Continuous Authentication and Gait Analysis

By Ben Baxter

1. Introduction to the Problem
   1. Continuous Authentication is a new and rapidly developing field. The current school of thought for simple authentication relies on biometrics to produce static or one time authentication of users. Research in continuous authentication mainly centers around using behavioral methods to constantly check and verify the user while they are interacting with the system. While this field is promising it still requires a large amount of training time before the method becomes effective as the user must be quantified and any false positives or negative manually eliminated to create a proper profile.
   2. Gait Analysis is an even newer field and because of the limitations in the initial technology used to capture it, cameras and foot sensors, it has not been able to gain widespread acceptance as an alternative to current methods. The time and cost to set the system up either at the point of interface would be counterproductive and it’s use as a point of entry based single authentication eliminates most of its advantages. Therefore, its current most prolific use is in diagnosing patients for a select few diseases that can be identified by a person’s gait.
   3. To bridge the gap between these two problems the system would have to be cost effect and non-intrusive. It would also have to be self-learning and correcting. Lastly it would need to be contained to the individual and not require the implementation of infrastructure to support it.
2. Literature Review

Literary Review of Continuous Authentication and Gait Analysis

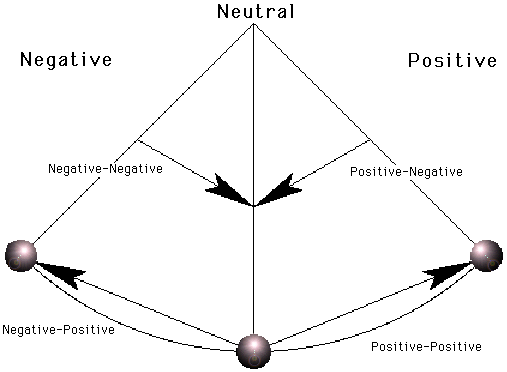
This paper for obvious reasons will be split into two sections. Both fields covered in this paper are so expansive that in the interest of clarity a few assumptions will be made about the scope of this analysis to reduce the content to a reasonable size. First for the gait analysis the focus will be on gait analysis done on arm swing alone and not full body. Second for continuous authentication only direct biometric authentication will be considered for this paper as other forms such as behavioral and recognition authentication do not provide any insight into the unique issues this study faces.

Continuous authentication is a new and constantly evolving field. Examples of current work in this field are keyboard, virtual keyboard, behavioral, gestural, temporal, biometric, recognition and tactile authentication. These are by no means the only avenues being pursued but they are the major one with the most supporting research. The current leading field in continuous authentication is based on behavioral biometrics ([a](http://ieeexplore.ieee.org/abstract/document/6459891/)) . The method for behavioral authentication depends on establishing a baseline for an everyday activity such as typing in a password, using the computer or typing on a smartphone. Once this baseline is established future interactions can be compared to determine if the same user is indeed accessing the system or an imposter. Simple biometric authentication has been in use for decades in the application of fingerprints, facial recognition (<b>) , retinal scans (<c>) , vein pattern analysis (<d>) , ear shape configuration ([e](http://ieeexplore.ieee.org/document/7893882/)) , and many others. However, these methods do not allow for continuous authentication in any convenient or reliable manner. Advances in multimodal authentication still use behavioral biometrics but they also combined it with older standard biometrics ([f](http://ieeexplore.ieee.org/document/7844515/)). This leads to the conclusion that while there are great advances being made there is no frontline so to speak. All the methods discussed only come into play only after access to restricted content has been granted however brief. The idea of a constant security that comes into effect almost from the moment the user enters the building is vastly preferable.

Gait analysis is divide into two separate but related fields. The first one is a security and identification field where the purpose is to reliably identify a person simply by using cameras and other technology to record their walking gait and match it to earlier collected information. This is not a well-researched field as of today. However, some early attempts have been made ([g](http://www.mdpi.com/2073-8994/8/10/100/pdf)). Work has also been completed to attempt to quantify the errors and issue of using accelerometer data in non-ideal conditions ([h](http://ieeexplore.ieee.org/document/7518029/)) .The second field is in medicine as when the first field was initially being explored there was a correlation to certain medical conditions that could be identified by the computer driven gait analysis that human doctors where incapable of themselves. In this manner, it has become a diagnostic tool for the medical field as well. Most of the leading work in arm swing and trunk analysis are being done to look for markers of Parkinson’s (<i>) / (<j>) / ([k](http://ieeexplore.ieee.org/document/7591269/)). However, none of these sources have made use of any technique for gathering the arm swing data other than motion capture camera technology. Further research into gait analysis all concentrates on measuring parts of the body not covered in this review.

