

# Gait Recognition in Mobile Security

Chase R. Ottomoeller  
Division of Science and Mathematics  
University of Minnesota, Morris  
Morris, Minnesota, USA 56267  
ottom005@morris.umn.edu

## ABSTRACT

This paper discusses a new form of mobile security, emphasizing both security and usability. Having a form of security of a mobile device that is unobtrusive can make that device easier to use. Using built-in accelerometers in smart phones as a way to collect biometric data is one of these forms. In this case the biometric data is a person's gait, or walking pattern. This increases the security of the device by allowing the authentication to not be something you have or something you know, but something you are. There are two examples that this paper will compare showing two forms of gait analysis. Though both examples need further work, they both implement working security schemes allowing the phone to verify the users identity correctly.

## Keywords

mobile authentication, gait recognition, biometrics, accelerometer

## 1. INTRODUCTION

Computational power in computers continues to increase year to year. One growing concern is security, including the security of mobile devices such as smart phones. The current methods to unlock smart phones will soon be vulnerable to brute force attack (trying all possible combinations). Common security authentication methods include: pin, password, swipe lock, token, and fingerprint. Most of these methods are either something a person has (card, token, key) or something a person knows (password). They all require a physical action to unlock the phone. A new form of mobile security involves biometrics. Using biometrics allows to identify a person based on their characteristics (fingerprint) and not by what they have or what they know. One form of biometrics being explored uses a person's gait (walking pattern).

This paper analyzes two different approaches for gait extraction and analysis. Each approach uses a general process to extract and analyse the data which includes: Pre-

## Algorithm Process

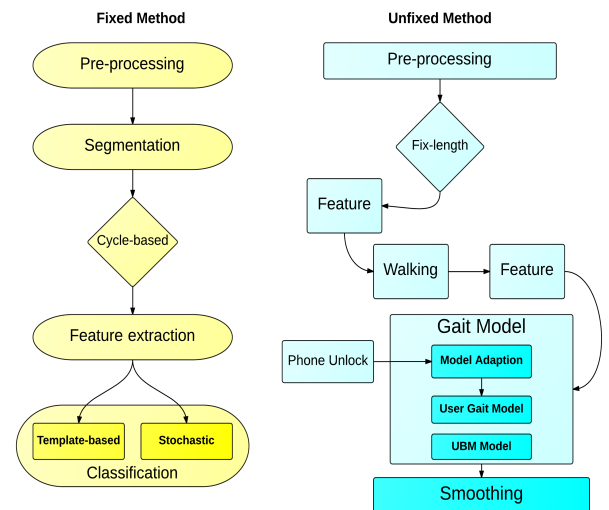


Figure 1: Different Approaches

processing, Feature Extraction, and Gait Analysis. *Pre-processing* is the process of taking the raw accelerometer data and preparing it for feature extraction. *Feature Extraction* is taking the pre-processed biometric data, and locating distinctive characteristics within that data. *Gait Analysis* is taking the features extracted from the data, and comparing them against a predefined template. This comparison is what determines if the current user is an imposter or not.

In the following sections, I will be comparing two approaches of gait recognition using accelerometers from a mobile device, such as a smart phone. These two approaches use different placement of the mobile device on the body. This difference in placement impacts the techniques that are used to recognize and analyze gait patterns. One approach has the smart phone in a fixed location on the waist. I will refer to this method as the *fixed* method. The other method has the smart phone in a more natural position on the body such as in a pocket. This method will be referred to as the *unfixed* method. Though both methods use a complex approach shown in figure 1, I will simplify comparisons between these methods. I will group them into three categories: Pre-Processing, Feature Extraction, and Gait Analysis.

