Using machine learning to detect pedestrian locomotion from sensor-based data

A Thesis Proposal
Presented to
the Faculty of the College of Computer Studies
De La Salle University Manila

In Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Computer Science

by

NGO, Courtney Anne

Solomon SEE
Adviser
Roberto LEGASPI
Co-Adviser

September 9, 2013

Abstract

The integration of low cost microelectromagnetic (MEM) sensors into smart phones have made inertial navigation systems possible for ubiquitous use. Many research studies developed algorithms to detect a user's steps, and to calculate a user's stride to know the position displacement of the user. Subsequent research have already integrated the phone's heading to map out the user's movement across a physical area. These research, however, have not taken into account negative pedestrian locomotion, wherein the user is moving but is not exhibiting any position displacement.

This research aims to solve this problem by collecting postive and negative pedestrian locomotion with data from phone-embedded sensors positioned in the research subject's front pocket. Using these data, a model will be built to classify negative pedestrian locomotion from positive ones.

Keywords: statistical computing model, digital signal processing, inertial navigation systems, MEM sensors, etc.

Table of Contents

1 Research Description

This chapter discusses the current technologies in inertial navigation. This also covers the objectives, scope and limitations of the research, significance of the research, and the research methodology.

1.1 Overview of the Current State of Technology

According to (Shala & Rodriguez, 2011), inertial navigation systems (INSs) determine the path taken by a person based on data gathered from inertial sensors. Inertial sensors that are usually used in these INSs include accelerometers and gyroscopes. Smart phones currently already employ these sensors as micro-electrical-mechanical systems (MEMS) devices, making it possible for INSs to be applied in smart devices and possibly for ubiquitous use.

Unlike other navigation systems, INSs do not continuously calculate for the path with the help of fixed measuring instruments. This distinguishes it among other navigation systems as it does not rely on access points unlike with Wi-fi routers in Wi-fi localization, satellites in the Global Positioning System (GPS), or markers in marker-based navigation. Compared to other navigational systems, INSs are independent of its environment, requiring less cost that otherwise would have incurred with the need of access points. This also implies less environment set-up as access points do not need to be installed for the navigation system to operate. Considering that it is a cheaper and simpler alternative, INS appears to be a more attractive approach to building navigation systems.

Using INSs in real-world situations, however, is limited because its MEMS devices are susceptible to noise and gradual drifts that cause cascading errors. Because of this, most existing inertial navigation systems integrate regular checking with access points with known positions such as satellites and Wi-fi routers to calculate the position of the mobile unit to compensate for these inaccuracies (Martin, Krosche, & Boll, n.d.).

Despite these, researchers continued to look into strategies to work around these limitations and make INS independent of these access points. Studies (Li et al., 2012; Shala & Rodriguez, 2011; Won Kim, Jin Jang, Hwang, & Park, 2004) have been using filters such as Kalman filter to address the issue of sensor drifts by providing estimates to reduce error in noisy data, giving more weight to those with stronger confidence. Apart from filters, other strategies include analyzing the signals to measure the step length, and to validate these steps through step detection, as evident in (Kothari, Kannan, Glasgwow, & Dias, 2012; Shala & Rodriguez, 2011; Moell & Horntvedt, 2012). Step detection is done in various ways, and past studies have used heuristic models in detection. Most look for peaks and valleys in the accelerometer signals that indicate movement or activity. If these spikes pass a certain threshold, studies such as (Kothari et al., 2012; Libby, 2008) consider them as steps. (Shala & Rodriguez, 2011) and (Moell & Horntvedt, 2012) also take into consideration the time gap between detected steps as one step could emit more than one peak and valley. By doing so, the algorithm would not falsely detect more than one step when only one has been made. Other studies like (Lee & Mase, 2001; S. Y. Cho & Park, 2006; Li et al., 2012) additionally incorporated false-step detecting lags, sling windows, and dynamic time warping (Parnandi et al., 2009?; Li et al., 2012; Thanh, Makihara, Nagahara, Mukaigawa, & Yagi, 2012a).

Although these step detection methods were able to give out promising results, they still limit the user from freely stopping and doing various movements in place. This is because their systems are not designed to detect if a person is truly moving from one position to another, making movements such as walking-in-place mislead their systems. In their experiments, subjects walked on pre-defined paths and were expected to continue and stop only when the experiment is finished. This presents a problem because in real-world situations, users are not bound to walk continuously.

Some studies (Susi, Renaudin, & Lachapelle, 2013; Renaudin, Susi, & Lachapelle, 2012) made data-based models to detect motion modes of handheld sensors. Motion modes refer to where the subject is holding the phone. It can be held in a swinging hand while the person is walking, it can also be positioned quite stably as when a person is walking while on the phone, and it can also be transferred from the bag to the subject's hand without the subject actually moving from his physical position. The last class is called "irregular movements", movements that move the phone, but the subject is actually not moving from position. Even though the two studies were able to classify irregular movements with a 94% accuracy, the studies did not elaborate what irregular movements were collected.

This research aims to do a more thorough study on irregular movements. In this research, positive pedestrian locomotion is defined as movements that include moving from one physical position to another on foot. Examples of these are walking, jogging, running, and climbing up and down the stairs. False pedestrian locomotions are movements that do not require moving from a position, such as standing. There are, however, some false pedestrian locomotion movements that can simulate movement from position, and can therefore trick the step detection algorithm into detecting that a genuine step has been taken. These activities include walking-in-place, jogging-in-place, running-in-place, and doing various exercises in place. For INSs to fully function in real-world applications where users are free to do these negative pedestrian locomotion movements, it is imperative that INSs would be robust enough to handle movements such as these. To not limit user activity when using INSs, there is a need for a model that can detect pedestrian locomotion.

1.2 Research Objectives

1.2.1 General Objective

This research aims to build a model that can detect pedestrian locomotion using data from inertial sensors

1.2.2 Specific Objectives

1. To study different features extracted from data taken with tri-axial accelerometers and gyroscopes that can be used in building the model;
2. To build a corpus of pedestrian locomotion by having research subjects participate in daily activities using inertial measured data;
3. To build a machine learning model that can detect pedestrian locomotion;
4. To evaluate the effect of features and window sizes on the performance of the model; and
5. To assess the effect of the model to an INS and its other modules.

1.3 Scope and Limitations of the Research

This research will concentrate on building a pedestrian locomotion detection model using data gathered from inertial sensors. An application will be developed on the Android platform to collect data, and will run on a Samsung Galaxy SII. Subjects will be requested to conduct experiments that will form the data corpus. The movement they will be performing includes positive pedestrian locomotion such as climbing up and down the stairs, and walking; and negative pedestrian locomotion such as walking- and running-in-place, bending down, turning around in place, and various exercises. These activities will not involve any equipment including treadmills, bicycles, and other objects.

The phone will be placed in the front pocket of the user, and a fixed orientation of the phone will be followed. The subjects would be limited to an age bracket of 20 years old to 49 years old. According to (Thanh, Makihara, Nagahara, Mukaigawa, & Yagi, 2012b), age becomes a factor for subjects under 20 years old, and over 50 years old. The difference comes from the former's stabilizing his gait, and the latter's degradation of physical strength. By limiting research subjects to the age bracket of 20 to 49 years old, the model to be generated can be used by a majority of people.

Unlike age, the same study found that gender is not a significant factor in a subject's gait. For this reason, this research will accept subjects of any gender. This study would collect data from a minimum of 30 subjects, following the example lead by (Anguita, Alessandro, Oneto, Parra, & Reyes-Ortiz, 2012). Most of the experiments will be conducted in Gokongwei and Bro Andrew Gonzalez buildings in DLSU-Manila. Research subjects who are non-DLSU students will perform the data collection in other spaces with a similar topology. The topology are therefore limited to level ground, and stairs.

1.4 Significance of the Research

This research will create a model that will aid existing INSs to be more robust. With the model this research aims to build, future INSs can allow its users to do more natural movement that otherwise would have mislead current algorithms. As these errors cascade during its use, it is imperative that true pedestrian locomotion be detected for proper path mapping to be accomplished. In line with this, the corpus to be built in this research differs from existing corpora. Aside from positive pedestrian locomotion movements that are already present in existing corpus, the corpus of this research will also include movements that are classified as negative pedestrian locomotion.

