# Unobtrusive Gait Verification for Mobile Phones

**Hong Lu**
Intel Labs
hong.lu@intel.com

**Jonathan Huang**
Intel Labs
jonathan.huang@intel.com

**Tanwistha Saha**
George Mason University
tsaha@masonlive.gmu.edu

**Lama Nachman**
Intel Labs
lama.nachman@intel.com

## ABSTRACT
Continuously and unobtrusively identifying the phone's owner using accelerometer sensing and gait analysis has a great potential to improve user experience on the go. However, a number of challenges, including gait modeling and training data acquisition, must be addressed before unobtrusive gait verification is practical. In this paper, we describe a gait verification system for mobile phone without any assumption of body placement or device orientation. Our system uses a combination of supervised and unsupervised learning techniques to verify the user continuously and automatically learn unseen gait pattern from the user over time. We demonstrate that it is capable of recognizing the user in natural settings. We also investigated an unobtrusive training method that makes it feasible to acquire training data without explicit user annotation.

### Author Keywords
mobile phones; gait recognition; activity recognition

### ACM Classification Keywords
H.1.2 User/Machine Systems: I.5Pattern Recognition;

## INTRODUCTION
Smart phones have evolved rapidly from a pure voice communication devices to a general purpose mobile computers and personal assistants. Security of mobile devices is becoming more crucial over time as these devices start to accumulate a lot of sensitive data about their users, such as emails, calendar, pictures, communication data, financial data and recently a lot of sensor data including location.

Typing a password is still the most common authentication mechanism, which is clearly cumbersome especially on the go. Recently new methods of authentication have been used including finger print recognition, drawing a pattern on screen, or face recognition. While these methods are trying to reduce the burden , they still require explicit user interactions and are not very conducive to high mobility scenarios. While a mobile assistant running on your phone might want to whisper in your ear (through your headset) the location of your next meeting that you are running late to, or the latest urgent message from your family that you missed, requiring you to take your phone out to authenticate yourself before it can help you is clearly undesirable. The recent advancement in alternative user interaction modalities like speech and gesture recognition promise to improve on-the-go experiences considerably, but will require a "lighter weight/touch" authentication to complement these solutions. Today devices offer binary security, for example you can activate the password on an iPhone for all apps except for the camera which can be accessed in the clear to reduce latency and improve the user experience. We imagine in the future to have many levels of security that tradeoff usability and accessibility given the risk imposed. Accessing a user's bank account clearly requires higher security than retrieving the location of the next meeting and occurs less frequently, hence creating an opportunity to improve the user experience accordingly.

We propose an unobtrusive, continuous, and implicit authentication method for smartphones leveraging gait analysis. Gait is a person's manner of walking. Gait analysis has been important in the field of healthcare for a long time. Detecting changes in walking can help identify early indicators of the onset of Parkinson disease and multiple sclerosis. It is also widely used in orthopedics, physical therapy and athletic training. Researchers from psychology, healthcare and biometrics [10, 18, 29] have widely acknowledged that human gait contains distinctive patterns that can be used for security purposes. While research is still underway, it has attracted interest as a method of authentication because gait authentication is non-invasive, does not require the subject's explicit cooperation, and provide continuous authentication while walking. Another advantage is that gait recognition is a biometric method – it relies on the biological or behavioral identification characteristics, similar to fingerprint or face recognition, it is not likely to be stolen or forgotten like a password. An impostor may observe how the authentic user walks but would still have difficulty to replicate the gait pattern.

Gait analysis could be used in two types of security tasks: verification or identification. In verification, the system validates a person's identity by comparing the captured data with


ISWC'14, September 13 - 17 2014, Seattle, WA, USA

http://dx.doi.org/10.1145/2634317.2634324

his enrolled gait pattern. It is a one-versus-others task. In identification, the system recognizes an individual among a group of users who have enrolled in the database. It answers the question “who is the walker”. The focus of this paper is primarily on verification, given the assumption that mobile phones are typically personal devices. Even though efforts are being made to make gait verification as accurate and usable as possible, it might never reach the same level of reliability as other stronger biometrics such as fingerprint or iris recognition. However it could be used as a method of continuous authentication for accessing less sensitive application/content on a smartphone (such as games,books, and newsstands) or be used in conjunction with other biometrics (e.g. speech recognition) [27] to provide more reliable authentication.

All existing research related to accelerometer based gait recognition has focused on fixed body placement and device orientation, such that the accelerometer's axes are aligned with the human body to facilitate analysis. Usually, the device is fixed on the shoe, ankle, lower leg, or around the hip or waist. However, to be practical in everyday scenarios, a gait verification system for a mobile phone needs to be able to deal with various device contexts (placement and orientation) encountered in typical phone usage. Also, a user's gait may vary over time due to changes in clothing (e.g. shoes and pants), terrain, mood, and health condition, making it impractical to enroll an individual's gait patterns under all possible conditions. It is therefore advantageous to have a means for systematically evolving gait models automatically using an unsupervised method to lower the enrollment/training burden. Since the mobile phone is a personal device, it is mainly used and carried by its owner. Therefore, it provides a unique opportunity for a gait verification system to learn from the user without explicitly bothering them with enrollment.

