Screen-based active user authentication

Mohammed E. Fathy\textsuperscript{a,\textemdash}, Vishal M. Patel\textsuperscript{a}, Tom Yeh\textsuperscript{b}, Yangmuzi Zhang\textsuperscript{a}, Rama Chellappa\textsuperscript{a}, Larry Davis\textsuperscript{a}

\textsuperscript{a}Center for Automation Research, University of Maryland, College Park, MD 20742, United States
\textsuperscript{b}University of Colorado, Boulder, CO 80309, United States

ARTICLE INFO

Article history:

Keywords:
Cyber Biometrics
Interaction Classification
Active Authentication

ABSTRACT

We investigate if screen-based recordings of computer interactions can be used for accurate active user authentication. A dataset of screen recordings of some PC interactions (MouseMoving, Typing, Scrolling, Other) of 21 users was collected and we ran a set of experiments to help our investigation. Low-dimensional feature vectors based on histogram of optical flows from each screen recording were used in our study. The first set of experiments investigated if these low-dimensional features can be used to recognize the type of interaction taking place in a particular recording and we found that linear SVM could succeed in achieving this with an accuracy of 91\% on 5 test users. The second set of experiments explored if classifiers trained on different types of recordings can be used to verify user identity. The results indicated that SVMs trained on Scrolling recordings can achieve moderately low FAR and FRR error rates of 20.7\% and 12.4\%, respectively. These preliminary results indicate that further research in using screen-based recordings for active authentication can lead to a reliable soft cyber biometric.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Biometrics deals with the problem of identifying individuals based on physiological or behavioral characteristics. Since many physical characteristics, such as face, iris, etc., and behavioral characteristics, such as voice, expression, keystroke, etc., are unique to an individual, biometric analysis offers a reliable and natural solution to the problem of identity verification. It has been shown that physiological biometrics techniques have been more successful for the problem of identity verification than behavioral characteristics. This is due in part to the fact that physiological features remain stable for long periods of time. On the other hand, behavioral characteristics are greatly influenced by one’s mood, stress or illness. This makes it somewhat instable for identity verification.

The current standard method for validating a user’s identity for authentication of a computer device requires humans to do something that is inherently difficult: create, remember, and manage long, complex passwords. Furthermore, as long as the session remains active, typical systems incorporate no mechanisms to verify that the user originally authenticated is the user still in control of the computer. Thus, unauthorized individuals may improperly obtain access to the computer if a password is compromised or if a user does not exercise adequate vigilance after initially authenticating on a device.
To deal with this problem, various cyber biometrics have been proposed in the literature. These methods capture the cognitive fingerprints of users. The rationale is that how individuals formulate their thoughts and actions are reflected through their behavior, and this behavior in turn can be used to characterize how the individual performs tasks using the computer. The most notable examples are keyboard dynamics [6] and mouse dynamics [7]. Some other examples of the computational behavior metrics of the cognitive fingerprint include eye tracking, how the user selects information, how the user searches for information, etc.

Active authentication is the process of verifying the user’s identity during the whole session instead of just at the login screen. We have proposed a novel active authentication technique that focuses on the unique aspects of the individual by utilizing a screen fingerprint. A screen fingerprint is acquired by first taking a screen recording of the computer being used by the operator and then extracting discriminative visual features from these recordings.

The screen fingerprint of an operator captures enough of the unique human interactions to be usable as a biometric for authentication. The qualities captured include cognitive abilities, motor limitations, subjective preferences, and work patterns. For example, how well the operator sees is a cognitive ability that can be captured visually by the size of the text shown on the screen. How fast the operator drags a window is a motor limitation that can be captured visually by the amount of motion detected on the screen. How organized the operator arranges multiple windows is a subjective preference that can be captured visually by the layout of salient edges identified on the screen. What suite of applications the operator uses is a work pattern that can be captured visually by the distribution of application-specific visual features recognized on the screen.

