of only 20 seconds, BoMP meets the European standard for access control with 99.5% TPR at 0.001% FPR when using the full $S21$ gallery.

Application scenario: Note that even though the time delay between the gallery and probe of Phase 1 data is very short, the excellent performance presented above can still find applications in continuous authentication. For example, we can extend the conventional password-based one-shot authentication to continuous authentication as follows. Starting from the onset of password typing, our system will record around 30 seconds of keyboard typing and use them as the gallery to learn a biometrics template instantaneously, which will be used to continuously perform TB-based authentication for the rest of the user session. When the user lefts the computer for an extensive period of time, the user session ends and a new session will start once a new user logs in via a password. The application scenario has a few favorable properties: 1) the short time delay (*e.g.*, a few hours) between the gallery and probe will result in very high authentication performances; 2) the biometrics template can be valid only during the current user session and deleted immediately once the session ends, which remedies the risk of compromised biometrics template.

B. Robustness with Phase 2

The Phase 2 data allows for additional experiments: testing the robustness of TB to intra-subject variations such as time and lighting, direct comparison with KD, the most relevant biometric to TB, and studying the effect of the probe interval.

Keystroke dynamics: We setup an identical experiment to compare TB with KD using Scheme 2. A single session is selected as the gallery for each subject and all remaining four sessions are used as probes. The probe length l_p is varied from 1 to 60 seconds, where each second uses 110 genuine matches and $3,190$ impostor matches to compute the EER. This protocol allows us to directly compare TB with state-of-the-art KD techniques. For TB we use the optimal parameters of $L = 4$ and $D = 400$.



(a) Typing behavior (b) Keystroke dynamics

Fig. 11. Average genuine and impostor match scores with Scheme 2. TB has a much large score margin than KD.

According to a recent survey of KD [6], the GP method [13] performs near the best for free text based authentication. It also can operate with short probe sequences and a single gallery, unlike some of the more advanced methods. It weights an R-measure and A-measure between n-graphs to create a similarity metric between a probe and a set of gallery typing sessions. R_2 measures the relative latency of all common digraphs between two sequences to ensure the order is similar, while A_2 measures the absolute latency of corresponding digraphs to ensure consistency of time. R_3 and A_3 are similarly defined for the trigraphs. We seek to find the optimal similarity metric for the GP method on our dataset by running the experiment on only the full 60-second probes. Table V demonstrates the EER of the GP method for various combinations of R_2, A_2, R_3, and A_3. We find that with our shorter probe length, the R_2 and A_2 combination performs the best as there are insufficient common trigraphs between sequences to positively effect the performance. We use the R_2 and A_2 combination for all the KD experiments described below.

Figure 11 shows the mean and standard deviation of genuine and impostor match scores for both TB and KD as the probe length increases for the experiment run with $M1$ as gallery and $M2$–$M5$ as probes. For TB, the impostor scores stay consistently low for any probe length, whereas the genuine scores increase over time. For KD, both impostor and genuine scores maintain a consistent average, but the variance, and hence the error, decreases. Figure 12 shows the average EER of TB and KD for the same experiment repeated with each possible gallery session. TB outperforms KD at all times, and

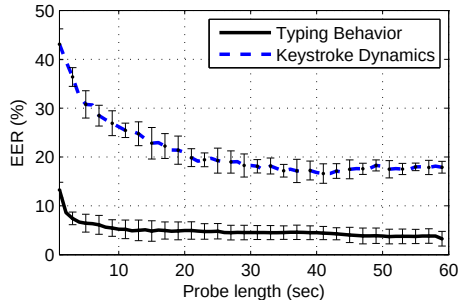


Fig. 12. EER comparison of TB and KD via Scheme 2. Error bars are reported from five runs, each with one of Mi as gallery and the rest four as probe.