## 2. BACKGROUND

Biometrics can be used as a new form of security that provides unique identification to a mobile user. Instead of relying on a password or physical key, a person can gain access to their device by a physical characteristic specific to them. Biometrics can be split into two categories, physiological and behavioural. Physiological biometrics include: Finger Scan, Facial Scan, Iris Scan, and DNA Matching. Behavioural biometrics include: Voice Recognition, Keystroke Scan, and Gait Recognition. In this paper, I will focus on gait recognition. The term gait refers to the walking pattern of a person. A person's walking pattern is cyclic in nature and may be composed of many gait cycles, where each gait cycle consists of at least two steps. [3]

In mobile security, when gait recognition is used, gait data is collected via the built-in accelerometer of the smart phone. The smart phone will use the collected data to analyze a person's walking pattern. If the phone can determine that the current user should have access, the phone will be unlocked. Otherwise the phone will switch to a different authentication method such as pin access. This method increases the security of the mobile device, because the form of unlocking the device is now a person characteristic (a biometric) and not something that is likely stolen or forgotten. Also, unlike other biometrics, such as a fingerprint that can be copied, an imposter would have trouble replicating a gait pattern.

## 3. PRE-PROCESSING

Once data is gathered, it needs to be pre-processed into a usable form which is done by separating the data into pieces. In the following sections I will explain the fixed and unfixed methods of pre-processing.

### 3.1 Fixed Method Pre-processing

With the phone on the waist in a fixed position, uniform sets of data called *walks* were collected. One walk is defined by walking a measured distance down a hallway. The experiment was conducted so that one set of data contained two "walks". In other words, one set of data was walking down a hallway, once down and once back. After extracted, the raw gait data is pre-processed using two steps: *linear interpolation* and *zero normalization*.

#### 3.1.1 Linear Interpolation

Since the built-in accelerometer does not output data in equal intervals, the data needs to be reshaped into equal intervals. This is done by applying *linear interpolation*. To avoid losing data during this change, up-sampling can be applied. Up-sampling avoids data loss by producing "an approximation of the sequence that would have been obtained by sampling the signal at a higher rate" [5]. Once the data has been reshaped, the acceleration measurements have to be adjusted to account for unstable accelerometer readings.

#### 3.1.2 Zero Normalization

The mobile device's accelerometer is used to measure 3 axes (X, Y, Z). The only axis needed, and the only axis affected by gravity is the X axis. Since the acceleration for the Y and Z axis are not stable over time, and they are not affected by earth's gravity, the acceleration data for the Y and Z axis have to be *zero normalized*. In other words the data for the Y and Z axes have to be set to zero. This is done by

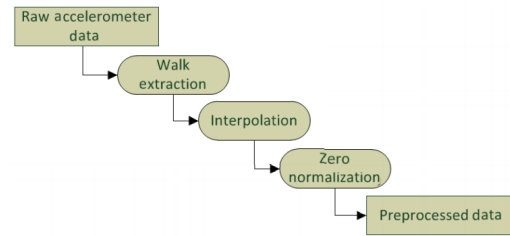


Figure 2: Fixed Method Pre-processing Step

simply subtracting their total acceleration from their average acceleration. An overview of the stages explained above are shown in figure 2.

### 3.2 Un-fixed Method Pre-processing

The previous method used a phone in a fixed location. This method expands from having the phone on the waist, to having the phone in a unfixed location, such as a pocket. Also, this method does not use a set length, such as a hallway, making it more flexible. To mark each walk this method uses a process called *framing*. Framing separates the data into sets of equal time. Once the data is segmented into frames, it is then projected onto a global coordinate system through a process known as *projection*.

#### 3.2.1 Framing

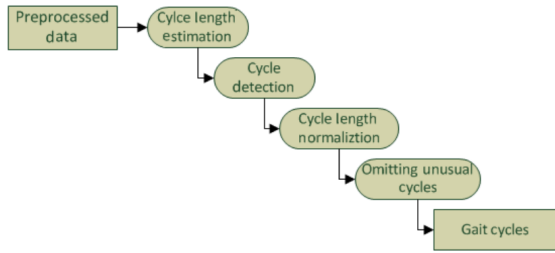
During Framing, sensor data is segmented into uniform frames for feature extraction and feature classification. This is similar to the pre-processing of the data for the fixed method. The fixed method had its data separated into "walks". This data, not having the separation of "walks", is separated into sections of 512 equal samples in length. This is done by separating sections every 5.12 seconds. They state the reason for 512 samples was "to balance between estimation accuracy and latency." In other words, this was to balance between the accuracy of matching a user and the time it took to compute the match. Also, not all frames move onto feature extraction. Frames that are below a chosen threshold, representing no movement, are dropped.