This paper presents a method to verify the user's gait on a mobile phone. To reduce the training effort, our system learns from the user implicitly over time as the user interacts with the device naturally. The contributions of this paper are as follows. We experimentally show that: 1) Gait verification can be done on smartphones in real life environments. 2) A gait model can be automatically adapted to specific users over time in everyday usage, thereby lowering the cost of enrollment. 3) The proposed gait verification pipeline can run in real-time on off-the-shelf Android mobile phones.

## RELATED WORK

Gait recognition has been studied as a behavioral biometric for more than a decade. It was first proposed by medicine and psychology researchers [6] [2] [14], who demonstrated human ability of recognizing family and friends by gait patterns [6], or inferring the walkers' gender from their gait [2].

Early machine based gait recognition research typically utilized computer vision techniques [3,16,22,23,26,28] by analyzing the movements of human body parts, e.g., the knee, the foot, the shoulder, etc. The features for classification are often spatio-temporal features (step length/width/interval, walking speed) and kinematic features (rotation of the hip/knee/ankle, joint angles of the hip/knee/ankle/thigh/trunk). An inherent advantage of vision-based gait recognition system is that it captures gait of the person from a distance when other types of biometrics (e.g. fingerprints) are not accessible. Currently the performance of the vision based gait biometrics is lower than other stronger biometrics(e.g. fingerprints), as it is still in research [4,21,22]. In a multi-biometric system, the performance could be improved when vision based gait recognition is combined with other biometrics, e.g., face recognition [15] or shoe pressure sensors [5]. Another type of appearance-based approach leverages radar [13]. It uses a continuous-wave radar to measure the Doppler shift of the signal bouncing off of people while they are walking. One advantage of a radar based system is that radar is less sensitive to the color and texture of the individual's clothing and the ambient light condition. Both vision and radar based solutions require deploying external sensors in the infrastructure, so they are not suitable for user authentication on mobile device.

Most accelerometer based gait authentication systems use pattern matching techniques [1, 7–9, 11, 18, 20]. An enrollment sample (usually a template of step or the histogram/FFT spectrum of a step) of the user is previously obtained and stored in the database. During authentication, the system acquires sensor data from an individual, segments out one step (often by local minimums and maximums), normalizes the data, and compares it to the enrollment templates in the database using a distance measurement, such as Euclidian distance, Dynamic Time Wraping (DTW), or correlation. This family of techniques works well with fixed body placement, usually on shoes [1], ankles [12], lower legs [11], waist [25,27], or hip [10,20,27] . This is due to the fact that the accelerometer's axes are aligned with the human body, and thus the vertical, forward-backward and sideways motion can be easily obtained. It is appropriate and light weight to implement in wearables, which usually have specific body placements and limited computation resources.

However, the assumption of fixed body placement and sensor orientation is not realistic with phone usage. We intend to relax the requirement of fixed placement by carefully designing our system to account for uncertainty of body placement and device orientation. We divert from the template matching technique and use a statistical modeling approach for two reasons (1) when the body placement (and orientation) of the sensor is not fixed, the segment of step cycle is less reliable due to placement and clothing (e.g. bounce in the pocket). The segmentation error will propagate into normalization and template matching phase, and greatly degrade the accuracy. (2) the user's gait may varied over time due to factors such as terrain, footwear, clothing, load on the body, mood, and fitness, etc. It is cumbersome, if not impossible, to enroll the user in all possible circumstances.

On the other hand, since a smart phone is carried mainly by its user, the system could learn from the user over time through online learning technique. Moreover, there is an opportunity for implicit labeling - when the system fails to recognize the user and the user has to unlock the phone by other means. This gives a hint of new gait patterns that the current gait model is not familiar with. Hence, it serves as a good op-

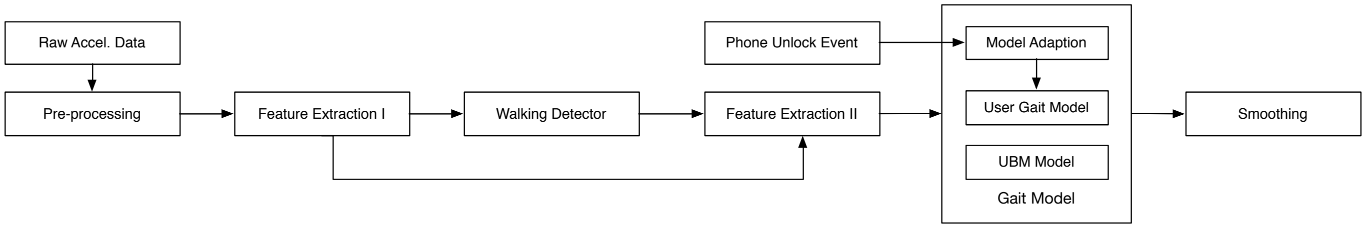


Figure 1: Algorithm Overview

portunity to update the gait model with the new pattern. We propose a statistical framework to perform gait verification and automatically personalize the gait models to learn from users over time as they interact with their mobile phones in a natural way.

