Report

Abstract

In this paper, we present an approach for identification based on biometric gait using acceleration sensor - called accelerometer. In contrast to preceding works, acceleration data are acquired from built-in sensor in mobile phone placed at the trouser pocket position. Data are then analyzed in time domain. In time domain, gait templates are extracted and Dynamic Time Warping (DTW) is used to evaluate the similarity score. With the sample data of 10 people, accuracy score of 80% is achieved.

Introduction

The explosion of mobility nowadays is setting a new standard for information technology industry. Mobile devices such as smart-phones and tablets more and more become popular, and hence, making people increasingly depend on them for their superior functionality. Such devices are commonly used for storage and retrieval of information like e-commerce and m-banking. However, through smart use of built in sensors in mobile phones, multiple other tasks can be performed. One such example is of identification of individuals using the accelerometer & gyrometer sensor in the smartphones. Currently, the most common methods to identify individuals in any area of surveillance is through cameras, face recognition or biometric thumbprint which require a pause in an otherwise crowded locations hence causing inconvenience in frequent use. However, these limitations can be solved using approaches based on another biometric parameter of humans i.e. Human Gait.

Human gait has been introduced as a particular style and manner of moving human feet. In a more detail level view, the mechanism of human gait involves synchronization between the skeletal, neurological and muscular system of human body. Therefore, gait characteristics will vary from people to people. Gait recognition has been studied as a behavioral biometric for decades. Its techniques could be typically divided into 3 categories: Machine Vision Technology (WVT), Floor Sensor Technology (FST), and Wearable Sensor Technology (WST). WST is recognized as the most approachable and newest of all. Sensors in WST are attached to human body in various positions, such as pockets, waist or shoes to record physical motions. WST takes advantage of mobile devices' sensing capabilities including GPS, accelerometer, and gyroscope sensor, etc. Thus, it will provide developers an edge over improving various techniques in identification.

In this project, we will try identification method based on WST using an integrated accelerometer on mobile phone. Moreover, because segmentation of gait cycles is the most important process in any gait analysis, we also provide a novel algorithm to partition gait cycles when the device is placed at trouser pocket.

Objectives

The objectives of this project include the following steps.

- Data Acquisition
- Noise Filtering
- Normalization
- Gait Cycle Partition
- Feature Extraction
- Feature Matching
- Results

Working Tools

The method consisted of these major software components:

1. Android application
2. Central web server
3. MATLAB scripts

1. Android Application

The android application was responsible for two major tasks:

a. Accelerometer data capture
b. Geocaching

Our development and test device is Samsung Galaxy S4. On this device we captured the accelerometer data using our app with a sampling rate of ~100Hz. Android fires the accelerometer sample event at a fluctuating rate but since our pattern matching algorithm is based upon dynamic time warping, the consistency of this sampling rate is not very critical. The sampling rate of ~100 Hz is sufficiently larger than our intended signal which is in the range of 0.5 Hz to 5 Hz.

The android application also has a geocache set at the SEECS entrance. Android OS monitors the phone’s location using GPS and whenever the location is within 200m of the app’s registered geocache, it notifies the app using a broadcast. The app receives
this broadcast and starts capturing the three accelerometer along with the three gyroscope

channels. It stores all of this data in local storage until it is able to connect with the NUST intranet. When it detects a connection with the local network, it looks for a web app at the dedicated web server's IP and uploads the recorded data to it.

2. Web Server

The central web server (in our case a normal laptop with the intranet's DHCP set IP) is running a simple PHP service. This service allows the android application to connect to it and upload a file containing six channels of accelerometer + gyroscope data. Whenever an upload is finished, the web server creates a local file with all of this data and triggers MATLAB's entry script on it. A single server instance is capable of servicing several application connections simultaneously. Although, for efficient load balancing, multiple parallel services with reverse proxy is the suitable and intended model.