#### 3.2.2 Projection

Each sample within a given frame contains an X, Y, and Z coordinate. Each coordinate is then projected into a vertical and horizontal variable. Once each sample is projected, the direction of gravity is determined by using a mean filter. Unlike the previous example, this process does not assume a fixed location or orientation of the mobile device. In order for the device to record accurate data from the accelerometer, the device's orientation (up, down, left, and right) must be accurate. If there is a significant change in the gravity variable then the orientation of the device has changed. This means the corresponding frame will be dropped and the horizontal and vertical axes will be adjusted accordingly.

### 3.3 Comparison of Pre-processing Approaches

Both the fixed and unfixed method split the data into equal intervals as well as modify the data so that it is in a useful form. The reason for the differences between the pre-processing methods stems from the differences in phone



**Figure 3: Feature Extraction**

placement. The data collected in the fixed method is already more organized than the data from the unfixed method. This is because the unfixed method has to account for multiple axes as well as data unrelated to gait. If the fixed method were to be implemented in more of a real life scenario then these methods become even more similar.

## 4. FEATURE EXTRACTION

*Feature Extraction* is extracting a set of data from a given frame to detect patterns of walking. Another way to define feature extraction is the process of extracting gait cycles from the data.

### 4.1 Fixed Method Feature Extraction

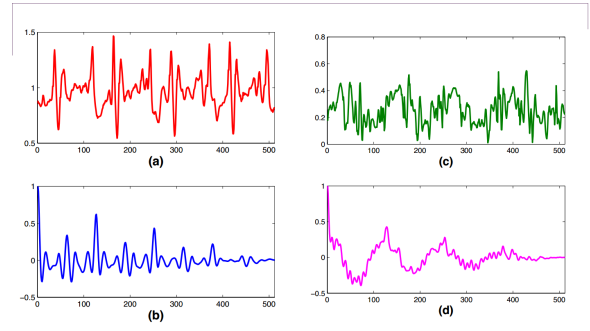
To extract gait cycles from the pre-processed data, the fixed method uses the following steps in order, as seen in Figure 3: Cycle length estimation, Cycle detection, Cycle length normalization, and Omitting unusual cycles.

#### 4.1.1 Cycle Length Estimation

Once the raw gait data is pre-processed, the next step is extracting the gait cycles. The first step in extraction is to automatically detect gait cycles in the walk. This detection is done by estimating the cycle length, done by computing the *salience vectors*. For this instance, a salience vector is the distance measurement from one chosen data point of a cycle to another chosen data point. A minimum and maximum salience vector is computed for each data point for a given walk. Each minimum salience vector is computed by counting the data values that are between the current data value, and the following smaller value in the walk vector. The same process is used for maximum salience vectors, but instead of calculating the distance to the next smallest value, it is calculating the distance to the next largest value. The number of values counted is the distance measurement for that data point. The minimum salience vectors with the greatest distance measurements are considered cycle starts. Sometimes there is more than one minimum salience vector, making the start of a cycle unclear. Maximum salience vectors help to verify the start of cycles, because they occur right after the cycle start.

#### 4.1.2 Cycle Detection

Once both minimum and maximum salience vectors are computed, they can be used to detect individual cycles. Figure 4 shows a set of gait cycles in black, and the salience vector distance measurement in grey. The start of each cycle is based on the minimum salience vector. Around times 750,



**Figure 5: (a) and (b) represent horizontal and vertical movement recorded by the accelerometer. (c) and (d) represent the horizontal and vertical movement after the data had been normalized.**

1150, 1450, and 1650 are spikes where the distance measurement for a data point is larger than the rest. These spikes are the minima calculated from the minimum salient vector. These minima mark the start of gait cycles. Also, if there are some cycles that are much longer than others, these minimum and maximum points are used to determine more cycle starts. This will split long cycles into smaller ones, giving more uniformity to the data cycles.

