Movement Recognition and Direction Detection of Pedestrians

Thesis submitted for the degree of

Doktor der Ingenieurwissenschaften (Dr. -Ing.)

Submitted to the Faculty of Electrical Engineering and Computer Science

of the University of Kassel

By M.Sc. Abdul Qudoos Memon

Kassel
Date of Defense: 12 April 2021

Abstract

Official records from the Federal Highway Research Institute (Bundesanstalt für Straßenwesen (BASt)) report that around 15,400 pedestrians were killed in traffic accidents in 2016 across thirty-six countries [1]. The car manufacturers and various research groups are currently working to improve pedestrians safety by developing passive and active pedestrian protection systems. Passive pedestrian protection aims to reduce the impact of a collision while the active approach tries to avoid collisions by using car-mounted sensors (such as radars and cameras) that require a direct line-of-sight to detect pedestrians. With cooperative-based approaches, cars and pedestrians exchange information of each other's movement including position, direction, and speed while on-board avoidance algorithms calculate the risk of collision and send warning signals, when appropriate. Cooperative-based approaches use wireless communication technologies for example Radio frequency communication [2], [3] or the GPS [4]to detect pedestrians, both wireless communication technologies works without the limitation of line-of-sight communication. However, [2] and [3] offers a limited range of communication and [4] offers a maximum sampling frequency of 1Hz which might not be fast to provide real-time updates in the movement history of pedestrians. In this dissertation, an alternative to these approaches, we developed a method that uses smartphone sensors to recognise the pedestrian movement (using the accelerometer) and direction (using the compass) accurately. The proposed approach works independent of the GPS and also improved the recognition time of pedestrians movement and direction in comparison to using the GPS.

Currently, smartphones are the most common devices people carry with them along with the tablets and smartwatches. Most smartphones are equipped with different types of sensors, including an accelerometer, magnetometer, gyroscope, compass, and GPS. In this work, we use the data acquired from a smartphone's accelerometer to identify the movement of a pedestrian and the smartphone compass to determine the direction of a pedestrian. Classification algorithms such as (J48) which is an open source java implementation of the C4.5 Decision Tree (DT) algorithm, k-nearest neighbor (kNN), Rule based (JRip) and Meta-level (bagging and boosting) are used to determine the type of movement, including slowing down to a stop, accelerating, and decelerating. We show that these classification algorithms achieve accuracies between 93.39% and 96.98% and are capable of recognising pedestrians movement within 500ms.

The direction of pedestrians movement is determined by using the accelerometer and compass sensor of a smartphone. To ensure the accuracy of a pedestrian movement direction, the smartphone's orientation must be pre-aligned to the pedestrian's orientation. If the smartphone and the pedestrian orientation is not aligned, then the direction obtained from the compass will not represent the actual movement direction of the pedestrian. Therefore, we present an algorithm that is independent of the smartphone's orientation that automatically aligns the smartphone orientation to the direction of movement when the pedestrian completes two steps while the smartphone is placed in a front trouser pocket. Our proposed automated algorithm reaches an accuracy of 96% and detects changes in pedestrians' directions within 250ms.

In this dissertation, we also studied the influence of the magnetic deviation on the compass while measuring the movement direction. To illustrate the influence of magnetic deviation, we designed a filter to differentiate between safe and endangered pedestrians based on their movement direction. We observed that magnetic deviations could influence the accuracy of the filter, so our algorithm compensates for these effects using the gyroscope of a smartphone. Finally, we investigate another important aspect that is the energy consumption of the smartphone sensors such as an accelerometer, gyroscope, magnetometer and compass, which allows us to analyse if a particular smartphone may be efficiently used for movement recognition and direction detection of pedestrians. We found that almost all investigated smartphones were capable of providing battery duration for about 24 hours excluding one smartphone model (SII GTI9100). We cannot tell which part of the (SII GT-I9100) consumes the energy. We can only assume that maybe there is the hardware failure, or the installed operating system may have the bug. Moreover, our analysis suggests that most smartphones could be used for daily sensing. Algorithms developed here are not limited to a pedestrian safety but may also be considered for additional applications, such as pedestrian dead reckoning and navigation.

Zusammenfassung

Nach offiziellen Angaben der Bundesanstalt für Straßenwesen (BASt), sind im Jahr 2016 insgesamt 15,400 Fußgänger bei Verkehrsunfällen in 36 Ländern ums Leben gekommen [1]. Automobilhersteller und Wissenschaftler arbeiten an passiven und aktiven Lösungen, welche zum Schutz von Fußgängern beitragen sollen. Passive Fußgängerschutzsysteme verfolgen den Ansatz, die Auswirkungen einer Kollision zwischen Fußgänger und Automobil zu mindern. Aktive Systeme hingegen verwenden Fahrzeugsensoren (z.B. Radar und Kameras), um Kollisionen von vornherein zu vermeiden, benötigen hierzu jedoch einen direkten Sichtkontakt. Fahrzeuge und Fußgänger tauschen Informationen über die Bewegung des jeweils anderen aus, wie bspw. Position, Richtung oder Geschwindigkeit. On-Board-Vermeidungsalgorithmen berechnen das Risiko einer Kollision berechnen und gegebenenfalls Warnsignale senden. Kooperative Ansätze verwenden drahtlose Kommunikationstechnologien, zum Beispiel Radiofrequenzkommunikation [2], [3] oder GPS [4], um Fußgänger zu erkennen, beides drahtlose Kommunikationstechnologien funktionieren ohne die Einschränkung der Sichtverbindung. Der Nachteil von [2], [3] liegt in der geringen Kommunikationsreichweite, während die geringe Abtastrate d.h. 1Hz bei GPS [4] eine praktikable Erkennung von Fußgängerbewegungen verhindert. Als eine Alternative zu diesen Ansätze, wird in dieser Dissertation eine Methode vorgestellt, welche Smartphone-Sensoren verwendet, um die Fußgängerbewegung (basierend auf Beschleunigungsdaten) sowie die Richtung (basierend auf Kompassdaten) genau erkennt. Der vorgeschlagene Ansatz arbeitet unabhängig von GPS und die Erkennungszeit der Fußgängerbewegung und Richtung im Vergleich zu einem GPS zu verbessert.

