MASTER

Habitual behavior monitoring using smartphones
impact of sensor input and personalized models on walking patterns

van Dort, M.P.

Award date:
2013

Technical University Eindhoven
Signal Processing Systems

Final report

30 September 2013

HABITUAL BEHAVIOR MONITORING USING SMARTPHONES

IMPACT OF SENSOR INPUT AND PERSONALIZED MODELS ON WALKING PATTERNS

Author:

Marijn van Dort

Supervisors:

Dr. Oliver Amft
Marco Altini

TABLE OF CONTENTS

ABSTRACT

The lack of physical activity is often related to cardiovascular diseases. Simple exercise like walking or climbing the stairs can already decrease the risk of this type of diseases considerably. Activity recognition can play an important role in this topic. Finding an easy way to exactly monitor our daily behavior gives us a complete insight in our daily routines and can motivate us to exercise. Modern smart phones are equipped with a rich set of sensors and are a new alternative platform for activity recognition. But smart phones are worn at different locations and orientations. People may carry their smart phone in the front pocket of their trousers or in their purse or bag. In this thesis we try to recognize basic activities like walking, biking or running and estimate walking speed. We used two different smart phone locations (front pocket trouser and purse), without telling the subject how to store the smart phone. One solution to cope with the above mentioned viabilities is to use orientation independent features and location specific models. We show that activity can be recognized with a feature set that only depends on a person's motion variation, with only a minor decrease in performance. We also investigated the accuracy of three different speed estimation algorithms under these unconstrained conditions. In this study we show that the main performance accuracy depends on step frequency performance. An improved step frequency algorithm is proposed which improves all walking speed estimation algorithms.

1 INTRODUCTION

The absence of exercise is often associated with higher risk of cardiovascular diseases. Recent studies show that the number of people who suffer from overweight and obesity are growing. The lack or decrease of physical exercise is often related to this increase of overweight, which may result in cardiovascular diseases. This is one of the reasons that researchers are interested in the research area of habitual behavior monitoring. Epidemiologists are also interested in how physical movement evolves during a person's live time in terms of intensity, type and duration.

A problem in this kind of research is monitoring the human behavior without affecting the behavior pattern. Therefore wearable monitor systems are needed, which are able to monitor the person's physical movement during the daytime. Modern smart phones have a comprehensive set of different sensors, which may make them suitable for this kind of task. Therefore, smart phone-based activity monitoring systems can play a major role in patient care.

Due to its wearable character a new range of applications became possible. Instead of not trying to interfere in human behavior, we can actively steer this behavior. The main motivation is to prevent diseases. These applications have the interest of insurance companies and other programs that try to reduce health care costs.

Walking is one of the easiest physical activities and is also the most common. It can be performed by almost every one. It is cheap, it can be performed almost everywhere and it is easily integrated in a daily routine. This is why this type of activity is often chosen to prevent lack of exercise. The aim of this project is to use a smartphone based system to monitor walking patterns during the daytime.

1.1 THESIS OUTLINE

The thesis consists of 9 chapters. In chapter 2 background information is given. Then chapter 3 continues with an overview of previous work related to this project. Chapter 4 explains the methodology. Chapter 5 is about the soft- and hardware. This is a more practical section, but also tries to explain Android implementation choices. Chapter 6 clarifies the conducted protocol and gives a brief resume of the data recording document. Chapter 7 and 8 presents the results and evaluates the performance of the algorithms. Chapter 9 is the final chapter which concludes the work and discusses future work.

2 BACKGROUND

Analysis of the human or animal walking gait may be the oldest science there is. This science was the hunter's main tool, long before any technologies existed. Trackers were able to analyze hunted characteristics by analyzing their footprints. A tracker could determine the hunted gender, weight, physical condition or if the hunted was hurt.

The walking we know '*nowadays*' is several million years old and maybe the most common motion in the human daily routine. A lot of research is done on this motion. This research is done in different research areas and with different purposes. Some examples are disability-injury programs, screening healthy persons for medical prevention, fall risk assessment and neurological disorders. Resent sciences like gaming and robotics are also interested in walking models.

The motion picture industry is especially interested in the emotion of walking. During recent years they came-up with physical models that really revealed the emotion of the animated "action" figure. The Boulic-Thalmann "BT" model is an (simple) example of such a physical model [1]. This model is based on the average motion of a large group of healthy persons.

Most of these gait analyses use gait capture devices which are advanced systems used within laboratory settings, to capture the walking motion with high accuracy. There is a lot of literature on basic and fundamental research on walking gait, based on the following accurate measurement systems.

Footswitches are devices placed under the feet. This allows us to measure foot strike and foot off events of both limbs and timing between these events. The advantage of this system is that it is wearable and can be used in long time research. **Gait mats** consist of an array of pressure switches embedded along the length of a walking/running strip. With this technology it is possible to measure cadence, stride, speed and basic gait phases. The participant is free of any sensors, so it does not interfere with the walking gait. The limitation is of course the length of the walking strip. A cheaper version of this system is a paper strip and ink pads attached to a person's shoes. **Video analysis** records the motion with video equipment. Then the video is analyzed frame by frame during a post video capturing stage. Every movement of the body can be tracked within the video frames. This is done manual or automated by tracking reflective markers (there are non-commercial 2D based systems available, capable of semi-automated tracking). With these video analysis it is possible to track a point in time (x(t),y(t)). **3D motion** is the most accurate system for measurements, performed within 3D space. The system obtains a comprehensive overview of the gait analysis, technics to track a point in space and time. Besides capturing the motion, 3D motion capturing also consists of advanced software. The system can automatically extract features of the motion (joint coordinates, velocities, accelerations and angles) and apply these automatically to diagnostic models.

2.1.1 WEARABLE SENSOR APPROACHES FOR WALKING GAIT ANALYSIS

Abovementioned systems developed over time. The current state of the art systems measure exactly where every point is in 3D space and how it moves over time. Complicated models are built from these analyses. This kind of measurements works well for small working (lab) environments. But these settings can interfere with the normal gait and give incorrect results. The laboratory setting may not agree with daily-life behavior. Therefore there is a need for ambulatory monitoring systems to measure the human behavior without the constrained laboratory settings. Wearable sensors may overcome these restrictions. With the advent of micro electromechanical sensors (MEMS) technology this new research area is added to the gait analyses. One goal of this area is to build ambulatory monitoring systems to measure a person's functioning without interfering in his life. Most wearable sensors measure physical quantities, the motion of a body part can be calculated from these quantities. By means of integration, but integration can introduce errors. Especially if drift errors add up, orientation errors grow unbound (these errors tends to be bigger with inexpensive sensors). Another problem is that the initial orientation is not known, which makes them less suitable for absolute orientation measurement. A solution to these problems is to determine the accurate cycle phases of the motion. These phases can be used to remove these errors. The foot ground event can be used as an absolute orientation reference.

Walking is a quasi-periodic motion, with the left and right leg out of phase. Often one walking cycle is defined as the event of the heel strike until the next heel strike of the same limb occurs (Figure 1). This event is called heel or foot strike, which is the time that “any part” of the foot strikes the ground. The most basic measurements of the walking motion are the step length (step stride, two steps), step frequency and walking speed. These can easily be measured by measuring the time a person walks a known distance. From this measurement the average walking speed can be calculated, the average number of steps, the number of steps per minute (cadence) and the average step length.

These types of measurement are nowadays often based on Inertial Measurement Units (IMU). An IMU typically consists of tri-axial accelerometer, gyroscope and magnetometer. These types of sensors are common today and present in almost all modern smartphones. This is one of the reasons that these smart phones are often used in this type of research. Smartphones are technical not the same as dedicated systems, which gives rise to other technical challenges. On the other hand epidemic research is always looking for an ambulatory monitoring system to measure the human behavior. Most people are not aware of their smartphone, which makes these smartphones really suitable for this kind of research. But the challenge becomes more difficult when using smartphones, because not all people use and wear their smartphone the same way. In the next section it is shown that most previous work is not suitable for this kind of freedom.

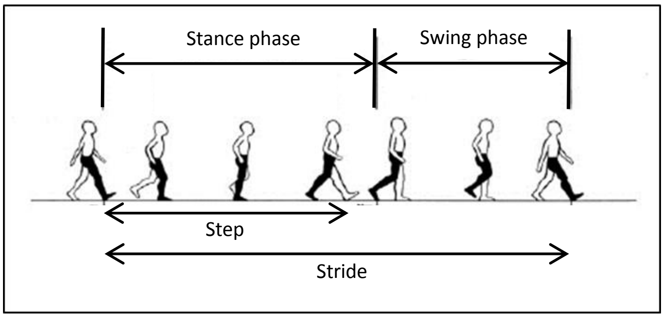


Figure 1 Normal gait for right limb (taken from Sutherland [2])

3 PREVIOUS WORK

Different strategies have been used by researchers to classify and estimate a person's walking speed. Often the problem is divided into two different sub-problems. Activity recognition and estimating the walking speed.

3.1 ACTIVITY RECOGNITION

The research of activity recognition is a lively research area. The first goal is to identify the type of activity. A more complicated task is to also assess the quality of this activity. Previous studies show that activity recognition can be performed with high recognition accuracy [13] however, the sensor was unrealistically attached to the body in a fixed position. More recent studies investigated some of the practical limitations due to the use of the smart phones in unconstrained settings, such as variability in phone location and orientation. The unconstraint environment in which the mobile phone is worn brings a new range of challenges: 1) Single sensor location, only one mobile phone to our disposal. 2) Different locations to wear the mobile phone. 3) Orientation of the mobile phone is unknown. People can put away their mobile phone in several orientations for a single location. 4) The rigid sensor connection, how closely the mobile phone follows body movement.

3.1.1 SINGLE PHONE SENSOR LOCATION

Most algorithms are based on multiple (2 to 10) sensor locations and have fixed location and orientation (thigh, sternum, shank, hip, upper leg, wrist, etcetera). However, literature has shown that basic activities can be identified with only a single sensor location [3]. Some of these locations correspond with common mobile phone locations, but most algorithms need a fixed location, fixed orientation and a rigid sensor connection. A rigid sensor connection ensures that the sensor follows the moving body part.

3.1.2 PHONE ORIENTATION AND LOCATION

Several studies investigated that the mobile phone can be worn at different locations and different orientations [4] [5] [6] [7] [8]. The orientation not only differs per location, but the orientation can also differ for a fixed location. Recent studies show two main approaches to solve these issues. 1) Transforming the coordinates system prior to the classification [5] [9]. 2) Finding features that are independent of orientation [7] [8] [10]. The first approach tries to rotate the sensor data into the same reference system as in which the classifier was trained. Studies show that this approach improves results, but only works well when the orientation variation is fixed. The second method seems to work better when the orientation is changing continually. The orientation independent features are often related to the motion intensity of the activity, by calculating the Root Mean Square (RMS) values or calculating the amplitude (interquartile range) of the resulting vector. The same strategy is used in solving the location problem. The advantage of the location independent method is that a single model can be used independent of the location. This approach is less sensitive to location misclassification because the location dependent method implies that first the location of the mobile phone is detected. Depending on the location the corresponding model is selected, this may lead to loss in speed accuracy. As already mentioned, orientation independent features are related to the activity motion intensity. But for different locations the measurable intensity differs, therefore we measure a complete other intensity range for another location. The measurable intensity variation is also seen when activities are performed over different (activity) intensities. This makes the activity recognition more complicated.

3.2 WALKING SPEED ESTIMATION

Estimating a pedestrian's step length and step frequency can be performed in numerous ways, but these methods strongly depend on the sensor location. Another problem is that most methods assume that the sensor is rigidly attached to the user's body, which is not correct for certain locations. The sensor is often placed on the foot, close to the Centre of Mass (CoM), hip or distributed over the leg. A good overview of the different locations used by different speed estimation algorithms is given in [11].

The different methods can be grouped into three main classes: The Kinematic models [12] [13] [14], Direct integration [26] [11] and Machine learning [15] [16]. Most previous studies are not suitable for the unconstrained smart phone situation. All kinematic models need accurate information over limb(s) orientation and limb(s) position in time. The direct integration method needs good walking gait event timing. These methods only work for a fixed sensor location and sensor connection. The unconstrained smart phone situation seems to be more suitable for the machine learning solution. In this approach the relation between speed and features is captured without building a physical model. Only this solution is at the current moment the least performing method of the three methods.

3.2.1 KINEMATIC MODELS AND DIRECT INTEGRATION

The kinematic models try to measure the displacement or angle of a certain body part and estimate the step length. Examples are the leg angle and the displacement of the hip. The direct integration is based on the deviation of the position from a known reference by double integration. The zero velocity update (ZUPT) is currently the most accurate method [11]. This method is called ZUPT, because it detects when the foot is on the ground and has zero velocity.

3.2.2 MACHINE LEARNING

These models do not use a physical model to estimate the walking speed. The human gait is seen as a black-box model. The black-box model contains a set of parameters which somehow have a relation between the walking speed and the measured data. This mapping can be established through training data. But after the parameter estimation, the model is easy and fast, so suitable for real-time implementation. This method is also suitable for different locations, but needs a learning stage per location. So a new location may need a complete redo of this learning stage. The accuracy depends on the completeness of the training data set. Environmental changes or walking motion not in the training set may lead to inaccuracies. Older studies are based on multiple sensor locations and fixed sensor locations. These solutions are not suitable for a smart phone. A reduced accuracy is reported when existing algorithms are used in this unconstrained location and orientation situation [17].