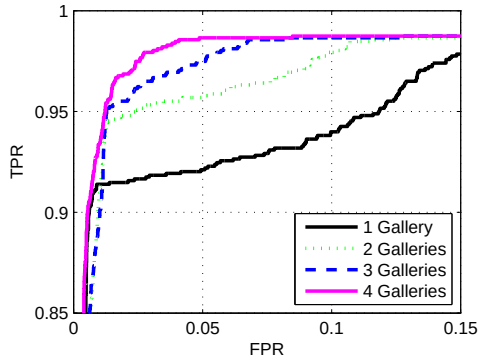


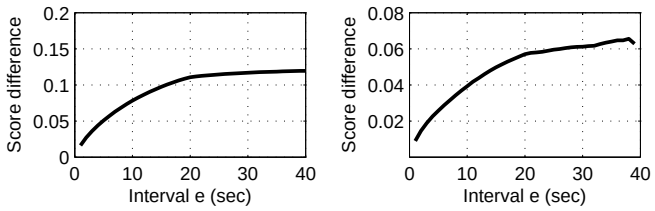
Fig. 13. ROCs for multiple gallery sessions on Phase 2.

even with a longer probe sequence, we would expect this trend to continue since TB uses the hand shape information along with the dynamic information.

We note that the performance of our KD experiment is lower than that of the GP method reported as 11.22% FPR and 98.66% TPR [38]. This performance difference is attributed to the fact that, we use 60-second probes and only one gallery session while $\sim$3-minute probes and 14 gallery sessions per subject are used in [38]. The identical experimental setup between our implementation of the GP method and our TB system still allows for a fair comparison.

Multiple gallery sessions: To see if multiple galleries can improve performance by capturing more of the variations for each user, we run an experiment with $l_p = 20$ and $e = 1$ under Scheme 1 with $M5$ as the probe. We begin with only using $M1$ as the gallery, and then add $M2$, $M3$, and $M4$. Figure 13 shows the ROC improving with additional gallery sessions. Specifically, the EER decreases from 7.2% to 2.5% when the number of gallery sequences increases from 1 to 4. Figures 15 (c) and (d) show the improvement in the genuine score distribution from 1 to 4 galleries.

Probe interval: So far, we have only used $e = 1$ interval between probes within a session in order to simulate the potentially high *frequency* required for continuous authentication and to increase the number of available probe samples. One potential criticism of this interval choice is the correlation between highly overlapped consecutive probes. While there is a certain level of natural correlation between any probes from the same user, we want to know if the correlation from



(a) (b)

Fig. 14. Mean absolute score difference of consecutive $l_p = 20$ probes with varying interval, e, for optimal parameters $L = 4$ and $D = 400$ with the full $S21$ gallery on Phase 1 (a) and Phase 2 (b).

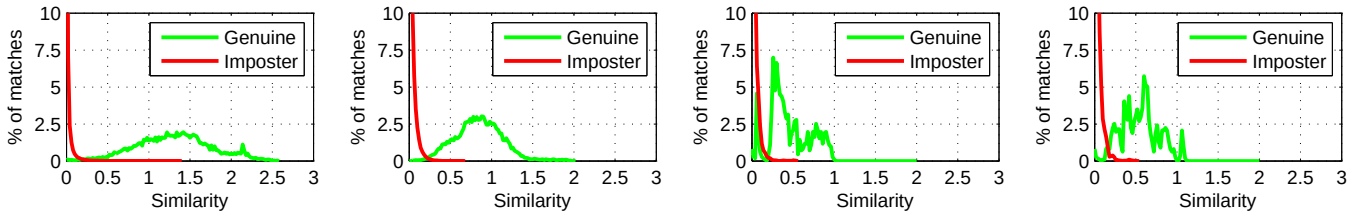
overlap inflates our performance. We look at the correlation by analyzing the score difference between consecutive probes with different intervals e.

Figure 14 presents the average absolute difference between the genuine match scores (Eqn. 4) of consecutive probes for different probe intervals. We see a few things from this figure. First, the score difference holds stable after 20 seconds, when there is no overlap. This indicates the stability of TB in general as the average genuine match score for Phase 1 is 1.04 and the average impostor match score is 0.22, which makes the average score difference only 15% of the match score margin. Second, between $e = 1$ and 20 seconds, we see a steady increase in the difference, which does indicate some additional correlation between overlapping probes.