## DESIGN OVERVIEW

This section provides an overview of the system work flow. We briefly describe each architectural component in turn, presenting a high-level overview of how they work in unison. In reality, the gait verification system waits until it gets a good amount of walking data, extract features, compares it to the particular user, and, if the pattern does not match, the user needs to fall back to conventional authentication method like finger print or passcode. Our proposed architecture comprises the following key components, shown in Figure 1:

**Preprocessing.** In this stage, accelerometer data is segmented into frames. Frame is the unit for feature extraction and classification. Not all frames are considered for processing. Energy level is used to filter frames that have no motion. Samples in admitted frames are projected onto global vertical (gravity) and horizontal direction to make the processing resilient to device orientation.

**Feature Extraction I.** A set of features are extracted from the projected frame for walking detection. For efficiency, this set of features are light weight yet still provide enough information for walking/non-walking classification.

**Walking Detector.** Gait analysis is only meaningful during walking. This stage discern the walking data from other activities. A decision tree classifier is applied using feature set computed by feature extraction I. Non-walking frames are dropped and will not be processed further.

**Feature Extraction II.** More relevant features are extracted for gait analysis. These features are more computationally intensive compared to feature set I. Feature set I and II are combined and used for gait verification.

**Gait Analysis.** We use a Gaussian Mixture Model - Universal Background Model (GMM-UBM) framework [24] for gait verification. The design objective is to 1) authenticate the user, 2) adapt the gait model such that it could account for different body placements and over time variance in the user's gait pattern. We adopt a technique similar to speaker verification. The authentication is done by comparing the likelihood score from a user gait model (representing the user's specific gait pattern) and a universal back ground model (which models human gait patterns in general). Naturally, if the walker is the user, the score from the user model should be higher than the UBM, and vice versa. The model adaptation component allows the model to learn from failure, so the user gait model can automatically personalize to the user over time, thereby lowering the training burden. Every time the algorithm fails to authenticate the user (false reject) and the user unlocks the phone by other means, data from previous walking session is used to update the gait model. In this process, the user just interact with the phone in natural ways and no explicit enrollment is required.

## ALGORITHMS

### Data Preprocessing

#### *Framing*

Segmenting sensor data stream into uniform frames is common practice for feature extraction and classification. In our system, the accelerometer samples at 100 Hz. The preprocessing starts with segmenting the raw accelerometer data into frames of 512 samples (5.12 seconds) with 50% overlap. The frame length of 512 samples are chosen to balance between estimation accuracy and latency. Not all the frames are admitted for further processing. All stationary (no motion) frames are dropped according to an energy threshold. Other non-walking activities and movements will also be skipped later by using activity recognition.

#### *Projection*

Each admitted accelerometer sample is projected onto a global coordinate system, which is insensitive to device orientation. Each sample in the frame, $a_i = (x_i, y_i, z_i)$, is project into $\vec{v}_i$ and $\vec{h}_i$ which are the components in the vertical and horizontal direction respectively.

First we low-passed accelerometer readings using a mean filter to estimate the direction of gravity [19]. Specifically, for each frame, gravity is estimated by

$$\vec{G} = (mean(X), mean(Y), mean(Z)) \quad (1)$$

where $X, Y$, and $Z$ are samples of respective axes in the frame. If $\vec{G}$ changes dramatically compared to the previous frame, the orientation of the device has been changed, so the frame will be dropped. Then we project each sample $a_i$ onto $\vec{v}_i$ and $\vec{h}_i$. $v_i$ is the projection along the vertical (gravity) direction. Its sign indicates the acceleration direction (up or down). $\vec{h}_i$ is horizontal. But its direction is not meaningful, because accelerometer along does not give us the absolute direction in the horizontal plane. So its magnitude $|\vec{h}_i|$ is used.

| | stage I |
|---|---|
| Time domain | mean, variance, skewness, kurtosis, energy, mean crossing rate, energy ratio between ver./hor. components |
| Frequency domain | spectrum peak, spectral entropy ratio between low freq. band energy and high freq. band energy |
| | stage II |
| Frequency domain | compressed sub-band cepstral coefficients compressed sub-band cepstral coefficients of autocorrelation |

Table 1: Motion Feature

$v_i$ and $\vec{h}_i$ are given by

$$v_i = \vec{a}_i \cdot \vec{G} \tag{2}$$

$$\vec{h}_i = \vec{a}_i - (v_i \times \vec{G}/|G|) \tag{3}$$

This operation is repeated for each of the 512 sample in the frame. The frame of raw accelerometer data is then convert to $\{(v_i, |\vec{h}_i|), i = 1, 2, \ldots, 512\}$, which is resilient to device orientation and used in feature extraction.

### Feature Extraction I

Our system uses a combination of time-domain, frequency-domain, and auto-correlation features. They are summarized in Table 1. We computed them in two stages - the set of feature required for walking detection is computed first; another set of more complex features are only computed on-demand based on walking activity detection. The same set of features are computed on both the vertical and horizontal components to characterize vertical and horizontal motion patterns.

Six time-domain features are computed on both directions: mean, variance, skewness, kurtosis, energy, mean-crossing rate. Then, the energy ratio between the vertical and horizontal components are also used. In the frequency domain, all features are based on spectrum analysis. Different activities have different energy distributions over the frequency spectrum. Walking usually has a peaks around $1-2$Hz, whereas other activities like running or in a moving vehicle usually show more energy in high frequency bands. So the ratio between lower freq. band (<3Hz) energy and higher freq. band (>3Hz) energy provides useful information as well. The features computed in stage I are used for walking detection.