This research would also be instrumental in propagating the application of INS in everyday activities. The significance of INSs's improvement would permit these systems to perform better than other approaches to navigation systems. As INSs only rely on the data gathered from the inertial sensors, and not on fixed instruments like access points and markers, it is a more environment-independent approach. This characteristic makes INSs appealing as it would entail less cost, and less environment set-up. Some applications that can be created using INSs include guide-map applications that can be used in public places, marketing applications for malls, and even a way to track mentally-handicapped patients around a vicinity.

A specific example of an application where the model can be used is considered in creating a museum guide mobile app. As a user enters the museum, the app can start to track the person's location and continue to keep track correctly even when the user stops to look at the museum pieces. Because the app can also derive the person's location via a starting point, it can act as a map and help the user find a specific museum artifact. Alternatively, it can also estimate which museum piece the user is looking at, and give him more information about the piece.

Furthermore, as users continue to use it, authorities would be able to identify "hot spots", or places that people frequent to. By doing so, they can study why these hot spots exist, and what can be done to improve them.

2 Review of Related Literature

This section discusses methodology, algorithms, features, and limitations of existing research.

2.1 Inertial Navigation Systems

INSs (Li et al., 2012; Shala & Rodriguez, 2011; Moell & Horntvedt, 2012) traditionally follow a sequence of phases to compute where a user has traversed: step detection, step length estimation, heading determination, and mapping. The final output of a INS is the path the system predicted the user roamed, which is usually accompanied with a blueprint map of the physical place. It is important that each phase would operate as accurately as possible as an error in one phase could cascade into the following phases. If errors are continuously accepted by the system, the final output would greatly reflect these errors.

When the user starts walking, the system starts collecting data from the inertial sensors: accelerometer, gyroscope, and orientation (to determine heading). The system starts with the step detection phase, which is responsible in determining whether a step has been taken or not. After it detects a step, it would call on the step length estimator to determine the distance of the step taken. It would also simultaneously call on the heading determiner to calculate for the heading of the phone. The distance and the heading would then be used to map the path of the user. After doing these processes continuously during the duration of the user's walk, the path of the whole walk would finally be generated.

The model this research aims to build would be placed before the step detection phase, as shown in Figure 2.1. If the pedestrian locomotion model detects a negative pedestrian locomotion, the system would just disregard the movement. If it does detect a positive pedestrian locomotion, the step detection phase would start to check if steps have truly been taken.

2.1.1 Step Detection Module

As mentioned before, the step detection module is responsible in determining if the user has taken a step. A conventional method of detection is a peak and valley detection as used in (Lee & Mase, 2001; Won Kim et al., 2004; Kothari et al., 2012; Libby, 2008), where each peak and valley must meet a certain threshold.

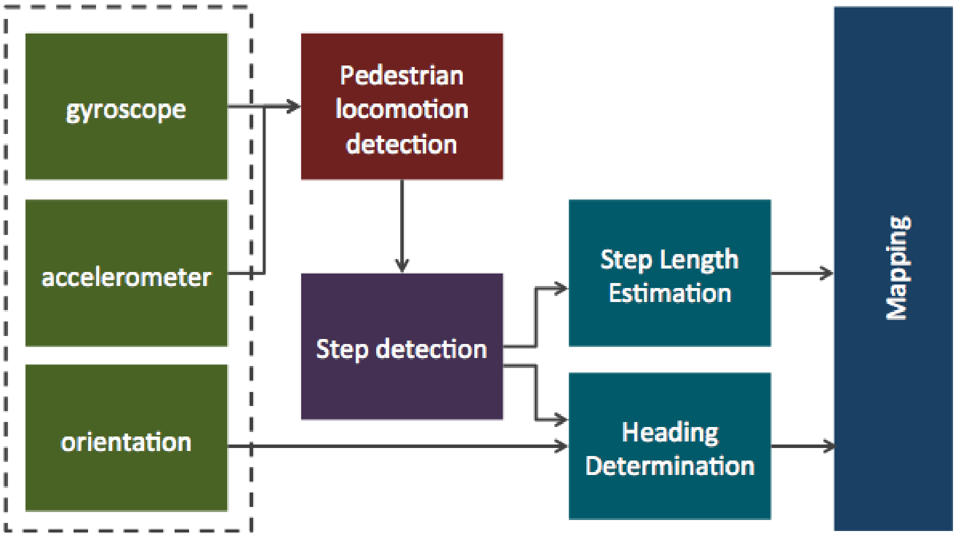


Figure 2.1: Conventional system flow of inertial navigation systems with the model this research aims to create.

In both (Won Kim et al., 2004) and (Lee & Mase, 2001), a pattern of expected vertical and horizontal acceleration when a person takes a step is used in a method called $peak-detectionmethodbasedoncombineddual-axialsignals$ (Ying, Silex, Schnitzer, Leonhardt, & Schiek, 2007). In their research, vertical acceleration corresponds to upward and downward movements, while horizontal accelerations pertain to forward and backward movements. For vertical acceleration, an upper threshold is expected to be reached during the first phase of a step, and another lower threshold at the second phase of the step. For horizontal acceleration, a lower threshold is expected to be achieved the second phase, and a upper threshold at the third phase of the step. Overall, four thresholds should be met at specific time periods of a step.

In (Li et al., 2012), they set two thresholds for changes in the acceleration magnitudes in lieu of peak and valley thresholds. The changes in one step must be more than $1.96m/s^2$, but less than $19.6m/s^2$.

Other step detection algorithms require a minimum time gap in between detected steps. Instead of applying the $peak-detectionmethodbasedoncombineddual-axialsignals$, studies such as (Shala & Rodriguez, 2011) gathered peaks exceeding $12.5m/s^2$ and made sure that only one step can be detected every 350ms. This would discard other peaks of the same step that have also exceeded $12.5m/s^2$. (Lee & Mase, 2001) also implemented a similar heuristic.

2.1.2 Stride Length Estimation Module

The stride length estimation module in an INS calculates the length of a step taken after one is detected. A number of studies (Lee & Mase, 2001; Won Kim et al., 2004; Li et al., 2012) made use of linear models to determine a step length. They have inferred that there exists a linear relationship between step frequency and step length, stating that a user takes bigger steps when walking faster. A frequency model was made in (Li et al., 2012)

(Li et al., 2012) developed a frequency model : $L_g = a * f + b$, where f defined as the walking frequency, and a and b are coefficients that are derived from collected data. In this manner, the coefficients can be altered to make the model specific to a user.

In (Won Kim et al., 2004), they were able to estimate that a step is 60cm long when it takes 0.675 seconds, and 80cm when it takes 0.662 seconds. They also considered the mean of acceleration and they were able to find a relation between it and step length. The mean was found to be around $2.8244 m/s^2$ when a 60cm step is taken, while it is around $5.438 m/s^2$ when taking a 80cm step. They were able to use both step frequency and acceleration mean as heuristics by applying the following equation to determine a stride length:

$$Stride = 0.98 * \sqrt[3]{\sum_{i=1}^{N} \frac{|Acc_{i,mag}|}{N}} \tag{1}$$

where
$Stride$ is the stride length
$Acc_{i,mag}$ is the magnitude of the accelerometer at time i
N is the window size

Aside from step frequency and acceleration mean, variations in acceleration were also taken into account in (S. Y. Cho & Park, 2006; D.-K. Cho, Mun, Lee, Kaiser, & Gerla, 2010; Nam, 2011). Furthermore, they stated that the models were also sensitive to inclination. Because of this, different models were generated for walking on level ground, up slope, and down slope.

2.2 Pedestrian Locomotion Classification

There are currently two studies (Susi et al., 2013; Renaudin et al., 2012) that have created models that consider negative pedestrian locomotion, though there a number of differences from the models and the model this research aims to create. (Susi et al., 2013; Renaudin et al., 2012)'s models considered different orientations which the phone might be held, but also assumed that the phone is generally handheld. Both of these studies classes are as follows:

- Quasi-stable : Subject is showing position displacement, but the phone is exhibiting a quasi-stable position (e.g. texting, phoning, phone is inside the bag)
- Swinging : Subject is showing position displacement, and the phone is also moving, positioned at the subject's swinging hand
- Irregular : Subject is not showing position displacement, but the phone is moving (e.g. subject is taking the phone out of his bag)

In the two studies, negative pedestrian locomotion are categorized under irregular movements. The studies, however, did not elaborate specifically what movements were collected that are considered irregular. Moreover, there is a difference with this study to the two studies as they concentrate on the phones being held in hand.