3. MATLAB Scripts

3.1 Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm for measuring the similarity between two temporal sequences which may vary in time or speed. It aligns the two sequences/signals under observation and assigns a similarity score by calculating the least distance between them. It warps the time scale of one signal by stretching or shrinking in order to make it as best matching to the second signal as possible. Hence it is a nonlinear matching algorithm. The steps for calculating DTW distance between two sequences (as published here) is:

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
   DTW := array [0..n, 0..m]

   for i := 1 to n
       DTW[i, 0] := infinity
   for i := 1 to m
       DTW[0, i] := infinity
   DTW[0, 0] := 0

   for i := 1 to n
       for j := 1 to m
           cost := d(s[i], t[j])
           DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                       DTW[i  , j-1],    // deletion
                                       DTW[i-1, j-1])    // match

   return DTW[n, m]
}
```

3.2 Butterworth Filter

The Butterworth filter is a type of signal processing filter designed to have as flat a frequency response as possible in the passband. It is available in Matlab in form of a built-in function and can be used readily by using the keyword:

[B,A] = butter(,);

It removes the noise from the data and provides a smooth signal with singular peaks successively.

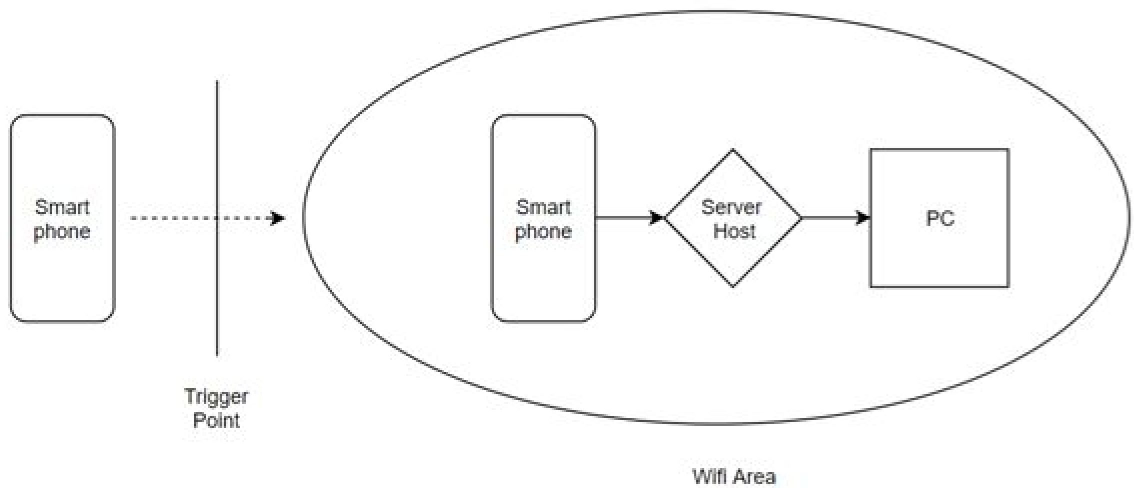
3.3 Peak-Finder

To address the location of gait cycles in the signal, points at the end of each gait cycle need to be located. To find such indexes, a built-in function was used in Matlab using the keyword:

[peaks,locs] = findpeaks(Signal);

Working

The general idea of seamless detection of an individual is to integrate geocaching and server connection through wifi in such a way that upon entering a tagged location in the Android app, the smartphone will automatically start reading the data from the accelerometer and gyro sensor. Readings of few seconds will be transferred to an automated host via the wifi connection and file will be received on a personal computer which runs the script to match the gait features. Templates for known individuals will be tested against the new data and those with matching results will be recognized. In this way, seamless detection of individuals can be taken out in institutions requiring security.



The main points of working process will be discussed here.

1. Data Acquisition

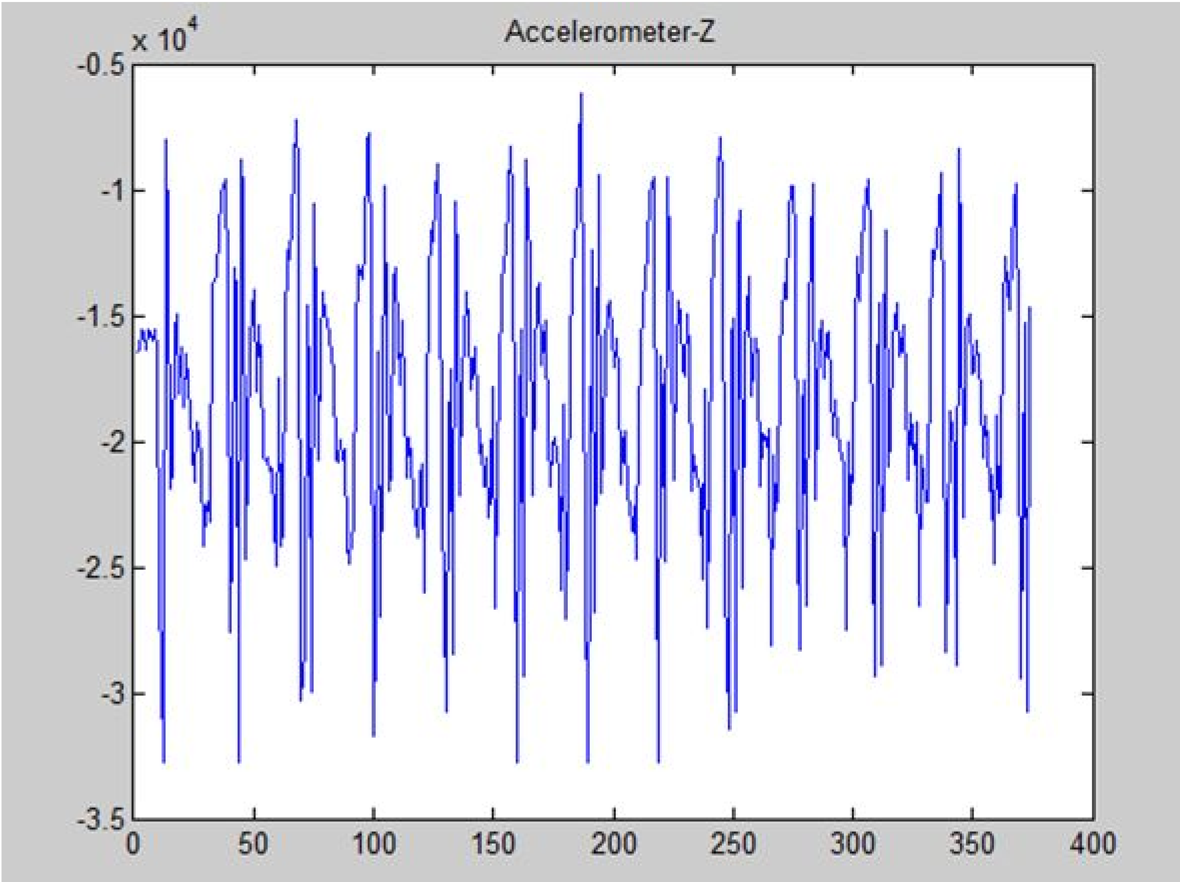
The proposed method for keeping smartphone in the pocket was decided after studying multiple positions. Particular attention was paid to the position of accelerometer. The mobile phone was vertically fixed at the pocket location as shown in figure. The position turned out to be the most appropriate for the mobile phone bearer as well for data accuracy.