In conclusion while the field of continuous authentication continues to advance in the realms of behavioral and camera based authentication, continuous passive biometric authentication remains the end goal of this line of research. Hopefully the introduction of a device small enough to be worn by a reasonable employee and capable of gathering a large amount of biometric data will advance this goal.

1. Problem Formulation (Research Question)
   1. The **ASA** is designed to represent asymmetry in arm swing magnitude between each arm. A value of 0.00 would indicate that both arms are moving exactly the same magnitude **ASA=(45°−arctan (ArmSwingmore∕ArmSwingless))90°×100%**
   2. To ensure that trunk rotation did not influence arm swing, we quantified trunk rotation to the left and to the right as the transverse plane angular rotation of the thorax with respect to the pelvic coordinate system. The sequence of rotations for this calculation was Z-X-Y (axial rotation, flexion/extension, sidebending). The magnitude of trunk rotation was quantified as the total side-to-side rotation of the thorax during a stride cycle. We then calculated the trunk rotation asymmetry (**TRA**) as follows: **TRA=(45°−arctan (TrunkRotationmore∕TrunkRotationless))90°×100%**
   3. In addition to arm and trunk movements, we calculated gait velocity as the velocity of the pelvic markers through the lab coordinate system in the direction of forward progress. Asymmetry of the lower limbs was defined as the asymmetry in stride time. The time, in seconds, from heel strike to subsequent heel strike was calculated for both sides. The stride time asymmetry (**STA**) was then calculated as follows: **STA=(45°−arctan (StrideTimemore∕StrideTimeless))90°×100%**
   4. The Problem that arises with all of the established equations and research in gait analysis is that it relies on being able to collect information about the entire body. Both sides, legs and trunk need to be analyzed in concert to gain a proper understanding. This leads to the question of how do you perform the same analysis with data from only one arm and only acceleration values at that. No ability to read positioning or other effects on the rest of the body. For example, when holding different weights in each hand the gait is modified to compensate. The TRA compensates for this in the ASA but without both sides of the equation these become useless.
   5. To combat this the plan for this research is to “discover” the algorithm that allows differentiation of the proper signature. To explain, the idea is to use the 30-entry data set of a single user to generate a training set in weka that then is able to distinguish the correct single data set from the 30 random volunteers with a false positive rate of less than 2% and a false negative rate of less than 0.3%. if these criteria can be met the technology will be considered viable.
   6. This in real world terms would be a user wearing the watch for a 1-3-day period going about their usual duties and then the data from the training period being used to generate a profile for the user. This is different from current behavioral training as it requires no active participation for the user for correction other than wearing the watch. After the training set is generated any unauthorized users wearing the watch would be detected within 30-40 seconds of movement.
2. Proposed Approach
   1. The approach suggested by this paper is to use a smartwatch, an inexpensive and readily available technology, to bridge the gap between these two disparate fields. If the accelerometer data generated by a user’s walking gait can be analyzed and shown to have a unique signature, this would allow the security apparatus of an organization to develop a new kind of ID that would both continuously authenticate the user as well as provide other useful data such as health biometrics through heart rate and location data using the built-in GPS.
   2. During the arm swing of a user there are 4 unique stages that can be identified and isolated to produce a set of features. The arm starts at neutral and the advances to the positive position in front of the body. In phase two the arm comes from the apogee back to neutral. In phase three the accelerates from neutral to negative until reaching apogee. In phase four the arm relaxes back to neutral. In subdividing the arm swing into these phase features may be generated on the differences in timing a force between phases and on the arm swing as a whole.



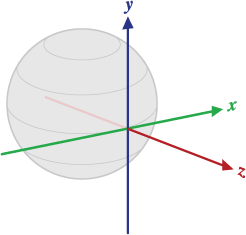
* 1. Because of the difficult nature of the data collection in this experiment the decision was made to collect a sampling of multiple arm swings from volunteers to create a cross reference of different samplings that the main data collection to compared against to prove the accuracy of the algorithm developed.
  2. Using a Java Feature Generation and filtering program created for this experiment the raw data will be parsed into a refined Linear data set that contains the raw data along with some extra calculated values. A secondary Non-linear data set file will be generated that contains all of the non-linear calculated features. They will be tied together by using the same reference number in the file name. Once the features have been generated the 30-entry single user data set will be fed into weka as a training set to generate a EM map and other data mining algorithms base line. These trained algorithms will then be fed the randomized 30 individual set and rated on their ability to identify the correct single set.
  3. If one of the algorithms is able to identify the correct set with a false positive rate of less than 2% and a false negative rate of less than 0.3% it will be considered for future research.