Some studies include personalized data into their models [18]. This data can be subject dependent data like height or weight. This type of data can be used as an algorithm setting. Other methods make use of a calibration (learning stage). This calibration improves results, but results depend on the accuracy of this calibration. Therefore automatic calibration methods are preferred.

3.2.3 ACCURACY OF DIFFERENT SPEED ESTIMATION ALGORITHMS

The accuracy performance differs per algorithm. The zero velocity update (ZUPT) has the best performance RSME 0.18 km/hour over a walking speed range 3 - 6 km/hour [19]. However this method is not suitable for a location independent implementation. The kinematic model also shows good results of RSME 0.2 km/hour, but this performance is obtained with multiple sensors. The machine learning models are the least performing methods. These types of models are difficult to compare, because the results depend heavily on the way the experiment is conducted. But to give an indication, RMSE 0.5 km/hour can be expected for this type of algorithms, within a limited experimental environment.

3.2.4 STEP FREQUENCY

There are a couple of algorithms for estimating the step frequency. Some could be useful to implement on a mobile phone, others are useless. Examples are the ZUPT or ZARU, these methods detect a zero velocity and/or zero angular rate within the walking gait. These techniques can only be used when the sensor is mounted to the foot or lower leg [26]. Another method is DFT, which allows decomposing of periodic signals into a sum of sinus waves. It has been described in numerous papers to estimate the step frequency [16]. It is also used to distinguish cyclic movement from non-cyclic movement. Another method which can be used to estimate step frequency is zero crossing detection [20]. As the name suggests, it detects when the signal changes from sign. At last there is the peak detection method. Signals of the sensors often show sharp peaks, with high amplitudes compared to the rest of the signal. These sharp peaks are somehow related to the foot strike event. They can be used to detect the foot contact [27]. This study gives insight into the change of the accelerator signal shapes over different walking speed and the relation to the stride cycle. Although this study has a good result in detecting the step frequency, it does not investigate how the signal shape changes and how the signal shape is influenced by the sensor coupling to the body for different locations. For example, a mobile phone worn in a purse can have a complete different signal shape when conditions are kept constant.

3.2.5 PERSONALIZATION OF SMART PHONE BASED MODELS

Some algorithms use a constant stride length while estimating the walking distance. The users must manually calibrate their stride length themselves [20]. Considering that the stride length is variable within the human walking patterns, such calibration may result in distance errors. Some studies show that there is a relationship between step frequency and stride length and that the stride length increases with the step frequency [21]. Other studies describe the step frequency walking speed relation with an exponential curve [22]. Other studies estimate the distances per step, the accuracy of these methods are compared in [16]. But all these methods need a subject dependent constant which may require manual input. Both active and semi-supervised learning of personal data is investigated in [18]. It is understandable that personal data improves model accuracy of speed estimation, but automatic approaches providing personalized data are still lacking.

3.3 OBJECTIVE

For the basic activity recognition we investigate if activity recognition is possible with orientation independent features alone. Because this type of feature is less informative, a performance penalty can be expected. Previous studies show that there are two promising solutions to solve the location and orientation issues for speed estimation, namely using different models per location or using a single model that is independent of the location. In this study these two solutions are explored and compared. We only selected methods which depend on a single constant. These methods are easily calibrated in a personalized calibration routine. The outcomes of these accuracies are compared to speed estimation methods that only depend on the person's length. Most studies only report the speed accuracy for the used location, but how accurate are they when the location is "misclassified"?

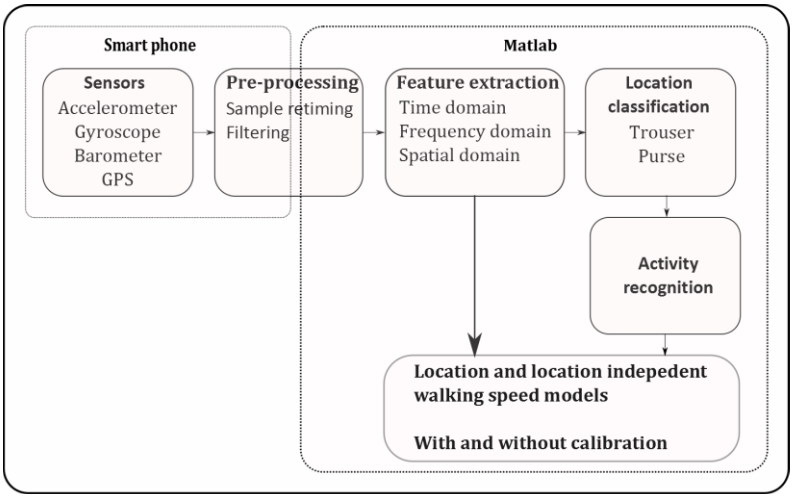


Figure 2 Block diagram of the different methods used. 1) Raw sensor reading on the smart phone. 2) Pre-processing. 3) Feature extraction. 4) Location classification (trouser and purse). 5) Basic activity recognition. 6) Speed estimation models, location and location independent methods.

4 Methodology

The approach we follow in this study will be based on a machine learning solution. Next to detecting steps and estimating walking speed, the actually walking periods must be classified. Also walking up-down the stairs is classified.

4.1 PRE-PROCESSING

We applied different low pass filters to filter out frequency bands and to remove the unwanted noise from the signal of interest. The sensor data stream are divided into consecutive frames of 4 seconds each, with an overlap of one second. Different window time lengths were tested. Most studies use time windows between 2.5 seconds and 5 seconds [10] [17]. Longer lengths will decrease step detection resolution with exchange of improved accuracy. A longer window time length gave no big improvement in accuracy. Shorter window time lengths resulted in decrease of performance. For each window 160 features where calculated. Appendix A summarizes the different features and gives a brief description per feature.

4.2 FEATURE EXTRACTION

The extracted features from the smart phone are used to derive location, basic activity and estimate walking speed. In this project we used two locations where a mobile phone often is worn, in the trouser and in the purse. The basic activity recognition was performed to classify 8 different activities, independently of the smart phone orientation Table 2. The accelerometer and gyroscope are used for capturing the walking gait. Several time features from the square sum of three components on x, y and z-axes and frequency domain features are extracted. The time domain features relate to the intensity of the activity. The primary features we use are the magnitudes of the low frequency portion of the signal spectrum. Because human walking is cyclic, the frequency component of the discrete Fourier transform (DFT) of the acceleration and gyroscope signal contains information about this walking motion. From the spatial domain the mobile angle variation is used. Some body parts make a rotational movement during the walking motion, which result in angle variation of the mobile. The simple single signal outputs are not used because of the orientation dependent nature of these features. The horizontal and vertical components (compared to the earth gravitation) of the acceleration are computed with the rotation matrix [5]. Some speed estimation methods relate the displacement of the up-down movement of the upper body to walking speed.

4.2.1 FEATURE NORMALIZATION

Because people are different, different normalizations are proposed. Normalization allows us to compare a single person to a group of people. There are useful normalizations proposed based on anthropometrics. In Table 1 the walking frequency is normalized [21], the goal is to compare a single person with a group of people.

Quantity	Symbol	Dimension	Dimensionless
Frequency	F	T^{-1}	$\frac{f}{\sqrt{\frac{g}{l}}}$

Table 1 Dimensionless step frequency (Hof, A. L. [23]), f is the step frequency, g the earth gravitation and l is the leg length.

4.2.2 FEATURE COMBINATION

For the feature combinations the Mutual Information Feature Select (MIFS) and Fisher composite algorithms where used. A nice review about this feature sub-set selection algorithms is given in [24] [25]. The idea of these algorithms is to find the best feature sub-set combination from the complete feature set.

4.3 ESTIMATING PDF PARAMETERS AND CLASSIFICATION METHODS

We choose the maximum likelihood technique as a parametric estimation for the unknown probability distribution function (PDF). So the unknown PDF function is approximated by a Gaussian distribution. The used classifier is based on the Bayes decision theory. More detailed information about the used methods can be found in [25].

4.4 THRESHOLD CLASSIFIER

The threshold method is the most commonly used method to classify static from dynamic movement. Both the linear accelerometer and gyroscope sensors are used for this classifier we took the average length of the vector of the sensor to make the features independent of the orientation. This results in a measurement that relates to the activity intensity, but is independent of the orientation. The feature value is the average over the complete window time length. From the test data the minimum intensity levels are learned. The minimum levels are used as threshold values in the threshold classifier. Levels below the thresholds are classified as static sedentary activity. When one value or both values are above or equal to the threshold levels, the feature vector is classified as a dynamic activity. This method can also contribute to reduce power consumption. By reducing sample rate at an early stage, the battery live time improves which may be the difference between monitoring a full day or monitoring only a couple of hours.

4.5 ACTIVITY CLASSIFICATION BY MEANS OF DECISION TREE

The decision tree is used to partition the activity problem into smaller sub-problems. Figure 3 depicts the used classification architecture. The first branch separates the sedentary activities from the dynamic activities. In the next branch the location of the mobile phone is classified. The tree then unfolds in multiple sub-trees all with a similar branch structure. Only a different feature set is used at the decision nodes.

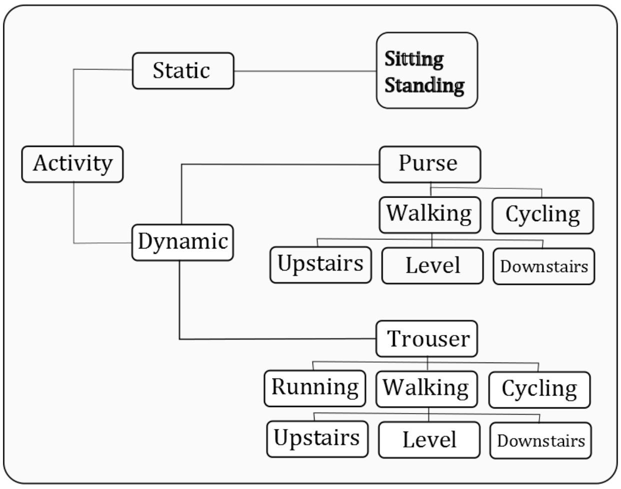


Figure 3 Used classification architecture.

4.6 SPEED ESTIMATION

All speed (v) estimation algorithms depend on a relation between the number of steps taken per time unit, step frequency (f) and an optional feature vector ($\vec{X}$) estimating the distance travel per step ($d(\vec{X})$) (1).

$$v = d(\vec{X}) * f \tag{1}$$

The speed estimation algorithm can be divided into two independent problems, the step frequency estimation and the step distance estimation. Most papers only report the overall performance of the estimated speed and do not investigate the sub accuracy of the two different problems. There are many studies that report walking speed accuracy. But due to the lack of a performance evaluation protocol, it is difficult to compare these results [11]. An example is the speed range, the largest errors are made at the edges of 2.5 – 6.0 km/hour, which makes it difficult to compare the results with experiments done over a more limited or other speed range.

4.6.1 STEP FREQUENCY

There are multiple ways to estimate the step frequency. In this project the following step frequency methods where investigated: Discrete Frequency Transformation (DFT), zero crossing and peak detection. These methods are applied on different signals: 1) total length of gyroscope and linear accelerometer vector, 2) angle variation of the mobile phone and 3) the up-down acceleration. For the DFT and zero cross algorithm the traces are first filtered with a low pass filter with a cut off frequency of 2.5 Hz because the step frequency is expected to be lower than this cut off frequency. For the peak detection no filtering is applied to preserve the peaks within the trace.

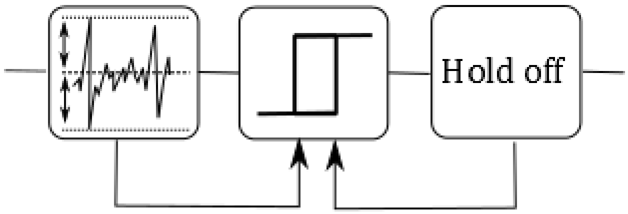


Figure 4 Peak detection

Figure 4 shows the schematic of the peak detection algorithm. In a sliding window with duration of 4 seconds the maximum and minimum signal levels are found. These levels are used as dynamic thresholds for the Schmitt trigger. These trigger levels are set to 0.75% of the maximum and minimum levels found in the time window. When the Schmitt trigger is triggered a hold off time is set to prevent retriggering due to short ringing effects.