Next comes the Android app to carry out the data acquisition process. The Android app is designed in such a way that background process is running all the time in the phone noting geolocation of an individual and upon geocache trigger the data-logging is carried out by reading sensor values for a short amount of time and are saved in a file. The data is in form of Z- Acceleration sample values in number of 30 Samples/sec.

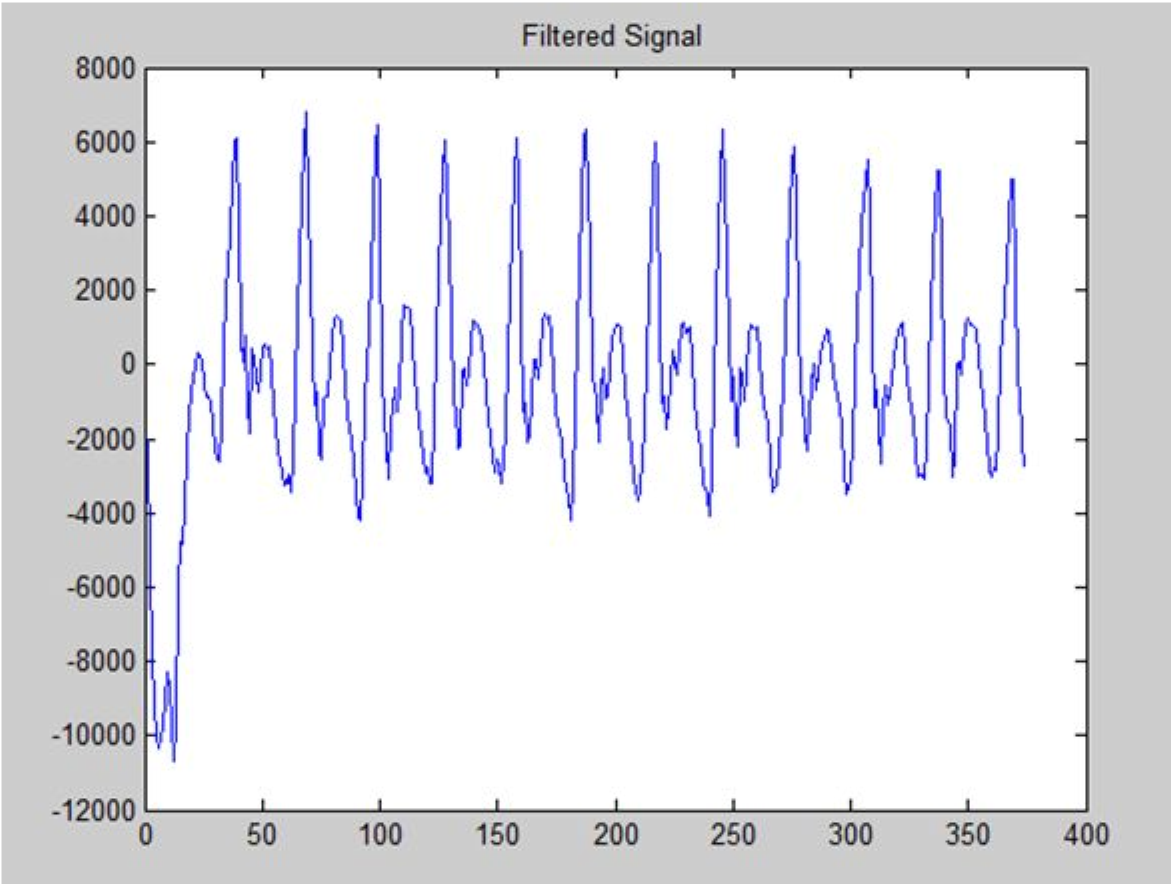
Each file upon creation is immediately sent to the server using the known wifi connection which is detected by the PC already connected already connected to the server. A script on PC organizes files for each individual separately to avoid mixing of files.

2. Noise Filtering

Noise elimination is carried out using Butterworth Filter which removes the high frequency noise bands and produces smooth signals. The resultant signal is easier to handle in feature extraction from before. As shown in figures:



Accelerometer-Z
x 10^4
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
0
50
100
150
200
250
300
350
400



2. Normalization

Normalizing signals is a very simple process in which each signal is limited to range of [-1,1]. The logic used to obtain this is given as follows:

```
if (abs(max(filtered))>abs(min(filtered)))
nfSignal = filtered./(max(filtered));
else
nfSignal = filtered./abs(min(filtered));
```

3. Cycle Detection

Detecting gait cycles in an otherwise long signal is a tricky process and required detailed study of the gait signals to find a differentiating element. Upon research, it was found that at the end of each gait cycle a surge or peak occurred in Z-Axis Accelerometer values of very high value which could be differentiated from other points. All that remained was to detect the locations of peaks in the signal and the indices were used to extract individual cycles in the complete signal.

Peak detection was carried by a built-in Matlab function which had the mathematical procedure as follows:

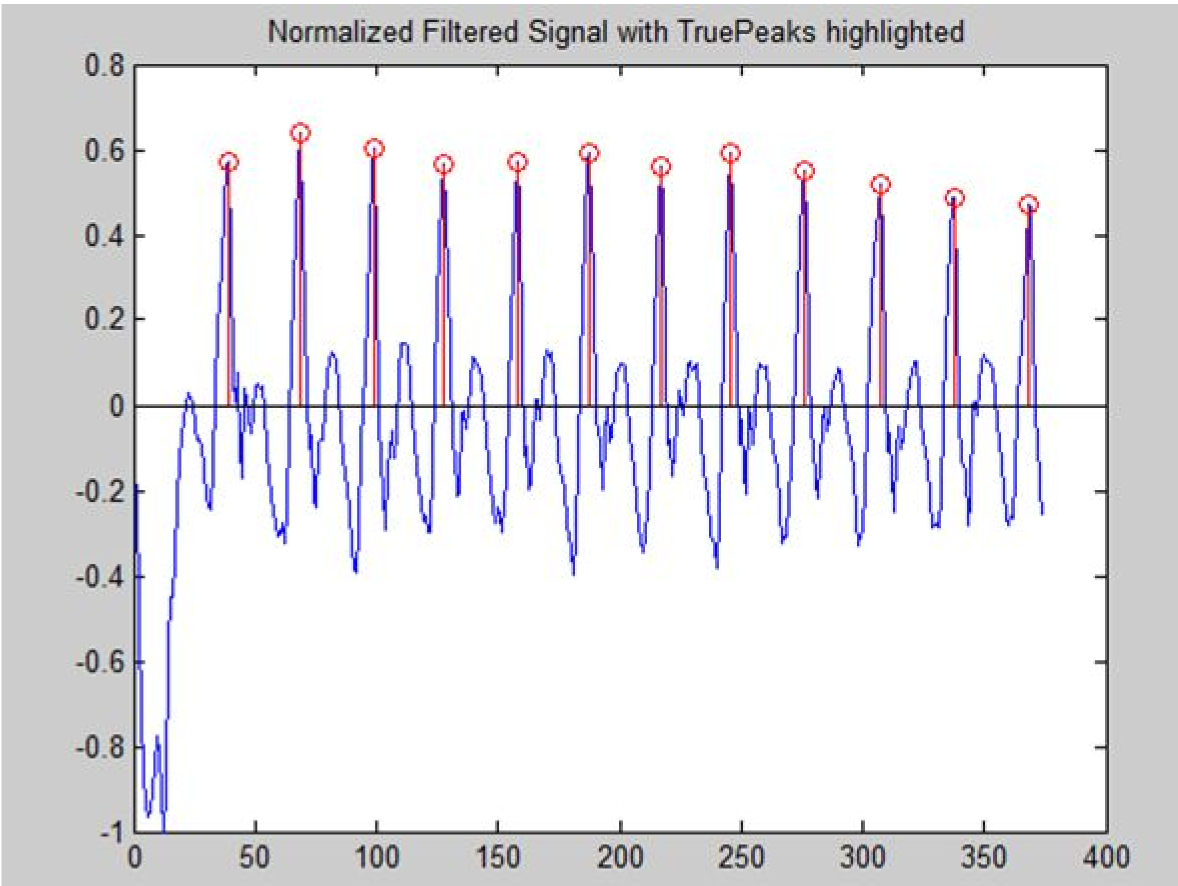
$$P = \{ d_i \mid d_i > d_{i+1} \wedge d_i > d_{i-1} \} \text{ with } i \in [1 \dots n]$$

Identification of peaks was done using simple logic of checking values of indices before and after each point on signal. However this would raise the issue of multiple unwanted peaks that did not have a very high value as opposed to peaks at end of each gait cycle. To counter this issue, a threshold was defined using the formula:

$$T = \mu + k\sigma$$

$$R = \{ d_i \in P \mid d_i \geq T \}$$

This threshold was used to differentiate gait cycle peaks from unwanted peaks.