### Walking Detection

As a front-end trigger, the activities of users are classified into one of the following classes: walking, non-walking motion (running, biking, in moving vehicle), and random movements (transitional motion), using the feature set I. We adopt an approach similar to [17]. Classification is done using a decision tree. We collected training data in various body placements (pockets, bags, belt clips, in hand, etc.) from both indoor and outdoor environments. A Markov Model smoother is applied on top of the sequence of inference results output by the decision tree to remove outliers.Using a smoothing length of 10 seconds, we obtain the confusion matrix shown in Table 2.

### Feature Extraction II

Once the walking detector confirms that the frame contains walking data. More relevant features are extracted for gait analysis. The first set of feature computed in this stage is the compressed sub-band cepstral coefficients (CSCC), which is

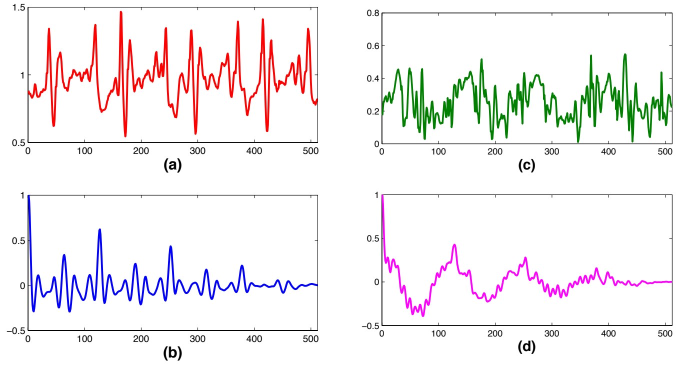


Figure 2: (a) projected vertical component. (b) normalized autocorrelation of the vertical component. (c) projected horizontal component. (d) normalized autocorrelation of the horizontal component.

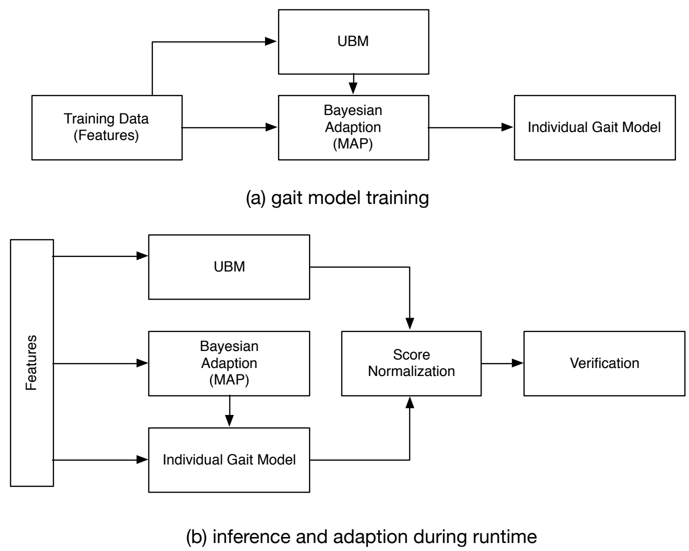


Figure 3: Algorithm workflow for (a) generate user gait model by MAP adaptation (b) runtime inference and individual model adaptation

inspired by MFCC feature set for audio analysis. MFCC is a compact set of features used to describe the characteristic of audio signal. It is widely used in speech recognition and speaker identification systems.

The compressed sub-band cepstral coefficients is computed by the following procedure, 1) compute the energy spectrum using the FFT spectrum which is already calculated in previous stage, 2) map the energy spectrum it into 26 bands using triangular overlapping bands and sum up the energy in each band, and 3) take the discrete cosine transform of the sub-band energy to form a compact 12-dimension vector representation. We use more bands in lower frequency than higher frequency in order to have better resolution in lower frequency, where most of the energy from human gait concentrates in. Like MFCC which encodes the pitch and formants structure of the voice, the compressed sub-band cepstral coefficients summarize the fundamental frequency of the movement and higher frequency vibrations in the data.

Another set of features extracted from this stage is autocorrelation features. Walking is a cyclic movement. Autocorrelation is useful in extracting the cycle, periodicity, and cadence of gait, even when the data is noisy. Figure 2 visualized projected vertical and horizontal components of a frame and their autocorrelation respectively. The data is collected from a phone carried in the front pocket of pants while the user is walking. It is known that the peak location in autocorrelation corresponds to the cycle length. We choose to use autocorrelation rather than the traditional method of segmenting the steps with a threshold based methods, because it is hard to make reliable step segmentation while the body placement is not fixed and the phone may not be tightly mounted. As shown in figure 2, due to bounce within the pocket, a local maximum or minimum could be spotted in the middle of a gait cycle, which could yield inaccurate step boundaries. In contrast, even with rather noisy input signal, the autocorrelation is still able to reveal the gait cycle robustly, especially with longer frame length. That's partially why we use a 5-sec frame to make sure that a frame captures at least several steps. Autocorrelation shows regular periodicity and structure in both components. Due to the bilateral symmetry of human gait, the autocorrelation will sometimes have a lower peak half way between each pair of major peaks. This effect is prominent in the vertical component, as Figure 2 (b) shows.