4.6.1.1 IMPROVED STEP FREQUENCY ALGORITHM

Different physical walking events (foot strike or thigh, upper body rotation) are qualified to be measured by a particular sensor. The foot strike is often shown in the trace as high spiky peaks. At certain locations this event is the most dominant measurable event in the sensor data. When the sensor is placed at the hip, the up-down movement of the hip can easily be detected, the force experienced by the sensor is directly linked to the walking motion. The frequency of the up-down movement then equals the step frequency. When the mobile phone is fixed on the user's torso, the rotational motion in the upper-body equals the stride frequency, which is half of the step frequency. A potential issue is that depending on the location different types of "*walking"* frequencies are measured. Over the dynamic walking range the dominant event may also change. At low speed 1 we may measure frequency 1 and at a certain increased speed 2 we may measure another frequency 2. For example at low speed when the mobile phone is in a purse the foot strike is often the dominant event, but when the walking speed increases the up-down movement becomes bigger. This is why some step frequency algorithms change from stride frequency to step frequency as the walking speed increases. The result is that the algorithm measures that the person is walking twice as fast. The exact speed at which this occurs depends per person. Walking the stairs often shows a four times higher frequency as the stride frequency, because the up-down acceleration of climbing the stairs becomes visible. Some mobile phone locations do not have a single dominant frequency. For example when the mobile phone is placed in the front pocket trouser most step-frequency algorithms will switch between the different events, which results in switching between reporting step, stride frequency or harmonics of the stride frequency.

The following algorithm was used to improve the frequency accuracy, by tackling the switching problem. In other words, the different frequency estimation methods agree on the frequency most of the time (all three reporting the same frequency). Only sometimes they switch to another mode and report another harmonic frequency (but unlikely all at the same time). The following algorithm tries to use this, when the three methods agree the algorithm locks to that frequency. When locked and the frequencies do not agree it tries to be close to the previous found frequency, by selecting the frequency which is closest to the previous frequency. If a person is walking at constant pace, the step frequency will not change so much. This resulted in the following algorithm.

```
1.  f_select = median(f1,f2,f3)
2.
3.  do {
4.      if f1 == f2 == f3
5.              f_select = f1
6.      else
7.              f_select = closest_to_freq(f1,f2,f3)
8.
9.      report f_select
10. }
```

$f1, f2$ and $f3$ are the estimated step frequencies from three different frequency estimation methods. In line 1 the algorithm is initialized by choosing the most likely frequency. If two frequencies agree then this frequency is selected. The variable f_select is the estimated walking frequency.

At line 3 the algorithm goes into a countless loop. The '*if*' statement at line 4 checks if all frequencies agree.

- Yes, Then this is the new walking frequency (line 5).
- No, the frequency (*f1, f2 and f3*) closest to the previous frequency is chosen as current walking frequency (line7).

4.6.2 Speed estimation with personalized calibration

Speed estimation algorithms are often calibrated to improve accuracy. In this project three different speed estimation formulas (2, 3 and 4) are used in the context of personalized calibration.

$$v = cf^{1.72} \tag{2}$$

This formula only depends on the step frequency, which is the main motivation for the formula. The walking speed is indicated by v, f is the step frequency and c the personalized constant. This formula is explained in [22]. This paper also explains for which conditions this formula is applicable. This formula is location independent.

$$v = c\sqrt[4]{A_{max} - A_{min}} \cdot f \tag{3}$$

Formula (3) is often used for walking speed estimation [28] [29]. The 4th root is a step length estimator. The 4th root takes the maximal (A_{max}) and minimal (A_{min}) vertical acceleration within the time window. The vertical acceleration is calculated by means of the rotation matrix scheme. The idea of this formula is that the vertical acceleration is related to the vertical displacement of the hip. Walking speeds can be estimated form the displacement of the hip [14].

$$v = c \cdot average(|\overrightarrow{a_{lin}}|) \cdot f \tag{4}$$

In the last formula (4) the step length is estimated by means of averaging the total length of the linear acceleration ($\overrightarrow{a_{lin}}$) vector, within the time window. This feature is used because it had the highest correlation with the actual reference speed. These last two formulas (3 and 4) need a different constant per location. So every possible location needs its own calibration routine.

4.6.3 SPEED ESTIMATION WITH GENERALIZED MODEL

All so methods which only depend on the person's length are investigated. The following approaches are applied: 1) step frequency nominalization Table 1, 2) the personal constant c which depends of the length and 3) linear regression model.

$$v = c(f')^{1.72} \tag{5}$$

In this formula (5) $f'$is the normalized step frequency. The constant c is the averaged constant found for the learning group.

$$v = c(l)\sqrt[4]{A_{max} - A_{min}} \cdot f \tag{6}$$

For this formula (6) the constant is transformed into a linear function depending on the subject length (l). This function is estimated from the test group.

$$v = h(l, f, p_4, A_{ver}) \tag{7}$$

The third formula (7) is a linear regression method depending on the length (l), step frequency (f), the power of the four biggest frequency components in the linear acceleration spectrum (p4) and the RMS value of the up-down acceleration A_{ver}.

4.6.4 SPEED ESTIMATION BY MEANS OF GPS CALIBRATION

Formula (2) is also used in an automatic calibration scheme. The idea is to solve the personalized constant during walking periods when GPS speed data are available.

4.7 TRAINING SET – PERFORMANCE ESTIMATION

To validate the activity classifiers and speed estimators the data set is divided into a training/learning set and test set. The parameter estimation is done on the learning set. The performance of the classifier is measured on the test set. All classifier performances are measured with the leave one person out method [21].

5 HARDWARE AND SOFTWARE

There are a couple of different smart phone operating systems available. We chose the android system. Not because it is one of the main players in the operating systems world, but because of the Context Recognition Network Toolbox (CRTN+) toolbox. This toolbox is especially developed for these kinds of experiments and available on the Android platform.

5.1 ANDROID SENSORS

The new smart phones contain a big range of sensors. This makes a number of potential new applications possible for the mobile phone. The introduction of sensors started at Android 1.5, in Android 2.3 new sensors and tools where added. This makes Android 2.3 the preferable Android version for sensor development. Android does not restrict mobile builders to specific hardware. Sensors on different mobiles may have different ranges, resolutions and sample rates. Also synthetic sensors have different implementations and may use different types of sensors for calculating the physical quantities. Generally it is preferred to use synthetic sensors. The synthetic sensor implementation often gives the best results for the specific hardware.

5.2 ROTATION MATRIX / ORIENTATION

Smart phones can determine their orientation within the earth gravitation field. Older devices only used the magnetometer to determine the orientation. This type of sensor is slow and sensitive for electromagnetic interference. Newer devices improved by using multiple sensors accelerometer and/or gyroscope. This is called sensor fusion, the best characteristics of the particular sensor type is used and combined into an improved synthetic sensor. Often the gyroscope is used for fast orientation variations. But for slow orientation variations it is not that accurate anymore, due to drift errors.

The developer has, starting from Android 2.3, more tools to work with the rotation matrix. This was the main motivation to update the CRTN+ toolkit to this Android version. The Android operation system has some functions available to do these kinds of calculations quickly.

The orientation angles (pitch, roll and yaw) are based on the content of this rotation matrix. These orientation angles have discontinuities in their output. This is why we do not use the orientation directly, but a derivative. Appendix C gives a more mathematical view on the rotation matrix.

5.2.1 INTERPOLATION

It is possible to request a sensor sample rate in the Android 2.3 API, but documentation reveals that the actual sample rate maybe higher than the requested sample rate (within the limit of the maximum sample rate). A sample rate test showed that the sample rate was not stable. On the mobile phones (Sony Ericsson Active, Samsung S3 and nexus 4) the sample rate varied. Therefore interpolation is used in this project to improve the sample rate. A constant sample rate is essential for further signal processing. To prevent discontinuities the cosine interpolation is chosen as method.

5.3 CTRN TOOLKIT

The CRNT+ is a software framework for easy development of activity recognition applications. The development of the toolbox started at the University of Passau. Later a sub-set of this toolbox is ported to the Android operating system by Jakob Weiger and extended by Frank Roberts at the University of Eindhoven [30].

This toolbox is a framework and aims to make these kinds of projects fast and easy. The toolbox globally consists of three parts:

1. The readers or sensors: The toolbox has a comprehensive set of sensors available and ready to use. These are not only phone internal sensors, but also external sensors connected by some kind of interface. The integration of new sensors is also made easy, due to the open framework.
2. The writers: The writers take care of the output. The output can be written to file or FTP and different sensor outputs can be merged into a single output stream.
3. The middle part, the service/user interface (UI) application. With this part the *service* can be started and stopped. UI capabilities for annotations (the labeling of the data) are included. The *service* is a special Android module, which runs in the background and runs independent of the phone usage.

The toolbox was based on version 2.2. This meant that not all internal phone sensors were available. Therefore we updated the toolbox to 2.3 and the missing sensors were added. Instead of toolbox phone timing we used internal phone timing, because the CRTN+ does not use the timestamp associated with the sample (this causes jitter, because timing depends on the processing time).

6 DATA RECORDING

6.1 DATA RECORDING PLAN

The data recording resulted in a dataset with more than 26 hours of annotated data from 20 participants. The group consisted of 14 male and 6 female self-reported healthy students or employees at Eindhoven University of Technology. Mean age was 29.4 years, mean weight was 70.2 kg and mean height was 1.76 meter. Participants were instructed on the study protocol. The participants were asked to perform 8 different activities: treadmill walking, climbing stairs, outdoor walking, uphill walking, running, biking, kitchen work and two stationary activities standing and sitting. Table 2 gives an overview of the different activities. Every activity was carried out twice, one time with the smartphone in the front pocket en one time with the smart phone in their bag. These locations are the most preferred locations to wear a smart phone. The recordings consisted of the two sessions per location of the mobile phone, a treadmill session at the gym and a free-living session carried out both in a home-like setup and outdoor session.

- The front pocket of the jeans, the body of the phone is mostly aligned with the participant's leg. This is the most comfortable way to wear the mobile phone. Hereby the y-as of the accelerometer will point upwards or downwards.
- The second place was a purse. In this experiment only a shoulder bag was used. The participant was not asked to put the mobile phone in a special way. This means that the orientation of the phone was random per participant.

Activity	Time (min)	Remark
Treadmill walking purse/trouser	8/8	Walking treadmill controlled 2.5 – 6.0 km/h
Treadmill running trouser	1	Running treadmill controlled 7.0 km/h
Climbing stairs purse/trouser	~1.5/1.5	Walking up and down three floors, self-paced
Outdoor walking purse/trouser	~10/10	Walking self-paced: 700 meters track
Walking uphill purse/trouser	~3/3	Walking treadmill angle 5,10 degree, speed 3.0 km/h
Biking outdoors purse/trouser	~10/10	Biking self-paced: 2200 meter track
Kitchen work trouser	1	Uncontrolled kitchen cleaning
Stationary activities trouser	2	Sitting, standing

Table 2 Overview of conducted activities

At the treadmill the participant performed two sessions of 8 and 9 minutes. During these sessions the speed of the treadmill was increased from a slow (2.5 km/h), moderate (4.5 km/h) to a brisk walking speed (6.0 km/h) in steps of 0.5 km/h. The treadmill was programmed to keep a constant speed for 1 minute and then increased speed by (0.5 km/h). For the trouser location the participant was also asked to run (7.0 km/h) for 1 minute. Climbing the stairs included two sessions for the two different locations of the mobile. Climbing the stairs was executed on one speed. The participant was asked to walk up and down the stairs at his own pace. The biking and outdoor walking activity consisted of an outdoor track over the campus. The participant was asked to walk and bike the indicated track at his own pace. The kitchen work and stationary activities were recorded in an office and kitchen environment. Every single activity performed by each participant was monitored by the supervisor.

6.2 DATA COLLECTION

For the data recording the nexus 4 phone was used. This phone was at the moment of this work one of the top of the bill smartphones and has a comprehensive set of sensors available. For this project the following sensors were used:

- Accelerometer
- (synthetic) Linear Accelerometer
- (synthetic) Gravitation
- Magnetometer
- Gyroscope
- Barometer
- GPS

This smart phone has an impressive sample rate of 200 Hz for the faster sensors accelerometer and gyroscope. During the data recording all sensors were recorded and saved to a text file on the SD-card of the mobile phone.

After the recording the data was transferred to an offline computer where the data was used as input for a couple of algorithms, which were written in Matlab.

- The data was retimed by means of interpolation.
- The time data series were labeled according to location and activity.
- The features were extracted per activity/subject.
- The features of a single subject were combined into a single subject file.
- The single subject files were combined into a single feature file for the complete group. This single feature file was the base for the classification and speed algorithms.

7 ACTIVITY CLASSIFICATION RESULTS

Figure 3 in section 4.4 depicts the used classification architecture. In the tree we try to answer every branch node question, until we reach a leaf node and determine the activity. We follow the tree structure in this chapter, by first trying to distinguish between sedentary and dynamic activities. Next the location was determined and then the walking activity was separated from running and biking. When the subject walked, the type of walking (up-down stairs or level walking) was distinguished.

7.1 CLASSIFICATION OF SEDENTARY - DYNAMIC ACTIVITIES

The sedentary activities were separated from the dynamic activities by means of the threshold classifier.

Threshold classifier			
Actual		**Sedentary**	**Dynamic**
	Sedentary	3.2%	0.3%
	Dynamic	0.1%	96.4%

Threshold classifier			
Actual		**Sedentary**	**Dynamic**
	Sedentary	90.4%	9.6%
	Dynamic	0.1%	99.9%

Metrics	
Norm. Acc.	95.2%

Table 3 Threshold Classifier

Table 3 depicts the performance of the classifier. The performance was measured with the leave- one-person out method. This classifier can be used to separate the sedentary activities from the dynamic activities, but the main motivation for this type of classifier was that it can reduce battery power consumption. Norm. Acc. is an abbreviation of the normalized accuracy metrics. This metric was chosen because of the class imbalance.