Neben Tablets und Smartwatches, sind Smartphones die von Personen am häufigsten mitgeführten Geräte. Sie sind mit verschiedenen Sensoren, unter anderem Beschleunigungssensor, Magnetometer, Gyroskop, Kompass und GPS ausgestattet. Basierend auf den in Smartphones verbauten Beschleunigungssensoren sowie Kompasse, wird in dieser Dissertation ein alternativer Ansatz entwickelt, welcher Bewegungsaktivitäten sowie Bewegungsrichtungen eines Fußgängers bestimmt. Die Bewegungsart, inklusive dem langsamen Gehen bis zum Halt, dem Beschleunigen und Abbremsen der Gehbewegung, wird mithilfe von verschiedenen Klassifizierungsalgorithmen, unter anderem (J48) die eine Open source-Java-Implementierung von C4.5 Decision Trees (DT) Algorithmus, K-Nearest-Neighbor (kNN), Rule-Based (JRip) und Meta-Klassifizierungsalgorithmen (Bagging, Boosting) erkannt. Es wird gezeigt, dass die verwendeten Klassifizierungsalgorithmen die Bewegungsart mit einer Genauigkeit von 93,39% bis 96,98% und innerhalb einer Zeitspanne von unter 500ms erkennen können. Die Richtung eines Fußgängers wird bestimmt, indem Beschleunigungsdaten mit Kompasswerten des Smartphones ergänzt werden. Um sicherzustellen, dass Bewegungsrichtungen korrekt erkannt werden, wird die Ausrichtung des Smartphones mit der des Fußgängers abgeglichen. Wird diese Ausrichtung des Smartphones nicht korrigiert, entspricht die vom Kompass stammende Richtung nicht mit der tatsächlichen Bewegungsrichtung des Fußgängers überein. Zur Korrektur der Ausrichtung wird ein Algorithmus vorgestellt, der unabhängig von der aktuellen Ausrichtung des Smartphones, diese innerhalb von zwei Schritten automatisch auf die Bewegungsrichtung des Fußgängers ausrichtet. Dieser Algorithmus erkennt Änderungen in der Bewegungsrichtung in einer Zeitspanne von unter 250ms und einer Genauigkeit von 96%.

Zusätzlich, wird in dieser Dissertation der Einfluss von magnetischen Störungen auf die vom Kompass errechnete Bewegungsrichtung untersucht. Um Störungen des Kompasses zu entgegnen, wurde ein Filter entwickelt, der zwischen gefährdeten und nicht gefährdeten Fußgängern auf der Grundlage ihrer Gehrichtung unterscheidet. Wie beobachtet können magnetische Störungen die Genauigkeit von diesem Filter beeinflussen. Daraufhin wurde ein Algorithmus entwickelt, der die Auswirkung von magnetischen Störungen auf Kompassdaten kompensiert. Schließlich wird der Energieverbrauch der Sensoren wie dem Beschleunigungsmesser, Gyroskop, Magnetometer und Kompass verschiedenen Smartphones untersucht. Es wird analysiert, ob ein bestimmtes Smartphone sich im Besonderen für die Bewegungserkennung und Richtungserkennung von Fußgängern eignet. Wir analysierten, dass fast alle untersuchten Smartphones ohne eine Smartphone Model (SII-GTI9100) in der Lage waren, die Akkulaufzeit von etwa 24 Stunden zu gewährleisten. Wir können nicht sagen, welcher Teil (SII GT-I9100) die Energie verbraucht. Wir können nur davon annehmen, dass möglicherweise ein Hardwarefehler vorliegt oder dass das installierte Betriebssystem den Fehler aufweist. Unsere Analyse legt nahe, dass die meisten Smartphones für die tägliche Wahrnehmung verwendet werden könnten. Die in dieser Dissertation vorgestellten Algorithmen beschränken sich nicht auf den Schutz von Fußgängern, sondern können auch in anderen Anwendungen wie Fußgänger-Koppelnavigation (Pedestrian Dead Reckoning), und zur Navigation eingesetzt werden.

Acknowledgement

First of all, I would like to express my special appreciation and thanks to my supervisor and mentor Prof. Dr.-Ing. Klaus David, Head of the Chair for Communication Technology (ComTec), University of Kassel, Germany, who provided me an opportunity to do Ph.D. in ComTec. His guidance and immense knowledge helped me during the time of research and writing of this thesis. His advice on both research as well as on my career has been invaluable. I am also grateful to him for his continuous support and motivation during my Ph.D. study, without which this work would not have possible. I could not have imagined having a better supervisor and mentor for my Ph.D. study.

I would like to thank Prof. Dr. rer. nat. Hartmut Hillmer, Prof. Dr. rer. nat. Bernhard Sick, and and Prof. Dr. rer. nat. Gerd Stumme for being part of my Ph.D. committee. I also want to thank all of you for valuable comments and suggestions.

The research and writing of this dissertation required the continuous support of many people during my 5-year journey. Beside my supervisor, I would like to thank, Dr. Rico Kusber, Dr. Alexander Flach, Dr. Sian Lun Lau, Dr. Immanuel Koenig, Andreas Jahn, Dennis Kroll and all other ComTec team members for their assistance and critical discussions during the solution of technical research questions and their sympathetic and moral support during this journey. I would also like to thank those who have participated in the data collection sessions.

I am truly grateful to Quaid-e-Awam University of Engineering Science and Technology, Nawabshah, Pakistan for the financial support, which offered me a scholarship to pursue higher education. I am also grateful to the Chair for Communication Technology (ComTec), University of Kassel, Germany for partial financial support during the end of this research, without which this work would not have completed.

Dedicated to

My loving parents (Late: Abdul Ghani Memon and Shamim Akhtar),

My caring wife, Samreen,

Our dearest kids, Emaan, Ehsaal and Abdul Ghani,

My respectable brothers Abdul Qadeer and Abdul Qayoom,

All family and friends,

For all their love and support and for putting me through the best education possible. I appreciate their sacrifices and I would not have been able to get to this stage without them.

