Continuous Authentication of Mobile User: Fusion of Face Image and Inertial Measurement Unit Data

David Crouse
Michigan State University
crouseda@msu.edu

Hu Han
Michigan State University
hhan@msu.edu

Deepak Chandra
Google Inc.
dchandra@google.com

Brandon Barbello
Google Inc.
bbarbello@google.com

Anil Jain
Michigan State University
jain@cse.msu.edu

Abstract

Mobile devices can carry large amounts of personal data, but are often left unsecured. PIN locks are inconvenient to use and thus have seen low adoption (33% of users). While biometrics are beginning to be used for mobile device authentication, they are used only for initial unlock. Mobile devices secured with only login authentication are still vulnerable to data theft when in an unlocked state. This paper introduces our work on a face-based continuous authentication system that operates in an unobtrusive manner. We present a methodology for fusing mobile device (unconstrained) face capture with gyroscope, accelerometer, and magnetometer data to correct for camera orientation and, by extension, the orientation of the face image. Experiments demonstrate (i) improvement of face recognition accuracy from face orientation correction, and (ii) efficacy of the prototype continuous authentication system.

1. Introduction

Since the first mobile telephone call in 1973, mobile phones have evolved from simple call-making implements to full-featured pocket computers, or *smartphones*[1] (see Fig. 1). Smart mobile devices encompass both smartphones and tablets running operating systems designed for smartphones. Mobile computing has expanded to the point where the number of mobile devices in use is nearly equal to the world's population. In 2013, 77% of such devices sold were smartphones.[2]

Figure 1. The evolution of the mobile phone.[5]

Mobile devices are rapidly becoming data hubs, used to store e-mail, personal photos, online history, passwords, and even payment information.[3] Despite all this sensitive data, 67% of users do not password protect their devices[4], leaving their personal information accessible to malicious individuals.

The 2013 Meeker Report[6] indicated that the average smartphone user checks their device 150 times per day. If unlocking the device takes 2 seconds, the typical user spends 5 minutes unlocking their device every day. This inconvenience leads individuals to make less security-conscious decisions like setting long timeouts before their phone locks or even leaving it unprotected. Current efforts to improve security convenience, such as Bluetooth login tokens, have critical vulnerabilities which make them less secure than simple PINs.

The current authentication model - *provide credentials*

[1] A smartphone is defined as a mobile telephone possessing a variety of sensors and that has an operating system capable of running applications more advanced than simple Java ME or BREW programs.

[2] www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

[3] www.cnet.com/how-to/how-nfc-works-and-mobile-payments/

[4] www.sophos.com/en-us/press-office/press-releases/2011/08/67-percent-of-consumers-do-not-have-password-protection-on-their-mobile-phones.aspx

[5] www.businessinsider.com/complete-visual-history-of-cell-phones-2011-5?op=1, www.google.com/nexus/5/.

[6] www.kpcb.com/insights/2013-internet-trends

once, lock when use is over - was developed for desktop machines, and it was assumed that users would de-authenticate upon leaving the vicinity. It is comparatively easy for a malicious individual to remove a mobile device from the control of the genuine user by snatching it from a table, or even a pocket or hand. If the user is logged in, the thief would be fully authenticated on the stolen device; through VPN and remote desktop apps, they may be able to gain access to the user's other devices. Current NIST guidelines for mobile security do not specify any form of automatic locking as long as someone is using the device, regardless of whether or not they are the authorized user. Instead they recommend waiting until the device is idle for a certain amount of time before locking it.[7] In the case of stolen devices, corporate systems such as Symantec Mobile Security rely on remote locking to deny unauthorized users access to mobile devices.[8] Such action can take several minutes, as the user must either use another device or call a technical support agent to initiate the lock.

The aforementioned flaws necessitate a fundamental overhaul of mobile security. User authentication needs to be both *unobtrusive* and *continuous*. By regularly conducting unobtrusive identity checks of the mobile user, continuous authentication verifies that it should still be in an authenticated state. With this system active, if the mobile device is stolen, it should quickly recognize the presence of an unauthorized user and lock itself. While biometric sensors for fingerprint, face, voice, and iris are available in mobile platforms, they are still used only for unlocking the device or authenticating users for mobile payments. Mobile devices possess other sensors (such as gyroscope, magnetometer, accelerometer, GPS, and touchscreen), which could assist in user authentication by deriving soft biometrics such as gait as well as typing/scrolling pattern. Figure 3 shows an example framework for mobile authentication based on physical and behavioral biometric traits; this framework is both continuous and unobtrusive.