We extract the same compressed sub-band cepstral coefficient features from the autocorrelation to summarize the shape and structure of the autocorrelation of a frame. Even though we didn't use features such as cycle length (step interval) explicitly, the information of the cycle length and periodicity of gait is contained in the compressed band energy coefficients of autocorrelation, just like pitch and formants are encoded in MFCC implicitly.

Again, all features are computed in both the vertical and horizontal components to capture the patterns in both directions. The computational load of feature set II is considerably heavier than feature set I. Therefore it is computed on-demand, after walking detection stage. Combined with the feature set I, we have a 87 dimension feature vector for gait verification.

### Classification Framework

We use a GMM-UBM verification framework, which is widely studied for speaker verification systems [24]. Fig. 3 shows a block diagram of the proposed algorithm. The algorithm can be divided into three major parts: offline UBM training, user gait model generation, and runtime inference and adaptation.

The UBM is a large universal background GMM trained from a data pooled from lots of subjects. It represents a subject-independent distribution of gait patterns under various conditions. In other words, the UBM is intended to represent how human beings walk in general. As a GMM, the UBM is parameterized as $\lambda(w, \mu, \Sigma)$ where $w, \mu, \Sigma$ are mixture weight, mean, and covariance matrix. Conceptually, the Gaussian components (parameterized by $\mu$ and $\Sigma$) in the trained UBM present different walking patterns exhibited by population of subjects in different conditions (terrain, placement, etc). The mixture weights, $w$, is the probability distribution over these walking patterns in the overall population. It can be seen as the prior distribution of walking patterns. For efficiency, our system uses diagonal covariance matrix. To avoid over fitting the training data, a variance limiting technique [24] is applied with the standard expectation maximization (EM) algorithm to train the UBM models.

The individual user's gait model is a GMM as well. Instead of employing the standard EM training, we derive the individual user's gait model by adapting the parameters of the UBM using Bayesian adaptation. This provides a tighter coupling between the user's model and UBM which not only requires less training data and produces better verification performance, but also facilities further adaptation during runtime to learn from the user as discussed later in this section. The training process is illustrated in Fig. 3 (a). The Maximum-a-Posteriori (MAP) adaptation [24] we used is a non-iterative online process and is therefore performed only once for new samples. It is designed to maximize the posterior probabilities of the GMM given the training observations from a user. MAP adaptation takes the UBM and training samples(features) from the user, $X = \{x_1, \ldots, x_n\}$, where $n$ is the number of samples. For the gaussian component $i$, the component's posterior probability given observation $x_t$ is

$$p(i|x_t) = \frac{w_i p_i(x_t)}{\sum_{j=1}^{M} w_j p_j(x_t)} \tag{4}$$

Using the posterior probability of component $i$, we can calculate the sufficient statistics for the weight, mean and variance:

$$n_i = \sum_{t=1}^{T} p(i|x_t),\ E_i(x) = \frac{1}{n_i}\sum_{t=1}^{T} p(i|x_t)x_t,$$
$$E_i(xx') = \frac{1}{n_i}\sum_{t=1}^{T} p(i|x_t)x_t x_t',\ \alpha_i = n_i/(n_i + r) \tag{5}$$

The updated parameters of the MAP adapted GMM are calculated as follows:

$$\hat{w}_i = [\alpha_i n_i/T + (1-\alpha_i)w_i]\gamma \tag{6}$$
$$\hat{\mu}_i = \alpha_i E_i(x) + (1-\alpha_i)\mu_i \tag{7}$$
$$\hat{\sigma_i}^2 = \alpha_i E_i(xx') + (1-\alpha_i)(\sigma_i^2 + \mu_i^2) - \hat{\mu}_i^2, \tag{8}$$

where $r$ is a relevance factor that determines how much relevancy the original model should hold. We set r to 16 empirically. $\gamma$ is a scaling coefficient that ensures $\sum_i^M \hat{w}_i = 1$.

Essentially, the MAP adaptation personalizes the UBM model by adjusting the shape of its Gaussian components and the mixture weight. For each walking style shown in the user's training data, MAP adaptation highlights (via the posterior probability) the corresponding Gaussian component in the UBM, and adjusts the Gaussian component's shape accordingly by update $\mu$ and $\sigma$, and increases its mixture weight $w_i$, such that the derived individual gait model fits the user's data (gait pattern) much better than the original UBM. In our system, the training data could be either supervised (user labelled walking data) or unsupervised (rely on the output the walking detector, and accumulate the data automatically over time).

During runtime, the features extracted from the walking frame is evaluated against both the derived user gait model and the UBM model, as illustrated in Fig 3 (b). The verification decision is made by comparing the log likelihood of the user gait model and the original UBM. If the walker is the user, the log likelihood of the user model will be higher than UBM model. Otherwise, if the walker is an impostor, the log likelihood of the UBM will be higher. Formally, the score, $\Delta$, for a feature vector X is computed as

$$\Delta = log(p(X|\lambda_{user})) - log(p(X|\lambda_{ubm})). \tag{9}$$

therefore the verification decision is made by,

$$\Delta \begin{cases} \geq \theta : \text{walker is user} \\ < \theta : \text{walker is imposer} \end{cases} \tag{10}$$

where $\theta$ is a threshold we set empirically as discussed in next section. In addition to the binary decision based on $\theta$, the score, $\Delta$, itself is a good indicator of the confidence level, it can be used to provide a soft decision.