Table of Contents

Abstract i

Zusammenfassung iii

Acknowledgement v

Table of Contents vii

List of Publications x

1 Introduction 1

1.1 Challenges 4

1.2 Research Methodology 5

1.3 Contribution 7

1.3.1 Pedestrian Movement Recognition 7

1.3.2 Pedestrian Movement Direction Detection (PMDD) Independent of Smartphone Orientation 8

1.3.3 Investigation and Compensation of the Magnetic Deviation on a Smartphone Compass 8

1.3.4 Energy Consumption of the Smartphone sensors................................. 9

1.4 Outline of the Dissertation 10

2 State of the art 11

2.1 User's movement direction detection approaches.. 11

2.1.1 Pedestrian Dead Reckoning using shoe-mounted Inertial Measurement Unit 12

2.1.2 Smartphone-based approaches 14

2.1.2.1 Smartphone handheld approaches... 15

2.1.2.2 Smartphone body attached approaches (fixed) 17

2.1.2.3 Velocity vector approaches based on smartphone sensors 18

2.1.3 Sensor Fusion: Smartphone sensors and map-based approaches 19

2.1.4 Comparison of Approaches 22

2.2 Existing approaches in activity recognition 25

3 Pedestrian Movement Recognition 27

3.1 Introduction 27

3.2 Movement Recognition using Accelerometer respectively GPS 29

3.2.1 Data collection.. 29

3.2.2 Feature extraction of accelerometer based movement recognition of a pedestrian ... 29

3.2.2.1 Sliding window data preparation ... 30

3.2.2.2 Calculation/transformation of feature extraction: ... 32

3.2.3 Classification algorithms used for movement recognition ... 32

3.2.3.1 Base-level classification Algorithms ... 33

3.2.3.2 Meta-level classification Algorithms ... 38

3.2.4 Results discussion of pedestrian movement recognition using accelerometer of a smartphone ... 42

3.2.5 Systems based on GPS only ... 45

3.3 Impact of the measured results on an accident scenario ... 48

3.4 Conclusion ... 50

4 Pedestrians Movement Direction Detection ... 51

4.1 Introduction ... 51

4.2 Methodology and Approach ... 55

4.2.1 Hardware specification of built-in sensors of a smartphone ... 55

4.2.2 Effect of smartphone orientation on pedestrians movement direction ... 56

4.2.3 Smartphone Orientation and Movement Direction Alignment (SOMDA) algorithm ... 59

4.3 Results and Discussion ... 64

4.3.1 Accuracy evaluation ... 64

4.3.2 Comparison of SOMDA and A-GPS ... 67

4.4 Conclusion ... 71

5 Investigation and Compensation of the Magnetic Deviation ... 72

5.1 Introduction: ... 72

5.2 Investigation of the magnetic deviation: ... 74

5.2.1 Experiment 1 (E1): Magnetic deviation due to the car at different distances ... 74

5.2.2 Experiment 2 (E2): Magnetic deviation due to the orientation of a car ... 75

5.2.3 Experiment 3 (E3): Magnetic deviation due to the orientation of the smartphone ... 76

5.2.4 Experiment 4 (E4): Magnetic deviation caused by a moving vehicle ... 76

5.3 Efficiency of filtering pedestrians ... 79

5.4 Compensation algorithm for magnetic deviation ... 82
5.5 Results and Discussion ... 86
5.6 Conclusion ... 95

6 Energy consumption of sensors of smartphones ... 96
6.1 Introduction ... 96
6.2 Methodology and Approach ... 97
6.2.1 Specification of used Smartphones ... 97
6.2.2 Limitations of used Smartphones ... 100
6.2.3 Measurement setup ... 100
6.2.4 Voltmeter and Ammeter (device based) measurements ... 102
6.2.5 Software-based measurements ... 102
6.3 Results ... 103
6.4 Discussion of the results ... 106
6.4.1 Duration of a battery charge using API and device based measurements ... 106
6.4.2 Comparison of API and device based measurements ... 108
6.4.3 Comparison of energy consumption and battery runtime using different sampling rate ... 110
6.5 Conclusion ... 112

7 Conclusion and Outlook ... 113
7.1 Summary and Conclusion ... 113
7.2 Outlook ... 118

List of Figures ... 120

List of Tables ... 123

Bibliography ... 124

List of Publications

Parts of the work conducted within the scope of this Ph.D. dissertation have published in conferences ([7], [13], [18] and [85]). These publications are listed below.

1 Flach, A. Q. Memon, S. L. Lau, and K. David, “Pedestrian movement recognition for radio-based collision avoidance: A performance analysis,” IEEE 73rd Vehicular Technology Conference (VTC Spring), pp. 1–5, Budapest, Hungary, May 15-18, 2011. https://doi.org/10.1109/VETECS.2011.5956722.

2 Q. Memon, S. L. Lau, and K. David, “Investigation and Compensation of the Magnetic Deviation on a Magnetometer of a Smartphone Caused by a Vehicle,” IEEE 78th Vehicular Conference (VTC Fall), pp. 1–5, Las Vegas, USA, September 2-5, 2013. https://doi.org/10.1109/VTCFall.2013.6692249.

3 König, A. Q. Memon, K. David, "Energy consumption of the sensors of Smartphones," The Tenth International Symposium on Wireless Communication Systems (ISWCS 2013), pp. 1-5, TU Ilmenau, Germany, August 27-30, 2013.

4 R. Kusber, A. Q. Memon, D. Kroll, and K. David, "Direction Detection of Users Independent of Smartphone Orientations," IEEE 82nd Vehicular Technology Conference (VTC2015 Fall), pp. 1-5, Boston, USA, 6-9 September 2015. https://doi.org/10.1109/VTCFall.2015.7390890.

1 Introduction

The work presented in this dissertation is part of the research group "Communication Technology-ComTec", which is an individual research and based on the earlier efforts from [5].

The recognition of pedestrian movements, such as accelerating, decelerating, slowing to a complete stop, and direction detection is the basis of many pedestrian safety and pedestrian dead reckoning applications. As motivation for our work in enhancing pedestrians safety, we shortly talk about a car-to-pedestrian collision, each year thousands of pedestrians are injured or killed in a car-related accidents. Official figures from the Federal Highway Research Institute (Bundesanstalt für Straßenwesen (BASt)) collected from 36 countries report that about 15,400 pedestrians died in traffic accidents in 2016 [1]. These accidents mainly occurred due to the inattention of the car driver, the misinterpretation of pedestrians speed and movement direction, or simply because pedestrians or drivers were unaware of the dangerous situation.