2.2.1 Corpus

The study (Renaudin et al., 2012) collected data from 12 subjects with an equal number of males and females, and with age ranging from 20-40 years old. The second study (Susi et al., 2013) gathered data from 4 subjects, with 2 men and 2 women.

Data was collected by asking the subjects to walk around a predetermined path that stretches for more than 150m. In the first study, (step length estimation), subjects also had to walk at varying speeds considering slow (0.8 km/h), regular (1.8km/h), and fast (4.0km/h) rates. In both studies, the path to be taken was consistently of the same level, and there was no indicated need that the path must be straight.

The subjects were able to do either of the classes during data collection. In order to keep track of the ground truth, or what the user was actually doing, an

IMU was either placed at the subject's foot or a proctor would administer a wheel sensor to keep track of the user's steps. All of these devices strictly follow GPS time so that synchronization would not cause a problem.

For both studies, an IMU containing a tri-axial gyroscope and a tri-axial accelerometer were used to collect data. Aside from the instances where the phone had to be placed in the bag (under quasi-stable), the IMU was generally handheld. Data from these sensors are collected at a frequency of 100Hz.

2.2.2 Pre-processing

After receiving the signals from the sensors, pre-processing of the sensors would be done before extracting features. Cutting-off outliers would help model generation later on in creating a model. Since it is known that human gait have a frequency of less than 15Hz for both accelerometers and gyroscopes, the data would undergo a Butterworth filter with a 15Hz cut-off.

It is important in these studies to know the orientation of the phone because these IMUs can be held in different ways. This could mean that the accelerometer's x-axis while the IMU is in a bag would not be the same as the accelerometer's x-axis while the IMU is at the hand. In order to treat this problem, the magnitude of each sensor's vector of data would be used instead. To do this, the Euclidean norm of the sensors's vectors would be computed.

$$sens_{i,mag} = \|sens_i\| = \sqrt{(sens_{i,x})^2 + (sens_{i,y})^2 + (sens_{i,z})^2} \tag{2}$$

where
$sens_{i,mag}$ is the magnitude
$sens_{i,x}$ is the x-axis value of the accelerometer/gyroscope at time i
$sens_{i,y}$ is the y-axis value of the accelerometer/gyroscope at time i
$sens_{i,z}$ is the z-axis value of the accelerometer/gyroscope at time i

The magnitude would still contain non-zero DC components, which were also removed afterwards.

A sliding window was then implemented with a 50% overlap. A step was estimated to take 1.28 seconds in the study, as it was not too long or too short for a step. In order to make sure that a step would occur in a span of a window, a sliding of 256 samples was chosen.

2.2.3 Feature Modeling

After pre-processing the raw data, features were extracted to prepare for model generation. The two studies used the same set of features for their classifier: energy, variance, and dominant frequencies. It is notable that these set of features were chosen because it maximizes the differences among inter-class data, and minimized the differences among intra-class data.

The energy was extracted as a feature because it can distinguish a static movement (e.g. standing still) from dynamic movements. The energy can be calculated as follows:

$$energy_i = \frac{1}{N} \sum_{0}^{N-1} {sens_{i,mag}}^2 \tag{3}$$

where
N is the length of the window
$sens_{i,mag}$ is the magnitude of the accelerometer/gyroscope at time i

Variance is also used as feature because it is sensitive to sudden increases which can be indicative of irregular movements.

Dominant frequencies are also used because it is indicative of the position of the IMU on the body. It can be calculated with Short Time Fourier Transform (STFT), a non-computation heavy algorithm. Spectograms shown in the studies exhibited a distinction of dominant frequencies of walking with the IMU (phone swinging in hand), as opposed to having the IMU in hand (talking on the phone). Peaks that are clear in the spectogram reflected the movements when the phone was place at the hand. Additionally, dominant frequencies are distinct from irregular to static to swinging, making it easier for the classifier to predict.

These gives each window a vector of 6 features, 3 features for each sensor.

2.2.4 Data Modelling

All of these features would be used in generating a classifier that can determine whether the phone is at the hand, is assuming a quasi-stable position, or is performing an irregular movement.

Both studies made use of a decision tree to create their models, and the generated model was used before step detection. The model's prediction was crucial in determining what kind of step detection algorithm would be used because there is a different set of heuristics depending on the phone's position.

2.2.5 Results

In the studies, a different set of subjects were asked to participate in evaluating the classifiers. There were also two kinds of evaluation done: controlled tests, and free motion tests. In controlled tests, the subjects had to follow a set of movements that he had to follow strictly. Free-motion tests, on the other hand, also restricts the subject to do a set of activities, but at the subject's choice of sequence.

The accuracies of the model were all high. In (Renaudin et al., 2012), quasi-stable movements were correctly predicted 100% of the time, and swinging was correctly predicted 98% (2% were mispredicted as irregular). In the study, irregular movements was only used as a default value, thus there is no accuracy rate given to this class.

Apart from the swinging and quasi-stable accuracy, the second study (Susi et al., 2013) was able to do free-motion tests, enabling the subjects to do irregular movements. The importance of free-motion tests were stressed as it tests the robustness of the model in a more realistic setting. Still, the model was able to do well. Swinging was predicted correctly 95% of the time, and quasi-stable movements 98% of the time. Irregular movements also reached an accuracy of 98%.

Table 2.1: Confusion matrix of the controlled tests in (Renaudin et al., 2012) and (Susi et al., 2013)

	Texting	Swinging
Texting	100%	0%
Swinging	0%	98% (2% irregular)

Table 2.2: Confusion matrix of the free-motion tests in (Susi et al., 2013)

	Swinging	Texting	Irregular
Swinging	95%	2%	3%
Texting	1%	98%	3%
Irregular	6%	0%	94%

2.3 Modified Step Detection Algorithms

As discussed in Section 1.1 and in Section 2.1.1, some approaches to step detection primarily assumes that the person holding the phone is walking. While their algorithms would be able to catch the instance a person stops walking due to low accelerometer activity, they generally do not take into account when the person is not moving from place but do generate a high accelerometer activity, such as the case in walking-in-place. The following algorithms, on the other hand, have incorporated additional measures that allow their algorithms to recognize if a person is doing the latter kind of activity.

2.3.1 Algorithms

(Thanh et al., 2012a) also had a different method of detecting steps. The research focused on creating a step detection model with data collected from 53 subjects from ages 15-70 years old. The classes are: walking on level-ground, climbing up a flight of stairs, climbing down a flight of stairs, climbing up a slope, and climbing down a slope; their sensors were placed at the waist, at ther person's back. They used support vector machines (SVM) and k Nearest Neighbor (KNN) with Dynamic Time Warping (DTW) as their distance function. According to (Thanh et al., 2012a), DTW calculates the similarity between two signals even if the two signals vary in time or speed. (Thanh et al., 2012a) claimed that by using DTW as their distance function, the speed at which a subject walks is irrelevant as DTW can detect similarities regardless of speed.

In some studies (Lee & Mase, 2001; Li et al., 2012), additional heuristics were implemented to prevent allowing false positive steps.

In (Lee & Mase, 2001), a lag parameter was added in their step detection algorithm. With the lag, the system can supposedly check if the step taken is not a step, but another body movement. It involves getting the z-axis of the accelerometer, which according to (Lee & Mase, 2001) is indicative of upward movements. The lag parameter is as follows:

$$lag = min_{j=0...N}(\sum_{n=0}^{N} z(n)z(n-j)) \tag{4}$$

where
lag is the lag parameter

N is the window length
$z(n)$ is the z-axis value of the accelerometer at time i

The lag must be greater than a threshold to pass the heuristic. As can be seen in the equation, the study assumed that other body movements would have less activity in the accelerometer's z-axis, and that walking would induce peaks in the z-axis.

(Li et al., 2012) further used DTW to detect as an added filter to detect false steps. Aside from (1) checking if peaks and valleys pass a certain threshold, (2) peaks and valleys must also not be too short, or (3) too long (maximum of 1 second). Acceleration's peak and valley's magnitudes are also considered, where (4) the magnitude must be within a minimum of 0.2g, and a maximum of 2.0g.