#### 4.1.3 Cycle Length Normalization

Once each start of a cycle is determined, the cycle distance is the distance from the start of one cycle to the start of the following cycle. Each cycle does not have the same length. Like in 3.1.1, by using linear interpolation the determined cycles are normalized to a set length. Equal cycle lengths are later required for some feature extraction techniques.

#### 4.1.4 Omitting Unusual Cycles

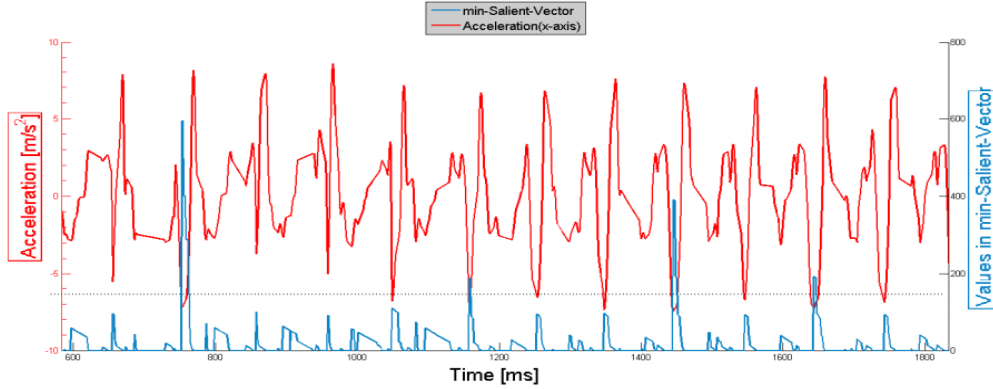
Some cycles, compared to the majority, are considered unusual. These could be caused by any motion or non-motion that does not correspond to walking. Such patterns are detected by “computing the distance using *Dynamic Time Warping* (DTW).” [2] DTW is “an algorithm for measuring similarity between two temporal sequences which may vary in time or speed.” [4] This means that cycles with an acceleration of half or less of the other cycles will be removed. Since this method positions the phone on the waist, and the data was collected in a controlled environment, there will be fewer unusual cycles removed in the fixed method than in the unfixed method. This is further explained in the following section.

## 4.2 Unfixed Method Feature Extraction

A different form of extraction used by [1] is separated into three stages: Feature Extraction I, Walking Detection, and Feature Extraction II. Feature Extraction I is used to get rid of any unwanted and incorrect data, such as data collected when the phone changes position. Walking Detection is used to specify what kind of motion the current data set represents, such as biking, running, or walking. Feature Extraction II pulls the gait data from the data set to be analysed.

### 4.2.1 Feature Extraction I

The first stage of the feature extraction is used to tell



**Figure 4: Minimum Salient Vectors**

whether the data represents walking or if the data represents some other activity. Differentiating between walking and non-walking helps further classify the data in 4.2.2.

Unlike the fixed method with the very specific context for what is and isn't walking, the unfixed method has data representing multiple types of motion. Motion such as riding a bike, or riding in a vehicle. For example, if the mobile device was collecting data when it was in a car going down the road, that data should not be used to extract features from. This is because this data is not gait data. This also holds true for collecting data on someone who is standing still. The data collected would not represent the data needed for gait recognition.

Telling whether the data collected is of someone walking or not is done by collecting vertical and horizontal features to produce motion patterns. Six time-domain features are computed for both directions. These six features are :mean, variance, skewness, kurtosis, energy, and mean-crossing rate. These features are based on spectrum analysis. In other words, doing "different activities have different energy distributions over the frequency spectrum." [1] Walking can be measured around 1-2Hz while driving in a car will output a higher frequency band. These features are used for walking detection in the following step.