The car manufacturers and various research groups address the challenge of improving pedestrian safety using two different approaches: passive and active pedestrian protection. The purpose of passive pedestrian protection is to reduce the impact of a collision [6] and [7], where the design of a car is modified to create softer impact zones in the front, such as rising hoods or pedestrian airbags [8], [9] and [10]. Alternatively, the aim of active pedestrian protection is to work towards collision avoidance. The active pedestrian protection approach uses cooperative-based techniques to detect pedestrians from a certain distance and to recognise their movements and direction. These techniques use algorithms that calculate the point of collision between the car and the pedestrian and send a warning signal to both participants. Based on the warning signal, the car driver reacts by applying the brakes or manoeuvring to avoid a collision. One of the examples of a cooperative based approach is presented in [4], which utilises the GPS to determine cars and pedestrians position and movement direction and applies the risk estimation algorithm to calculate the collision probability. An advantage of using the GPS is that it does not requires direct-line-of-sight communication between a car and a pedestrian and still works even if the pedestrian is hidden to the moving car, as shown in Figure 1-1. In this dissertation, we investigate if the GPS data provides timely enough updates of the pedestrian movement. The answer based on our analysis is "Conditionally yes, providing the pedestrian speed remains constant”. However, it is likely in real scenarios that pedestrians will change their speed or direction frequently, such as walking at a varying rate of speed, sudden acceleration to catch a bus or tram or stop at a curb while a car passes. In these cases, use of the GPS will cause a delay in determining updates of the pedestrian movement.

We present observations suggesting that GPS of a smartphone functioning at a maximum sampling frequency of 1Hz, which might not fast to provide real-time updates in the movement history to avoid collisions between cars and pedestrians. For example, every time a pedestrian changes his speed or direction, the GPS response in updating the movement and direction information takes 3s to 4s (see section 3.2.5). Based on our analysis, the approach presented in [4] may result in an incorrect estimation of the collision risk because of the time lag in feedback. During this time delay, the pedestrian may be already endangered as the risk estimation algorithm has failed to send the warning signal in time or send the false warning signal as the pedestrian new information is not updated. To improve the recognition time of the pedestrian movement and direction, we present an alternative to using GPS that uses an accelerometer and a compass, and we investigate how the approach can timely recognise pedestrian movement (using the accelerometer) and direction (using the compass). An early recognition of pedestrians movement and direction can benefit applications such as pedestrian safety, navigation, and pedestrian dead reckoning.

Multiple investigations exist for detecting user’s activities and the movement direction based on the sensors such as the accelerometer, compass, gyroscope and the GPS. The aforementioned sensors can be used as standalone or integrated into a sensor board. These sensors can be installed on the user's body or clothing, such as on the user’s belt or shoes. The accelerometer and gyroscope measure the acceleration and angular rate on different axes. The compass measures the movement direction, whereas the GPS determines the position and heading of the user. Nowadays, these sensors are also readily available on smartwatches and tablets. Therefore, smartwatches and tablets can also determine user’s movement and direction. According to the United Nations report, in 2018 the world population has reached to 7.63 billion [118], and by the end of 2019, 3.8 billion people use the mobile phone all over the world [119], which is the half of the world population. Currently, smartphones, tablets, and smartwatches are the most common types of devices carried by people. A survey in [14] shows, the estimated number of smartphone users increases to 2.53 billion by the end of 2018. Moreover, this number is continuously increasing and will reach 2.87 billion by 2020. This survey includes all the people of different age who possess the smartphone and uses it once in a month. Therefore, we propose the use of smartphone sensors as a matter of convenience because smartphones are so common device people carry with them along with tablets and smartwatches. Most smartphones are now equipped with different types of sensors, including an accelerometer, magnetometer, gyroscope, compass, and GPS. For this research, we use data acquired from the smartphone accelerometer to identify the pedestrian movement and the smartphone compass to determine pedestrian direction. In particular, we address the following questions and review additional details regarding challenges and methods to overcome these problems:

1. Using smartphone sensors such as an accelerometer and compass, can we determine the direction of pedestrian movements regardless of a smartphone orientation and without any help from the GPS?
2. Using a compass of a smartphone, we determine the movement direction of a pedestrian. Magnetic interference influences the compass. Such deviations might happen while determining the movement direction of a pedestrian. How can we solve the magnetic deviation problem in a compass of a smartphone caused by surrounding magnetic material?
3. With the utilisation of smartphone sensors, what is the energy consumption of these sensors? Is it feasible to use the smartphone sensors to determine the pedestrian movement and their direction?

This chapter is organised into four sections. In Section 1.1, we describe in detail the specific problems related to our work. In Section 1.2, we present the methodology about how to solve these problems. In Section 1.3, we give the summary of contributions. Finally, in Section 1.4, we present the structure of the dissertation.

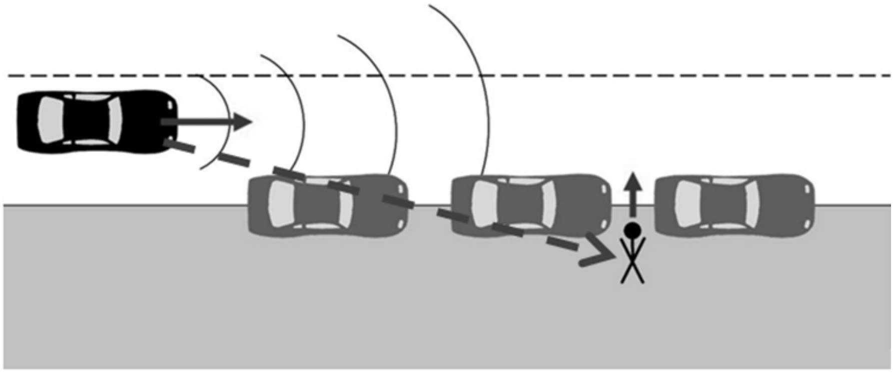


Figure 1-1 Typical accident scenario [12]

1.1 Challenges

In [5] and [12], the authors analysed a typical accident scenario, as shown in , where a pedestrian is hidden to the moving car travelling at a speed of 50km/h. To avoid this type of accident, Flach presented a collision avoidance approach that exchanges position, speed, and direction of the car and the pedestrian using a radio-based communication system. In [5], Flach also discussed the importance of the time required for processing to avoid a collision between cars and pedestrians by categorising the time into different intervals, including system time available, reaction time, and braking time. Here, the system time available is the total time for detection of pedestrians and their position, transmission of position updates, calculation of collision probability, and the sending of a warning message in case of a dangerous situation. Reaction time is the time in which the driver reacts and applies the brakes. Braking time is the time interval when the brakes are applied and the car coming to a complete stop.