In this paper, we outline our development of an unobtrusive continuous authentication system based on face matching. Performance and accuracy for unconstrained face matching is improved by integrating data from the device's accelerometer, gyroscope, and magnetometer (collectively called the Inertial Measurement Unit or IMU) to correct camera sensor orientation and hence face image. Our approach is generalizable to any mobile device with a front-facing camera and IMU.

[7] nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

[8] www.symantec.com/content/en/us/enterprise/fact_sheets/b-mobile-security-DS-21260542.pdf

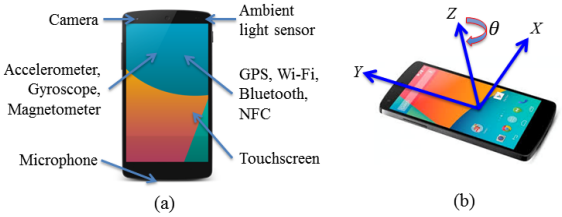


Figure 2. Sensors (a) and axes (b) of a Nexus 5 Android phone.

2. Background

Android is currently the most popular mobile phone operating system, with 84.7% of the current market share.[9] Unlike iOS, the second most popular system, Android allows device sensors to be accessed from background services. Android devices also commonly have multi-core processors, allowing continuous authentication to run seamlessly in the background while the user accesses other applications. Development of the proposed continuous authentication system will be focused on the Google Nexus 5.[10] The Nexus line incorporates the full set of device sensors and allows for low-level operating system modification, an important characteristic for continuous authentication research and prototype building.

Continuous authentication monitors and authenticates the user throughout the use of the device [16]. While passwords and tokens are commonly used in one-shot authentication, the characteristic of periodical monitoring in continuous authentication excludes them for our task. Instead, continuous authentication systems have been seeking biometric trait-based approaches.

The published literature on continuous authentication can be grouped into two main categories: computers (desktop and laptop) and smart mobile devices. Continuous authentication systems periodically verify the identity of a user by collecting both physical (e.g., fingerprint [15], face [15, 8]) and behavioral (e.g., keystroke dynamics [14, 7, 10], and mouse movement [6, 13]) traits. For mobile devices, the touch screen and internal sensors (e.g., gyroscope, accelerometer, GPS, and Bluetooth) become the important sources for collecting behavioral traits such as touch pattern [2, 3, 17], motion pattern [1], gait [9, 18], location and context [4] information.

Obtrusive authentication requires users' cooperation which significantly limits its usability. By contrast, unobtrusive authentication does not require cooperative input by the user; the system performs authentication by automatically collecting person specific information, such as various biometric traits and other sensory data. Compared with ob-

[9] www.idc.com/prodserv/smartphone-os-market-share.jsp

[10] The Nexus 6, the successor to the Nexus 5, was not available at the time this study began.

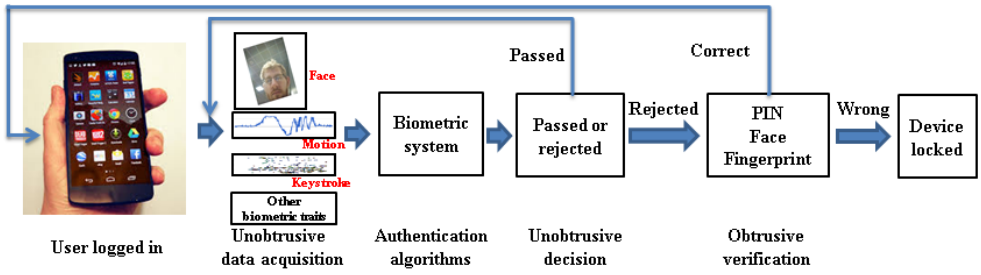


Figure 3. A general framework of unobtrusive continuous authentication on mobile based on biometric system.