The proposed technology exploits the synergy between recent advances in pixel-level screen analysis [4], [13], [14] and vision-based biometrics. Vision-based biometrics such as face and iris recognition have become more reliable. Yet, its dependence on hardware sensors often limits its applicability. On the other hand, pixel-level screen analysis has received a lot of attention in human computer interaction in the past two years. One attractive advantage of pixel-level screen analysis is its wide applicability, since the screen buffer can be accessed on all platforms at the software level. However, pixel-level screen analysis has not been used as a modality for biometrics. To the best of our knowledge, this is the first attempt to combine vision-based biometrics and pixel-level screen analysis in a complementary manner for the purpose of active user authentication.

We investigate in this paper how well different interactions can be classified visually. Once classified, we investigate how well each type of interaction can be used to continuously authenticate the individual. In order to study the effectiveness of screen fingerprints, we have put together a dataset in which screen recordings of different individuals were collected while doing some common tasks (e.g. typing, scrolling, mouse-based dragging and window resizing). We describe that dataset and present some initial results using different configurations of SVM and K-Nearest-Neighbor classifiers. Based on our limited experiments using this dataset, we have found that screen fingerprints can indeed capture enough of the unique human traits to be usable as a biometric for authentication.

1.1. Organization of the paper

This paper is organized as follows. A motivation for screen fingerprints is presented in Section 2. The dataset employed in this paper and the collection procedure are described in Section 3. The design of features we tried in our work is presented in Section 4. Section 5 details the procedure for automatically inferring the type of interaction type in a given screen recording. Section 6 investigates how well screen recordings can be used to verify the user identity. Concluding remarks are given in Section 7.

2. Screen fingerprints

Over the past few years, computer vision techniques have been successfully applied to the analysis of the graphical user interfaces shown in a screen recording to support a wide range of applications including automation [12], search [14], software testing [4], and tutorial [13]. Some applications perform batch analysis after screen recordings are acquired, such as searching online documentation about the interfaces in a screen recording [14], [13]. Some applications operate in real-time while screen recordings are made. For example, Yeh et al. has developed the Sikuli visual automation tool [12] that can observe a screen recording in real-time, identify an interface component by appearance, and send automation command (e.g., click) to that component. This tool has made a significant impact on software engineering in that it is currently used by dozens of companies to automate GUI testing. Active authentication based on screen fingerprints is a novel application of screen recording analysis that has not been attempted before.

A typical scenario of active authentication using screen fingerprints proceeds as follows. First, an operator of the computer logs on using an initial authentication mechanism such as entering a password. While the operator is using the workstation, the screen of the computer is being observed. Screen recording are taken within short observation windows. Each time a screen recording is taken, the recording (a video) is visually analyzed to extract a screen fingerprint aimed to identify the person who is using the computer. This observed screen fingerprint is compared to the reference screen fingerprint previously measured and stored for the authorized operator. If a match is established between the observed and reference fingerprints, the operator is actively authenticated. Now suppose the operator steps away and leaves the workstation unattended. An adversary may gain physical access to the computer. While the adversary is using the computer, a screen recording is taken to extract a screen fingerprint. However, the observed screen fingerprint no longer matches the reference screen fingerprint. As a result, active authentication fails. The workstation may lock itself up to prevent further unauthorized use by the adversary.

Screen fingerprints offer several advantages over other potential modalities for active authentication as described below:
2.1. Advantages over language-based techniques

Language-based techniques such as those based on computational linguistic and structural semantic analysis seek to authenticate computer operators based on verbal cues such as the words and phrases an operator uses in digital communication (e.g., emails, memos). These stylometry techniques [2], [3] do not work well in situations when operators’ primary responsibilities do not involve personal communicating (e.g., data entry) or when operators mainly use mouse or touch-based interfaces (e.g., Photoshop). Our proposed modality can deal with these situations because it relies on visual cues that are always observable on a computer screen regardless of the types of applications operators use.

2.2. Advantages over motor-based techniques

Motor-based techniques seek to authenticate computer operators based on kinetic cues such as how fast an operator types [11] or moves a mouse pointer [10]. These techniques cannot support operators who use voice or touch as the primary input modality. Our proposed modality can authenticate operators who do not use a mouse or keyboard because it does not depend on specific input devices.