An approach presented in [4] uses the GPS to determine the position, speed, direction of the car and the pedestrian. Such approach works well in a typical accident scenario where the speed of the pedestrian remains constant. However, if the pedestrian changes speed, then the approach presented in [4] may not be able to avoid the collision because the GPS-based system must first determine the new pedestrian speed, which adds a time delay of 3 to 4s (see section 3.2.5) before recalculating the collision risk. Based on the time available to avoid a collision, which can be 2.1s as suggested in [5] and [12], and due to the GPS delay of 3 to 4s (see section 3.2.5) in recognising new movement updates, it may be late to avoid the collision. In this case, the GPS sampling frequency of 1Hz might not fast to provide real-time updates in the movement history of pedestrians. Therefore, an enhanced solution is required to avoid the delay and to recognise the pedestrian movements and detect their direction sooner.

In this dissertation, we propose to use other sensors of a smartphone, including the accelerometer to recognise the pedestrian movements and the compass to determine movement directions. We identified problems using smartphone sensors while determining movement and direction of pedestrians, which are mentioned below. In Section 1.2, we discuss the solutions to these problems.

1. To infer the correct direction of the pedestrian movement from the compass azimuth, the smartphone orientation must be the same as that of the pedestrian's orientation. When users/pedestrians carry a smartphone, such as in the trouser pocket, their orientation (user/pedestrian frame) differs from the smartphone orientation (device frame) resulting in data collected from the smartphone not being in alignment with the user's orientation. For example, data collected from the smartphone accelerometer for the X-axis relative to the device frame may not align with the user's acceleration along the relative X-axis of the user/pedestrian frame. We call this issue as the "smartphone in the pocket problem."
2. The compass of a smartphone is sensitive to the magnetic interference caused by nearby magnetic materials. The magnetic interference introduces an error as it deflects the actual magnetic North measured by the compass, which will then influence the accuracy of direction detection of the pedestrian movement.
3. Finally, we investigate another important consideration regarding the energy consumption of smartphone sensors. This area is not thoroughly studied and is challenging because the smartphone battery capacity is limited and smartphone manufacturers consider energy consumption of sensor chipsets proprietary information. Our investigation allows us to determine if a particular smartphone may be efficiently used for movement recognition and direction detection of pedestrians in terms of power consumption.

1.2 Research Methodology

In Chapter 3 (see section 3.2.5) and [13], we observed that if the pedestrian changes the walking speed, which is often the case in real life scenario, in this case, the GPS causes delay of 3 to 4s to recalculate the new speed and direction. In contrast to [4], we avoid the delay in determining pedestrian movements and direction detection caused with the GPS by proposing to incorporate data from other sensors inside a smartphone, such as an accelerometer, gyroscope, and compass. Smartphones are typically equipped with types of sensors that will be useful in determining the early changes in the pedestrian's movement and direction. In chapter 3 and our previous paper [13], we analysed the recognition of the pedestrian's movement such as accelerating, decelerating, and slowing to a stop using the GPS of a smartphone. In those results (see Section 3.2.5), we show that with the GPS it takes 3 to 4s to recognise the pedestrian movement. Similarly, we show in Chapter 4 (see section 4.3.2) that the movement direction detection of pedestrians, covering a full direction range from 0-360 degrees with respect to North, is also possible within 3 to 4s using the GPS of a smartphone. Also, we determine that if the pedestrian is standing in place, rotating clockwise or counter clockwise, the GPS is not able to detect the turning movement, as the GPS requires a minimum of two successive locations to determine the direction. Finally, we analysed that the GPS takes 3 to 4s to provide real-time updates in the movement history of pedestrians, which is slow and especially a concern for pedestrian safety applications where the time constraint is limited, i.e., 2.1s [5] and [12] to avoid a collision between a car and a pedestrian.

As mentioned in the previous section 1.1, our alternative approach to the GPS is to recognise pedestrians movement (using the accelerometer), and determine direction (using the compass) offer several challenges. How we address those issues are summarised below and further discussed in Chapter 3 to 6.

1. To resolve the “smartphone in the pocket problem,” we propose the algorithm, which uses sensors data independently of the smartphone’s orientation. The algorithm automatically aligns a smartphone orientation to the direction of movement after the pedestrian completes two steps when the smartphone is placed in a front trouser pocket. The accelerometer of a smartphone is used to determine the pitch and roll angle with respect to a device coordinate system (also referred as device frame) and also used to determine the step peak detection. Based on the pitch, roll and step peak detection information, the algorithm identifies a smartphone orientation and then aligns the smartphone orientation to the movement direction.
2. To manage magnetic interference with the smartphone compass, we propose the algorithm, which compensates for the magnetic deviations detected. Our compensation algorithm utilises additional data from the smartphone’s gyroscope so it can differentiate between the magnetic deviation and the rate of turn. The algorithm first checks if there is a change in the compass data while simultaneously no turn is observed from the gyroscope data, then the algorithm categorises the detected change as a magnetic deviation and compensates accordingly.
3. Two techniques are used to investigate the energy consumption of smartphone sensors. First, a voltmeter and ammeter are connected to the battery and the smartphone in series connection to record the flow of current and then determine

the actual power consumption of a specific sensor. Second, software installed on the smartphone that records the energy consumption from the internal smartphone Application Program Interface (API). There is no certain way to calculate the energy consumption of smartphone sensors because smartphones does not have the hardware capability to calculate the current. Therefore, we choose two different approaches to investigate and validate the energy consumption of used sensors.

1.3 Contribution

1.3.1 Pedestrian Movement Recognition

In chapter 3, we present an approach to recognise pedestrians movement, including acceleration, deceleration and slowing to a stop using the accelerometer of a smartphone. To determine the accuracy of the movement recognition, we use a variety of base-level classification algorithms, including Decision Tree (DT), K-nearest neighbor (kNN), and Rule-based (JRip). Meta-level classifiers further improved these accuracies, and in our results, we show that the classification algorithm achieves recognition accuracies between 93.39% and 96.98%. All classifiers complete the desired recognition within 250ms, and the feature extraction process requires less than 200ms. It is possible to obtain complete movement recognition under 250ms by adjusting the values of the sampling rate, window length, and overlapping percentage.