7.2 SMART PHONE LOCATION RECOGNITION

In this stage we tried to determine the location of the mobile phone. Figure 5 shows a bar graph of the Fisher Discriminant Ratio (FDR) [25] single feature measure. The higher the FDR value the better the discriminating power of individual features between the two locations. The feature set contained both orientation dependent as independent features. The graph shows that two features have a high outcome. All of the high scoring features depended on the orientation.

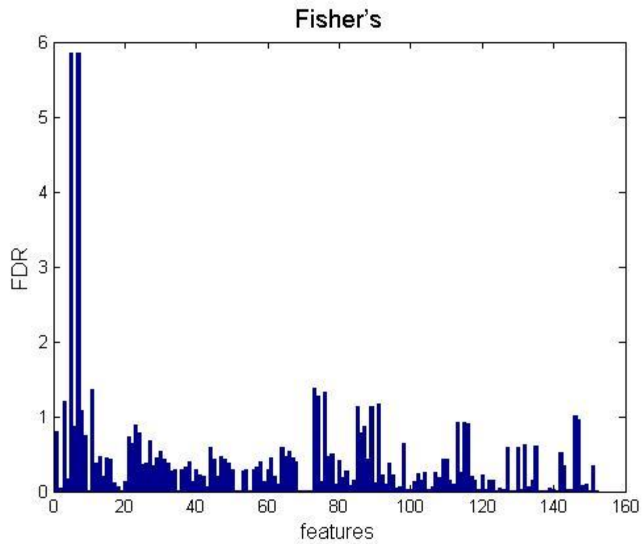


Figure 5 FDR [21] measure for location classifier. The mobile phone located in the front pocket, the y-axis was almost always pointing towards the earth gravity (average value about ±1g). For the purse this was the least possible orientation (average value about 0g). So it is easy to understand why this feature scored so high.

Table 4 shows the performance for different feature selection methods, type of sensor(s), orientation and dependent-independent features. The performance was measured with the Bayesian classifier True Positive Rate (TPR). Not only the results differ per used method, but also the number of features needed for the result. Orientation dependent means that only one orientation dependent feature was allowed, in this case the average of the y-axis. The result improved by adding this orientation depended feature by 6% - 10%. There is discussion if it is allowed to use orientation dependent features [16]. Therefore only orientation independent features were used.

	Type	Average	Nr features
1	**divergence 1 *orient - independent***	85.1%	8
2	**divergence 2 *orient - independent***	83.7%	7
3	**Fisher composite *orient - dependent***	91.1%	11
4	**Fisher composite *orient - independent***	84.8%	10
5	**MI composite *orient - dependent***	91.3%	4
6	**MI composite *orient - independent***	85.9%	5
7	***Orient - dependent***	91.3%	4
8	***Orient - independent***	90.4%	3
9	***Orient - dependent* Accelerometer**	88.3%	4
10	***Orient - independent* gyroscope**	83.3%	5

Table 4 performance for different sub selection methods [21].

Table 4 also shows the performance if only one sensor type was used (only gyroscope or accelerometer (9,10)). The accelerometer performs better than the gyroscope, but needs an orientation dependent feature for this result. Both methods alone performed worse than when both sensors were combined.

The number of features needed for the classifier differs per method. These methods (5,8) were selected because of the performance and because they required the least number of features:

1. Power of first harmonic gyroscope, this measures the cyclic rotation in movement.
2. Amplitude angle variation, a low frequency measure for rotational movement.
3. Ratio between harmonics H1 and (H2, H3 and H4) gyroscope.

All these features are related to the rotation of the movement.

Orientation dependent			
Actual		**Trouser**	**Purse**
	Trouser	41.2%	3.9%
	Purse	6.3%	48.5%

Orientation dependent			
Actual		**Trouser**	**Bag**
	Trouser	91.3%	8.7%
	Purse	11.5%	88.5%

Metrics	
Norm. Acc.	89.9%
TPR	91.3%
TNR	88.5%

Orientation independent			
Actual		**Trouser**	**Purse**
	Trouser	37.6%	4.0%
	Purse	11.9%	46.5%

Orientation independent			
Actual		**Trouser**	**Bag**
	Trouser	90.4%	9.6%
	Purse	20.4%	79.6%

Metrics	
Norm. Acc.	85.0%
TPR	90.4%
TNR	79.6%

Table 5 Performance metrics location

Table 5 shows performance metrics for the location classifier based on the orientation dependent and orientation independent feature set (to show the difference between these two types of features).

Figure 6 displays how the errors were distributed over the different activities for the orientation independent feature set. The walking speed varies from 3.0 to 6.0 km/h for both purse and trouser locations. In general the intensity measured for the purse location is smaller than the intensities measured for the trouser location. The high walking speed of the purse lay in the same vector space as the low walking speed of the trouser. The classification error showed the biggest error for the 'walking 6.0 km purse' activity, which was misinterpreted as 'slow walking trouser'. The result shows a potential problem for the orientation independent features which are based on the motion intensity. The bag classifier performance decreases from 88.5% to 79.6%. The trouser classifier was less sensitive: 91.3% to 90.4%.

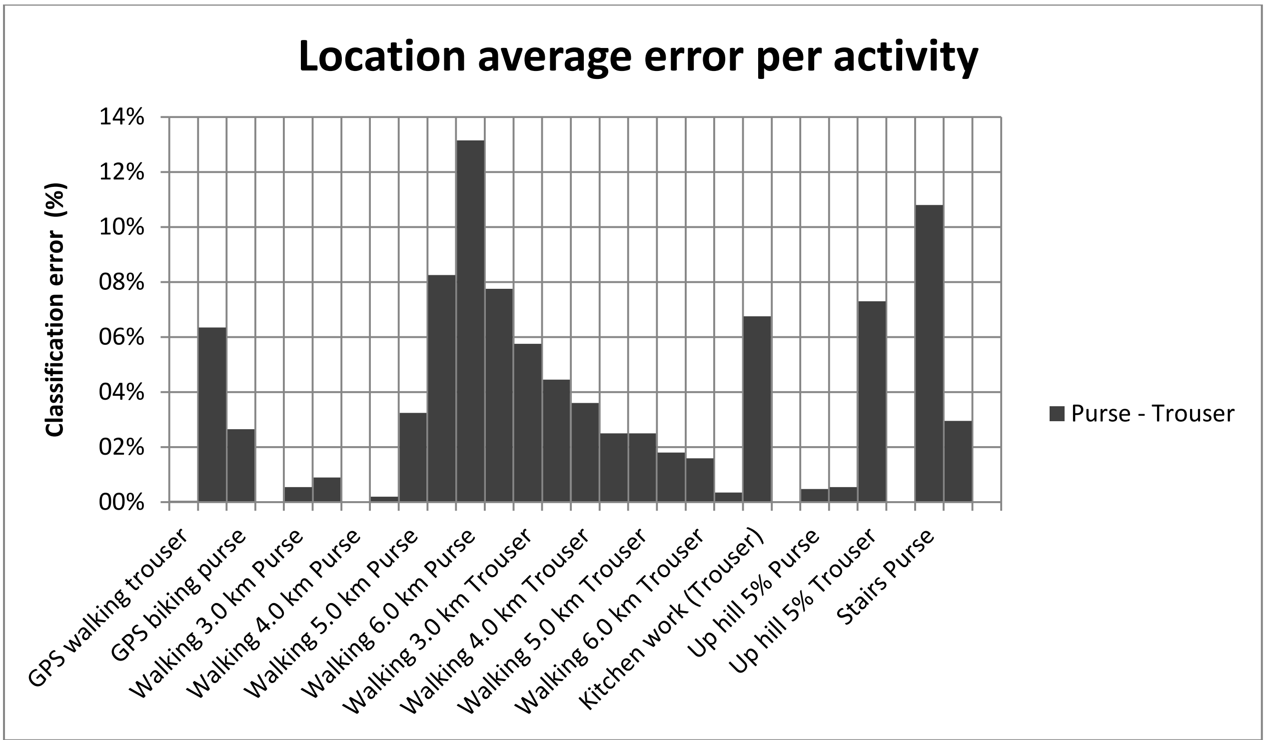


Figure 6 Average errors per activity for location classifier

7.2.1 Activity Type Recognition walking, running and biking

After the location classification we tried to classify if the performed activity was walking, running or biking. For the purse location only walking and biking was classified.

7.2.1.1 Purse

The following features were used for the classification of walking and biking with mobile phone in the purse (found with the MIFS):

1. Motion intensity measured linear acceleration (Length of sensor vector).
2. Motion intensity value of up and down acceleration (Length of sensor vector).
3. Motion intensity gyroscope, rotation movement (Length of sensor vector).
4. Ratio peak value and RMS value linear up and down acceleration. Walking motion often shows high acceleration peaks compared to the average value.
5. Walking step frequency.
6. Ratio peak value and RMS value gyroscope.
7. Speed from GPS.

Without GPS			
		Walking	Biking
Actual	Walking	70.5%	7.4%
	Biking	4.1%	17.9%

Without GPS			
		Walking	Biking
Actual	Walking	90.5%	9.5%
	Biking	18.8%	81.2%

Metrics	
Norm. Acc.	88.9%

With GPS			
		Walking	Biking
Actual	Walking	74.6%	0.4%
	Biking	0.0%	24.9%

With GPS			
		Walking	Biking
Actual	Walking	99.4%	0.6%
	Biking	0.1%	99.9%

Metrics	
Norm. Acc.	99.6%

Table 6 Performance metrics walking/biking purse with and without GPS feature. The performance in the version without GPS speed was reasonable, but less accurate than for projects done in less unconstrained environments. The result really improved when the GPS speed was added and resulted in better performance than algorithms that do not use GPS.

7.2.1.2 Trouser

The following feature set was used for the classification of walking, biking and running with mobile phone in the trouser:

- Motion intensity measured linear acceleration (Length of sensor vector).
- Walking step frequency.
- Inter quartile range ratio vertical/horizontal linear acceleration.
- Motion intensity gyroscope, rotation movement (Length of sensor vector).
- Largest four DFT powers linear acceleration.

Without GPS			
	Walking	Biking	Running
Walking	23.5%	2.3%	0.1%
Biking	1.0%	28.8%	2.8%
Running	6.5%	3.3%	31.7%

Without GPS			
	Walking	Biking	Running
Walking	90.7%	9.0%	0.4%
Biking	3.2%	88.3%	8.6%
Running	15.7%	7.9%	76.4%

Metrics	
Norm. Acc.	85.1%

With GPS			
	Walking	Biking	Running
Walking	71.9%	0.3%	4.6%
Biking	0.3%	22.9%	0.0%
Running	0.3%	0.0%	4.0%

With GPS			
	Walking	Biking	Running
Walking	93.6%	0.4%	6.0%
Biking	1.4%	98.6%	0.0%
Running	5.9%	0.3%	93.8%

Metrics	
Norm. Acc.	95.4%

Table 7 PERFORMANCE METRICS WALKING/BIKING/RUNNING TROUSER with and without GPS feature. Identical as for the purse location, the performance in the version without GPS speed was reasonable, but less accurate than for projects done in less unconstrained environments. The result really improved when the GPS speed was added and resulted in better performance than algorithms that do not use GPS.

7.2.2 Activity Type Recognition Normal Walking and (up-down) Stairs

For climbing (up-down) stairs the barometer sensor was added as an extra feature to the feature set. This feature was measured in a larger computation window length than the other features, because of the noise on the barometer reading. The calculation was done in a 6 second window, instead of the normal 4 seconds window.

It is a good compromise because it takes about 6 seconds to climb the stairs between two floors and this window gave good results. The following feature set was used for the classification of walking (up-down) the stairs (found with the MIFS):

1. Ratio between peak/RMS up-down acceleration.
2. Elevation barometer.

Trouser				Trouser				Metrics	
	Walking	Up	Down		Walking	Up	Down	Norm. Acc.	67.8%
Walking	20.1%	6.4%	1.2%	Walking	72.4%	23.2%	4.4%		
Up	2.2%	25.1%	11.9%	Up	5.6%	64.1%	30.4%		
Down	1.1%	9.9%	22.1%	Down	3.4%	29.8%	66.8%		

Trouser (barometer)				Trouser (barometer)				Metrics	
	Walking	Up	Down		Walking	Up	Down	Norm. Acc.	93.4%
Walking	25.0%	1.9%	0.9%	Walking	90.1%	6.7%	3.2%		
Up	0.7%	36.7%	1.7%	Up	1.9%	93.7%	4.4%		
Down	0.3%	0.9%	31.9%	Down	1.0%	2.6%	96.4%		

Purse				Purse				Metrics	
	Walking	Up	Down		Walking	Up	Down	Norm. Acc.	66.3%
Walking	18.4%	7.4%	2.3%	Walking	65.6%	26.3%	8.1%		
Up	9.4%	26.3%	1.6%	Up	25.1%	70.5%	4.3%		
Down	10.5%	2.4%	21.7%	Down	30.4%	6.9%	62.8%		

Purse (barometer)				Purse (barometer)				Metrics	
	Walking	Up	Down		Walking	Up	Down	Norm. Acc.	94.7%
Walking	27.6%	0.3%	0.1%	Walking	98.6%	1.0%	0.4%		
Up	1.0%	35.0%	1.3%	Up	2.8%	93.8%	3.4%		
Down	1.2%	1.7%	31.8%	Down	3.3%	4.9%	91.8%		

Table 8 PERFORMANCE Walking Up-down stairs for both locations, with and without barometer feature. The results show that no reasonable classifier performance was possible without the barometer. Table 8 demonstrates that the barometer feature improved this classification.