DTW comes in as a fifth heuristic which acknowledges the similarity of steps taken with the right leg, and similarity of steps taken with the left leg. In this condition, the similarity of the last step taken with the left/right foot and the current step taken with the left/right foot must be greater than a threshold. If the result is negative, a sixth heuristic compares the current left step with the next left step. If these two signal's similarity passes the threshold, the current left step would be considered a step. This heuristic can help in detecting false steps because if a subject stops to walk-in-place, there would be a difference in the signals. But if the subject continues to walk-in-place, there will still be the possibility that "walking-in-place" steps would be considered similar to one another, and thus detected as steps.

2.3.2 Corpus

Building a corpus is not required to create a inertial navigation system, as shown in studies like (Lee & Mase, 2001) where the heuristic model were modeled after studies on gait. Although in other heuristics based models, data were collected to create their heuristic models. In (Li et al., 2012), they collected data from 40 subjects, ending up with a total of 10,000 steps to create their step detection algorithm.

For data-based models, the number of subjects varied from study to study. Some studies collected data only from 4 subjects (Susi et al., 2013), 12 subjects (Renaudin et al., 2012), some with 53 subjects (Thanh et al., 2012a). The biggest inertial-based gait database (Thanh et al., 2012b) had 736 subjects, composed of 382 males and 354 females, with ages ranging from 2 to 78 years old.

For studies like (Li et al., 2012), 50 subjects participated for data collection. The research collected just enough data to build a generic model since personalization would still be needed to make the model work for a specific subject.

2.3.3 Results

In general, adding a heuristic-based or data-based model to detect negative pedestrian locomotion resulted to a more effective system.

The accuracy in (Lee & Mase, 2001) after adding a lag parameter reached an accuracy of 96.3% and 95.8% for detecting level ground and climbing down the stairs respectively. Though the climbing up class only had a 54.2% accuracy, the accuracy went up to 83.3% after using personalized data.

Table 2.3: Performance of the generic model discussed in (Lee & Mase, 2001)

	Level	Up	Down	Missing
Level	96.3%	1.4%	0%	2.3%
Up	38.9%	54.2%	0%	6.9%
Down	2.8%	0%	95.8%	1.4%

Table 2.4: Performance of the personalized model discussed in (Lee & Mase, 2001)

	Level	Up	Down	Missing
Level	96.3%	1.7%	0%	2.0%
Up	11.1%	83.3%	0%	5.6%
Down	0%	2.8%	95.8%	1.4%

In (Li et al., 2012), the number of mispredicted false steps went down from 29 to 14 after adding the data-based model. The model also performed well even when testing the model with freestyle walking (subjects can walk in any style they want) where the mispredicted steps were reduced from 2.51 to 1.22 steps.

In the study, false negatives are more important than false positives. False positives, or the number of mispredicted false steps, can be further checked with the step detection algorithm. Even if a false step was considered a step in the pedestrian locomotion model, there is still the possibility that the false step would be detected as false by the step detection algorithm. The false negatives increased from 0.4 to 0.5. But as stated in the study, the benefits outweighed the disadvantages.

3 Research Methodology

This subsection will describe the research methodology of the study.

3.1 Research Concept Formulation

At this stage, the researcher would read on existing studies and explore possible topics that can form the concept of the research.

3.2 Review of Related Literature

A review of previous literature would be done to know the current state of research in pedestrian locomotion classification. The researcher would also be reading studies on digital signal processing to analyze how feature vectors were designed in other studies.

3.2.1 Research Approach.

The study will generally be following a quantitative approach. Raw data collected with a smart phone's accelerometer and gyroscope will be processed using machine learning algorithms. Data collection, particularly class labelling, will be administered by the researcher.

To build the model, it is important to collect a large volume of data that can represent both positive and negative pedestrian locomotion. Therefor a number of test subjects will be asked to participate in a set of activities to contribute to the data corpus.

3.3 Data Collection

3.3.1 Data Collection Activities.

The activities the test subjects will participate revolve around movements that causes positive and negative pedestrian locomotion. For this research, no instruments apart from the smart phone will be used. This means that using equipment such as treadmills is not included in the scope of the research.

For the positive pedestrian locomotion, subjects are asked to do each of the following activities for 5 minutes:

- walk
- climb up the stairs
- climb down the stairs

For the negative pedestrian locomotion, subjects are asked to do each of the following activities for 5 minutes:

- stand
- walk-in-place
- leg swings
- turn around
- random movement
- twisting
- bending
- leaning on balls and heels of feet
- sitting

3.3.2 Research Subjects

Based on (Thanh et al., 2012b), age can affect gait recognition for certain age groups. Children under 10 years old and those from ages 10 to 19 had unstable gaits.

Subjects that are members of the "under 10" and "10-19" age brackets affected the gait recognition negatively brought about their unstable gait as these particular subjects are still learning how to walk. Similarly, subjects that are 50 years old and above also have unstable gait due to a degradation in physical strength. The gait recognition, however, was able to perform similarly and well for the rest of the age brackets. For this research, subjects are targeted to be from the age bracket of 20-49. The subjects can be of any gender, and must be of no physical affliction that may affect his gait.

Following the research done by (Anguita et al., 2012), this research would also require a minimum of 30 subjects for the experiments.

3.3.3 Data Collection Instruments

In order to collect gyroscope and accelerometer readings, a Samsung Galaxy S2 will be used as an instrument. The device already has a tri-axial accelerometer and tri-axial gyroscope built inside it. An Android application would be developed to collect these readings, and store them for model building afterwards. Only one device would be used per experiment.

3.3.4 Data Collection Procedure

Research subjects will be asked to participate in the research by doing all of the movement for the indicated time duration. The phone will be placed on their front right pocket throughout the length of the activity. The orientation of the phone is placed so that the screen of the phone faces the thigh of the subject, and the top of the phone is pointed to the bottom of the pocket.

The program is already set to collect readings at a frequency of 100Hz. According to (Gyllensten & Bonomi, 2011; Ermes, Parkka, Mantyjarvi, & Korhonen, 2008), 20Hz was found to be enough to recognize human gait. With an initial collection of 100Hz at the data collection, the frequency can be lowered during post-processing depending on the performance of the model generated. Ideally, a lower frequency is better as it would not consume more power as a higher frequency collection would.

During data collection, the program will store three values each from the gyroscope and the accelerometer, corresponding to the three axes.

3.4 Data Pre-processing

Before features can be extracted, the raw data first needs to be filtered and pre-processed. The data would first undergo a Butterworth filter with a 15Hz cutoff, following the example of (Susi et al., 2013) and (Renaudin et al., 2012). According to (Mathie, 2003), human gaits only induce signal frequencies of less than 15Hz.

A sliding window would then be implemented with a 50% overlap following the methodology of (Susi et al., 2013; Renaudin et al., 2012). Three window sizes of lengths 128-, 200-, and 256-samples will be used. A program would be developed to automatically pre-process the data.

3.5 Feature Extraction

After pre-processing, the features can be extracted. During this phase, additional literature will be read to learn more about which features can be used for the model.

Currently, there is a shortlist of features that are being considered for the study. Each window would be a summarized to a feature vector containing the following features:

- mean of the accelerometer x-axis
- mean of the accelerometer y-axis
- mean of the accelerometer z-axis
- standard deviation of the accelerometer x-axis
- standard deviation of the accelerometer y-axis
- standard deviation of the accelerometer z-axis
- energy of the accelerometer x-axis
- energy of the accelerometer y-axis
- energy of the accelerometer z-axis
- mean of the gyroscope x-axis
- mean of the gyroscope y-axis
- mean of the gyroscope z-axis
- standard deviation of the gyroscope x-axis
- standard deviation of the gyroscope y-axis
- standard deviation of the gyroscope z-axis

- energy of the gyroscope x-axis
- energy of the gyroscope y-axis
- energy of the gyroscope z-axis

A program would be developed by the researcher to automatically extract the time and frequency domain features after pre-processing. One advantage of choosing these features is that they are not computational heavy, making it possible to use the model on-line.

3.6 Model Generation

The feature vectors will be fed to a machine learning algorithm to build the classifier. Following the example of (Susi et al., 2013; Renaudin et al., 2012) and (Li et al., 2012), this research will be using decision trees and support vector machines (SVM) to generate the data-based model. WEKA would be used to create these models which is capable of running algorithms like C4.5 and SVMs.

Different models using different combinations of features will be generated to know which model is the most effective in detecting pedestrian locomotion.