Based on our results presented in Chapter 3 (see section 3.2.4), the combined base- and meta-level classification algorithms determine the pedestrian’s movement such as stop, deceleration, or acceleration in less than 500ms, while GPS takes 2s to achieve the same determinations only if the pedestrian is walking at a constant speed. If the pedestrian changes the walking speed, GPS takes 3 to 4s (see section 3.2.5) to achieve the same movement. Therefore, when comparing with the GPS sampling frequency of 1Hz, our proposed approach has an advantage of being 1.5s when the pedestrian is walking at a constant speed and 3.5s faster in determining the pedestrian movement when the pedestrian is walking at varying speeds. The information acquired from the accelerometer only recognises the different movements of a pedestrian but not the direction of movement. Therefore, we propose the addition of a compass sensor to determine the pedestrian movement direction in conjunction with movement recognition.

1.3.2 Pedestrian Movement Direction Detection (PMDD) Independent of Smartphone Orientation

To infer the movement direction of a pedestrian using the compass of a smartphone, the orientation of the smartphone with respect to the pedestrian's movement direction must be the same. If this is not the case, then the compass azimuth will provide misleading data of the movement direction as shown in our results (see Section 4.2.2). Since pedestrians will likely carry a smartphone with different orientations, we investigate how a range of orientations of the smartphone in the pocket results in different data even though the test path for collecting the data remains the same for all smartphone orientations. In our experiments, we show that most smartphone orientations are not the same as the pedestrian movement direction, which leads to inaccurate movement direction detection. Therefore, we developed the algorithm called "Smartphone Orientation and Movement Direction Alignment" (SOMDA) that accurately aligns the smartphone orientation to the movement direction of the pedestrian. The SOMDA algorithm uses both the accelerometer and the compass of a smartphone and is independent of the smartphone orientation as well as the GPS. It determines the orientation of a smartphone in a pocket based on the pitch and roll angle in conjunction with the step peak detection obtained from the accelerometer. Once the orientation of the smartphone is identified, then the algorithm aligns the smartphone orientation to the movement direction of the pedestrian. We demonstrate that the SOMDA algorithm reaches 96% accuracy (see Chapter 4).

1.3.3 Investigation and Compensation of the Magnetic Deviation on a Smartphone Compass

In [17], a weakness in the compass of a smartphone is identified such as the magnetic deviation that is caused due to the interference of magnetic material. As shown in the results (see Section 5.2), the existence of magnetic deviations influences the performance of a compass. As an example, we design a filter that is able to differentiate safe and endangered pedestrians in a car to pedestrian collision avoidance scenario based on their movement direction. The pedestrian moving on sidewalks parallel or away from the road are being filtered out and not considered as endangered. Similarly, pedestrians moving towards the road are considered as endangered. The designed filter assumes that pedestrians moving on sidewalks are equally distributed in all directions. In this case, filter recognises as the 50% of pedestrians as endangered as they are moving towards the road and 50% as safe as they are moving away from the road. Moreover, we show that the existence of magnetic deviation on a compass influences the efficiency of safe pedestrians. We propose the algorithm called “Magnetic Deviation Compensation Algorithm (MDCA)” to compensate for magnetic effects, which makes use of the smartphone’s gyroscope. The gyroscope can differentiate between the rate of turn and the presence of a magnetic deviation in a compass. The MDCA is shown to compensate for the magnetic deviation successfully. The compensation algorithm first checks the change in the compass data, if the algorithm see any changes in compass, then it will directly jump to gyroscope data to validate the change in compass data is a turn. In parallel, the algorithm looks for the change in the gyroscope data, and if any observed, it considers the change in compass as a turn. Similarly, if there is no change seen in the gyroscope data, the algorithm considers the change in compass data as a magnetic deviation, which is compensated accordingly. For further detail, see Chapter 5.

1.3.4 Energy Consumption of the Smartphone sensors

Our investigation of the energy consumption of smartphone sensors is another important aspect because in this research smartphone sensors are used to provide the pedestrian movement recognition and direction detection. The idea of investigating the energy consumption using two different methods is to validate the accuracy of energy consumption of individual or combined sensors. Firstly, we use the voltmeter and the ammeter (called as hardware-based measurements), which connects with the battery and the smartphone in a series connection. The voltmeter and ammeter are also connected to the computer using USB cable, PC-link data recording software is installed on a computer to record the flow of current when the sensors are switched ON. To validate the energy consumption obtained using hardware-based measurements, we use the self-written software, which is installed on a smartphone and measures the energy consumption from the internal smartphone Application Program Interface (API) when an individual or combination of sensors are switched ON. We show that our investigation confirms that a particular smartphone can be used effectively for pedestrian movement recognition and direction detection along with how long the battery will maintain a sufficient charge during sensing activity (see Chapter 6). Moreover, we conclude that almost all investigated smartphones managed to cover a sensing time of approximately 24 hours and could be expected to work for daily sensing and context surveys excluding one smartphone model (SII GT-I9100). We cannot tell which part of the (SII GT-I9100) consumes the energy. We can only assume that maybe there is the hardware failure, or the installed operating system may have the bug [18].

1.4 Outline of the Dissertation

This dissertation is organised into seven chapters. In Chapter 2, we describe in detail state-of-the-art. In Chapter 3, we present our approach to recognise the pedestrian movement using the smartphone accelerometer. We used different classification algorithms to ensure an accuracy of the pedestrian movement. In Chapter 4, we present the algorithm that is independent of the smartphone’s orientation that automatically aligns the smartphone orientation to the direction of movement after the pedestrian completes two steps and assumes the smartphone is located in a front trouser pocket. In Chapter 5, we investigate the magnetic deviation in a smartphone compass and analyse how this influences the accuracy of the designed filter that distinguishes between a safe and endangered pedestrians in a typical accident scenario. Therefore, we develop an additional algorithm to compensate for magnetic deviation effects. In Chapter 6, we present an investigation of the energy consumption of the smartphone’s sensors using two different approaches. We present the energy consumption of smartphone sensors using the hardware and software-based measurements. Finally, in Chapter 7, we present the overall conclusion of this work.

2 State of the art

In this chapter, we present and discuss state of the art conducted within the scope of this dissertation. Existing research is categorised into the following to facilitate our review, which includes comparisons with how previous work relates to the research presented in this dissertation.