2.3. Advantages over application-based techniques

Application-based techniques seek to authenticate computer operators based on usage cues such as which applications or features an operator is using. However, these techniques are difficult to scale because each application must be specifically instrumented in order to track its usage. Often such elaborate instrumentation requires access to the application’s source code or special application programming interface (API). Comprehensive coverage is hard to attain because some proprietary or legacy applications do not provide source code or API for instrumentation. Our proposed modality is able to provide wide coverage over most applications without additional instrumentation. As long as an application is visible on a computer screen, it can be captured in a screen recording. Some of the distinctive visual properties of an application can be extracted to be part of an operator’s screen fingerprint.

3. Dataset

3.1. Dataset preparation

We collected a dataset of screen recordings of PC interactions of 21 users. Each person was asked to perform 4 types of tasks: dragging icons, typing, scrolling and resizing. The person was directed to repeat the different tasks 5 times but in a permuted order. We developed a Java program to guide the user through the data collection process. The program recorded the screen at a rate of 12 frames per second. We found that the rate did not affect the system responsiveness and was good enough to capture the visual dynamics of the different types of interactions. The four tasks collected are described as follows:

**Drag-Drop:** The user is asked to drag a set of files into a certain directory, one file at a time. This is an instance of the MouseMoving interaction type.

**Typing:** The user is asked to type specified paragraphs into a typing window. Clearly, this is an instance of the Typing interaction type.

**Scrolling:** The user is asked to scroll through a document and count the number of times a certain letter appears in the section titles. The type of interaction here is Scrolling.

**Resizing:** The user is asked to resize an image window so that it takes (roughly) the left half of the screen. This is also an instance of the MouseMoving interaction type.

Figure 1 shows a sample screenshot from each task. Before starting the actual task, there may be some frames where the user is switching context to start doing the task. Similarly, there may be some frames after finishing the actual task where the user is terminating the current task. Accordingly, we manually specified for each recorded video the start and end times within which the core interaction takes place. This divides each recording into one segment from the core interaction type (Typing, MouseMoving or Scrolling) and up to two segments of Other interaction. We end up with 1243 instances of the 4 different types of interactions (Other, Typing, MouseMoving and Scrolling). The average recording length is 2 s for Other, 43 s for typing, 12 s for MouseMoving, and 83 s for Scrolling. The typical data collection session for a particular user lasted between 20-25 min.

4. Features: Average histogram of oriented optical flows

After taking a screen recording, the next step in the active authentication pipeline is to extract a feature vector that, ideally, can distinguish different kinds of interactions (e.g. Scrolling versus MouseMoving) and discriminate different users (legitimate versus illegitimate).

One of the most popular techniques of measuring the change in visual appearance between two consecutive frames, $f_t$ and $f_{t+1}$ is the optical flow $w_t$. It can be thought of as a velocity field over the image that describes the visual motion at every pixel [8]. We believe that features based on optical flow are good at discriminating among different interaction kinds. Indeed, Scrolling is characterized by vertical (and/or horizontal) visual motion; whereas, MouseMoving is characterized by continuous motion in all directions rather than just vertical or horizontal. Optical flow can also discriminate different users as it is sensitive to their different interaction rates.

We use the following procedure to extract a feature vector $x$ from a screen recording $r = \{f_0, f_1, \ldots, f_{T-1}\}$. First, we downscale each frame $f_t$ to $160 \times 100$ resolution and use the downscaled frames to calculate the optical flows $\{w_0, w_1, \ldots, w_{T-2}\}$ using the implementation in [9]. We then calculate a Histogram of the oriented Optical Flow (HOF), denoted by $h$, from each $w_t$ using the implementation in [5]. The HOF vector $h$ consists of $B$ orientation bins. Each bin receives the sum of the magnitudes of all the optical flow entries in $w_t$ with orientations within the angular range of that bin. Thus, the HOF vector is not affected if the location of interaction changes on screen. We restrict the number of orientation bins $B$ of $h$ to 100 but we neither normalize the histograms nor subdivide the images into grids as
Finally, we compute \( x \) as the arithmetic mean of all the HOF vectors:

\[
x = \frac{1}{T - 1} \sum_{t=0}^{T-2} h_t
\]

The division by the number of HOF vectors \((T - 1)\) makes the Average HOF (AHOF) feature vector \( x \) invariant to the length of the recording. For all the experiments reported in this paper, we use the 100-dimensional AHOF vectors extracted from the screen recordings as features. The distribution of AHOF vectors extracted from the screen recordings of user \( u \) while doing the interaction type \( c \) defines the screen fingerprint of the user \( u \) for that particular interaction type \( c \). We try in Section 6 different standard models (e.g. KNN and SVM) to learn the AHOF distribution of each user by using the individual AHOF vectors as training samples.