8 STEP FREQUENCY AND SPEED ESTIMATION PERFORMANCE

As mentioned before, the speed estimation algorithm can be divided into two independent problems, the step frequency estimation and the step length (formula 1). Most studies only report the overall performance of the estimated speed and do not investigate the sub-accuracy of the two different problems. This was the main motivation to measure the performance independent, instead of the overall performance.

8.1 STEP FREQUENCY PERFORMANCE

The first task was to investigate how the different step frequency algorithms perform when used in the unconstrained environment.

location	2.5 km/h	6.0 km/h	Δfreq/(km/h)
Trouser	1.20	2.27	0.18
Purse	1.10	2.12	0.12

Table 9 Min/Max per location.

Table 9 displays the minimum and maximum step frequencies per location of the mobile phone, found in the data set. The last column gives the average frequency steepness found for this group. To evaluate the different algorithms formula (2) was used in this section, together with the treadmill data. As explained in [22], this formula works when a person is constrained in walking speed, like on a treadmill. The constant c was solved for the single speed (4.5 km/h). Then the difference between the actual speed and the estimated speed was calculated over the complete range. Then the error was calculated with the root mean square error (RMSE) for all the different speeds.

8.1.1 DFT

This method was used on different features. We investigated the performance for the following features:

1. Length of the linear accelerometer vector.
2. The up and down acceleration.
3. The mobile angle variation.
4. Length of the gyroscope vector.

A problem was the resolution of the DFT method. Figure 7 illustrates this problem. The green trace is the actual treadmill speed, the red trace is the estimated speed by means of the DFT method and the blue a reference method which is explained later. In the figure we can differentiate the different speeds walked during the treadmill course. It is clear that the speed prediction is at its best at 4.5 km/h, because this speed was used for the calibration. The DFT method predicts the 4.5 km/h speed perfect, except for a small portion where there was a deviation of almost 1 km/h. It seems as if the speed estimation resolution was too big, the red line shows discrete steps of about 1 km/h. This was due to the frequency resolution of the DFT method, which relates to the computation window length. The resolution can be reduced by increasing the computation window length. Formula (8) gives the relation between window length and frequency resolution.

$$\Delta f = \frac{1}{Computation\ window\ (sec)}\ Hz \qquad (8)$$

The last column of Table 9 gives the average step frequency steepness. It is clear that the chosen window length was too short for accurate speed estimation. But increasing the window length will result in a poor step detection resolution. A resolution of 0.5 km/hour already needs a window length of about 11 seconds (step frequency), when the stride frequency is used the window length becomes about 22 seconds.

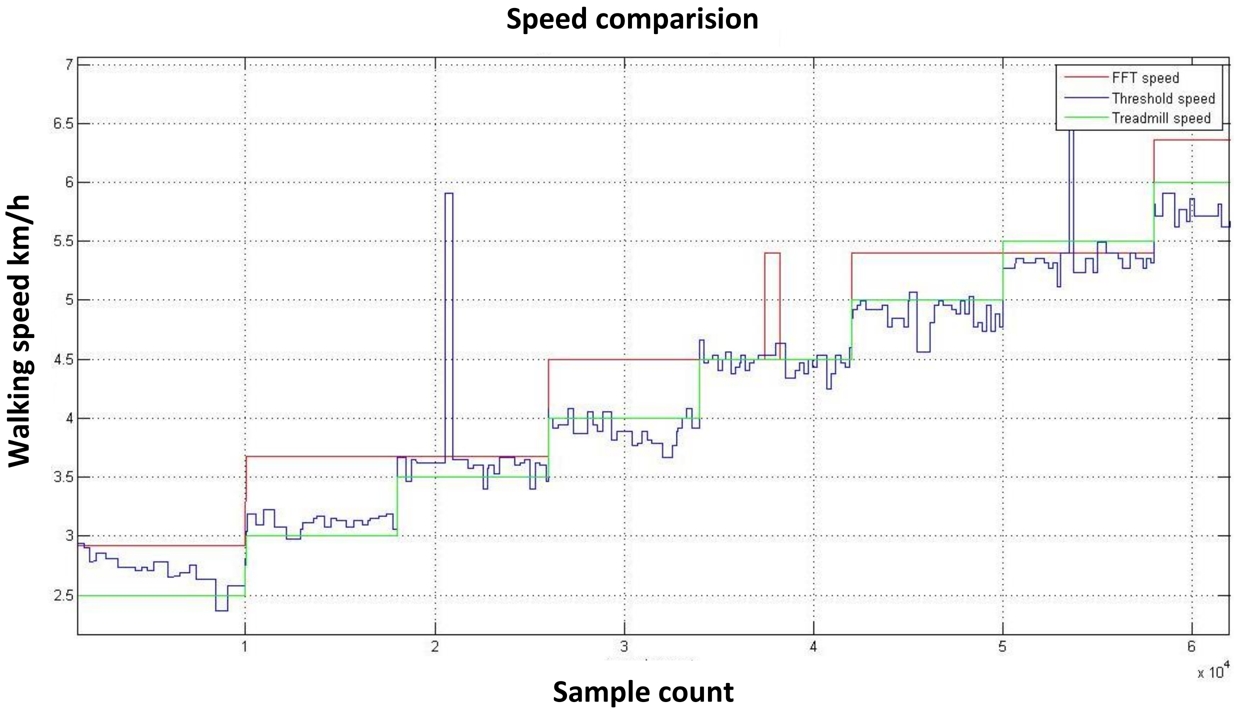


Figure 7 Speed method comparison

Table 10, gives an overview of the RMSE for the different features and the two locations.

	Purse (RSME)	Trouser (RSME)	
DFT Gyroscope	1.15	0.92	km/h
DFT angle varation	1.21	1.33	km/h
DFT up-down accel	4.36	6.02	km/h
DFT linear accel	3.93	2.59	km/h

Table 10 (DFT method) RMSE different signals – locations. The length of the gyroscope vector gave the best result for both locations. The up-down acceleration and linear acceleration gave the worst result.

8.1.2 ZERO CROSSING DETECTION

The zero crossing detection method has no resolution problem as described in the DFT method. But this method is more susceptible to switching between stride and step frequency.

The zero crossing detection methods were tested on the following features:

1. Length of the linear accelerometer vector.
2. Up and down acceleration.
3. Length of the gyroscope vector.

Zero crossing detection	Purse (RSME)	Trouser (RSME)	
Gyroscope	5.83	5.71	km/h
Up-down acceleration	2.09	2.78	km/h
Linear acceleration	19.45	5.08	km/h

Table 11 (Zero crossing method) RMSE different signals and location. The zero crossing detection error was bigger than in the previous DFT method. Therefore this method is less suitable for location independent step frequency implementation.

8.1.3 Peak detection

The characteristics of the peak detection method are similar to the zero crossing detection method, namely a good resolution but more susceptible to switching. The peak detection method was tested on the following features:

1. The mobile angle variation.
2. Up and down acceleration.
3. Length of the gyroscope vector.

Peak detection	Purse (RSME)	Trouser (RSME)	
Gyroscope	1.35	1.41	km/h
Angle varation	0.95	0.41	km/h
Up-down accel	1.20	6.06	km/h

Table 12 (Peak detection METHOD) RMSE different signals and locations. This method performs better than the previous method (DFT and zero crossing), because the RSM errors are smaller. Although we want to accomplish a better performance.

8.1.4 Improved Step frequency algorithm

Table 13 displays the improved results as obtained with the improved step frequency algorithm, described in section 4.5.1.1. This method improved with 50% in comparison to the peak detection method.

Peak detection	Purse (RSME)	Trouser (RSME)	
Improved algorithm	0.41	0.25	km/h

Table 13 Improved step frequency algorithm.

A big advantage of this technique is that it does not increase the computation length (walking resolution). It works within 2 strides or 4 steps, the previous walking stride plus the current walking stride. The greatest benefit of the peak detection over the DFT method is the shorter time window in which a frequency can be detected. This method improves the accuracy of the frequency, without compromising too much on this important aspect of the peak detection method.

8.2 Speed estimation methods

In this chapter a comparison was made between the following speed estimation methods indicated by formula (2,3 and 4). Formula 2 was chosen because it only depends on the step frequency. The Weinberg (3) method has proven itself for locations were the up-down movement of the upper body was measured (mobile phone placed on hip or torso). This method was implemented often in projects and used in research, also in the context of location independent speed estimation [16]. Last formula (4) was chosen because up-down linear accelerations high correlation with the actual speed.

8.2.1 Speed estimation methods with calibration

The used calibration method was already introduced in the previous section (8.1). The unknown parameters were solved for the single speed 4.5 km/h. Then the difference between the actual speed (treadmill speed) and the estimated speed was calculated over the complete walking speed range. This gave an indication of how well the method performed when we have the possibility of a personalized calibration (section 3.2.5). Table 14 shows the performance of the different formulas.

	RMSE							
Formula	Correct classification		Miss classification					
	Purse[2]	Trouser[3]	Purse[4]	Trouser[5]	Pu[6]	Tr[7]	Combined[8]	
2	0.41	0.25	0.25	0.41	0.38	0.27	0.32	km/h
3	0.42	0.28	0.73	1.03	0.48	0.35	0.42	km/h
4	0.37	0.38	0.66	0.81	0.43	0.42	0.43	km/h

Table 14 Performance personalized calibration corrected for classification.

The second and third column show the RSME measured for the particular locations of the mobile phone. This is the error that is normally reported in studies. But location classifiers are not 100% error free, so sometimes the location is misclassified which results in an extra error added to the previous error. We quantified the errors for the wrong location and displayed them in column 4 and 5. As expected, these results are worse than the results in column 2 and 3. This error is not realistic and agrees with 100% misclassification. For a more realistic error estimation the actual location misclassification must be used. In column 6 and 7 we show this error when the location classification of the previous chapter is used (Table 5). For this we used the true positive rate (TPR) [21] from the table. The RSME in column 6 and 7 was calculated with formula (9,10).

$$Error\ (trouser) = Err(trouser)[3] \cdot TPR + \overline{Err(trouser)[5]} \cdot (1 - TPR) \tag{9}$$

$$Error\ (purse) = Err(purse)[2] \cdot TNR + \overline{Err(purse)[4]} \cdot (1 - TNR) \tag{10}$$

The last column(8) averages the results from column 6 and 7. This is not realistic because most people only wear their mobile at a preferred location. This averaged number gives the error when the mobile is worn equally at both locations. The actual error will be between the error stated in column 6 and 7, depending on how the mobile is distributed over the locations. The results are comparable with previous studies, the only interesting finding is that the methods are comparable in accuracy. But the implementation difficulty differs considerably per method.

8.2.2 PERSONALIZED MODEL

The results in the previous section 8.2.1 were quite good, but need a personalized calibration. This calibration is not preferred, so can we get similar or comparable results when methods are used without this calibration? In this chapter we tried to estimate the walking speed without this calibration stage and used methods that only depend on the length of the person. For this formulas (5, 6 and 7) were used. Table 15 shows the performance for the three formulas.

Formula	Correct (RSME)	Combined (RSME)	
5	0.52	0.52	km/h
6	1.01	1.12	km/h
7	0.48	0.57	km/h

Table 15 Performance no calibration

The second column shows the RMSE when no location classification error is made (100% correctly classified). In the third column we tried again to indicate the error for location misclassification, same error as for column (8) section 8.2.1. The overall performance was worse than the previous result. This is explainable because we did not use the subject dependent constant.

8.2.3 AUTOMATIC CALIBRATION BY MEANS OF GPS

If we compare the results of the two previous sections, we notice a decrease in speed accuracy when we do not apply personalized calibration. But the use of a calibration routine is not desirable. Therefore we investigated if we could use GPS speed to calibrate.

The first question is: how accurate are the GPS data? Previous results revealed that the average RMSE was between 0.32 km/h and 0.57 km/h depending on the used method. GPS datasheets revealed that speed accuracy is about 0.5 km/h when the conditions are perfect. In real life the conditions are never perfect, so speed error can be expected to be higher than 0.5 km/h. To estimate the GPS error, we used our own data set and formula (2). The constant c was learned from the treadmill data as in the previous sections. Next we used this constant c and formula (2) to estimate the speed of the outdoor walking data set. Then we compared this result to the GPS speed.