How does this correlation affect our experiments? If we examine the ROC curve for the Phase 1 experiment using the $S21$ gallery without any overlap on the probes, *i.e.*, $e = 20$, we obtain better results with perfect classification of genuine matches and only 1 out of $30,086$ impostor matches falsely classified. Here, the accuracy is so high that we need sufficient samples to make accurate claims. For the Phase 2 experiment with one gallery session at $e = 20$, we obtain an EER of 7.5% nearly equivalent to the one with overlap, whose EER is 7.2%. Here, we see a negligible difference from removing the overlap, so we use as many possible probes with both Phases.

Similarity distribution: We have already compared the similarity distributions of genuine and impostor match scores for BoP and BoMP with 1 gallery; now we compare Phase 1 with Phase 2 in Figs. 15 (b) and (c). BoMP of Phase 2 has a similar impostor match score distribution, but the genuine score has a secondary peak with a low score, which is hypothesized to be caused by two factors. First, lighting conditions in the lab vary widely based on the time of day for data collection. Our foreground segmentation considers lighting, but still has error detecting the fingertips in low light situations. Second, subjects wear different length sleeves on different days. We currently cutoff the arms when sleeves are absent, but do not reconstruct missing information when long sleeves occlude part of the hands. Hand modeling and fitting can help solve both issues by reconstructing the missing hand information due to improper segmentation or occlusion. We also demonstrate that multiple gallery sessions helps improve the genuine score distribution as demonstrated in Fig. 15 (d).

Efficiency: Execution in real-time and beyond is important for any continuous authentication system. Experiments are run in Matlab on a standard Windows 8 desktop computer with



(a) BoP in Phase 1 (b) BoMP in Phase 1 (c) BoMP in Phase 2 w. 1 gal. (d) BoMP in Phase 2 w. 4 gal.

Fig. 15. Similarity score (K_w for BoW, Eqn. 4 for BoP and BoMP) distributions of genuine matches and impostor matches.

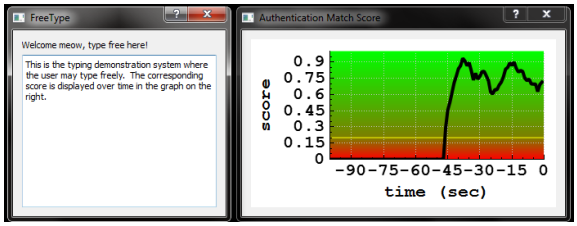


Fig. 16. GUI of our real-time TB-based continuous authentication system.

an AMD A10 APU at 3.4 GHz with 8 GB of RAM. As stated previously, Phase 1 experiments run 31–33 FPS depending on l_p and l_g, with $L = 4$ and $D = 400$. With a Matlab implementation, Phase 2 experiments run at 32.7 and 32.6 FPS with 1 and 4 gallery sequences respectively. In theory, given a probe of length n frames and a gallery of length m frames, the time is $O(n)$ for computing the feature vectors, $O(Dn)$ for assigning to codewords and probe normalization, and $O(n)$ for incremental similarity computation. In practice, it spends 79%, 20%, and $< 1\%$ of time on these three parts respectively. The extremely efficient similarity computation of BoMP enables the potential advantage of a large number of gallery sessions, yet still achieving real-time efficiency.

C. Continuous Authentication Demonstration System

During the summer of 2013, we implemented a demonstration system of the proposed TB-based continuous authentication in Microsoft Visual Studio. Our demo allows an arbitrary user to enroll the system by typing free text for 30 seconds, and a biometrics template is created immediately. During the testing, once a user claims his identity, continuous authentication is performed while the user is typing on the keyboard. As shown in Fig. 16, as the user types in the left window, our system continuously displays the computed similarity score, and genuine matches are claimed when the scores are above the yellow threshold line. A user may type completely different texts during the enrollment and testing. When the hands are off the keyboard, the score will be zero immediately. When a user just starts typing, the score will typically goes above the threshold in less than 5 seconds for genuine matches.

Our demo runs at 30 FPS on a conventional Apple Macbook laptop, with constantly less than 10% CPU load, which translates to 300+ FPS. Our demo was well received and won the *Best Demo Award* at the IEEE International Conference on Biometrics: Theory, Applications and System 2013 [1]. From September 2013 till now, we have been showing and testing our demo to at least 200 subjects, including conference attendees, high school students in outreach events, campus visitors, etc. Our observation is that the system is very robust, and rarely makes any (false positive or false negative) errors. We use this demo system as a way to not only disseminate our work, but also continue our database collection.