5. Interaction classification

To decide how subsequent identity verification will proceed, it is helpful to know the type of interaction occurring in each recording. Supervised machine learning techniques can be used to train a classifier that can automatically label the interactions in the recordings. This is because each recording in the dataset is labeled not only by its user ID but also by the type of interaction. The two machine learning techniques we tried are Support Vector Machine (SVM) [1] and AdaBoost [15]. For SVM, we used the soft-margin, linear kernel version and we handled multiclases using the One-Versus-All (OVA) strategy where we trained a separate SVM to classify each interaction type against all the other interaction types. For AdaBoost, we used the single-node decision tree as the weak classifier and handled multiclases by implementing the simple variation of AdaBoost called SAMME described in [15].

5.1. Experimental evaluation

We used the interactions of 14 randomly selected users to train both SVM and AdaBoost. The parameter tuning was performed on the interactions of 2 other users and the testing was done on the interactions of the remaining 5 users. The classification accuracy was as high as 91% for SVM versus 80% for AdaBoost. The confusion matrices of both techniques are also shown in Table 1 and 2, respectively. In both matrices, each row indicates that most of the matching errors result from confusing the row’s interaction type as Other (e.g. 6% of the MouseMoving instances in Table 1 are misclassified only as Other). This is explained by the fact that there are many more instances in the Other class than all other interaction classes. This type of error (i.e. non-Other instances being misclassified as Other) is more acceptable in practice than all the remaining types of errors (i.e. instances misclassified as non-Other) because such types of errors will result in invoking a verification classifier on a recording of an incompatible type (e.g. running the Scrolling-based verification classifier on an Other recording).
Table 1: Confusion matrix for interaction classification via SVM.

<table>
<thead>
<tr>
<th>True/Predicted</th>
<th>Other</th>
<th>Typing</th>
<th>MouseMoving</th>
<th>Scrolling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
<td>0.93</td>
<td>0.04</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>Typing</td>
<td>0.08</td>
<td>0.92</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>MouseMoving</td>
<td>0.06</td>
<td>0.00</td>
<td>0.94</td>
<td>0.00</td>
</tr>
<tr>
<td>Scrolling</td>
<td>0.36</td>
<td>0.00</td>
<td>0.00</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 2: Confusion matrix for interaction classification Ad-aBoost.

<table>
<thead>
<tr>
<th>True/Predicted</th>
<th>Other</th>
<th>Typing</th>
<th>MouseMoving</th>
<th>Scrolling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
<td>0.94</td>
<td>0.03</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>Typing</td>
<td>0.44</td>
<td>0.56</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>MouseMoving</td>
<td>0.48</td>
<td>0.00</td>
<td>0.48</td>
<td>0.04</td>
</tr>
<tr>
<td>Scrolling</td>
<td>0.44</td>
<td>0.00</td>
<td>0.04</td>
<td>0.52</td>
</tr>
</tbody>
</table>

6. Identity verification

We experimentally investigated whether all interaction types are equally powerful in verifying user identity. Before describing the details of the experiment, we present a few definitions. Denote by \( S(u,c) \) the set of all recordings of interaction \( c \in C = \{ \text{Other}, \text{Typing}, \text{MouseMoving}, \text{Scrolling} \} \) performed by user \( u \in U \). Let \( S(c) \) be the set of all instances of interaction \( c \) in the dataset. That is, \( S(c) = \bigcup_{u \in U} S(u,c) \).

6.1. Biometric performance measures

We measure the detection error metrics commonly used in the evaluation of biometrics. These are defined below:

- **FAR**: False Acceptance Rate is the fraction of illegitimate samples (i.e. negatives) that are incorrectly accepted (i.e. classified as positives). The complementary fraction of FAR is the True Rejection Rate (TRR = 1 − FAR). Maximizing TRR is equivalent to minimizing FAR.