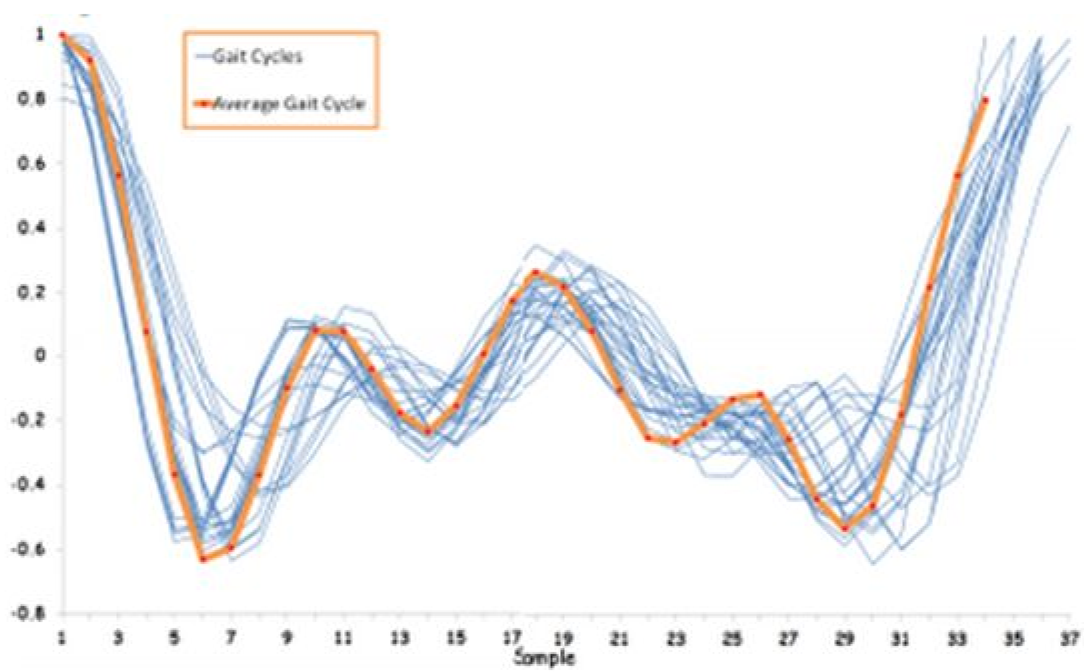


4. Feature Extraction

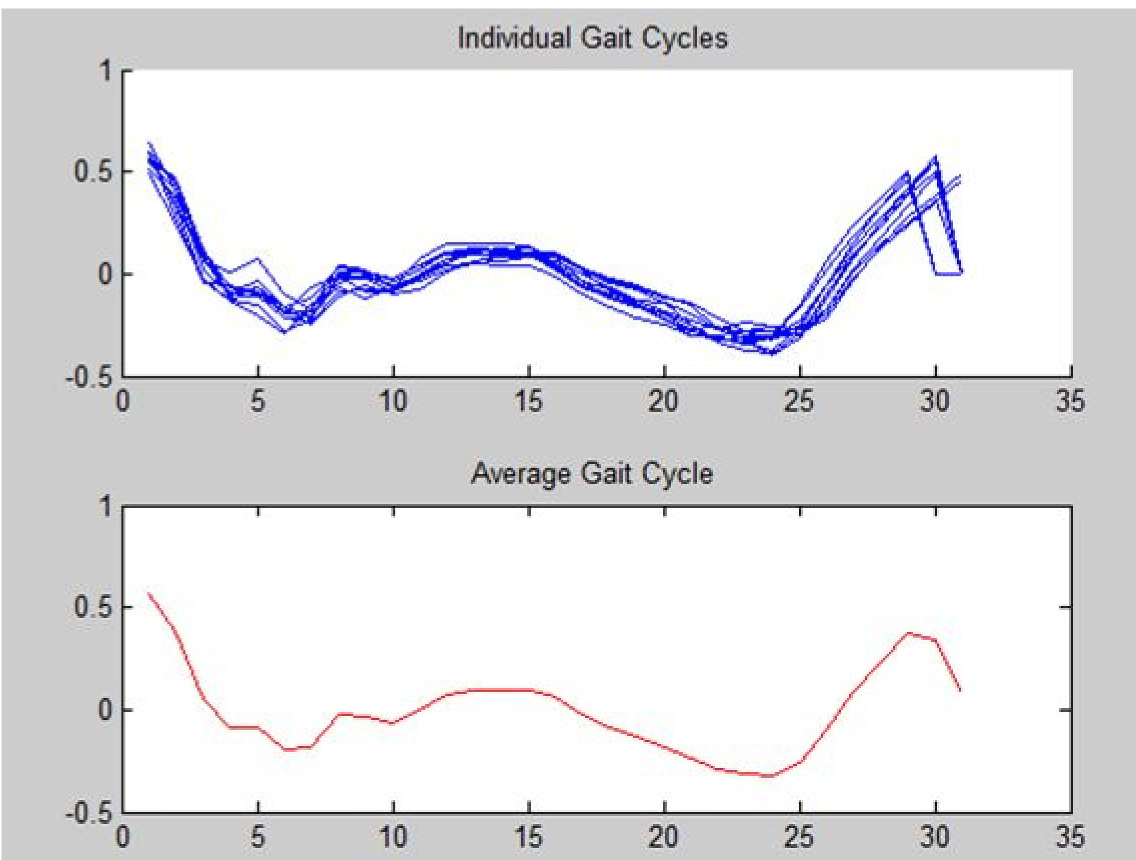
Determining unique features in any data requires immense amount of testing. In case of this data, it was rather simple. The feature extraction from gait signal was achieved by calculating average gait cycle.

A cycle is called an average cycle if it is the most similar to all other cycles

The calculation of average gait cycle was done using Dynamic-Time Warping algorithm which is mentioned earlier. Gait cycles were fed individually to the algorithm and it determined the average gait cycle. Resultantly, required unique feature was extracted.



Gait Cycles
Average Gait Cycle
Sample



Individual Gait Cycles
Average Gait Cycle

4. Feature Matching

Using DTW algorithm once again, newly acquired average gait cycles were tested against templates of individuals already saved before testing the identification method. The difference between the test and template list was saved as a score.

Minimal score determines the template of the individual and identification is matched.

However, threshold was determined for all user scores below which minimal value rendered zero number of matching and the user was unidentified. The threshold value varies upon the number of template samples taken.