VII. Conclusions and Future Directions

This paper presented a novel approach of continuously validating the identity of a user in real time through the use of typing-based behavior biometrics. We investigated a number of methods to compute the similarity of two typing sequences. In particular, we proposed a novel BoMP approach to efficiently compute the high-order co-occurring phrases that are composed of words across both the temporal and feature spaces. We collected a multi-phase, multi-session, and multi-model (visual, audio, and keystroke timing) database of keyboard typing by 63 unique subjects. Through extensive experiments, we demonstrated excellent performance at operational points most relevant to authentication applications, as well as explained where and why BoMP improves upon the prior work. We also demonstrated superior performance over keystroke dynamics, with a much shorter probe sequence, which offers far less authentication delay. Finally the success of our ultra-real-time demo system indicates again the promise of this novel biometric modality.

As a novel exploration of TB, our approach focuses on the behavioral traits that can be observed through how an individual operates the keyboard. Similar to the fact that you leave a fingerprint when touching something with your finger, in behavior biometrics when you operate something you leave a pattern based on how your mind processes information, which is called a "cognitive fingerprint" by DARPA's Active Authentication program [2]. Just like conventional fingerprints, the key challenge with cognitive fingerprints is whether the pattern is consistent within an individual and discriminative between individuals. Indeed typing behavior has demonstrated the potential to meet such challenge, with carefully designed data collection, extensive experiments, and a successful real-time demo system.

We observe that TB can achieve excellent performances (TPR=99.5% when FPR=0.001% for text independent test with 20-second probes) Such performance is even more notable considering the fact that we have not utilized *supervised learning* approaches, or any *appearance* features, both of which are known to boost the performance of biometrics

authentication. Hence, this demonstrates that our novel continuous biometric modality is promising and can be further improved. Note that the already excellent performance in our experiments and demo system suggests us to leave the aforementioned two directions as future work, rather than implementing them in the current system. Although the number of subjects (51 and 30) is preferred to be larger, it is on par with the number of subjects (51) in the well-known CMU benchmark keystroke database [16]. In addition to enjoying the benefits of being a non-intrusive continuous authentication modality with short verification time, typing behavior also has additional favorable properties. For example, being a behavioral biometric, it is inherently robust to spoofing because the dynamics of typing is especially hard to imitate.

There are many interesting directions to further the development of typing behavior. First, we will continue to enroll subjects into unconstrained typing. Second, we will perform keyboard-based calibration so as to compensate the potential varying camera angles and keyboard position among the typing sequences. Third, we will study how the type of keyboard affects the authentication performance. Fourth, given sufficient amount of data, we can also employ machine learning methods to learn various parameters or weights in the similarity calculation, such as the different weights for $K_k(\mathbf{V}, \mathbf{V}')$. Fifth, we will incorporate appearance features into the sequence matching, which can be especially useful when users make pauses during typing. Sixth, we will explore the use of hand movement while operating the mouse. Seventh, applying object tracking or image alignment [18], [19] to hands can parse the hands' structure and enable advanced features for fingers and dorsum. Finally, we view TB as an exploration of the visual aspect of keyboard typing, rather than a replacement of KD. Hence, we can fuse TB with KD to achieve greater authentication performances and robustness in a diverse set of application scenarios.

ACKNOWLEDGMENT

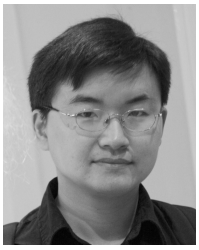
The authors thank the volunteers who participated in the collection of the keyboard typing database at Michigan State University. The authors also thank the associated editors and reviewers for their efforts and constructive comments.