The model will function before the step detection module as shown in before in Section 2.1. This approach is distinct from the one taken by the algorithms exposed in Section 2.3. Unlike their proposals where additional measure are incorporated in the step detection module, this research expects the model to operate as a module on its own.

3.7 Data Analysis

In data analysis, the models would be tested and their results will be analyzed and compared to one another. In this light, the effectivity of the model can be assessed. The following are characteristics of the model that will be experimented on:

Analyses would be made on the effects of varying window sizes have on the effectivity of the model. The effects of different window size on the effectivity it will have on time domain features over frequency domain features.

Additional analyses would also be made on the effect different window sizes have over the potency of time domain features against frequency domain features, and vice-versa.

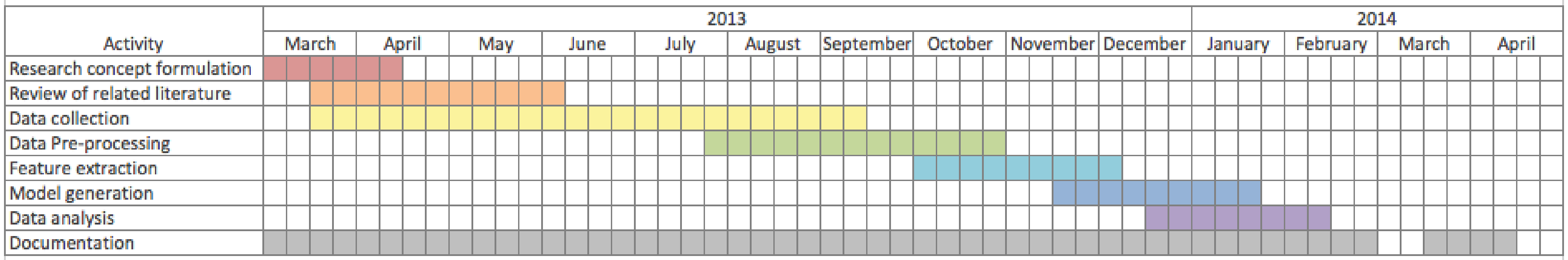
A mobile app will be developed composed of a step detection module, a stride length estimation module, a heading determination module, and a mapping module. The final result of the app is a map of a user's traversal across a space generated using inertial data. A similar app will be developed with an additional pedestrian locomotion module. A comparison between the two apps will allow this research to observe the effects of adding a pedestrian locomotion model as opposed to having none.

Afterwards, conclusions can be made and recommendations can be suggested for further research.

3.8 Documentation

The researcher will be documenting the progress and results of the study throughout the duration of the research.

3.9 Calendar of Activities



4 Theoretical Framework

This subsection will describe the theoretical framework of the study.

4.1 Sensors

The data gathered from the sensors are already normalized as provided by the phone. The measuring units are dictated in each sensor discussion.

4.1.1 Gyroscope

Gyroscopes are inertial sensors that can measure angular velocity (Aaron Burg, n.d.). Traditional gyroscopes have a mass that spins along an axis independent of the external frame. As the mass vibrates and the external frame is set to place, the mass preserve its initial orientation and will continue to resist changes to its orientation as long as the mass is spinning. These devices are useful in maintaining the balance of a machine such as aircrafts when without any orientation reference (Bryan, n.d.).

Traditional gyroscopes have been in use since the 18th century primarily for marine navigational purposes. The devices soon also proved beneficial in aeronautics exercises by 1916 (Aaron Burg, n.d.). The arrival of silicon machines prompted the entry of microelectromechanical systems (MEMS) technology to create millimeter-size versions of gyroscopes (Aaron Burg, n.d.; *Nanocomputers and swarm intelligence*, 2008). Current commercially-sold smart phones produced contain MEMS gyroscopes that can measure the angular velocity from three axes. In MEMS, the the spinning mass is replaced by a vibrating plate. When the plate is rotated, an output voltage proportional to the angular momentum is triggered (STMicroelectronics, n.d.). MEMS gyroscope typically give its results in radians per second (r/s).

4.1.2 Accelerometer

Accelerometers are devices that measure proper acceleration forces (LLC, n.d.). Accelerometers contain a mass suspended on a spring that interacts with fixed segments of the device. If the accelerometer experiences a positive acceleration in the x-axis, the mass would interact with the respective fixed segment. The measured displacement of the mass is then given as the acceleration of the x-axis (Android, 2013).

Smart phones currently commercially sold contain a MEMS 3-axis accelerometer. These devices are typically used to know if the phone is in landscape or portrait mode, is being tilted for gaming use, or is experiencing free fall which will trigger the device to cease disk interaction (Inc., 2012). MEMS accelerometers provide the acceleration of the device in meters per seconds square (m/s^2).

4.1.3 Difference between devices

Although all smart phone devices use MEMS sensor chips for their accelerometers and gyroscopes, chips used between devices differ from one another. These chips do not always adhere to the same standard of measuring acceleration and angular velocity, allowing the possibility of conflicting range of values. Aside from using different chips, calibration is also a factor. If two devices are calibrated differently, or are not using the same chip type, a model created over the recorded values of one device are not expected to operate similarly when used in the other device.

Figure 4.1 compares the collected values of a Samsung Galaxy S2 and a CD-R King B1-EP2. The data exhibited are the Euclidean norm of the accelerometer data. Though the difference can turn to be minute when compared to other devices, it can also be as large as shown in the figure. In this example, it is evident that the CD-R King B1-EP2 made use of an accelerometer that responds slower to movement.

One solution to go around the problem of different value ranges across devices is to normalize the values from 0 to 1. Although this tones down the difference between signals, it is still not a prime solution. Figure 4.2 below shows the effect of normalizing the data displayed in the figure before.

4.2 Feature Modelling

This subsection discusses how features will be modelled in preparation for modelling.

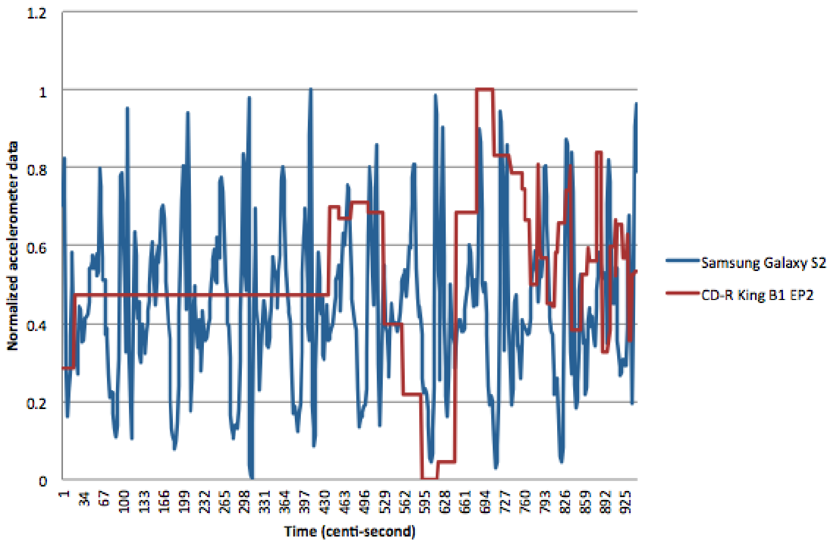


Figure 4.1: Graph showing the difference between values collected by a Samsung Galaxy S2, and a CD-R King B1-EP2, while performing the same movement.

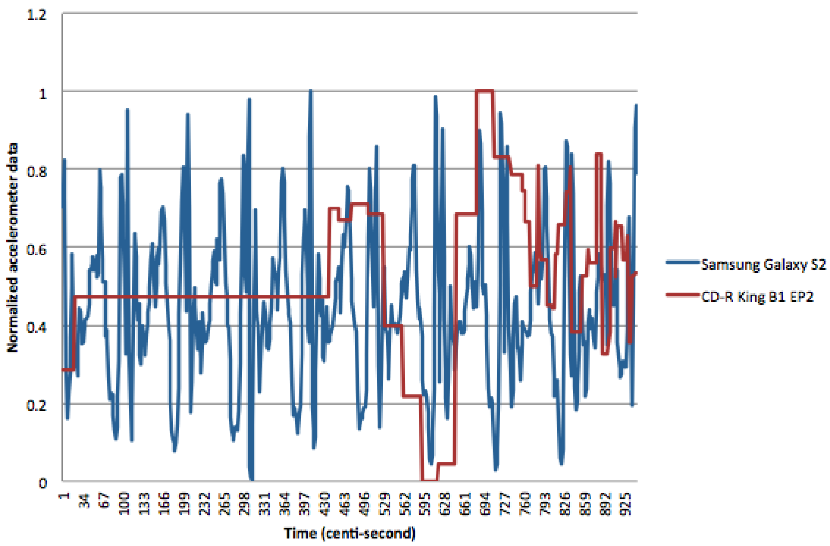


Figure 4.2: Graph showing that normalized values of data collected by two separate phones are still perceptible to variance.