MAP adaptation is also used at the runtime to adapt the gait model further to learn new gait patterns of the user. A phone tends to be used by a single person and the user is unlikely to share the device with others often, therefore gait detected by the walking detector could be candidates for adaptation. Even more importantly, the system could learn from its failure. False negatives, detected by an alternative authentication mechanism like passcode or facial recognition, indicates a potential new walking pattern which is not emphasized in the current user model. As such the longer the system observes the user the better it adapts to user's profile. For each of the candidate walking session, the first and last a few frames are not used for adaption, because they contains the acceleration and deceleration phases where gait is less stable.

The user model itself is adapted from UBM, which contains many walking patterns to begin with. The adaptation during training process does not remove any of the components in the original UBM, therefore, the user model still contains all the components (patterns) but just assigning them very low weights. Intuitively, the runtime adaptation works by highlighting the existing underweighted Gaussian component corresponding to the new pattern of the user, and adjusting it by giving it higher weight and update its mean and variance to fit to the user's data better. Comparing to re-training the user model using EM, which requires storing all the training data and require lots of computation, this adaptation method is online, light weight, and requires less user data, yet converges faster. Also, the adaptation is unsupervised in a sense, it leverages the walking detector's inference result and user's unlock action, no extra user interaction is required for adaptation specifically.

## EXPERIMENT

To evaluate the proposed gait verification system in different settings, we collected data sets with off-the-shelf android phones (including Samsung Galaxy S3, S4, Google Nexus 5 and Intel Xolo) in both controlled environments and real-world environments. Three data sets are used for evaluation:

| classified as → | sedentary | walking | other motions | random |
|---|---|---|---|---|
| walking | 0.014837 | 0.949555 | 0.011869 | 0.023739 |
| biking | 0.008180 | 0.021267 | 0.910702 | 0.059851 |
| running | 0.000000 | 0.000000 | 0.970588 | 0.029412 |
| vehicle | 0.056477 | 0.000582 | 0.935953 | 0.006987 |

Table 2: Confusion matrix of the walking detector

1. the data set used for training walking detector and UBM. It consists of 47 subjects, including 19 females and 28 males. The subjects performed the following activities with equal proportion: stationary(sitting, standing), walking, biking, running, vehicle, random movements. They are instructed to carry the phone in different body placements and perform the activity naturally. Not all the subject performed all the activities. In total we have about 18 hours of data.

2. the data set used for evaluating supervised training of gait model. We collected 12 subjects'(5 females and 7 males) gait data with annotations (start/end, body placement, speed). The subjects were told to carry the phone in at least 2 of their natural placements (such as pant front/back pocket, jacket pocket, belt clip, jacket pocket, and bags). The data collection is performed in multiple sessions in order to capture different placements and walking speed (slow, normal, fast). We got about 1 hour of walking data from each subject.

3. the data set used for evaluating unsupervised training. We collect this data set in real-world setting with longer time horizon. Accelerometer data is collected 24/7 from 8 subjects who contributed data for two to three weeks. Their height ranges from 1.6 to 1.95 meters, and their weight ranges from 60 to 115 kilograms. Their phones are programmed to start data collection automatically on boot. Unless the user intentionally turns off the data collector or the battery runs out, accelerometer data is continuously collected in the background. The subjects just use the phone as their primary phone, and no specific instruction is given to them. The data is unlabeled. On average, we extracted 5 hours of walking data from each of the subjects using the walking detector.

These three data sets are collected in three different geolocations and do not overlap each other.

Table 2 shows the confusion matrix of the walking detector with 10s smoothing window. The result is derived from 10-fold cross validation on data set 1. Overall, the precision of the walking detector is about 98% and the recall is about 95%. As a front-end trigger of gait analysis, the walking detector is tuned for higher precision in order to reduce false positives.

Figure 4 (a) and (b) illustrates the distribution of the score (log-likelihood ratio, $\Delta$) of authentic users and impostors respectively. For each of the 12 users, 1/3 of the data is used to learn the user gait model and the other 2/3 is used for testing. The log-likelihood ratio is the main criteria for gait verification. Intuitively, the score of authentic user is higher than the score of impostors. The user's score is well above zero,whereas the impostors' score centers around zero. A score close to zero indicates that the user gait model is not