REFERENCES

[1] http://www.btas2013.org/awards/.
[2] http://www.darpa.mil/Our_Work/I2O/Programs/Active_Authentication.aspx.
[3] A. Ahmed and I. Traore. A new biometric technology based on mouse dynamics. *IEEE Trans. Dependable and Secure Computing*, 4(3):165–179, 2007.
[4] A. Altinok and M. Turk. Temporal integration for continuous multimodal biometrics. In *Proc. Workshop Multimodal User Authentication*, pages 131–137, 2003.
[5] D. Asonov and R. Agrawal. Keyboard acoustic emanations. In *Proc. of IEEE Symp. on Security and Privacy*, pages 3–11, May 2004.
[6] S. P. Banerjee and D. Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *J. of Pattern Recognition Research*, 7(1):116–139, 2012.
[7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE T-PAMI*, 24(4):509–522, Apr. 2002.
[8] P. Dollar, V. Rabaud, G. Cottrell, and S. J. Belongie. Behavior recognition via sparse spatio-temporal features. In *PETS Workshop*, pages 65–72, Oct. 2005.
[9] N. Duta. A survey of biometric technology based on hand shape. *Pattern Recognition*, 42(11):2797–2806, Nov. 2009.
[10] European Committee for Electrotechnical Standardization (CENELEC). European standard alarm systems. access control systems for use in security applications. part 1: System requirements. Technical Report EN 50133-1, 2002.
[11] J. Farquhar, S. Szedmak, H. Meng, and J. Shawe-Taylor. Improving "bag-of-keypoints" image categorisation: Generative models and PDF-kernels. Technical report, University of Southampton, 2005.
[12] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531, 2005.
[13] D. Gunetti and C. Picardi. Keystroke analysis of free text. *TISSEC*, 8(3):312–347, Aug. 2005.
[14] M. S. Hossain, K. S. Balagani, and V. V. Phoha. New impostor score based rejection methods for continuous keystroke verification with weak templates. In *BTAS*, pages 251–258, 2012.
[15] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Trans. CSVT*, 14(1):4–20, Jan. 2004.
[16] K. S. Killourhy and R. A. Maxion. Comparing anomaly detectors for keystroke dynamics. In *Proc. of the Int. Conf. on Dependable Systems and Networks (DSN)*, pages 125–134, Lisbon, Portugal, July 2009.
[17] X. Liu, T. Chen, and B. V. K. V. Kumar. On modeling variations for face authentication. In *FG*, pages 384–389, 2002.
[18] X. Liu, T. Yu, T. Sebastian, and P. Tu. Boosted deformable model for human body alignment. In *CVPR*, pages 1–8. IEEE, 2008.
[19] S. Lu, D. Metaxas, D. Samaras, and J. Oliensis. Using multiple cues for hand tracking and model refinement. In *CVPR*, volume 2, pages II–443. IEEE, 2003.
[20] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV*, 2000.
[21] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE T-PAMI*, 23(2):228–233, Feb. 2001.
[22] S. Mitra. Gesture recognition: A survey. *IEEE T-SMC C, Appl. and Rev.*, 37(3):311–324, 2007.
[23] T. Mustafic, S. Camtepe, and S. Albayrak. Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. In *ICJB*, 2011.
[24] K. Niinuma, U. Park, and A. K. Jain. Soft biometric traits for continuous user authentication. *IEEE Trans. Information Forensics and Security*, 5(4):771–780, Dec. 2010.
[25] J. A. Ouellette and W. Wood. Habit and intention in everyday life: The multiple processes by which past behavior predicts future behavior. *Psychological Bulletin*, 124(1):54–74, July 1998.
[26] K. Revett, H. Jahankhani, S. T. de Magalhães, and H. M. D. Santos. *Global E-Security*, volume 12 of *Communications in Computer and Information Science*, chapter A Survey of User Authentication Based on Mouse Dynamics, pages 210–219. Springer Berlin Heidelberg, 2008.
[27] J. Roth, X. Liu, A. Ross, and D. Metaxas. Biometric authentication via keystroke sound. In *ICB*, 2013.
[28] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-fei. Spatial-temporal correlatons for unsupervised action classification. In *IEEE Workshop on Motion and video Computing*, pages 1–8, Washington, DC, Jan. 2008.
[29] A. Serwadda, Z. Wang, P. Koch, S. Govindarajan, R. Pokala, A. Goodkind, D. Guy Brizan, A. Rosenberg, V. Phoha, and K. Balagani. Scan-based evaluation of continuous keystroke authentication systems. *IT Professional*, PP(99):1–1, 2013.
[30] D. Shanmugapriya and G. Padmavathi. A survey of biometric keystroke dynamics: Approaches, security and challenges. *Int. J. of Computer Science and Information Security*, 5(1):115–119, 2009.
[31] C. Shen, Z. Cai, X. Guan, Y. Du, and R. Maxion. User authentication through mouse dynamics. *IEEE Trans. Information Forensics and Security*, 8(1):16–30, 2013.
[32] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar. Continuous verification using multimodal biometrics. *IEEE T-PAMI*, 29(4):687–700, Apr. 2007.
[33] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, 2003.
[34] J. Sun, X. Wu, S. C. Yan, L. F. Cheong, T. S. Chua, and J. T. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.
[35] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.
[36] Unknown. Biometrics metrics report. Technical report, U.S. Military Academy, Dec. 2012.
[37] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of american sign language. *CVIU*, 81(3):358–384, 2001.
[38] K. Xi, Y. Tang, and J. Hu. Correlation keystroke verification scheme for user access control in cloud computing environment. *The Computer Journal*, 54(10):1632–1644, July 2011.
[39] R. Yang, S. Sarkar, and B. Loeding. Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition