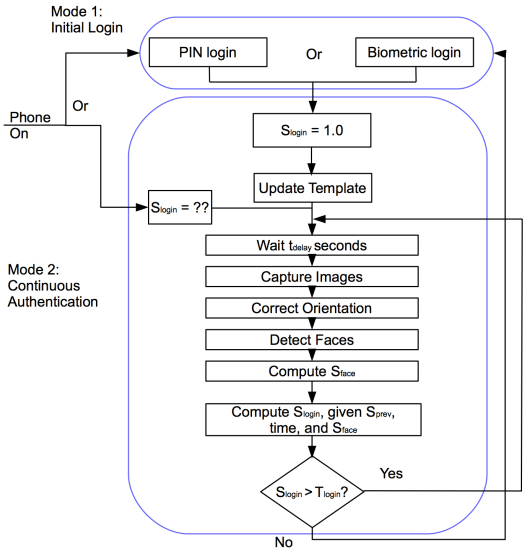


Figure 4. Flowchart of the implementation of the proposed unobtrusive continuous authentication system based on unconstrained face images and IMU data. t_{delay} (default 30 seconds) controls how often the user's identity is verified, while T_{login} (default 0.6) controls the minimum confidence needed to keep the user authenticated.

trusive authentication based on PIN or token, an unobtrusive authentication system may provide relatively low-security access control, but has a higher usability. For the mobile device domain, the usability is critical, because users do not want to break from their workflow to verify their identity.

The main contributions of this paper are: (i) Real-time orientation correction of face images captured unobtrusively on a mobile device using gyroscope, accelerometer and magnetometer data, thereby improving the face matching accuracy. (ii) A person-specific (one-vs-all) method for discriminating the genuine mobile device user from a malicious individual. (iii) A prototype continuous authentication system on a mobile device.

3. Proposed continuous authentication system

Figures 3 and 4 illustrate the proposed mobile device continuous authentication system using unconstrained face images and IMU data. Ideally, such a system would be truly continuous, monitoring for impostors at all times. However, this is not feasible due to battery and processing power constraints. Instead, periodic sampling is used to determine the user's identity, with t_{delay} seconds between two sampling sessions, lasting t_{sample} seconds. Each sampling session is compared with the enrollment face image set (e.g., multiple face images of the genuine user). Simply thresholding the face match score S_{face} for each sample removes a great deal of granularity and will inevitably result in a large number of false accepts and rejects. Rather, a function f_{map} of the face score is used to increment or decrement a second score, the login score S_{login}. This score represents the confidence in the user's identity and is set to 1.0 when the user logs in. At this point, the face template can be updated if necessary, as this is the only time that the authorized user has been obtrusively verified to be using the device.[11] If the login score ever drops below a confidence threshold T_{login} (in this case 0.6), the user is immediately logged out of the device. The confidence in the user's identity should not simply change when a face score is obtained, but should rather decrease in the time between such instances. This decrease is specified by the integral of a function f_{dec} over the time between sessions. f_{dec} can be constant or can vary with time since the last login. The latter allows for greater control over the system behavior.

3.1. Uprightness correction

It is unreasonable to assume that mobile device users will keep their devices oriented perfectly vertical at all times. If the camera is oriented with some z-axis rotation (see Fig. 2(b)) relative to the face, face detection and matching will become more difficult [11]. In most cases, a face will be oriented so that the major axis has a small angle from vertical in face roll. Given this assumption, a correction of the input image so that a face will appear at a similar angle

[11] The issue of face spoofing is not addressed here

from vertical in the image will significantly improve matching accuracy. Such a correction can be accomplished either through interpretation of image data or through fusion with additional sensor data.

As noted above, Android devices are equipped with an IMU which can be used to determine the in-plane device rotation θ w.r.t. the z-axis. This information, along with the width and height of the input and output images, is used to define a rotation matrix M:

$$M = \begin{bmatrix} 0 & 0 & \frac{w_o}{2} \\ 0 & 0 & \frac{h_o}{2} \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & -\frac{w_i}{2} \\ 0 & 0 & -\frac{h_i}{2} \\ 0 & 0 & 1 \end{bmatrix},$$

where w_i and h_i are the width and height of the raw image to be rotated, respectively; w_o and h_o are the width and height of the output image, respectively. These are specified separately to allow for a larger output image so that none of the input image data is "cut off" by the rotation.

One challenge in obtaining an accurate value of θ, and by extension M, is that the Android IMU does not provide data at user-requested times. Rather, it samples the sensors at one of several selectable fixed rates (ranging from 5 to 200 Hz on the Nexus 5) and provides only those values. Slower sampling rates are more desirable because they consume less battery power than high sampling rates; however, lower sampling rates lead to higher error in rotation. The motion of the device is continuous, and fixed rate sampling, especially at a lower data rate than the camera provides, neglects the change in orientation experienced between the capture of IMU data and camera data, which may be significant. If the corrected images are viewed as a continuous stream, this phenomenon manifests itself as a "jittering" of the image.

Using the times the orientation and image are captured, the estimate of θ can be refined. After image data is captured, the system waits for the next IMU value to be reported. The device rotation θ_{img} at the instant of capture is then interpolated.