#### 4.2.2 Walking Detection

Using the features calculated from Feature Extraction I, classification can be done using a decision tree. There are three activities that the data can be classified as: walking, non-walking, and random movements. Non-walking motion is running, biking, or as used in the above example, moving in a vehicle. Random motion is a motion such as turning or skipping. A Markov Model smoother is applied to the new set of data to get rid of any outliers.

#### 4.2.3 Feature Extraction II

This section is still a little vague, but it might have to be that way to save space.

Feature extraction II is done only on the data collected that is determined to be gait data. In this stage, more relevant features are extracted for analysis. The first analysis to be done is the compressed sub-band cepstral coefficients

(CSCC). This is done in the following three steps. 1) The energy spectrum is computed using the *fast fourier transform* (FFT) spectrum. The FFT is an algorithm that computes the Discrete fourier transform (DFT) and its inverse. 2) The study then maps the energy spectrum into 26 bands, using triangular overlapping bands, summing up the energy in each band. 3) The processing then takes the discrete cosine transform of the sub-band energy to form a 12-dimension vector representation. This, in all, summarizes the fundamental frequency of the movements and the higher frequency vibrations in the data. [3]

### 4.3 Comparing Feature Extraction Methods

Both of these methods of feature extraction try to determine what parts of the data represent a walk cycle. The fixed method is simpler because it does not have to worry about different variables affecting the data. Though this method is simpler, it has the stipulation of the mobile device being clipped to your waist. The fixed method also only uses data from the controlled environment of walking down a hallway. The non-fixed method has to worry about more variables affecting the data, but implements a real life situation of just having the mobile device in a pocket.

## 5. GAIT CLASSIFICATION

This last step is taking the extracted features and verifying that those features match a stored set of features; the stored set of features being the users. The fixed method uses Support Vector Machines(SVM's). The basic idea of SVM's is taking dimensional data and separating it into two classes. This involves the use of a Gaussian kernel defined in the following section. The unfixed method uses a Gaussian Mixture Model - Universal Background Model(GMM-UBM) framework to verify an individual's gait. The main concept with this technique is to use scoring. Verification of a user's identity is done by comparing the likelihood score from a user's gait pattern to the universal background model. The universal background model represents human gait patterns in general.

### 5.1 Fixed Method Gait Classification

SVMs are used to classify gait data in the fixed method. SVMs are often used for biometric recognition, where imposter and genuine data have to be determined. First the SVM finds a *hyperplane*. A hyperplane is a subspace that is one dimension less than its “ambient space”. For example, there could be a set of data represented by 3 axis (X,Y,Z). The hyperplane would be the same data plotted on a 2 axis space. So, the SVM will take the data, and find a hyperplane that linearly separates  $\mathcal{D}$  dimensional data into two classes. The dimensional data has to be linearly separable for the SVM to work. Since the gait data is not linearly separable, a “kernel induced feature space” [2] is introduced in the SVM. A kernel function maps non linearly separable data to a high-dimensional space. Once mapped to a high-dimensional space the data is linearly separable and therefore can be used by the SVM to find a hyperplane. This means that depending on the kernel function you use, the better classification accuracy it will produce. There are kernel functions that would require the use of fixed length gait cycles. Since gait cycles are not usually of a fixed-length, a different kernel is used. By using the DTW distance, the problem of fix-length input feature vectors can be solved as well as more accurate finding of similarity in the data. Two different approaches using SVM and DTW can be used to classify gait cycles: Pre-computed data matrix and Pre-computed kernel.

### 5.1.1 Pre-computed data matrix

Gait cycles are represented by the DTW distance. The DTW distance is the distance between two gait cycles. The benefit of the DTW distance is that gait cycles with different lengths can be compared. A DTW matrix is then computed by taking the sample DTW distance and compared to all other samples. This matrix is needed as an input for the SVM.