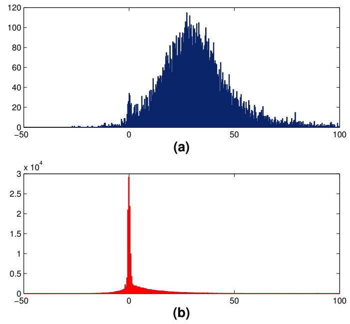


Figure 4: the distribution of score of (a) the authentic user (b) impostors

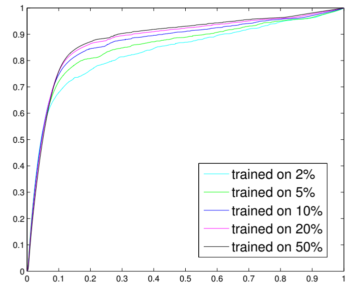


Figure 5: ROC curve of supervised training method

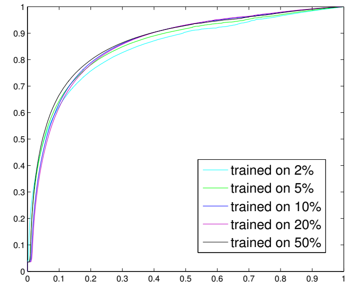


Figure 6: ROC curve of unsupervised training method

familiar with the input gait pattern. Most likely, the user does not yet exhibit that gait pattern, so the Gaussian component corresponding to that pattern is not adapted much by the training process (even though its weight might be reduced), therefore the log-likelihood from user model is not very different from the log-likelihood from UBM. The score is a good indicator of the confidence level, but the final decision depends on the threshold $\theta$, which controls the trade off between incorrect rejection and false admission. The proper value of $\theta$ might be set according to the application scenario. In this evaluation, we focus on the equal error rate (EER), which is a measurement of the overall performance.

Figure 5 presents the ROC curve of the gait verification system with user annotated data. We experimented with different proportions of data for training, ranging from 2% to 50%, the rest of data is used for testing. As expected, using larger amount of training data increases the accuracy. The benefit however levels off after more then 10% of data is used from training. The ERR is about 14% when 20% of annotated data is used for user gait training. The placement of the device greatly shapes the gait pattern that it senses. Some placements are better than others for gait analysis purpose. Taking a closer look into the errors, we noticed that body placements where the phone is tightly coupled with body motion (e.g., pants pocket, hip clip) are better than placements where the phone is loosely correlated with motion of gait (e.g., holding in hand, where the motion of gait is compensated by movement of the hand).

A similar experiment is repeated on data set 3, and the result is shown in Figure 6. The data is unannotated, so we first applied the walking detector on the data to identify walking frames. We also vary the amount of data for training. Clearly, the spread between different training percentage is narrower than that of supervised data set. It is due to the fact that we have abundant data for training with the unsupervised approach. It demonstrated one of the key advantages of the unsupervised method - it is not constrained by user's willingness to annotate data, therefore it could leverage larger amount of unlabeled training data. Over time, the quantity of data available could compensate for the disadvantage of lower quality data. Another observation is that even when 20% of the data is used for training, the accuracy is still about 5% lower than in the previous supervised case. The reason could be twofolds (1) the training data selected by walking classifier is less clean due to the unsupervised nature of this approach, and (2) this data set spanned longer time period and was collected in totally uncontrolled environment, thereby it contains richer real-world conditions and more realistic walking styles of users. Both factors make the gait analysis more difficult.

| Stage | Avg. Runtime (microsecond) |
|---|---|
| projection & admission | 855.156 |
| feature extraction I | 3872.707 |
| walking detector | 41.667 |
| feature extraction II | 9100.677 |
| gait analysis | 13993.379 |

Table 3: Runtime Benchmark

To best understand the cost of running gait verification on a mobile phone, we conducted a detailed component benchmark test, as shown in Table 3. We implemented and benchmarked a prototype gait verification system on Google Nexus 5 running Android 4.4.2. The runtime shown in Table 3 is measured with one single processing thread. Note, however, that the stages are only engaged when necessary, e.g., in the case of sedentary, the motionless frames would be dropped immediately without reaching the first projection&admission stage. Feature extraction and gait analysis (verification and adaption) are two heaviest operations. However, even when the whole pipeline is fully engaged, it only takes less than 30 millisecond to process a frame (5 seconds) of accelerometer data. During full operation, the CPU usage is below 3% (including overhead of a simple UI), whereas the memory usage is about 8MB which again includes Android UI and service overhead. Our results show that it is feasible to implement a continuous gait verification system on off-the-shelf phones. Insights and results from this initial implementation will serve as a basis for the further development and optimization.

We are aware that gait verification requires continuous accelerometer sensing, which could keep the phone CPU awake and drain the battery overtime, but we do believe the problem of power consumption could be mitigated by the adoption of task-offloading techniques, such as Apple's M7 motion processor and batch sensing model offered in Android 4.4.

## CONCLUSION AND FUTURE WORK

We presented an adaptive method for gait verification in diverse scenarios for smartphones. Unlike conventional authentication methods, gait verification does not require user's explicit input. Using a non-iterative MAP adaptation scheme for Gaussian Mixture Models, we demonstrated that it is feasible to customize the gait model using only unlabeled data at a low computational overhead. Our proof-of-concept software demonstrates that it can run on off-the-shelf Android smart phone in real time.

As part of our future work, and building on our initial prototype implementation, we plan to conduct a user study in a larger scale. On one hand, we could harvest a diverse range of gait data in the wild from heterogeneous participants for better UBM training. The performance of gait verification greatly depends on the quality (generality) of the UBM. The UBM trained on a relatively small number of subjects in this work could be a bottle neck of current implementation. With a more representative UBM, we expect to see an improvement in the verification accuracy. On the other hand, the behavior of a gait verification system depends heavily on the setting of its parameters. Trade-offs can be made to balance performance and usability. For example, longer smoothing window and tighter threshold help to improve the verification accuracy, but require longer time to verify the user, thus degrading the usability. Conducting a user study will help us understand the user experience and give us insight for setting practical trade-offs between verification accuracy and system usability.