Formally, given the rotation measurements θ_i and θ_j from the IMU, taken at times t_i and t_j ($t_i < t_j$), respectively, a linear interpolation for the device rotation θ_{img} at time t_{img} ($t_i < t_{img} < t_j$) when the image was captured can be calculated as

$$\theta_{img} = f(\theta_i, \theta_j, t_i, t_j, t_{img}) = \theta_i \frac{t_j - t_{img}}{t_j - t_i} + \theta_j \frac{t_{img} - t_i}{t_j - t_i}.$$

Once the rotation matrix M is obtained, it is applied to the image. To handle clipping in the case where the output image dimensions are insufficient to hold the rotated image, M^{-1} is used to transform the coordinates (x, y) of each pixel in the *output* image to the corresponding *input* image coordinates (x', y'). The transformed coordinates do not exactly correspond to pixels in the input image, so bilinear interpolation is used to estimate the true color of the rotated pixel. Examples of the original and corrected face images of one subject are shown in Fig. 5.

Figure 5. Examples of face image orientation correction using gyroscope, accelerometer and magnetometer data. (a) Captured face images, and (b) corrected face images in which faces are kept upright during the rotation of the mobile device.

This method only corrects for sensor orientation, not face orientation. Since phone users may sometimes incline their heads at the same angle as the device screen in order to better view it, the assumption of head uprightness may not always hold. To this end, it is best that the orientation corrected image be matched in addition to, rather than instead of, the uncorrected image to maximize face detection probability and matching performance.

3.2. Person-specific modeling

Considering the unconstrained and uncooperative scenarios in continuous authentication, relying on a single uprightness-corrected face image is not reliable. Therefore, we propose to match all the uprightness-corrected face images in a session to the enrollment face images of the mobile device user. Using a score-level fusion (with sum rule) of the face match scores from individual testing and enrollment face images, we generate a match score between a session of testing face images and the device user's enrolled face images. We use the commercial off-the-shelf (COTS) PittPatt SDK as our baseline face matcher.

Continuous authentication is composed of periodic one-shot authentication, each of which determines whether the current user is the enrolled user or not. This can be formulated as a one-vs-all classification problem. Therefore, a person-specific model should be used to take into account changes in face appearance over time to represent the "one" class. All the enrollment face images of a mobile device user are used as the samples of "one" class. To get a better representation for the "all" class, we used the entire collection of face images in the Labeled Faces in the Wild (LFW) database which includes $13,233$ face images of $5,749$ subjects.[12]

Given the training data of the "one" and "all" classes of a mobile device user, we extract features using Biologically

[12] vis-www.cs.umass.edu/lfw/

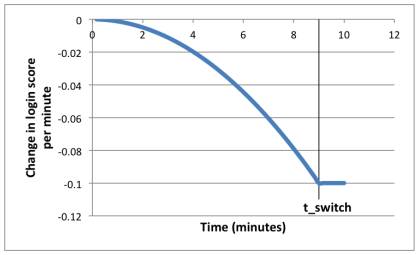


Figure 6. Function f_{dec}(t) mapped from 0 to 10 minutes

Inspired Model (BIM) [12] from six different facial components (the forehead, eyebrows, eyes, nose, mouth and the whole face) of each subject, where the facial components are located using a method proposed in [5]. Different from [5], we use PittPatt instead of Viola-Jones face detector to detect the (unconstrained) face before performing facial landmark detection. We train a one-vs-all SVM classifier with RBF kernel for each of the six facial components, and the six per-component classifiers are fused in score-level (z-score normalization is applied) to produce a person-specific classifier.

3.3. Confidence functions

As mentioned above, two functions, f_{map} and f_{dec} are being used to compute the confidence score. f_{dec} maps the decrease in the confidence score over time, accounting for the decline in confidence in the user's identity between sessions. f_{dec} is a piecewise function with a single parameter, t_{logout}, which is defined as the time the device should take to log the user out if no face match data is obtained.

$$t_{switch} = 1.5 * t_{logout} - 6$$
$$f_{dec}(t) = \begin{cases} -\frac{-0.1}{t_{switch}^2}t^2 & t < t_{switch} \\ -0.1 & t \geq t_{switch} \end{cases}.$$

f_{dec} dictates the slope of S_{login} at any given time (except for discontinuities when face match data is obtained). The quadratic portion corresponds to a cubic decline in login confidence. Confidence that the genuine user is in control of the device should decline slowly after login, when the identity of the user is confirmed, and should decline faster as time goes on. The slope is bounded at -0.1 per minute to prevent a loss of confidence that is too rapid. f_{dec} is plotted in Fig. 6. All times are specified in minutes. If faces are not detected during a given image capture, f_{dec} completely governs the behavior of the system. In cases where there are multiple sessions where there is no face detected, the reaction will vary based on the time since login. The system is more tolerant to multiple sessions without a face that are encountered when the device has not been in use for long than it is once the device has been used for several minutes.