- **FRR**: False Rejection Rate is the fraction of legitimate samples (i.e. positives) that are incorrectly rejected (i.e. classified as negatives). The complementary fraction of FRR is the True Acceptance Rate (TAR = 1 − FRR). Maximizing TAR is equivalent to minimizing FRR.

When a biometric is evaluated, it typically assigns a real score to each instance in the test set. These scores are then mapped into decisions (accept/reject) based on a real threshold \( \theta \). Different thresholds can lead to different detection error rates (FAR, FRR) (or equivalently accuracy rates (TRR, TAR)). In this paper, we select the threshold that achieves the best tradeoff between FAR and FRR. To do this, we define a new performance metric that we call F1-DET (short for F1 of Detection Error Tradeoff). F1-DET is simply the harmonic mean of TRR and TAR. The idea of F1-DET is analogous in concept to the traditional F1 score and how it is used to conservatively evaluate different (precision, recall) pairs. In other words, F1-DET is high only when both TRR and TAR are high (or equivalently FAR and FRR are both low). If either TRR or TAR or both are low, the corresponding F1-DET will also be low.

6.2. Experimental evaluation

We tried different configurations of K-Nearest Neighbours (KNN) and soft-margin SVM as biometric classifiers. For each user \( u \) and interaction \( c \), we train a biometric to verify the legitimate instances in \( S(u,c) \) against the illegitimate instances in \( S(c) − S(u,c) \). We do this by running a K-fold cross validation process with \( K = |S(u,c)| \). The data set \( S(c) \) is divided into \( K \) parts where each part contains one legitimate (positive) sample from \( S(u,c) \) and \( 1/K \) of the illegitimate (negative) samples in \( S(c) − S(u,c) \). In the ith fold, we train on all parts except the ith which is used for testing. After completing all folds, we evaluate FAR, FRR and F1-DET at all possible threshold values and set FID(\( u, c \)) to the highest F1-DET score. In addition, we set FAR(\( u, c \)) and FRR(\( u, c \)) to the pair corresponding to FID(\( u, c \)).

Table 3 shows for each classifier and interaction class \( c \) the scores F1D(\( c \)), FAR(\( c \)), and FRR(\( c \)) where F1D(\( c \)) is the average of FID(\( u, c \)) taken over all users \( u \). FAR(\( c \)) and FRR(\( c \)) are defined in a similar fashion. It is easy to see that Scrolling leads to the best detection accuracy compared to other classes of interaction. This is true for all classifiers although Scrolling performance is best with SVM. Typing has the lowest detection accuracy (i.e. F1D) compared to other types of interaction. Figure 2 shows for each interaction \( c \) the box-plot of the F1D(\( u, c \)) scores achieved by the classifier that was found to give the highest average F1D score for interaction \( c \) in Table 3. The figure leads to observations similar to those derived from Table 3. In addition, it shows that the lowest F1D scores achieved by the linear SVM for Scrolling tend to be better than the lowest F1D.
scores achieved by the RBF-Kernel SVM although Table 3 indicated that both have the same average F1D score for Scrolling.

7. Concluding remarks

While the experiments in Section 5 show that a classifier can visually determine the interaction type in a screen recording at a very good accuracy (91% with SVM), the experiments in Section 6 indicate that not all interaction types are equally good at verifying the user’s legitimacy. Only screen recordings of Scrolling lead to moderately low detection error rates (FAR = 20.67% and FRR = 12.38%) while other types achieve relatively high error rates. Based on our results, a screen-based biometric should be activated only during Scrolling interactions as other interactions are less reliable. Although the verification performance obtained may not be as high as some of the other longer-established modalities such as mouse dynamics, screen output can enhance the security of a multi-modal system in case there is little of the data that other modalities monitor as it is the case during scrolling when both typing and mouse movement are usually absent. It is also worth noting that the performance of the other modalities have been the target of research for much longer time (33 years for keystroke dynamics [6] and 9 years for mouse dynamics [7]) and the performance of screen fingerprints can be further improved beyond the reported results by investigating richer features and other classifiers.

Acknowledgment

This work was supported by cooperative agreement FA87501220199 from DARPA.

References