using nested dynamic programming. *IEEE T-PAMI*, 32(3):462–477, Mar. 2010.
[40] E. Yärük, H. Dutağaci, and B. Sankur. Hand biometrics. *Image and Vision Computing*, 24(5):483–497, 2006.
[41] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2):213–238, 2007.
[42] L. Zhang, L. Zhang, and D. Zhang. Finger-knuckle-print: A new biometric identifier. In *ICIP*, pages 1981–1984, 2009.
[43] L. Zhang, L. Zhang, D. Zhang, and Z. Guo. Phase congruency induced local features for finger-knuckle-print recognition. *Pattern Recognition*, 45(7):2522–2531, July 2012.
[44] Y. Zhang and T. Chen. Efficient kernels for identifying unbounded-order spatial features. In *CVPR*, pages 1762–1769. IEEE, 2009.
[45] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen. Spatio-temporal phrases for activity recognition. In *ECCV*, pages 707–721. Springer, 2012.
[46] N. Zheng, A. Paloski, and H. Wang. An efficient user verification system via mouse movements. In *CCS*, pages 139–150, 2011.
[47] Y. Zhong, Y. Deng, and A. K. Jain. Keystroke dynamics for user authentication. In *CVPRW*, 2012.
[48] L. Zhuang, F. Zhou, and J. D. Tygar. Keyboard acoustic emanations revisited. In *CCS*, pages 373–382, 2005.
[49] J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Computing Surveys*, 38(6):1–56, 2006.

Joseph Roth is currently pursuing a Ph.D. degree with the Computer Vision Lab from the Department of Computer Science and Engineering at Michigan State University, East Lansing, MI. He received his B.S. in Computer Science from Grand Valley State University, Allendale, MI, in 2010. His research interests are computer vision and biometrics.



Xiaoming Liu is an Assistant Professor in the Department of Computer Science and Engineering at Michigan State University (MSU). He received the B.E. degree from Beijing Information Technology Institute, China and the M.E. degree from Zhejiang University, China, in 1997 and 2000 respectively, both in Computer Science, and the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University in 2004. Before joining MSU in Fall 2012, he was a research scientist at General Electric Global Research Center. His research areas are face recognition, biometrics, image alignment, video surveillance, computer vision and pattern recognition. He has authored more than 70 scientific publications, and has filed 22 U.S. patents. He is a member of the IEEE.

Dimitris Metaxas (M'93-SM'98) received the B.E. degree from the National Technical University of Athens Greece, Athens, Greece, in 1986; the M.S. degree from the University of Maryland, College Park, in 1988; and the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 1992.

He is a Professor with the Department of Computer Science, Rutgers University, New Brunswick, NF. He is directing the Computational Biomedicine Imaging and Modeling Center. His research interests include the development of formal methods upon which computer vision, computer graphics, and medical imaging can advance synergistically.