Table 1. Confidence function parameters.

Parameter	Value
t_{logout}	10 minutes
T_{login}	0.6
t_{sample}	10 seconds
t_{delay}	30 seconds

Table 2. Rules used to determine the influence of a face match score from a session on the login score. The scores at individual FARs are calculated from an independent dataset.

Values of S_{face}	Changes of S_{login}
$S_{face} \geq$ score at 1% FAR	$1.0 - S_{logout}$
$S_{face} =$ score at 5% FAR	$0.1 \times (t_{delay} + t_{sample})$
$S_{face} =$ score at 10% FAR	0
$S_{face} \leq$ score at 20% FAR	$-(1.0 - S_{logout})$

During the continuous authentication, the face matching score S_{face} of each session (calculated as session-session match score) must be transformed into some increment or decrement to S_{login}. Values for false acceptance rate (FAR) and true acceptance rate (TAR) at various values of S_{match} from a pilot study are used to define the cubic transformation function f_{map}. Table 2 defines the points to which the function is fit.

The resulting fit function is used as f_{map}, which in this case uses scores from PittPatt SDK as values of S_{face}

$$f_{map} = \begin{cases} 0.4 & FAR(S_{face}) \leq 1\% \\ f'(S_{face}) & 1\% < FAR(S_{face}) < 20\% \\ -0.4 & FAR(S_{face}) \geq 20\% \end{cases},$$

where $f'(S_{face}) = 0.608S_{face}^3 - 0.853S_{face}^2 + 0.648S_{face}$.

These two functions (f_{map} and f_{dec}), along with the time of the previous session, t_{prev}, are used to determine the login score at time t_{ses}

$$S_{login}(t_{ses}) = S_{login}(t_{prev}) + \int_{t_{prev}}^{t_{ses}} f_{dec}dt + f_{map}(S_{face}).$$

As t_{delay} gets longer, it becomes more possible that f_{dec} will cause the login score to dip below zero between sessions. To prevent the impostor from having extra time with the device from that point to the next session evaluation, a future (*e.g.* at time t_{fut}) estimate of S_{login} is calculated using t_{delay} and t_{sample}

$$S_{login}(t_{fut}) = S_{login}(t_{ses}) + \int_{t_{ses}}^{t_{fut}} f_{dec}dt,$$

where $t_{fut} = t_{ses} + t_{delay} + t_{sample}$. The score at t_{fut} is compared with the login threshold T_{login} to determine if continued access to the device is permissible. If it falls below this threshold, the device is automatically locked. Note that due to the variable amount of time needed to calculate S_{match}, t_{fut} may not be the actual time that the login score is next calculated.

3.4. Prototype Implementation

We provide a prototype implementation of the proposed continuous authentication system on the Nexus 5. A session begins with the capture of images and correction of their orientation. One challenge is that the application of M^{-1} and bilinear interpolation of the output image, while simple operations, are time consuming to execute for the 2×10^6 pixels in a typical image. Android provides a native interface to the system hardware via RenderScript, a Just-In-Time compiled language for performing large numbers of parallel operations. By porting the rotation code to RenderScript to run on the device Graphics Processing Unit (GPU), image uprightness correction is made to operate at 15 frames per second (fps) on the Nexus 5. For the overall system, images are captured using still mode instead of video mode for better image quality, averaging 3.9 fps image acquisition speed.

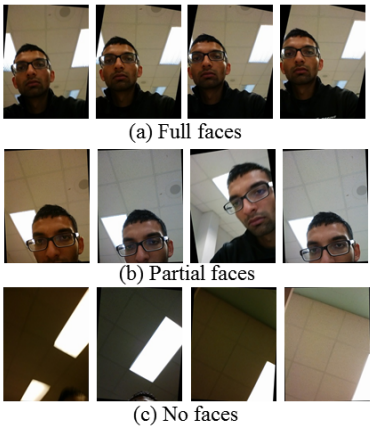
Following image acquisition, faces are extracted from the images and sent to a server for evaluation. S_{face} is calculated on a session-session basis by a matcher[13] and is transmitted back to the device. As shown in Figure 4, the enrollment template is updated every time the user logs into the device, provided at least 5 faces can be extracted from the initial session. The last 5 enrollment sessions are used for matching.