### 5.1.2 Pre-computed Kernel

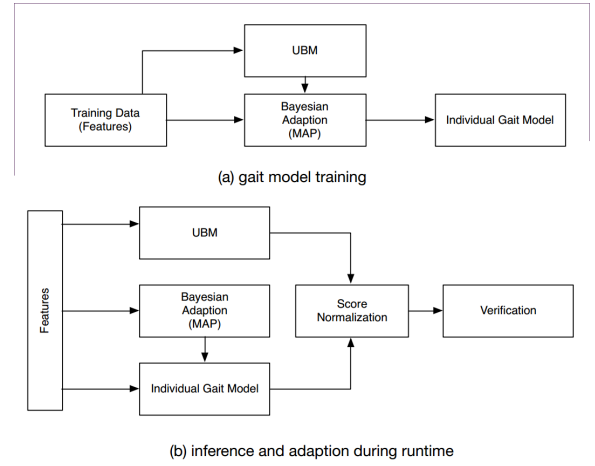
Using Gaussian Dynamic Time Warping (GDTW) kernel equation (1) as the kernel matrix also allows to classify the gait data. Though this kernel matrix is a symmetric matrix, it is not guaranteed that it has all positive eigenvalues.

$$K(x, z) = \exp(-\gamma \|DTW(x, z)\|^2) \quad (1)$$

## 5.2 Unfixed Method Gait Classification

The framework used for classifying gait cycles is a Gaussian Mixture Model-Universal Background Model (GMM-UBM). The algorithm can be split into three primary sections: UBM training, user gait model generation, runtime inference and adaptation.

The UBM is a large, Universal background GMM that is trained from a large source of data. This means a UBM represents the general walk pattern of humans. The UBM is defined as  $\lambda(\omega, \mu, \Sigma)$  where  $\omega$  represents the mixture weight,  $\mu$  represents the mean, and  $\Sigma$  represents the covariance matrix. In other words,  $\omega$  is the prior distribution of walking patterns while  $\mu$  and  $\Sigma$  are different walking patterns given by the population in different conditions. For efficiency the *covariance matrix* is used. A covariance matrix “generalizes the notion of variance to multiple dimensions.” Also, to avoid over fitting the training data, a standard expectation maximization (EM) algorithm is applied. The EM is what trains the UBM models. For example, if given a collection of training vectors, the EM estimates the parameters to most



**Figure 6: Algorithm work-flow for (a) generate user gait model by MAP adaptation (b) runtime inference and Individual model adaptation**

likely to occur. The EM algorithm does this through iteration. Each time the algorithm iterates it refines the GMM parameters, increasing the likelihood of the estimated model for the given feature vectors.

The individual user’s gait model is a GMM, but instead of using EM training, a Bayesian adaptation is used. This is simply relating the odds of one event to the odds of another. The use of the Bayesian adaptation is both used for performance as well as for its ability to learn from the data. The Maximum-a-Posteriori (MAP) is the adaptation used. The MAP adaptation adjusts the Gaussian components and mixture weight to personalize the UBM model.

It is during runtime when the extracted features are compared against the user gait model and the UBM creating a score. High scores result from the user model being higher than the UBM model and visa versa for low scores. The score are computed as

$$\Delta = \log(p(X|\lambda_{user})) - \log(p(X|\lambda_{ubm})) \quad (2)$$

In this equation  $\Delta$  is equal to the user’s score,  $X$  is the feature vector, and  $\lambda$  represents the type of model the feature vector is being compared against. MAP is also used at runtime to learn new gait patterns from the user. It does this from false negatives given to the device by an alternative form of authentication.

## 6. PUTTING IT ALL TOGETHER

I have compared two forms of using gait patterns as a form of mobile security. One experiment used a fixed location for the device while the other used multiple locations. Both use different methods of extracting the gait data as well as well as analysing the data, but each method does use off the shelf Android phones. This allows for each method to use similar, if not the same accelerometers, giving both methods access to the same data. Neither experiment is perfect yet, but they both show promise in using walking patterns as a form of mobile security.

### 6.1 Fixed Method Experiment



The fixed method did two experiment setups: Template-based classification and Machine learning based classification. The data set was made from 51 subjects (39 males, 12 females) that walked down a 18.5 meter long corridor.