4.2.1 Features

Time and frequency domain features that will be extracted from the raw accelerometer and gyroscope data will include a signal window's mean, standard deviation, and signal. The frequency domain features will be gathered after the raw signals are with short-time Fourier transform (STFT), using the same temporal window used in extracting time domain features.

Mean The mean of a set of numbers is normally used to estimate where a sample point is "located" when the set is arranged in an ascending order, thus making it a measure of location(Ronald E. Walpole, 2010). Although calculating for the mean has its advantages, it is easily influenced by outliers(Ronald E. Walpole, 2010).

$$Ave_{i,w} = \frac{1}{N}\sum_{i=0}^{N-1} sens_{i,w} \tag{5}$$

where
N is the window size
$Ave_{i,w}$ is the mean of the w-axis of the accelerometer/gyroscope over window i
$sens_{i,w}$ is the w-axis value of the accelerometer/gyroscope at time i
w can either be the x, y, or z-axis

Using preliminary data collected in this research, the means of the unfiltered Euclidean norm of the accelerometer data were collected as shown in Figure 4.3. Each mean was gathered over a window of 200 samples, with a 50% overlap. The results show that there is a distinction between the means collected when a subject is walking as opposed to when a subject is standing and walking-in-place. Based on the figure, the signals tend to reach higher frequencies when performing positive pedestrian locomotion movements. As compared to the subsequent features, the mean is less distinguishing of positive and negative pedestrian locomotion movements.

Standard Deviation Getting the variance of a data set aids in understanding the position of a sample point relative to the set's mean(Walpole, 2009). The standard deviation is equivalent to the square root of the variance of a given data set (Ronald E. Walpole, 2010). It is usually used in advantage of its sensitivity to the variance of the upper and lower data samples (Walpole, 2009).

$$Stddev_{i,w} = \sqrt{\frac{1}{N-1}\sum_{i=0}^{N-1}(sens_{i,w} - Ave_{i,w})^2} \tag{6}$$

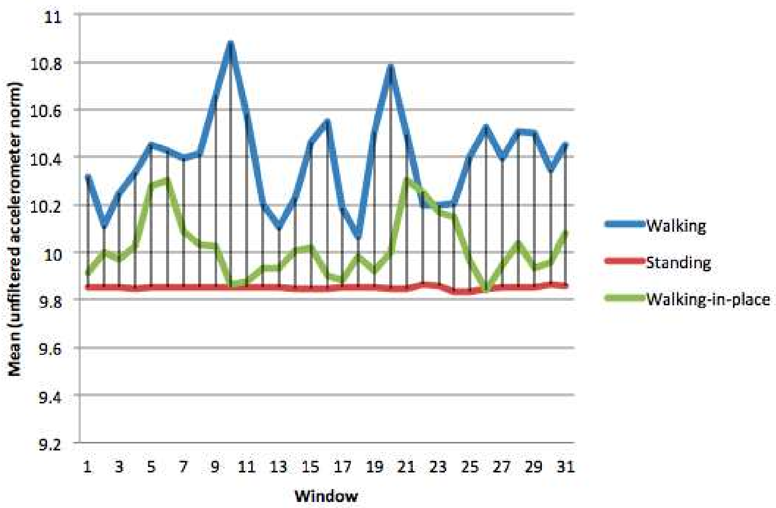


Figure 4.3: Graph showing that means collected while walking are distinct from the means collected when standing or walking-in-place.

where
N is the window size
$Stddev_{i,w}$ is the standard deviation of the w-axis over window i
$sens_{i,w}$ is the w-axis value of the accelerometer/gyroscope at time i
w can either be the x, y, or z-axis
$Ave_{i,w}$ is the mean of the w-axis of the accelerometer/gyroscope over window i

The standard deviations collected from the preliminary data shows a wide distinction between positive and negative pedestrian locomotion. Using the same data in gathering the means, the standard deviations of the unfiltered Euclidean norm of the accelerometer data were computed. As seen in Figure 4.4, the signals tend to fluctuate more when walking, as opposed to standing. The difference in walking and walking-in-place is explained by a higher y-axis activity when doing the former.

Energy The energy of a signal is usually interpreted as the strength of the signal. Given this, a signal's energy can be measured by calculating the area under the curve (Melissa Selik, 2004). To treat the negative samples that accompany digital signals, the square of the signal is computer over a temporal window.

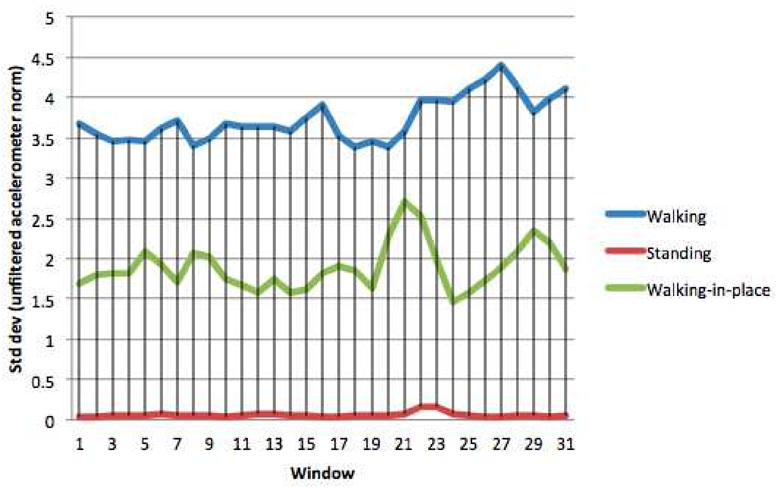


Figure 4.4: Graph showing that the standard deviations collected while walking are distinct from the means collected when standing or walking-in-place.

$$Energy_{i,w} = \sum_{i=0}^{N-1} sens_{i,w}^2 \tag{7}$$

where
N is the window size
$Energy_{i,w}$ is the standard deviation of the w-axis over window i
$sens_{i,w}$ is the w-axis value of the accelerometer/gyroscope at time i
w can either be the x, y, or z-axis

The formula used to determine the signal energy is derived from the formula used to identify energy in Physics.(Tan, 2008)

A preliminary test was also done to gauge the efficacy of using energy as a feature. Based on the same data set, the energies of the unfiltered Euclidean norm of the accelerometer data were calculating using the same window size. A distinction between walking and standing or walking-in-place can be clearly seen in Figure 4.5, similar to the results displayed by the previous two features. It is evident in this experiment that positive pedestrian locomotion movements contain more energy as opposed to its negative counterpart.

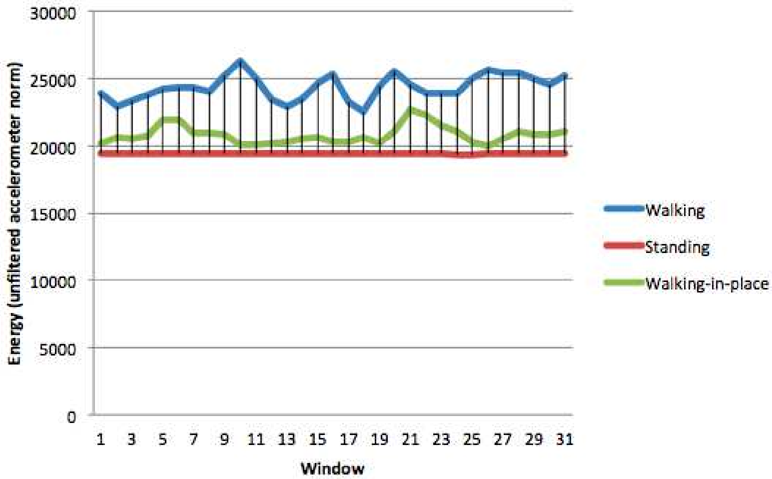


Figure 4.5: Graph showing that energies collected while walking are distinct from the means collected when standing or walking-in-place.