4. Experimental results

4.1. Face matching accuracy

Using a customized mobile application running on an Android smartphone, we collected images of 10 participants over a period of 1-6 weeks. The application captured images at variable framerates (3.9 fps on average) and saved IMU orientation correction data. Images were captured for one minute at a time every time the user turned on their device and every eleven minutes subsequently (ten minute spacing between sessions). In total, we corrected over $250,000$ images from about $3,600$ sessions, with an average of about 70 images per session.[14] The average numbers of images and sessions per subject are about $25,000$ and 360, respectively.

Effectiveness of the proposed uprightness correction method and the person-specific model was validated by reporting the face verification performance on this dataset, which we divided into an enrollment set (target set) and a testing set (query set). We randomly select 30 sessions from each subject to make the enrollment set. The remaining sessions (about 330 sessions) of each subject are used for testing. This results in about 6,300 genuine session pairs and 56,000 impostor session pairs for each subject. We do the random selection 5 times, and report the average performance with standard deviation. The reason why only 30 sessions of each subject are used for enrollment is to better replicate the scenarios in real applications. A mobile device user is less likely to spend a long time (e.g., more than a few minutes) for enrollment. We perform experiments by using each session of images as a data point. Due to the unconstrained nature of the smartphone dataset, about half of the images do not contain a face (see examples in Fig. 7(c)).


(a) Full faces

(b) Partial faces

(c) No faces

Figure 7. Example images of one subject from our database under different scenarios, where (a) full faces, (b) only partial faces, and (c) no faces are captured in the images.

- **Uprightness correction improves face matching accuracy**. We perform face matching on both the original images and the uprightness corrected images using the baseline matcher (PittPatt SDK). High true acceptance rate (TAR) at low false acceptance rates (FAR), e.g., 1%, is important for the targeted applications. The ROC curves from the original images and the uprightness corrected images in the FAR range of 0.1%–1% are shown in Fig. 8(a). We can see that the proposed automatic uprightness correction method improves the face matching accuracy. For example, uprightness correction leads to 6% higher performance at 0.1% FAR (40% vs. 34%). Example sessions where the uprightness correction improves the face matching performance are shown in Fig. 9(a).

- **Person-specific model outperforms the COTS matcher**. We then compare the proposed person-specific model with the COTS matcher on the uprightness corrected images. The ROC curves of these two methods in Fig. 8(b) show that the proposed person-specific model outperforms the COTS matcher (4% higher performance than the COTS matcher at FARs of 0.1%–1% on average).

  A score level fusion of the proposed person-specific model with the COTS matcher achieves better perfor-

[13] For the system evaluated in Section 4.2, the PittPatt matcher was used.

[14] The numbers of original and corrected images are both $250,000$.

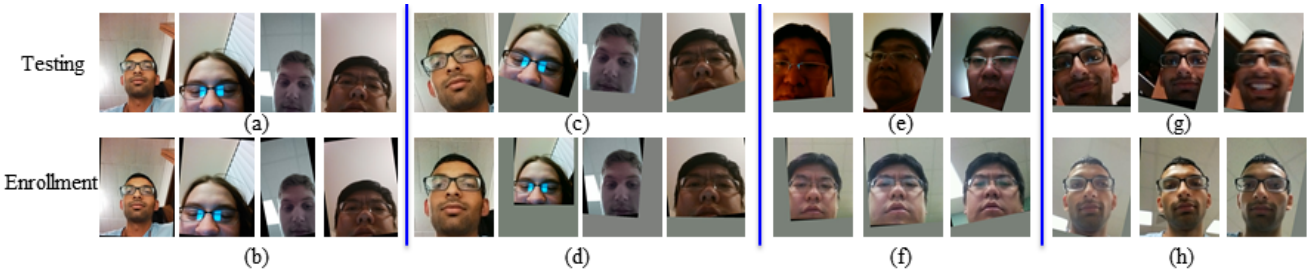


Figure 9. Examples of face detection and matching by the proposed approach. (a, c) show the original images without uprightness correction, and the face detection results, respectively; (b, d) show the uprightness corrected images and the face detection results, respectively; (e, f) show impostor face session pairs that are wrongly accepted by COTS, but correctly rejected with the fusion of COTS and person-specific model; (g, h) show genuine face session pairs that are wrongly rejected by COTS, but correctly accepted with the fusion of COTS and person-specific model. (e,f,g,h) are shown at 1% FAR.