- User’s movement direction detection approaches (Section 2.1)
  - Pedestrian Dead Reckoning (PDR) approaches using a shoe-mounted Inertial Measurement Unit (Section 2.1.1)
  - Smartphone-based approaches (Section 2.1.2)
    - Smartphone-based handheld approaches (Section 2.1.2.1)
    - Smartphone-based body attached (fixed) approaches (Section 2.1.2.2)
    - Velocity vector approaches based on smartphone sensors (Section 2.1.2.3)
  - Sensor fusion-based approaches (Section 2.1.3)
- Movement detection (activity recognition) of pedestrians using an accelerometer (Section 2.2)

2.1 User’s movement direction detection approaches

Multiple investigations exist for detecting the movement direction of users based on sensors such as the compass, accelerometer, gyroscope, and the GPS. These sensors are integrated into smartphones, the Inertial Measurement Unit (a sensor board combining the accelerometer, magnetometer, gyroscope, and compass to measure or sense the acceleration, movement direction, angular rate, and surrounding magnetic field), and self-designed sensor boards. Identifying the movement direction of a user/pedestrian can benefit applications, such as pedestrian safety, navigation, and pedestrian dead reckoning.

In this section, we present state of the art on the movement direction detection of users. To the best of our knowledge, no investigation is publically available that studies the influence of the orientation of the smartphone, self-designed sensor board, or Inertial Measurement Unit while simultaneously detecting the movement direction of the user. Furthermore, we found no publication that demonstrated the detection of the movement direction (with respect to North) of a user independent of smartphone orientations. The work we present in this dissertation determines the movement direction of a user/pedestrian independent of a smartphone orientation when carried in a trouser pocket. All of the previous approaches described in Section 2.1 determine the movement direction of the user based on the following predetermined assumptions.

- Fixed position and known orientation of the self-designed sensor board as determined by:
    - mounting on a shoe.
    - mounting on a belt.
- Fixed position and known orientation of the smartphone as determined by:
    - Smartphone carried (fixed) in a pocket with a known orientation.
    - Smartphone carried in hand with a known orientation.
- User movements are inferred from the GPS or an accelerometer data.

Moreover, in Section 2.1.4, we present comparisons with how previous work relates to the research presented in this dissertation.

2.1.1 Pedestrian Dead Reckoning using shoe-mounted Inertial Measurement Unit

In this section, we summarise the Pedestrian Dead Reckoning approach that determines the movement direction using the shoe-mounted Inertial Measurement Unit. First, we present related approaches, and then in section 2.1.4 we discuss their relation to existing research as well as the research presented in this dissertation. We also present state of the art on determining movement direction and estimation of the user's subsequent position and on error correction in the user's movement direction due to bias in the sensor data. Pedestrian Dead Reckoning uses a combination of multiple sensors, such as an accelerometer, magnetometer, gyroscope, or the GPS to estimate the position of a user. It first determines the user steps and then calculates the displacement. Finally, it determines the user's heading and estimates the new position of a user based on the previously calculated displacement, heading estimation, and known location. The major disadvantage of Pedestrian Dead Reckoning approach is that it requires a known initial location via another sensing mechanism, such as the GPS or WiFi, or the user must manually set it through the digital maps in an applications graphical interface (GUI). Moreover, the pedestrian dead reckoning approach will not function properly if the initial location is unknown.

In [19], Stirling et al. proposed a method utilising a self-designed sensor board integrating the accelerometer, magnetometer, and gyroscope sensor, which is installed on a pedestrian’s shoe. Using the sensors data, proposed method determined the step detection, step length, displacement, and movement direction of the user carrying the self-designed sensor board. A similar approach was presented in [20], which uses a commercially available Inertial Measurement Unit, XSense MTX, instead of the self-designed sensor board. The XSense MTX is attached to a pedestrian shoe, similar to [19], and the user’s heading is determined based on the sensor’s data. In [29], another similar approach to [19] and [20] was presented, that uses the self-designed sensor board, Nevmote, which contains an accelerometer, magnetometer, and gyroscope sensor. The Nevmote is firmly attached to a user’s body, such as on the belt, and the approach determines the step detection, estimation of step length, and movement direction of a user.

In [23], an Inertial Navigation System (INS) was installed on a pedestrian’s shoe to determine the movement direction of the user. The Inertial Navigation System combines the motion sensor, such as an accelerometer, and the rotation sensor, such as a gyroscope, and processes the data to calculates the position, orientation, and velocity of the object. Moreover [23] presented a method to correct drift errors in the sensors installed on the Inertial Measurement Unit.

In [21], Enez et al. presented an approach that estimates the position and heading of a user by using the accelerometer and gyroscope installed on the Inertial Measurement Unit attached at a fixed position, such as on the user's foot, similar to [19], [20], and [23]. To minimise the heading drift error in the Inertial Measurement Unit sensors, they proposed a drift elimination method.

In [22], a localisation approach using a pedestrian dead reckoning technique was presented, which uses the Inertial Measurement Unit installed on a shoe, also similar to approaches from [19], [20], [21], and [23]. Here, the Inertial Navigation System determines the movement direction of a user by integrating the accelerometer and gyroscope data. A 3-axis magnetic field sensor is simultaneously used to correct the heading error, comparing the user heading obtained from the accelerometer and gyroscope sensors data and compensating for differences in the inertial navigation System. The approach requires a known initial position via another sensing mechanism, such as the GPS or WiFi, or the user must manually set it through the application’s graphical user interface (GUI).

In [25], Tobias et al. presented a prototype based on the low-cost foot-mounted Inertial Measurement Unit. The system calculates the position of a pedestrian using the double integration of the accelerometer data along with the angular rate as determined from the gyroscope. The Inertial Measurement Unit produces the drift in the sensor data due to the integration of data over time. Therefore, to enhance the localisation accuracy, their prototype used the Zero Velocity Update (ZUPT) [25], [26] and [27], and Zero Angular Rate Update (ZARU) [25] and [28], which both correct for drift errors in position.

In [24], another approach was presented for a pedestrian navigation based on the shoe-mounted Inertial Navigation System. In this work, Godha et al. determined the position and movement direction of a user using two different Inertial Measurement Units to evaluate the accuracy of a low-cost MEMS-based Inertial Measurement Unit and medium cost-level Inertial Measurement Unit in comparison to moderate-cost GPS-based unit available for a pedestrian's navigation. Their results show similar accuracies using the low-cost MEMS-based Inertial Measurement Unit as compared to the GPS data.