1. Data Collection
   1. Physical Methods
      1. The devices used for collection where a Motorola nexus 5 and a Motorola moto 360 smartwatches. The application used to collect and export the data sets was a sensor dashboard program developed for the London android hackathon of 2014 by Juhani Lehtimäki, Benjamin Stürmer, Sebastian Kaspari.
      2. Data collection was conducted on 30 volunteers including the research assistant collecting the information. A further 30 data sets where collected on the research assistant with the same parameters
      3. The parameters for the collection was for the volunteers to attach the watch to the dominate hand while carrying the paired smartphone in their non-dominate hand
      4. For the purposes of this experiment it was determined that a blind study was necessary for proper data collection as awareness of arm swing data being take would affect the subjects arm swing. Therefore, the subjects where told the data of interest was heart rate data from walking.
      5. After each session, the data was uploaded from the device and then the device was wiped clean for the next session.
   2. Output
      1. Device ID
         1. Device ID reporting. Same device for all collections so consider superfluous information
      2. Timestamp
         1. A Timestamp object using a milliseconds time value. The integral seconds are stored in the underlying date value; the fractional seconds are stored in the nanos field of the Timestamp object.
         2. For clarity Filter program removes the start time from all timestamps giving a start to stop time.
      3. Raw X,Y,Z
         1. The raw output from the accompanying data sensor type with a magnitude ranging both negative to positive
      4. Datatype
         1. 1 – Accelerometer
            1. The length and contents of the [values](https://developer.android.com/reference/android/hardware/SensorEvent.html#values) array depends on which [sensor](https://developer.android.com/reference/android/hardware/Sensor.html) type is being monitored (see also [SensorEvent](https://developer.android.com/reference/android/hardware/SensorEvent.html) for a definition of the coordinate system used). [Sensor.TYPE\_ACCELEROMETER](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ACCELEROMETER):All values are in SI units (m/s^2)
            2. values[0]: Acceleration minus Gx on the x-axis
            3. values[1]: Acceleration minus Gy on the y-axis
            4. values[2]: Acceleration minus Gz on the z-axis
            5. A sensor of this type measures the acceleration applied to the device (**Ad**). Conceptually, it does so by measuring forces applied to the sensor itself (**Fs**) using the relation:

**Ad = - ∑Fs / mass**

* + - * 1. In particular, the force of gravity is always influencing the measured acceleration:

**Ad = -g - ∑F / mass**

* + - * 1. For this reason, when the device is sitting on a table (and obviously not accelerating), the accelerometer reads a magnitude of **g** = 9.81 m/s^2
        2. Similarly, when the device is in free-fall and therefore dangerously accelerating towards to ground at 9.81 m/s^2, its accelerometer reads a magnitude of 0 m/s^2.
      1. 3 – Orientation
         1. All values are angles in degrees.
         2. values[0]: Azimuth, angle between the magnetic north direction and the y-axis, around the z-axis (0 to 359). 0=North, 90=East, 180=South, 270=West
         3. values[1]: Pitch, rotation around x-axis (-180 to 180), with positive values when the z-axis moves **toward** the y-axis.
         4. values[2]: Roll, rotation around the y-axis (-90 to 90) increasing as the device moves clockwise.
      2. 4 – Gyroscope
         1. All values are in radians/second and measure the rate of rotation around the device's local X, Y and Z axis. The coordinate system is the same as is used for the acceleration sensor. Rotation is positive in the counter-clockwise direction. That is, an observer looking from some positive location on the x, y or z axis at a device positioned on the origin would report positive rotation if the device appeared to be rotating counter clockwise. Note that this is the standard mathematical definition of positive rotation and does not agree with the definition of roll given earlier.
         2. values[0]: Angular speed around the x-axis
         3. values[1]: Angular speed around the y-axis
         4. values[2]: Angular speed around the z-axis
         5. Typically the output of the gyroscope is integrated over time to calculate a rotation describing the change of angles over the time step, for example:
      3. 9 – Gravity
         1. A three dimensional vector indicating the direction and magnitude of gravity. Units are m/s^2. The coordinate system is the same as is used by the acceleration sensor.
      4. 10 -- Linear Acceleration
         1. A three dimensional vector indicating acceleration along each device axis, not including gravity. All values have units of m/s^2. The coordinate system is the same as is used by the acceleration sensor.
         2. The output of the accelerometer, gravity and linear-acceleration sensors must obey the following relation:
         3. acceleration = gravity + linear-acceleration
      5. 15 – Game Rotation Vector
         1. Identical to [TYPE\_ROTATION\_VECTOR](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ROTATION_VECTOR) except that it doesn't use the geomagnetic field. Therefore the Y axis doesn't point north, but instead to some other reference, that reference is allowed to drift by the same order of magnitude as the gyroscope drift around the Z axis.
         2. In the ideal case, a phone rotated and returning to the same real-world orientation will report the same game rotation vector (without using the earth's geomagnetic field). However, the orientation may drift somewhat over time. See [TYPE\_ROTATION\_VECTOR](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ROTATION_VECTOR) for a detailed description of the values. This sensor will not have the estimated heading accuracy value.
         3. For Reference See below Type Rotation Vector

The rotation vector represents the orientation of the device as a combination of an *angle* and an *axis*, in which the device has rotated through an angle θ around an axis <x, y, z>.