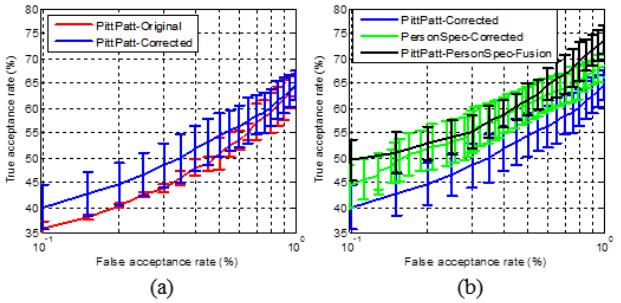


Figure 8. Performance of the proposed face matching approach. (a) The proposed automatic uprightness correction method improves the face matching accuracy; (b) The proposed person-specific (PersonSpec) method outperforms the COTS matcher. The fusion of the proposed person-specific method and COTS leads to further improvement.

mance. At 1% FAR, the fusion leads to the performance of 73% TAR, compared to the 64% for the COTS matcher. Example sessions where the fusion improves the performance are shown in Figs. 9(b,c).

4.2. Authentication performance

While the previous experiment was carried out in a highly unconstrained set of environments, the verification of authentication performance was conducted in an office environment. 24 subjects used a device equipped with the prototype continuous authentication application. The subject operated the device for up to 15 minutes, and the time the user was logged out was noted. For impostor trials, the user operated the device for about a minute, then handed the device to an impostor, who used it until they were logged out. Each participant performed both trials 3-5 times with different impostors. Results are summarized in Figure 10. For all of these trials, t_{logout} was set to 10 minutes, mirroring suggested idle logout time from HIPAA.[15]

[15] www.hipaa.com/2009/06/access-control-automatic-logoff-what-to-do-and-how-to-do-it/

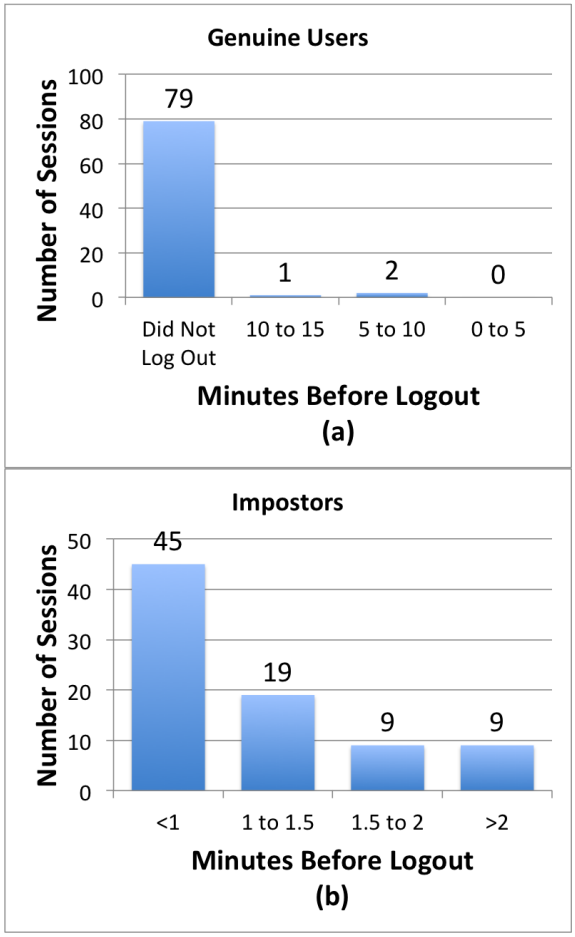


Figure 10. Results of Genuine(a) and Impostor(b) Tests

As Figure 10 (a) shows, in over 96% of the trials, the genuine user retained access to their device for the entire duration of the test. Of the remaining 3 trials, the user retained device access for at least 5 minutes. The early logouts can be attributed in two cases to PittPatt attempting to match a partial face, and in the third to the user holding

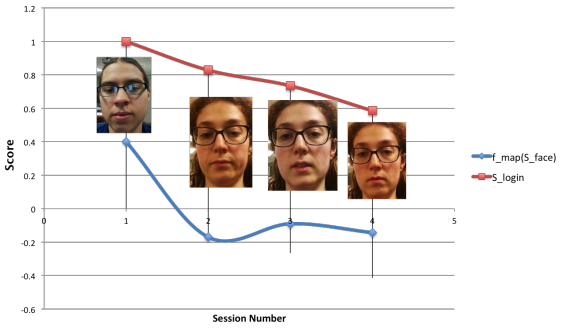


Figure 11. Impostor trial with sample face images.

their head at an angle. Our future work in unconstrained matching will attempt to detect and address these scenarios.