### 6.1.1 Template-based classification

For the template-based classification a interpolation of 100 Hz was used. Also, for this experiment a Piecewise Linear Approximation (PLA) was used before the cycle length estimation step. A PLA is a function that is used to approximate a curve connecting points along the curve with straight lines (interpolating linearly). Also, the PLA uses the Sliding Window And Bottom-up (SWAB) approach. The SWAB approach starts at the leftmost detected segment, applies the PLA function, and moves onto the new set of data using the sliding window approach. It is this PLA representation of the walks that will undergo the remaining steps. From the final processed data, the best cycle with the lowest DTW distance is chosen. This cycle is known as the feature cycle. the remaining cycles are known as probe cycles. Once all the reference and probe cycles are computed two classes are made: genuine and impostor. These classes are made by comparing the two classes against one another in respect to their DTW distances. If at least 50% of one walk votes for the genuine class then accept, else reject. The accuracy of this method is measured by the Equal Error Rate (EER). This is the rate at which both the acceptance and rejection errors are equal. For this template-based classification the EER was 22.49% for same-day, 29.4% for different-day and 33.3% for mixed days.

### 6.1.2 Machine learning based classification

The machine learning classification uses the normal approach to processing the data along with the added approaches described in section 5.1. Once the gait cycles are extracted they are separated, 80% go into a training data set while 20% go into a testing data set. The DTW distance matrix is computed using the training data set.

## 6.2 UnFixed Method Experiment

The unfixed method experimented using 3 data sets all using android phones. The first data set included 49 people, 19 female and 28 male. These subjects performed various tasks that included: standing, sitting, walking, biking, running, driving, and random movements. The second data set was made up of 12 people, 5 female and 7 male. These subjects did tests in a more controlled environment with different phone placements, walking distances, and walking speeds. The third data set was to collect data in a real-world setting. This involved 8 people that collected data for 5 hours each.

## 7. CONCLUSION

I have compared two forms of gait recognition methods using very similar strategies to create an unobtrusive security process for mobile phones using stock android phones. The difference between these two methods are: placement of the mobile device, and their different implementations of the strategies to which their algorithms were built. Though the fixed method was more geared to a controlled setting and the unfixed method was geared to more real world implementations, they both show promise for future work.

The fixed method used a new method to extract gait cycles using Piecewise Linear Approximation. The fixed method also used two new approaches using Support Vector Machines and Dynamic Time Warping to classify gait cycles. Though only gait cycles of the same length were used, DTW allows for future work with gait cycles of variable lengths. Future work will involve recording at higher sampling rate (varying frequencies) as well as placing the phone in more realistic positions on the body.

The unfixed method used a different approach when dealing with gait data. There was the use of MAP and Gaussian Mixture Models, customizing the gait model. Future work will involve training the UBM better by sampling from a larger data set. There are also trade-offs to be made balancing performance over usability. Using a longer smoothing window could help improve accuracy but will take longer to verify the user.

## 8. ACKNOWLEDGMENTS

I would like to thank Kristin Lamberty, and Elena Machkasova for their advice and feedback as well as Melissa Helgeson for additional feedback and proofreading.

## 9. REFERENCES

- [1] H. Lu, J. Huang, T. Saha, and L. Nachman. Unobtrusive gait verification for mobile phones. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers, ISWC '14*, pages 91–98, New York, NY, USA, 2014. ACM.
- [2] M. Muaaz and R. Mayrhofer. An analysis of different approaches to gait recognition using cell phone based accelerometers. In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, MoMM '13, pages 293:293–293:300, New York, NY, USA, 2013. ACM.
- [3] M. Sujithra and G. Padmavathi. Next generation biometric security system: An approach for mobile device security. In *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, CCSEIT '12*, pages 377–381, New York, NY, USA, 2012. ACM.
- [4] Wikipedia. Dynamic time warping — wikipedia, the free encyclopedia, 2014. [Online; accessed 25-October-2014].
- [5] Wikipedia. Upsampling — wikipedia, the free encyclopedia, 2014. [Online; accessed 25-October-2014].