## REFERENCES


1. Bamberg, S. J. M., Benbasat, A. Y., Scarborough, D. M., Krebs, D. E., and Paradiso, J. A. Gait analysis using a shoe-integrated wireless sensor system. *Information Technology in Biomedicine, IEEE Transactions on 12*, 4 (2008), 413–423.
2. Barclay, C. D., Cutting, J. E., and Kozlowski, L. T. Temporal and spatial factors in gait perception that influence gender recognition. *Perception & Psychophysics 23*, 2 (1978), 145–152.
3. BenAbdelkader, C., Cutler, R., Nanda, H., and Davis, L. Eigengait: Motion-based recognition of people using image self-similarity. In *Audio-and Video-Based Biometric Person Authentication*, Springer (2001), 284–294.
4. Bolle, R. *Guide to biometrics*. Springer, 2004.
5. Cattin, P. C., Zlatnik, D., and Borer, R. Sensor fusion for a biometric system using gait. In *Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on*, IEEE (2001), 233–238.
6. Cutting, J. E., and Kozlowski, L. T. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of the psychonomic society 9*, 5 (1977), 353–356.
7. Derawi, M. O., Bours, P., and Holien, K. Improved cycle detection for accelerometer based gait authentication. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, IEEE (2010), 312–317.
8. Derawi, M. O., Nickel, C., Bours, P., and Busch, C. Unobtrusive user-authentication on mobile phones using biometric gait recognition. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, IEEE (2010), 306–311.
9. Gafurov, D. A survey of biometric gait recognition: Approaches, security and challenges. In *Annual Norwegian Computer Science Conference* (2007), 19–21.
10. Gafurov, D. *Performance and security analysis of gait-based user authentication*. PhD thesis, University of Oslo, 2008.
11. Gafurov, D., Helkala, K., and Søndrol, T. Biometric gait authentication using accelerometer sensor. *Journal of computers 1*, 7 (2006), 51–59.
12. Gafurov, D., Snekkenes, E., and Bours, P. Improved gait recognition performance using cycle matching. In *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, IEEE (2010), 836–841.
13. Gürbüz, S. Z., Melvin, W. L., and Williams, D. B. Detection and identification of human targets in radar data. In *Defense and Security Symposium*, International Society for Optics and Photonics (2007), 65670I–65670I.
14. Johansson, G. Visual motion perception. *Scientific American* (1975).
15. Kale, A., RoyChowdhury, A. K., and Chellappa, R. Fusion of gait and face for human identification. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 5, IEEE (2004), V–901.
16. Little, J., and Boyd, J. Recognizing people by their gait: the shape of motion. *Videre: Journal of Computer Vision Research 1*, 2 (1998), 1–32.
17. Lu, H., Yang, J., Liu, Z., Lane, N. D., Choudhury, T., and Campbell, A. T. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ACM (2010), 71–84.
18. Mantyjarvi, J., Lindholm, M., Vildjiounaite, E., Makela, S.-M., and Ailisto, H. Identifying users of portable devices from gait pattern with accelerometers. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 2, IEEE (2005), ii–973.
19. Mizell, D. Using gravity to estimate accelerometer orientation. In *2012 16th International Symposium on Wearable Computers*, IEEE Computer Society (2003), 252–252.
20. Nickel, C., Busch, C., Rangarajan, S., and Mobius, M. Using hidden markov models for accelerometer-based biometric gait recognition. In *Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on*, IEEE (2011), 58–63.
21. Nixon, M. S., Carter, J. N., Grant, M. G., Gordon, L., and Hayfron-Acquah, J. B. Automatic recognition by gait: progress and prospects. *Sensor Review 23*, 4 (2003), 323–331.
22. Nixon, M. S., Carter, J. N., Shutler, J. D., and Grant, M. G. New advances in automatic gait recognition. *Information Security Technical Report 7*, 4 (2002), 23–35.
23. Niyogi, S. A., and Adelson, E. H. Analyzing and recognizing walking figures in xyt. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, IEEE (1994), 469–474.
24. Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. Speaker verification using adapted gaussian mixture models. *Digital signal processing 10*, 1 (2000), 19–41.
25. Rong, L., Jianzhong, Z., Ming, L., and Xiangfeng, H. A wearable acceleration sensor system for gait recognition. In *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, IEEE (2007), 2654–2659.
26. Su, H., and Huang, F.-G. Human gait recognition based on motion analysis. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol. 7, IEEE (2005), 4464–4468.
27. Vildjiounaite, E., Mäkelä, S.-M., Lindholm, M., Riihimäki, R., Kyllönen, V., Mäntyjärvi, J., and Ailisto, H. Unobtrusive multimodal biometrics for ensuring privacy and information security with personal devices. In *Pervasive Computing*. Springer, 2006, 187–201.
28. Wang, L., Ning, H., Tan, T., and Hu, W. Fusion of static and dynamic body biometrics for gait recognition. *Circuits and Systems for Video Technology, IEEE Transactions on 14*, 2 (2004), 149–158.
29. Wang, L., Tan, T., Hu, W., and Ning, H. Automatic gait recognition based on statistical shape analysis. *Image Processing, IEEE Transactions on 12*, 9 (2003), 1120–1131.