As indicated in Figure 10 (b), the impostor had access to the device for less than 1 minute in the majority of trials, and in 89% had access for less than 2 minutes. With one genuine-impostor pair, the off-the-shelf matcher continually identified the impostor as a genuine user, allowing continued access to the device. However, it is important to note that this worst-case scenario merely offers the same behavior as an unprotected phone, where any impostor has unlimited access to the device as long as they continue to use it.

Figure 11 illustrates the effect of an impostor on the face and login score. Note that although the face similarity score is only moderately negative, the impostor is still logged out relatively quickly.

5. Conclusions and Future Work

We have presented a robust continuous authentication system for mobile devices. The fusion of camera with the device's IMU allows for enhanced face matching performance, boosting the effectiveness of the system. The proposed person-specific model enables robust performance on face images captured unobtrusively (in unconstrained environments without altering user behavior). The dual confidence functions proposed will allow for use of increased granularity of face data and allow for modeling of changing confidence even between authentications. Experiments have shown significant reduction in the time that impostors have access to a device, as well as the high degree of usability for genuine users. In future work, we will modify the system to perform all functions on-device, eliminating the need for a separate server for matching. Additionally, a larger database of subjects will be collected to evaluate matching performance on unconstrained subjects.

Acknowledgements

This research was supported by a grant from Google ATAP.

References

[1] C. Bo, L. Zhang, X. Li, Q. Huang, and Y. Wang. Silentsense: silent user identification via touch and movement behavioral biometrics. In *Proc. MobiCom*, 2013.

[2] N. L. Clarke and S. M. Furnell. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*, 6(1):1–14, 2007.

[3] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Trans. Inf. Forensics Security*, 8(1):136–148, Jan. 2013.

[4] K. Halunen and A. Evesti. Context-aware systems and adaptive user authentication. In *Proc. AmI Workshops*, 2013.

[5] H. Han, B. Klare, K. Bonnen, and A. K. Jain. Matching composite sketches to face photos: A component-based approach. *IEEE Trans. Inf. Forensics Security*, 8(1):191–204, Jan. 2013.

[6] Z. Jorgensen and T. Yu. On mouse dynamics as a behavioral biometric for authentication. In *Proc. ASIACCS*, 2011.

[7] F. Monrose and A. D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Generation Computer Systems*, 16(4):351–359, Feb. 2000.

[8] K. Niinuma, U. Park, and A. K. Jain. Soft biometric traits for continuous user authentication. *IEEE Trans. Inf. Forensics Security*, 5(4):771–780, 2010.

[9] A. Primo, V. V. Phoha, R. Kumar, and A. Serwadda. Context-aware active authentication using smartphone accelerometer measurements. In *Proc. IEEE CVPR Workshops*, 2014.

[10] J. Roth, X. Liu, A. Ross, and D. Metaxas. Biometric authentication via keystroke sound. In *Prof. ICB*, 2013.

[11] S. Z. Li and A. K. Jain (Eds.). *Handbook of Face Recognition, 2nd ed.* Springer, 2011.

[12] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Proc. IEEE CVPR*, 2005.

[13] C. Shen, Z. Cai, and X. Guan. Continuous authentication for mouse dynamics: A pattern-growth approach. In *Proc. IEEE/IFIP DSN*, 2012.

[14] S. J. Shepherd. Continuous authentication by analysis of keyboard typing characteristics. In *Proc. ECSD*, 1995.

[15] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar. Continuous verification using multimodal biometrics. *IEEE Trans. Pattern Anal. Mach. Intell*, 29(4):687–700, 2007.

[16] I. Traore and A. Ahmed. *Continuous Authentication Using Biometrics: Data, Models, and Metrics*. IGI Publishing Hershey, PA, USA, 2011.

[17] X. Zhao, T. Feng, W. Shi, and I. A. Kakadiaris. Mobile user authentication using statistical touch dynamics images. *IEEE Trans. Inf. Forensics Security*, 9(11):1780–1789, 2014.

[18] Y. Zhong and Y. Deng. Sensor orientation invariant mobile gait biometrics. In *Proc. IJCB*, 2014.