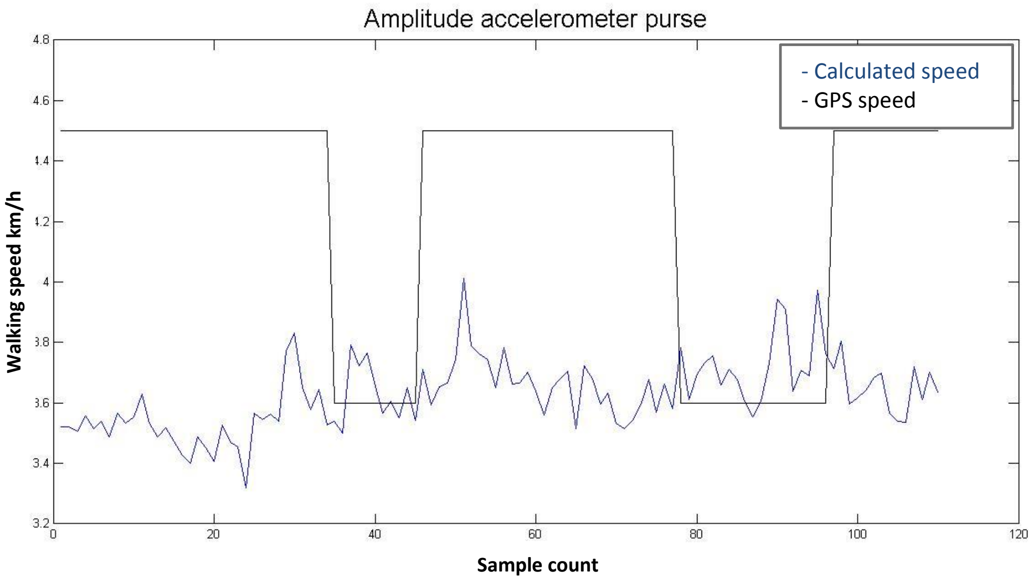


Figure 8 Estimated walking speed versus GPS speed. The blue trace shows the estimated walking speed and the black trace shows the GPS speed. It is immediately clear that these results do not match.

difference GPS and estimated speed		
	Mean (km/h)	**Max (km/h)**
Trouser	0.84	2.53
Purse	0.67	1.40

Table 16 Difference between the GPS speed and our own speed calculation. The results (RSME) are higher than the 0.5 km/h found in the GPS datasheet. It is doubtful whether we can use the GPS for calibration.

In the next stage we reversed the procedure: if GPS speed is available can we use it as calibration? Therefore we used the outdoor walking data to estimate the constant c in formula (2), followed by the evaluation of the accuracy of this constant c on the treadmill data. We used the following calculation: when a new GPS speed value was available, the walking frequencies of the last 4 features (12 seconds) were averaged. This average value was used to solve the constant c. This new constant c was averaged with the current constant, which resulted in a new value. So after the outdoor walking we took the learned constant and used this set constant to estimate the treadmill speed. Then the difference between these speeds was calculated. Figure 9 illustrates this process. The red trace is the constant c found for the treadmill calibration and the blue trace is the evolution of the constant during the outdoor walk. In the ideal situation the blue trace becomes closer to the red trace.

Calibration GPS (frequency) -reference Treadmill		
	Average RMSE	
Trouser	0.37	km/h
Purse	0.51	km/h

Table 17 shows the RMSE difference between the treadmill speed and speed estimation based on GPS calibration.

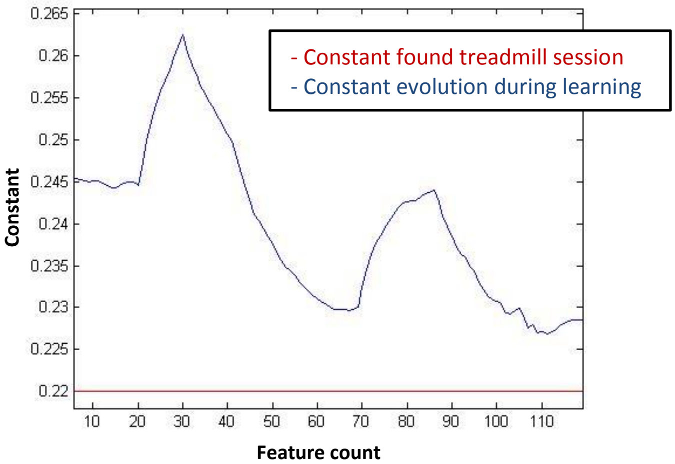


Figure 9 constant evolution during calibration

The result was much closer to the un-calibrated (the method that only depends on the length of the person) results than to the calibrated result. So it is still necessary to find a better calibration scheme to improve walking speed estimation based on automatic calibration by means of GPS.

9 CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

In this thesis, walking activity recognition is conducted at multiple walking speeds. Other activity recognition studies distinguish activities at a single or limited intensity range. On the other hand, walking speed estimation is often conducted over an excessive speed range, but mostly does not investigate the walking activity classification. In this study these two areas are combined. The speed intensity range complicates the performance of location classification. The classifier for the trouser location is more accurate than the purse location (90.4% to 79.6%). This has to do with the high walking intensity for the purse location is quite similar to the low walking intensity for the trouser location. The main reason is that the location and activity recognition only used a feature set that is independent of orientation and depends on the motion variation of the mobile. Single feature measures show that these types of features are less informative. But this study shows that good classifiers can be built on top of this type of features. Because these features depend on the motion intensity they reduce performance during activities at different locations in combination with different motion intensities.

The outcome of this classification is used to compare different speed estimation methods. Two main methods where investigated, dependent and independent on the location of the mobile phone. The main error of the speed estimation algorithms came from the step frequency estimation. This why an improved step frequency estimation algorithm was developed. The step frequency method improved accuracy for all speed estimation methods. This method is also independent of both location and orientation, which makes the method robust. Most improved step frequency algorithms increase the computation window length, by using some kind of averaging. This means that the subject must walk for a longer time before an accurate walking frequency is detected. The proposed step frequency algorithm works within 2 strides or 4 steps.

The performances of the different speed estimation methods are comparable. But the speed estimation which only depends on walking frequency is much easier to implement. It is also a realistic algorithm to implement on current mobile phones. If a personalized calibration scheme is used or automatic calibration scheme is developed, then this system only shows a small decrease in performance compared to single location specific methods like ZUPT (0.18 to 0.32 km/h). A personalized calibration improves the speed estimation. Unfortunately the idea to use GPS for this calibration was not successful.

9.2 FUTURE WORK

During this study I spent a lot of time on finding good walking gait timing. To know the exact timing is a really good way to improve speed estimation, as it is used in the zero velocity update (ZUPT) method. But exact timing is difficult to establish in the context of location and orientation independency. I did not succeed in this. In this study it is shown that personalized calibration scheme really improves speed estimation. The simple GPS scheme didn't result in the desirable outcome. Therefore there is a need for an improved calibration method. Also all methods show that the error distributions over the different subjects are not equal. The methods work better for most subjects than the presented results, but for a small subset of subjects the methods perform much worse than the presented results. Somehow this small group does things different. It is not said that improving the accuracy for the bigger group will also result in improvement of the smaller group. So we have to be careful with performance measurements based on the average over the complete group (improving for the bigger group often leads to decrease in performance for the smaller group, this is not reflected in the average result). And maybe future speed estimation algorithms need also group/subject based models. The up-down stairs classifier really improved with the extra barometer feature. The mobile phone has a lot of different sensors. Classification and speed estimations can be done in multiple ways. Classification performance may improve by combining these different methods (Wi-Fi speed estimation maybe used to distinguish kitchen activity from location movement). Also the internet connection can play a role by using location context information, like local temperature or other location specific information. Knowing that the person is in-doors makes it less likely that a person is biking.

APPENDIX A EXTRACTED FEATURES

This appendix shows all the features extracted from the data. Gray number indicated that this feature is orientation dependent and not used in this project.

Name	Description	number
Avg(ax^2)	RMS value x-as accelerometer	20
iqr(ax)	Inter quartile range x-as accelerometer	21
mean(ax)	Average x-as accelerometer	22
std(ax)	Standard deviation x-as accelerometer	23
Avg(ay^2)	RMS value y-as accelerometer	24
iqr(ay)	Inter quartile range y-as accelerometer	25
mean(ay)	Average y-as accelerometer	26
std(ay)	Standard deviation y-as accelerometer	27
Avg(az^2)	RMS value z-as accelerometer	28
iqr(az)	Inter quartile range z-as accelerometer	29
mean(az)	Average z-as accelerometer	30
std(az)	Standard deviation z-as accelerometer	31
Avg((ax.ay.az)^2)	Average length of total acceleration	32
iqr((ax.ay.az))	Inter quartile range total acceleration	33
mean((ax.ay.az))	Average length of total acceleration	34
std((ax.ay.az))	Standard deviation total acceleration	35
Avg(lax^2)	RMS value x-as linear accelerometer	36
iqr(lax)	Inter quartile range x-as linear accelerometer	37
mean(lax)	Average x-as linear accelerometer	38
std(lax)	Standard deviation x-as linear accelerometer	39
Avg(lay^2)	RMS value y-as linear accelerometer	40
iqr(lay)	Inter quartile range y-as linear accelerometer	41
mean(lay)	Average y-as linear accelerometer	42
std(lay)	Standard deviation y-as linear accelerometer	43
Avg(laz^2)	RMS value z-as linear accelerometer	44
iqr(laz)	Inter quartile range z-as linear accelerometer	45
mean(laz)	Average z-as linear accelerometer	46
std(laz)	Standard deviation z-as linear accelerometer	47
Avg((lax.lay.laz)^2)	Average length of total linear acceleration	48
iqr((lax.lay.laz))	Inter quartile range total linear acceleration	49
mean((lax.lay.laz))	Average length of total linear acceleration	50
std((lax.lay.laz))	Standard deviation total linear acceleration	51
Avg(rax^2)	RMS value of rotated x-as accelerometer	52
iqr(rax)	Inter quartile range rotated x-as accelerometer	53
mean(rax)	Average rotated x-as accelerometer	54
std(rax)	Standard deviation rotated x-as accelerometer	55
Avg(ray^2)	RMS value of rotated y-as accelerometer	56
iqr(ray)	Inter quartile range rotated y-as accelerometer	57
mean(ray)	Average rotated y-as accelerometer	58
std(ray)	Standard deviation rotated y-as accelerometer	59
Avg(hora^2)	RMS value of rotated z-as accelerometer	60
iqr(hora)	Inter quartile range rotated z-as accelerometer	61
mean(hora)	Average rotated z-as accelerometer	62
std(hora)	Standard deviation rotated z-as accelerometer	63
Avg(vera^2)	RMS value of vertical accelerometer	64
iqr(vera)	Inter quartile range vertical accelerometer	65
Mean(vera)	Average vertical accelerometer	66
std(vera)	Standard deviation vertical accelerometer	67
Avg((hora/vera)^2)	RMS value of ratio horizontal/vertical accelerometer	68
iqr(hora/vera)	Inter quartile range ratio horizontal/vertical accel	69

Mean(hora/vera)	Average ratio horizontal/vertical accelerometer	70
std(horla/verla)	Std ratio horizontal/vertical accelerometer	71
Avg(rlax^2)	RMS value of rotated x-as linear accelerometer	72
iqr(rlax)	Inter quartile range rotated x-as linear accelerometer	73
mean(rlax)	Average rotated x-as linear accelerometer	74
std(rlax)	Standard deviation rotated x-as accelerometer	75
Avg(rlay^2)	RMS value of rotated y-as linear accelerometer	76
iqr(rlay)	Inter quartile range rotated y-as linear accelerometer	77
mean(rlay)	Average rotated y-as linear accelerometer	78
std(rlay)	Standard deviation rotated y-as linear accelerometer	79
Avg(horla^2)	RMS value of rotated z-as linear accelerometer	80
iqr(horla)	Inter quartile range rotated z-as linear accelerometer	81
mean(horla)	Average rotated z-as linear accelerometer	82
std(horla)	Standard deviation rotated z-as linear accelerometer	83
Avg(verla^2)	RMS value of vertical linear accelerometer	84
iqr(verla)	Inter quartile range vertical linear accelerometer	85
Mean(verla)	Average vertical linear accelerometer	86
std(verla)	Standard deviation vertical linear accelerometer	87
Avg((horla/verla)^2)	RMS value of ratio horizontal/vertical linear accel	88
iqr(horla/verla)	Inter quartile range ratio horizontal/vertical lin accel	89
Mean(horla/verla)	Average ratio horizontal/vertical linear accelerometer	90
std(horla/verla)	Std ratio horizontal/vertical linear accelerometer	91
Avg(gyx^2)	RMS value x-as gyroscope	92
iqr(gyx)	Inter quartile range x-as gyroscope	93
mean(gyx)	Average x-as gyroscope	94
std(gyx)	Standard deviation x-as gyroscope	95
Avg(gyy^2)	RMS value y-as gyroscope	96
iqr(gyy)	Inter quartile range y-as gyroscope	97
mean(gyy)	Average y-as gyroscope	98
std(gyy)	Standard deviation y-as gyroscope	99
Avg(gyz^2)	RMS value y-as gyroscope	100
iqr(gyz)	Inter quartile range y-as gyroscope	101
mean(gyz)	Average y-as gyroscope	102
std(gyz)	Standard deviation y-as gyroscope	103
Avg((gy.gyy.gyz)^2)	Average length of total gyroscope	104
iqr((gyx.gyy.gyz))	Inter quartile range total gyroscope	105
mean((gyx.gyy.gyz))	Average length of total gyroscope	106
std((gyx.gyy.gyz))	Standard deviation total gyroscope	107
iqr(angleG)	Inter quartile range orientation from gravitation	108
mean(angleG)	Average orientation from gravitation	109
std(angleG)	Standard deviation orientation from gravitation	110
amplitude totall angle	Amplitude angle (orientation) from gravitation	111
fft freq grav	Frequency with the largest power gravitation	112
fft pwr1 grav	Largest power gravitation	113
fft pwr4 grav	Largest four powers gravitation	114
fft pwr4 % grav	Percentage of four largest pwr's compared to total pwr	115
fft pwr B0 grav	Power in Band 0 gravitation	116
fft pwr B1 grav	Power in Band 1 gravitation	117
fft freq la	Frequency with the largest power linear acceleration	118
fft pwr1 la	Largest power linear acceleration	119
fft pwr4 la	Largest four powers linear acceleration	120
fft pwr4 % la	Percentage of four largest pwr's compared to total pwr	121
fft pwr B0 la	Power in Band 0 linear acceleration	122
fft pwr B1 la	Power in Band 1 linear acceleration	123
fft freq rla	Frequency with the largest power rotated linear accel	124
fft pwr1 rla	Largest power rotated linear acceleration	125
fft pwr4 rla	Largest four powers rotated linear acceleration	126