4.2.2 Windowing

Choosing a window size is a vital part in creating a model that deals with signals. In this research, three window sizes will be experimented with. The results shown in Section 4.3 use a window size of 200. The final results will also include experiments with window size of 128, 200, and 256; all three will be treated with a 50% overlap.

Based on an initial trial done calculating the time gaps between steps, the experiment yielded a mean of 0.52 seconds between steps. This would translate to 52 data samples before another step is expected to be taken. The window size is doubled to ensure that at least one step is recorded in each window, which gives 104. Consequently, it would be better to process signals of a length that is an exponent of 2 in Fourier transform. Given these, 128 is used as it is the nearest number to 104 that is an exponent of 2.

The 200-sample size window is based on the 100Hz sampling rate of the data collection program. This temporal window size will capture a 2-second frame of motion. With reasons akin to the the 128-window size, a window size of 256 is also considered as it is the number that is an exponent of 2 nearest 200.

4.3 Machine learning algorithms

4.3.1 J48

J48 is the WEKA implementation of Quinlan's C4.5 algorithm(Quinlan, 1993). The C4.5 decision tree creates a decision tree based on the attribute values of the available training data. Whenever it encounters a set of items for training, it identifies the attribute with the highest information gain, and designates this attribute as the root node. An attribute is said to have the highest information gain if it is able to discriminate and classify the various instances most clearly. The process is repeated by calculating the information gain of all the other attributes that might branch off from its parent node. This is done until all the data instances are able to follow a path from the root node to a leaf node. There is also the chance that the tree would not be able to accomodate all data instances due to noise.

Using WEKA, a cursory J48 model was created using preliminary data, and a 10-fold cross validation test was done. Even though the data has not yet been completely collected, an accuracy of 87.9% was achieved. Furthermore, a 92.6% recall rate of positive pedestrian locomotion is evident, which is important as the step detection algorithm succeeding the pedestrian locomotion model can further filter out false steps.

4.3.2 SVM

Support Vector Machines (SVMs) is a non-probabilistic binary linear classifier. Given a dataset, the SVM algorithm charts the values into a higher-dimensional space. Using the input data instances or the support vectors as representation, the algorithm would proceed to find the optimal hyperplane separating the different instances of the classes. An optimal hyperplane is similar to linear regression in that it maximizes the gap among the classes. Setting its kernel to a higher exponent would allow the hyperplane be more flexible to the data its trying to separate. In this paper, a linear function would be used.

An important aspect of SVMs is to find the maximum-margin hyperplane, and this entails solving for the optimum distance of it and the data inputs. Due to the complexity of the optimization algorithms, the learning takes longer and usually requires external quadratic programming solvers. John Platt at Microsoft Research came up with an algorithm that solves this problem in 1998. His algorithm is called sequential minimal optimization (SMO) (Platt, 1999), which strategically cuts down the optimization problem into smaller 2-dimensional sub-problems, and the subproblems are solved analytically. The SMO chooses two Lagrange multipliers iteratively and optimizes the pair. One of these multipliers should violate the Karush-Kuhn-Tucker (KKT) conditions, conditions of which are necessary for non-linear programming to be optimal. Convergence is ensured with SMOs.

As with J48, a preliminary SMO model was also created using the the collected data so far. A 10-fold cross validation test resulted to a 84.4% accuracy. There is also a high recall of positive pedestrian locomotion cases, reaching 91.7%, rendering both J48 and SMO find algorithms to create a sufficient pedestrian locomotion model.

4.4 Inertial Navigation System Modules

4.4.1 Step Detection Module

The step detection module would detect steps from accelerometer signals once the pedestrian locomotion model determines that the user is moving from one place to another.

The accelerometer signals would be scoured for a value greater than threshold α. As shown in Figure 4.7, a lot of values would surpass the set threshold. In order to discard false peaks, a second threshold β is introduced. Threshold β is the minimum time gap between two steps. Before a step is identified, the time gap between the said step and the previous step must be greater than threshold β. In Figure 4.9, the green arrow represents a step whose time difference from its predecessor is being assessed. The initial step would be relieved of satisfying threshold β as it does not have a previous step to refer to. The step detection process is shown in Figures 4.6 to 4.9.

These two thresholds will be determined after collecting data.

4.4.2 Stride Length Estimation Module

The stride length estimation module would start calculating for the step length once the Step Detection Module has determined the user made a step. A linear model would be created as studies have shown before that a linear relationship exists between stride length and step frequency. This module would update the step frequency along with the Step Detection Module. A linear model would be generated after collecting data.

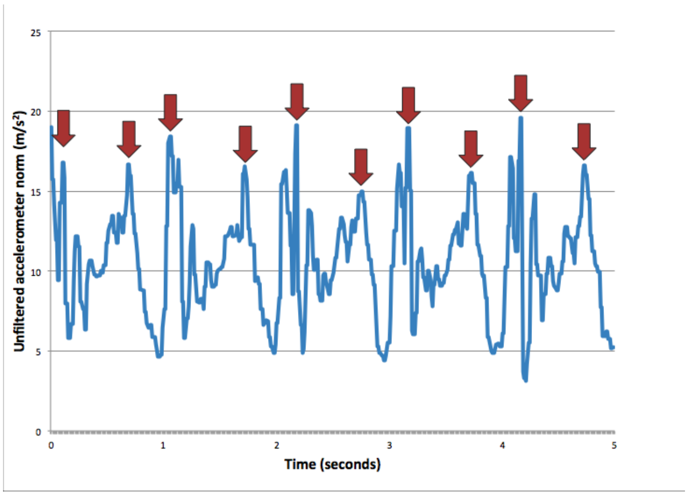


Figure 4.6: Peak generated by steps are indicated by red arrows.

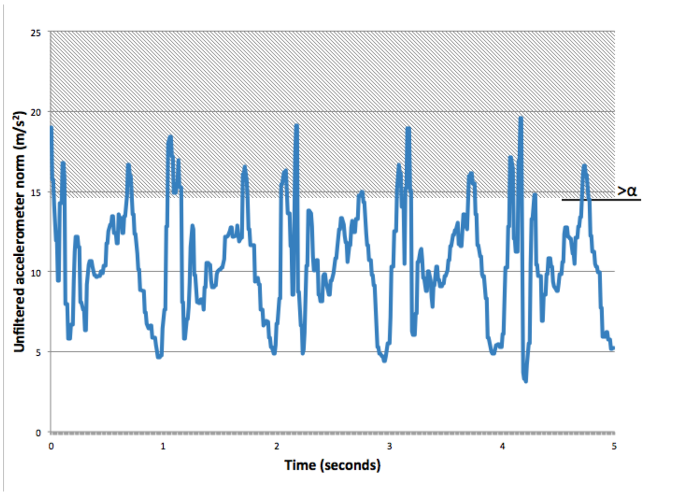


Figure 4.7: The shaded area in the chart shown the data points that have pass threshold α.

4.4.3 Heading Determination Module

This module would work side-by-side with the Stride Length Estimation Module after the Step Detection Module determines a step has been taken. It is responsible of approximating the direction the user is heading. In this research, the orientation y-axis data would be used to determine the heading. The values can range between $0°$ to $359°$.

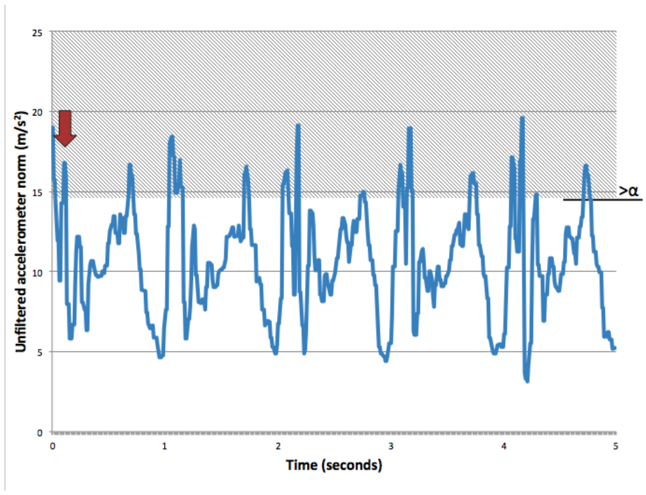


Figure 4.8: The initial step only needs to pass threshold α as there is no previous point to refer to for threshold β.