The three elements of the rotation vector are <x\*sin(θ/2), y\*sin(θ/2), z\*sin(θ/2)>, such that the magnitude of the rotation vector is equal to sin(θ/2), and the direction of the rotation vector is equal to the direction of the axis of rotation.

The three elements of the rotation vector are equal to the last three components of a **unit** quaternion <cos(θ/2), x\*sin(θ/2), y\*sin(θ/2), z\*sin(θ/2)>.

Elements of the rotation vector are unitless. The x,y, and z axis are defined in the same way as the acceleration sensor.

The reference coordinate system is defined as a direct orthonormal basis, where:

X is defined as the vector product **Y.Z** (It is tangential to the ground at the device's current location and roughly points East).

Y is tangential to the ground at the device's current location and points towards magnetic north.

Z points towards the sky and is perpendicular to the ground.

values[0]: x\*sin(θ/2)

values[1]: y\*sin(θ/2)

values[2]: z\*sin(θ/2)

values[3]: cos(θ/2)

values[4]: estimated heading Accuracy (in radians) (-1 if unavailable)

* + 1. Sensor name
       1. All Acc so the information is superfluous.
    2. Accuracy
       1. 0
       2. 1
       3. 2
       4. 3

1. Feature Engineering
   1. Timing
      1. Total Arm swing execute time
      2. Phase 1-4 time to execute
      3. Differential timing between phases, 2-3-4-1, 3-4-1-2, 4-1-2-3, etc.
   2. Min and max
      1. Local Max values for x,y,z
      2. Global Max values for x,y,z
      3. Local Min values for x,y,z
      4. Global Min values for x,y,z
      5. Max time for phase completion
      6. Min time for phase completion
      7. Min/max for Full arm swing
   3. Phase differential
      1. Min/max x,y,z for individual phases
      2. Variance values for x,y,z on Linear basis
   4. Averages
      1. Session averages for all values listed above.
2. Machine Learning Classification / Prediction
3. Performance Evaluation
4. Experimental Work for Validation
5. Discussion
6. Conclusion and Future Work
   1. Conclusion
   2. Future work
      1. Isolate and identify what cause shift in reporting sensor
      2. Remake collection app with better reporting and finer sensor controls as well as true raw data output.
      3. Develop algorithm to rate and quantify data points based on data type and accuracy to create a standardized reporting measure
      4. Two watch collection to offset issues of gait analysis and allow use of current equations in calculations
      5. Comparison with smartphone accelerometer data to determine if a watch is truly needed.
      6. Larger sample sizes and most participants.
7. References
   1. Behavioral Biometrics (BB) -- <http://ieeexplore.ieee.org/abstract/document/6459891>
   2. Facial Recognition (FR) – <http://ieeexplore.ieee.org/document/6528223>
   3. Retinal Scans (RS) – <http://ieeexplore.ieee.org/document/7847561>
   4. Vein Pattern analysis (VPA) – <http://ieeexplore.ieee.org/document/7882698>
   5. Ear Shape Pattern Recognition (ESPR) – <http://ieeexplore.ieee.org/document/7893882>
   6. Combined Behavioral and biometric Authentication (CBBA) – <http://ieeexplore.ieee.org/document/7844515>
   7. Gait Analysis for security and identification using smartphones (GASIS) – [www.mdpi.com/2073-8994/8/10/100/pdf](http://www.mdpi.com/2073-8994/8/10/100/pdf)
   8. Smartphone Gait Analysis using Accelerometer (SGAA) – <http://ieeexplore.ieee.org/document/7518029>
   9. Arm Swing Analysis for Identification of Parkinson’s (ASAIP) – <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5053872>
   10. Kinematic analysis of arm and trunk movements in the gait of Parkinson’s disease patients based on external signals (KAATGPES) – <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4713791>
   11. Implementation of a smartphone as a wireless gyroscope platform for quantifying reduced arm swing in hemiplegia gait with machine learning classification by multilayer perceptron neural network (ISWGRAS) -- <http://ieeexplore.ieee.org/document/7591269>
   12. Arm Swing Magnitude and Symmetry during Gait in the early stages of Parkinson’s disease (ASMS) -- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2818433>
   13. Android Sensors -- <https://developer.android.com/reference/android/hardware/Sensor.html>
   14. Android Sensor Descriptions -- <https://developer.android.com/reference/android/hardware/SensorEvent.html#values>