fft pwr4 % rla	Percentage of four largest pwr's compared to total pwr	127
fft pwr B0 rla	Power in Band 0 rotated linear acceleration	128
fft pwr B1 rla	Power in Band 1 rotated linear acceleration	129
fft freq gyro	Frequency with the largest power rotated linear accel	130
fft pwr1 gyro	Largest power rotated gyroscope	131
fft pwr4 gyro	Largest four powers gyroscope	132
fft pwr4 % gyro	Percentage of four largest pwr's compared to total pwr	133
fft pwr B0 gyro	Power in Band 0 gyroscope	134
fft pwr B1 gyro	Power in Band 1 gyroscope	135
corr accel	Correlation between accelerometer axis	136
corr gyro	Correlation between gyroscope axis	137
corr accel gyro	Correlation between accelerometer and gyroscope	138
ampl orien var	Amplitude angle variation calculated from orientation	139
rla fft/fft thres	Frequency FFT closest to Threshold frequency cal	140
rla ampl h0..5	Power of five harmonics of thres freq rla	141
H0/(H..)	Ratio between rla H0/(H1..H4)	142
lra FMAX	Largest power frequency of linear rotated acceleration	143
lra AMAX	Power of largest frequency	144
grav fft/fft thres	Frequency FFT closest to Threshold frequency cal	145
grav ampl h0..5	Power of five harmonics of thres freq	146
grav H0/(H..)	Ratio between grav H0/(H1..H4)	147
grav FMAX	Largest pwr frequency of linear rotated acceleration	148
grav AMAX	Power of largest frequency (gravitation)	149
gyro fft/fft thres	Frequency FFT closest to Threshold frequency cal	150
gyro ampl h0..5	Power of five harmonics of thres freq	151
gyro H0/(H..)	Ratio between H0/(H1..H4)	152
gyro FMAX	Largest pwr frequency of linear rotated acceleration	153
gyro AMAX	Power of largest frequency	154
diff phase H1 H2 acc	The phase difference between harmonics of accel	155
diff phase H1 H2 grav	The phase difference between harmonics of gravitation	156
diff phase H1 H2 gyro	The phase difference between harmonics of gyroscope	157
diff phase H1 acc-gyro	The phase difference between first harmonic acc-gyro	158
diff phase H1 acc-grav	The phase difference between first harmonic acc-grav	159
diff phase H1 gyro-grav	The phase difference between first harmonic gyro-grav	160
Peak/RMS lhor acc	Ratio peak and RMS value linear horizontal accel	161
Peak/RMS gyro	Ratio peak and RMS value gyroscope	162

APPENDIX B DATA SET

This chapter explained the different files and data set, format and other stuff.

Every activity is recorded in its own file. This means that every subject has 16 separated files. The type of activity is indicated by the filename. The prefix T_ means the trouser location and P_ the other purse location.

(*)_kitchen.csv	Contains random the kitchen, dishwashing, walking to trash bin.
T_SittingStanding.csv	Only trouser first sitting then standing.
(*)_Stairs.csv	Up-Down three stairs in potential building.
(*)_Walking_0deg.csv	Treadmill session over different speeds
(*)_Walking_5deg.csv	Treadmill uphill two angles 5,10 deg.

The GPS outdoor walking and biking are recorded into two separate files. One file contains the GPS data, the other file the sensor data.

(*)_GPSWalking_Location.log	Outdoor walking
(*)_GPSWalking_Sensor.log	
(*)_GPSBiking_Location.log	Outdoor biking
(*)_GPSBiking_Sensor.log	

These files contain the raw sensor data, which is not retimed as described in the thesis. The files with the suffix '*(filename)_retimed.mat*' are as the name suggested retimed. The file is a time series Matlab object. Ts.time contains the retimed time. The ts.Data contains 34 rows of sensor data.

The files with the suffix '*(filename)_Labeled.mat*' The retimed file is labeled:

```
**************************************************************************************
* 2. Timeseries format
**************************************************************************************
        ts.time            = time
        ts.data(:,1:3)     = acceleration x,y,z
        ts.data(:,4:6)     = gravitation x,y,z
        ts.data(:,7:9)     = linear acceleration x,y,z
        ts.data(:,10:12)   = magnetic sensor x,y,z
        ts.data(:,13:15)   = rotated acceleration x,y,z
        ts.data(:,16:18)   = rotated linear acceleration x,y,z
        ts.data(:,19:21)   = orientation
        ts.data(:,22:24)   = gyroscope x,y,z
        ts.data(:,28:30)   = Barometer presure,height,constant
        ts.data(:,31)      = speed raw GPS
        ts.data(:,32)      = speed calculated from travelled distance
        ts.data(:,34)      = type activity
                             Purse_GPS_Biking        = 54;
                             Purse_GPS_Walking       = 22;
                             Purse_STAIRS            = 84;
```

```
Purse_V25_A0            = 100*2.5 + 21 + 0; %20
Purse_V30_A0            = 100*3.0 + 21 + 0; %20
Purse_V35_A0            = 100*3.5 + 21 + 0; %20
Purse_V40_A0            = 100*4.0 + 21 + 0; %20
Purse_V45_A0            = 100*4.5 + 21 + 0; %20
Purse_V50_A0            = 100*5.0 + 21 + 0; %20
Purse_V55_A0            = 100*5.5 + 21 + 0; %20
Purse_V60_A0            = 100*6.0 + 21 + 0; %20
Purse_V30_A5            = 100*3.0 + 37 + 5; %36
Purse_V30_A10           = 100*3.0 + 37 + 10;%36

Trouser_GPS_Biking      = 50;
Trouser_GPS_Walking     = 18;
Trouser_KITCHEN         = 96;
Trouser_STANDINGSITTING = 176;
Trouser_STAIRS          = 80;
Trouser_V25_A0          = 100*2.5 + 17 + 0; %20
Trouser_V30_A0          = 100*3.0 + 17 + 0; %20
Trouser_V35_A0          = 100*3.5 + 17 + 0; %20
Trouser_V40_A0          = 100*4.0 + 17 + 0; %20
Trouser_V45_A0          = 100*4.5 + 17 + 0; %20
Trouser_V50_A0          = 100*5.0 + 17 + 0; %20
Trouser_V55_A0          = 100*5.5 + 17 + 0; %20
Trouser_V60_A0          = 100*6.0 + 17 + 0; %20
Trouser_V30_A5          = 100*3.0 + 33 + 5; %36
Trouser_V30_A10         = 100*3.0 + 33 + 10;%36
Trouser_RUN             = 100*7.0 + 144 +0;
```

****

This labeled file is the base of the feature extraction files. For every activity there is a separate feature file made. This file contains a struct object with time,data (171 rows/features) and header explaining the rows. Header number are the same as in appendix A. The first 19 rows contains info and are not actual features:

```
1 'subjectID'          # between 1 and 20
2 'weight'
3 'height'
4 'age'
5 'windowType'         # ignore
6 'windowName'         # ignore
7 'windowLength'       # in seconds
8 'windowOverlap'      # overlap in procent
9 'locationName'       # purse / trouser
10 'activityName'
11 'locationID'
12 'activityID'
13 'activityallName'
14 'activityallID'      # see above activity numbers
15 'speed GPS'
16 'speed Loc'
17 'uphill angle'
18 'height'             # ignore
19 'future',...         # ignore
```

The Features.mat file combines these single files into one feature file per subject. These files are located in the personal directory. The ALL directory contains a combined Features.mat file where all 20 subjects are combined into a single feature file.

M1	male
M2	male
M3	male
M4	male
M5	male
M6	male
M7	male
M8	male
F9	female
M10	male
M11	male
F12	female
F13	female
M14	male
F15	female
M16	male
F17	female
M18	male
M19	male
F20	female

APPENDIX C THE ANDROID EMULATOR AND DDMS

C.1 PROBLEMS CONNECTING PHONE ECLIPSE

It always takes me at least half a day to connect a new phone to eclipse. Most of the time the phone is not visible in the DDMS-manager and window device manager reports an error message. The following sequences of instruction work most of the time to solve this problem. And hopefully it will solve our problem.

1. Uninstalled Google USB Driver from Eclipse in the Android SDK Manager.
2. Uninstalled the driver from Device Manager - click box for "delete driver from my computer"
3. Unplugged and re-plugged the phone into the computer.
4. Windows "improperly" installed drivers for the "*new phone*".
5. The "*new phone*" is now showing up in My Computer like a drive.
6. Reinstall Google USB Driver in SDK Manager *{SDK directory}/extras/google/usb-driver*.
7. Update "*new phone*" driver in Device Manager.
8. Everything works.

C.2 SOME OTHER NICE THINGS TO KNOW

This explains the finding I had on the android emulator. It may be useful chapter for somebody who starts Android development. The emulator is a powerful tool and can be used for rapid development, but it is no substitute for testing on a true device. The emulator can be configured to emulate different types of android devices. Especially customizing the hardware setting is a nice feature. You can test your software on different hardware configurations. Some of the configurations can be done the GUI of the Android Virtual Device Manager, but the actual hardware of the device is described in the config.ini file. Every emulator configuration has a directory containing this config-file and extra files which emulate for example the SD-card. This directory is a meaningful place for configuring the emulator.

There are also nice useful android configurations available made by fellow developers (http://androidbook.blogspot.com/2010/08/creating-useful-avds.html).

To switch between portrait and landscape modes of the emulator, use the 7 and 9 key on the numeric keypad of the host machine (or Ctrl+F11 and Ctrl12 keys). Which is very useful to test how the layout performs between the two different orientations.

You can launch a second emulator to interact with the first emulator to simulate calls, text messages and such. The second emulator must have a different configuration as the first one. The tool does not allow you to start multiple instances of the same AVD configurations.

The reason I started reading the emulator documentation was for the reason that I started to develop a GPS tracker. This does not work indoors when developing on the phone.

The Dalvik Debug Monitor Service (DDMS) is a debugging tool provided with de Android software development kit. The DDMS provides access point into emulator or the actual phone for the purpose of debugging. It can also be used for file and process management.

With the DDMS is possible to interact with processes. Every Android application runs in its own VM with its own user id. For the application VM we monitor threads and heap, stop processes and force garbage collection.

You can take screen captures of the emulator and the device. This can be quit useful for reports and documentation. Take the following steps, to capture a screenshot.

1. Choose the emulator or device
2. Press the multicolored square picture button.

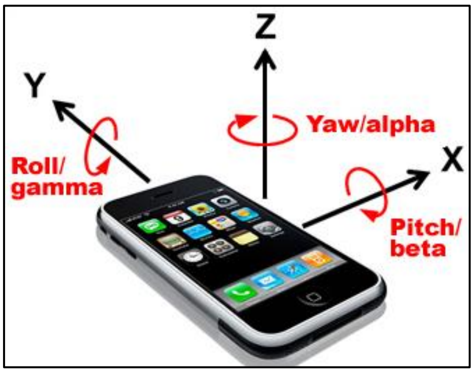
APPENDIX D ROTATION MATRIX

To estimate how the mobile device is orientated within the earth gravitation field we use the rotation matrix. The output of the accelerometer is defined by the following formula (E1).

$$G_p = \begin{pmatrix} G_x \\ G_y \\ G_z \end{pmatrix} = R(G - a_l) \tag{E1}$$

In this formula a_l is the linear acceleration. If we how assume that the mobile device lays still and has no linear acceleration. Then R is the rotation matrix, which describing the orientation of the mobile device relative to the earth's coordinate frame. If the mobile is lying flat on the table will give an output of -1g in the z direction (E2).

$$G_p = R(G - 0) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{E2}$$



The orientation of the mobile device can be defined by its roll, pitch and yaw rotations from. The notation and definition comes from nautical and aerospace world.