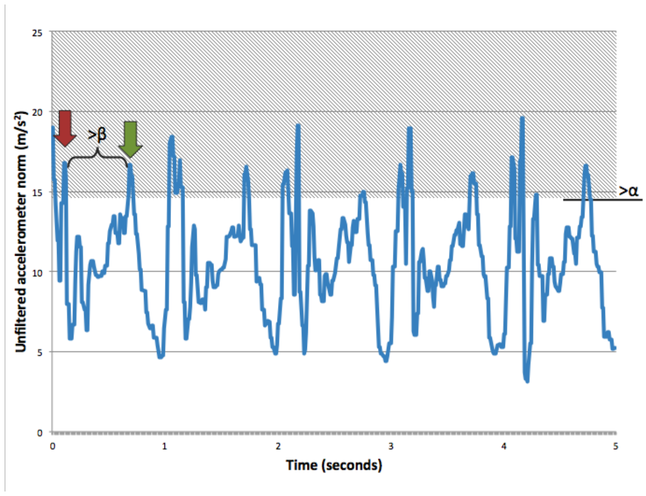


Figure 4.9: Proceeding steps need to have a time gap greater than threshold β between itself and the previous step.

4.4.4 Mapping Module

The mapping module outputs a series of points indicating a user's traversal across a space. It would receive inputs from the stride length estimator module and heading determination module, and would have knowledge of the coordinates of the previous point. The coordinates of the initial point would be set to (0,0).

The new point would be calculated as:

$$x_{cur} = l * cos(a) + x_{prev} \tag{8}$$

$$y_{cur} = l * sin(a) + y_{prev} \tag{9}$$

where
x_{cur} is the x-coordinate of the current point
y_{cur} is the y-coordinate of the current point
x_{prev} is the x-coordinate of the previous point
y_{prev} is the y-coordinate of the previous point
l is the stride length
a is the heading

The final predicted path would be relative to the user's initial position, and would be superimposed manually over an actual map of the venue. An actual map would not be used to determine the path of the user.

References

Aaron Burg, B. S. M. W., Azeem Meruani. (n.d.). *Mems gyroscopes and their applications: A study of the advancements in the form, function, and use of mems gyroscopes.* Lecture Notes.

Android, G. (2013). *Sensors overview.* Available from `http://developer.android.com/guide/topics/sensors/sensors_overview.html`

Anguita, D., Alessandro, G., Oneto, L., Parra, X., & Reyes-Ortiz, J. (2012). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Iwaal* (p. 216-223).

Bryan, G. H. (n.d.). On the beats in the vibrations of a revolving cylinder or bell. In *Proc. of cambridge phil. soc.*

Cho, D.-K., Mun, M., Lee, U., Kaiser, W. J., & Gerla, M. (2010). Autogait: A mobile platform that accurately estimates the distance walked. In *Percom* (p. 116-124). IEEE Computer Society. Available from `http://dblp.uni-trier.de/db/conf/percom/percom2010.htmlChoMLKG10`

Cho, S. Y., & Park, C. G. (2006). Mems based pedestrian navigation system. In *The journal of navigation* (pp. 135–163). Seoul: The Royal Institute of Navigation.

Ermes, M., Parkka, J., Mantyjarvi, J., & Korhonen, I. (2008). Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Trans. Info. Tech. Biomed.*, *12*(1), 20–26.

Gyllensten, I. C., & Bonomi, A. G. (2011). Identifying types of physical activity with a single accelerometer: Evaluating laboratory-trained algorithms in daily life. *IEEE Trans. Biomed. Engineering*, *58*(9), 2656-2663.

Inc., A. (2012, August). *Mac notebooks: About the sudden motion sensor.* Available from `http://support.apple.com/kb/ht1935`

Kothari, N., Kannan, B., Glasgwow, E. D., & Dias, M. B. (2012). Robust indoor localization on a commercial smart phone. *Procedia CS*, *10*, 1114-1120.

Lee, S.-W., & Mase, K. (2001). Recognition of walking behaviors for pedestrian navigation. In *Proceedings of the 2001 ieee international conference on control applications* (pp. 1152–1155).

Li, F., Zhao, C., Ding, G., Gong, J., Liu, C., & Zhao, F. (2012). A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 acm conference on ubiquitous computing* (pp. 421–430). New York, NY, USA: ACM. Available from `http://doi.acm.org/10.1145/2370216.2370280`

Libby, R. (2008, June). *A simple method for reliable footstep detection in embedded sensor platforms.* (normal checking with low pass filter)

LLC, D. E. (n.d.). *A beginner's guide to accelerometers.* Available from `http://www.dimensionengineering.com/info/accelerometers`

Martin, J. D., Krosche, J., & Boll, S. (n.d.). Dynamic gps-position correction for mobile pedestrian navigation and orientation. In *Proceedings of the 3rd workshop on positioning, navigation and communication.*
Mathie, M. (2003). *Monitoring and interpreting human movement patterns using a triaxial accelerometer*. University of New South Wales. Available from `http://books.google.com.ph/books?id=f87DtgAACAAJ`
Melissa Selik, A. B., Richard Baranuik. (2004, August). *Signal energy vs signal power.* Online.
Moell, V., & Horntvedt, A. (2012). *Positioning for mobile phones using wlan and accelerometer data*. Unpublished master's thesis, Lunds Universitet.
Nam, Y. (2011). Map-based indoor people localization using an inertial measurement unit. *J. Inf. Sci. Eng.*, *27*(4), 1233-1248.
Nanocomputers and swarm intelligence. (2008). London: John Wiley and Sons.
Parnandi, A., Le, K., Vaghela, P., Kolli, A., Dantu, K., Poduri, S., et al. (2009?). Coarse in-building localization with smartphones.
Platt, J. C. (1999). Advances in kernel methods. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), (pp. 185–208). Cambridge, MA, USA: MIT Press. Available from `http://dl.acm.org/citation.cfm?id=299094.299105`
Quinlan, J. R. (1993). *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
Renaudin, V., Susi, M., & Lachapelle, G. (2012). Step length estimation using handheld inertial sensors. *Sensors*, *12*(7), 8507–8525. Available from `http://www.mdpi.com/1424-8220/12/7/8507`
Ronald E. Walpole, S. L. M. K. Y., Raymond H. Myers. (2010). Scientists and engineers: Guide to probability and statistics. In (chap. 1). Pearson Education South Asia Pte. Ltd.
Shala, U., & Rodriguez, A. (2011). *Indoor positioning using sensor-fusion in android devices*. Unpublished master's thesis, Kristianstad University, School of Health and Society.
STMicroelectronics. (n.d.). L3g4200d mems motion sensor: ultra-stable three-axis digital output gyroscope [Computer software manual].
Susi, M., Renaudin, V., & Lachapelle, G. (2013). Motion mode recognition and step detection algorithms for mobile phone users. *Sensors*, *13*(2), 1539–1562. Available from `http://www.mdpi.com/1424-8220/13/2/1539`
Tan, S. (2008). Linear systems. In (chap. 8). University of Auckland.
Thanh, T. N., Makihara, Y., Nagahara, H., Mukaigawa, Y., & Yagi, Y. (2012a). Inertial-sensor-based walking action recognition using robust step detection and inter-class relationships. In *Icpr* (p. 3811-3814).
Thanh, T. N., Makihara, Y., Nagahara, H., Mukaigawa, Y., & Yagi, Y. (2012b). Performance evaluation of gait recognition using the largest inertial sensor-based gait database. In *Icb* (p. 360-366).
Walpole, R. E. (2009). Introduction to statistics. In (chap. 2). Pearson Education

South Asia Pte Ltd.

Won Kim, J., Jin Jang, H., Hwang, D. H., & Park, C. (2004). A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning System*. (shoe embedded)

Ying, H., Silex, C., Schnitzer, A., Leonhardt, S., & Schiek, M. (2007). Automatic step detection in the accelerometer signal. In S. Leonhardt, T. Falck, & P. M√§h√∂nen (Eds.), *4th international workshop on wearable and implantable body sensor networks (bsn 2007)* (Vol. 13, p. 80-85). Springer Berlin Heidelberg. Available from http://dx.doi.org/10.1007/978-3-540-70994-7_14