When there is no linear acceleration applied on the mobile phone the Roll and the Pitch can be calculated from the accelerometer values. The Yaw can be calculated from the accelerometer and must be determined from the magnetometer.

The following rotation matrixes can be defined for the roll, pitch and yaw.

$$roll(\emptyset) = \begin{pmatrix} \cos\emptyset & 0 & -\sin\emptyset \\ 0 & 1 & 0 \\ \sin\emptyset & 0 & \cos\emptyset \end{pmatrix} \tag{E3}$$

$$pitch(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \tag{E4}$$

$$yaw(\gamma) = \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{E5}$$

There are six possible orderings of these three rotation matrices. Only two orderings have compute meaning. The outputs of the accelerometer have just two degrees of freedom, so is not possible to solve the matrix for three unique values of the roll, pitch and yaw-angles. Four of these sequences are rejected, because they depend on all three angles. Two of these two rotation sequences depend only on the roll and pitch angles.

Once the roll and pitch angles are known, the tilt can be compensated by rotating in roll and pitch using the matrices.

$$\begin{pmatrix} G_x \\ G_y \\ G_z \end{pmatrix} = roll(\emptyset)pitch(\theta)yaw(\gamma)\begin{pmatrix} A_{xas} \\ A_{yas} \\ A_{zas} \end{pmatrix} \tag{E6}$$

APPENDIX E DATA RECORDING PLAN

E.1 INTRODUCTION

This data recording plan describes the steps to be followed in order to acquire the relevant data that will be used during the project entitled "*Habitual Behavior Monitoring using Smartphones: Impact of Sensor Input and Personalized Models on Walking Patterns*" [31]. The aim of this project is to use the smartphone, to assess these walking patterns. The goal is to measure the total duration, speed and speed-distribution of walk bouts. The following sensors are used, accelerometer, gyroscope, magnetometer, barometer and GPS to monitor the participants. The data is recorded on a smartphone running Android OS using the CRNTC+ framework. Once the recording is finished, the data is transferred to the computer. The computer will process the information in order to extract relevant features of the participants walk bouts, number of steps, duration, speed and speed-distribution.

In section 2 the goal of this assignment is presented. Section 3 the required hardware and placement is explained. The last section 4 discusses the detailed data recording plan.

E.2 GOAL

The aim of this data recording plan is to describe in a precise manner the protocol used to collect and transmit relevant data. This data is necessary for the development of a walking pattern monitor. Such recording plan will be carried out using a mobile phone (smartphone running Android). The idea is to illustrate the different steps that participants take for acquiring the necessary recording data that will be used to develop and validate a number of algorithms used to extract relevant features from the data.

During the data recording plan the mobile phone is worn at two different places, this document explains how and where the mobile phone should be placed, during the recording sessions.

E.3 RECORDING HARDWARE SETUP

E.3.1 REQUIREMENTS

- Samsung S3 with Android 2.3 or higher and the CRN Toolbox Center+ application installed.

E.3.2 SENSOR PLACEMENT

Most people have two preferred places to wear their mobile phone which depends on gender. The male population prefers to wear their mobile in the pockets of their pants. Unlike the female population which like to wear their mobile in their bag. During the data recording the mobile phone is worn at these two different locations:

- The front pocket of the jeans, the body of the phone will be aligned with the participant's leg. This is the most comfortable way to wear the mobile phone. Hereby the y-as of the accelerometer will point upwards or downwards.
- The second place will be a bag. In this experiment only a shoulder bag is used. The participant is asked to put the mobile phone in the bag, at the beginning of the experiment. This means that the orientation of the phone depend per participant.

E.4 Data recording plan

Twenty participants will be asked to perform 3 different activities that involve a treadmill, climbing stairs and an outdoor route. Every activity is conducted twice for two different locations.

For the treadmill the participant will perform two sessions of 7 minutes. During this session the speed of the treadmill will increase from slow walking (2.5 km/h), moderately (4.5 km/h) to brisk walking speed (6.0 km/h) in steps of (0.5 km/h). The treadmill is programmed to keep every speed constant for 1 minute and then increase speed by (0.5 km/h).

To simulate uphill walking the treadmill is set under two different angles 5%, 10% and one speed. The uphill walking will be performed at a speed of (3.0 km/h).

The second activity, "climbing stairs" includes two sessions for the different locations of the mobile. Climbing the stairs will not include different speeds. The participant is asked to rise and downwards on their own preferred speed.

The last activity consists of an outdoor track over the campus. The participant is asked to walk the indicate track at its own preferred speed. The track will be around 2 km.

If possible the data recording can be recorded with two mobiles. Both locations can be recorded in a single run. The indoor treadmill recording can be extended, by decreasing the step size for the flat walking and increasing the number of speed for the uphill walking.

Every single activity performed by each participant will be monitored by the author of this work. The data recording will take place in the "Student Sport Center Eindhoven" and the "tue-campus".

The general procedure of treadmill recording

1. The angle of the treadmill is adjusted and the treadmill is programmed to the speed course.
2. The participant will hold the smartphone in his hand in order to start the recording by pressing the start button inside the application, indicated by the supervisor.
3. The participant puts the mobile on the indicate location.
4. The participant will stand still for duration of 10 seconds, at the beginning of the activity. This is going to be done for the purpose of synchronization.
5. The treadmill is started and goes through course of the different walking speeds.
6. The participant will exercise for 7 minutes. The participant is allowed to stop at any moment if she/he can't accomplish the activity.
7. The participant will stand still for duration of 10 seconds, indicated by the supervisor.
8. The participant will take the mobile out of hers/his pocket or bag.
9. The participant will press the stop button inside the application to stop the recording.

This procedure is repeated for the different treadmill angles and the different mobile locations.

The general procedure of stairs recording

1. The participant will hold the smartphone in his hand in order to start the recording by pressing the start button inside the application, indicated by the supervisor.
2. The participant puts the mobile on the indicate location.
3. The participant will stand still for duration of 10 seconds, at the start of the activity. This is going to be done for the purpose of synchronization.
4. Participant climbs and lowers the stairs. In its own pace.
5. The participant will stand still for duration of 10 seconds, indicated by the supervisor.
6. The participant will take the mobile out of hers/his pocket or bag.
7. The participant will press the stop button inside the application to stop the recording.

This procedure is repeated for the two different mobile locations.

Outdoor track

1. The route is explained to the participant
2. The participant will hold the smartphone in his hand in order to start the recording by pressing the start button inside the application, indicated by the supervisor.
3. The participant puts the mobile on the indicate location.
4. The participant will stand still for duration of 10 seconds, at the start of the activity. This is going to be done for the purpose of synchronization.
5. Participant will walk the route in its own pace.
6. The participant will stand still for duration of 10 seconds, indicated by the supervisor.
7. The participant will take the mobile out of hers/his pocket or bag.
8. The participant will press the stop button inside the application to stop the recording

This procedure is repeated for the two different mobile locations.

BIBLIOGRAPHY

[1] R. Boulic, N. Magnenat-Thalmann and D. Thalmann, "A Mathematical implementation of a "Global Human Walking model with Real-Time Kenematic personification"," *The Visual Computer: International Journal of Computer Graphics - Special issue on computer animation 1989/90,* vol. 6, no. 6, pp. 344 - 358, Dec. 1990.

[2] D. Sutherland, K. R. Kaufman and J. R. Moitoza, Kinematics of normal human walking, Baltimore: Human walking, 1994.

[3] X. Long, Y. Bin and R. M. Aarts, "Single-Accelerometer-Based Daily Physical Activity Classification," in *31st Annual International Conference of the IEEE EMBS*, Minnesota, 2009.

[4] Y. Shi, Y. Shi and J. Liu, "A rotation based method for detecting onbody," in *UbiComp Proceedings of the 13th international conference on Ubiquitous computing Pages 559-560*, New York, 2011.

[5] A. Henpraserttae, S. Thiemjarus and S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," *Body Sensor Networks (BSN),* pp. 41 - 46, 2011.

[6] A. Vahdatpour, N. Amini and M. Sarrafzadeh, "On-body device localization for health and medical monitoring applications," *Pervasive Computing and Communications (PerCom),* pp. 37 - 44, 2011.

[7] L. Sun, B. L. D. Zhang and L. Shijian, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," *Ubiquitous intelligence and computing ,* vol. 7, pp. 548 - 562 , 2010.

[8] P. Siirtola and J. Roning, "Recognizing human activities userindependently on smartphones based on accelerometer data," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 1, no. Special Issue on Distributed Computing and Artificial Intelligence, pp. 38-45, 2012.

[9] A. Anjum and M. U. Ilyas, "Activity recognition using smartphone sensor," in *Consumer Communications and Networking Conference (CCNC)*, Las Vegas, 2013.

[10] J. Yang, „Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," *Interactive multimedia for consumer electronics (IMCE),* vol. 1, pp. 1-10 , 2009.

[11] S. Yang and Q. Li, "Inertial Sensor-Based Methods in Walking Speed Estimation," *sensors,* vol. 12, no. 5, pp. 6102-6116, 2012.

[12] S. Miyazaki, "Long-Term Unrestrained Measurement of Stride Length and Walking Velocity Utilizing a Piezoelectric Gyroscope," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING,* vol. 44, no. 8, pp. 753-759, 1997.

[13] S. Sakka, C. Hayot and P. Lacouture, "A generalized 3D inverted pendulum model to represent human normal walking," IEEE, Memphis, 2010.

[14] J. Kappi, J. Syrjarinne and j. Saarinen, "MEMS-IMU based pedestrian navigator for handheld devices," *The Institute of Navigation,* vol. 14, pp. 1369 - 1373, 2001.

[15] H. Vathsangam, A. Emken, D. Spruijt-Metz and G. S. Sukhatme, "Toward Free-Living Walking Speed Estimation Using Gaussian Process-based Regression with On-Body Accelerometers and Gyroscopes," *Pervasive Computing Technologies for Healthcare,* vol. 14, pp. 1 - 8, 2010.

[16] J. Park, A. Patel, D. Curtis, S. Teller and J. Ledlie, "Online Pose Classification and

Walking Speed Estimation using Handheld Devices," *ACM Conference on Ubiquitous Computing* , pp. 113-122 , 2012.

[17] A. Brajdic and R. Harle, "Walk Detection and Step Counting on Unconstrained Smartphones," *ACM international joint conference on Pervasive and ubiquitous computing* , pp. 225-234 , 2013.

[18] B. Longstaff, S. Reddy en D. Estrin, Improving activity classiffication for health applications on mobile devices using active and semi-supervised learning, Pervasive Computing Technologies for Healthcare, March 2010.

[19] J. Alvarez, R. Gonzalez, D. Alvarez, A. loprez and J. Rodriguez-Uria, "Multisensor approach to walking distance estimation with foot inertial sensing," *Sensor,* vol. 12, no. 8, p. 10536–10549, 2007.

[20] K. D. Cocker, G. Cardon en I. D. Bourdeaudhuij, K. De Cocker, G. Cardon, and I. De Bourdeaudhuij, Br J Sports Med, 40(8):714–716,, 2006.

[21] W. Zijlstra, Assessment of spatio-temporal parameters during unconstrained, European Journal of Applied Physiology, 92(1):39–44, 2004.

[22] J. Bertram and A. Ruina, "Multiple walking speed frequency relations are predicted by contrained optimization," *Thoretical Biology,* pp. 209, 445-453, 2001.

[23] L. Hof, „Scaling gait data to body size," Gait Posture, Groningen, 1996.

[24] R. Battiti, „Using Mutual Information for Selecting Using Mutual Information for Selecting," *IEEE TRANSACTIONS ON NEURAL NETWORKS,* vol. 5, nr. 4, p. 537, 1994.

[25] C. Bishop, Pattern recognition, Springer, 2009.

[26] I. Skog, P. Händel, J. Nilsson and J. Rantakokko, "Zero-velocity detection—An algorithm evaluation.," IEEE Trans. Biomed. Eng, 2010.

[27] W. Zijlstra and A. L. Hof, "Assesment of spatio-temporal gait parameters from trunk accelerations during human walking," *Gait & Posture,* vol. 18, no. 2, pp. 1-10, 2003.

[28] H. Weinberg, „Using the ADXL202 in Pedometer and Personal Navigation Applica-," Analog Devices, Norwood.

[29] J. Park, A.Patel, D. Curtis, S. Teller en J. Ledlie, „Online Pose Classification and Walking Speed Estimation using Handheld Devices," *UbiComp,* 2012.

[30] G. Spina, J. F. Roberts, P. Weppner, Lukowicz and O. Amft, "CRNTC+: A smartphone-based sensor processing framework for prototyping personal healthcare applications," *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare: IEEE,* 2013.

[31] Marco Altini, „Habitual Behaviour Monitoring using Smartphones: Impact of Sensor Input and Personalised Models on Walking Patterns".

[32] A. Bellemans, "Power demands in walking and pace optimization," *American Journal of Physics,* vol. 49, pp. 25-27, 1981.