2.1.2 Smartphone-based approaches

Smartphones are equipped with many different types of sensors, such as an accelerometer, magnetometer, gyroscope, and GPS. These sensors are used for various applications, including pedestrian safety, navigation, and pedestrian dead reckoning. Smartphones also provide users/pedestrians with navigation capability without the need to carry a separate GPS unit. Researchers leverage these sensors to determine the movement direction of users and, in this section, we present the state of the art of work. To the best of our knowledge, there is no previous investigation studying the influence of the smartphone's orientation or position while detecting the movement direction of a user. We separate the smartphone-based movement direction approaches in the following three categories. Moreover, in Section 2.1.4, we present comparisons with how previous work relates to the research presented in this dissertation.

- Movement direction is obtained from smartphone sensors when a smartphone is carried in hand with known smartphone orientations (we call this category as the "Smartphone handheld approaches").

- Movement direction is obtained from smartphone sensors when a smartphone is carried in a trouser pocket or attached to a belt (we call this category as the “Smartphone body attached approaches”).
- Movement direction is obtained from the calculation of velocity vector data, which is determined by integrating accelerometer values of a smartphone. These approaches are independent of the position and orientation of a smartphone but inaccurate due to noise in the accelerometer is integrated.

2.1.2.1 Smartphone handheld approaches

In [32], Qian et al. presented an indoor localisation approach using the inertial sensors of a smartphone, which determines the location of a pedestrian using the Pedestrian Dead Reckoning technique. Pedestrian Dead Reckoning detects the steps of a user, determines the length of each step and estimates the user’s heading and position. Principal Component Analysis [30] and [31] were used to avoid error in the inertial sensors while detecting the movement direction of the user. This process first transforms the accelerometer data into global coordinates and then integrates the data to determine the heading of the user. Moreover, the positioning of the user is obtained through the Pedestrian Dead Reckoning technique. To remove the drift in the heading estimation, the already obtained position of the user is matched with a floor map using the particle filter. The drawback of using the inertial sensors (accelerometer and gyroscope) of a smartphone is that the error increases when the data from these sensors are integrated to calculate the movement direction. However, we assume that the inertial sensors require support from the other sensors, such as the compass or the GPS, to minimise error while determining the movement direction. In [32], sensors’ data were collected in a predefined way, such as when being held in hand or placed near the ear in a calling position, and the orientation of the smartphone was not changed during the measurements.

In [37], an indoor pedestrian positioning method was presented based on the accelerometer, magnetometer, and gyroscope sensors of a smartphone. This approach detects a step, estimates a step length and heading, and minimises errors when the user takes a turn. They used the magnetometer and gyroscope individually to estimate the user’s heading and combine these to minimise the error in the heading estimation. The approach relies on the specific position and orientation of the smartphone, i.e., carrying a smartphone in hand with the similar orientation as the user movement direction, are required to perform these calculations.

In [33], Zhu et al. presented the Accurate Pedestrian Tracking system (APT), which is based on a pedestrian dead reckoning algorithm using the accelerometer and gyroscope of a smartphone. A Pedestrian Dead Reckoning algorithm uses the accelerometer data to detect the steps of the user followed by the calculation of the travelling distance assuming that the length of the steps are constant. The gyroscope is used to estimate the heading by integrating the angular velocity. The APT system requires the initial position of the pedestrian and a location map to work correctly. The major drawback of such an approach is that it assumes the user's steps length is constant to calculate the distance. However, this is not a reasonable assumption as every user will have different walking styles, speeds, and step size. Moreover, the user heading is estimated using the gyroscope of a smartphone, and we assume that when gyroscope sensor data is integrated to determine movement direction, the inherent errors are also integrated.

In [45] Khandelwal et al. performed step detection, determination of step direction and step length of the user through the use of the accelerometer and compass sensors in a smartphone. The accelerometer data is used to detect steps and calculate their length. The compass sensor is used to determine the step direction. Compass identifies the direction of North and also gives the azimuth. The compass azimuth is the angle between north and a smartphone orientation when it is held in a default position, i.e., parallel to the ground. To align the compass azimuth and the user's orientation, Khandelwal et al. used the GPS to calculate the user's initial movement direction, and an offset is calculated between the user's direction of movement and compass azimuth. Moreover, the offset is added to the compass azimuth to align the compass azimuth and the movement direction of a user. Khandelwal et al. have used GPS only once (i.e., beginning of measurements) to align the compass azimuth to the user's movement direction. After alignment, they assume that the smartphone orientation remains the same throughout measurements. However, we think that this is not the real-life case. This approach will not work correctly when a smartphone is placed in a loose trouser pocket. For example, when the user starts moving straight ahead, the smartphone starts wobbling inside a loose trouser pocket, which results in different smartphone orientation for each step and simultaneously different compass azimuth. Moreover, we think that with the use of GPS, the determination of movement direction of a user can be delayed because the GPS receiver takes "time to first fix (TTFF)" to get the initial position. In [46] Paonni et al. show that even if the GPS receiver has clear view to the sky, it takes TTFF for cold start is 100s, warm start is 50s, and hot start is 2s.

Furthermore, in an urban environment, the presence of tall buildings often increases the TTFF and diminishes the accuracy of the GPS positioning.

2.1.2.2 Smartphone body attached approaches (fixed)

In [34], Li et al. designed an indoor localisation algorithm based on smartphone sensors, which controls the drift in the gyroscope data and variability in a user walking profiles. The algorithm uses the accelerometer, magnetometer, and gyroscope to provide movement directions of the user carrying a smartphone in their pocket. The authors assume that the phone and the user orientations remain the same or a heading offset is known. The algorithm detects the steps, estimates the steps length and heading, and an inference point is determined. The inference point is defined as “the point in time with the particular combination of movement characteristics that repeats at every step, i.e., mid-stance phase during walk cycle” [34]. To avoid the drift and sensitivity of the smartphone movement inside the pocket, Li et al. only used the sensors’ data at the time of the inference point when the reference foot is on the floor. However, this approach is not practical due to reasons such as the user cannot easily align a smartphone orientation to their own movement’s orientation, especially when carrying a smartphone in a pocket. In this case, the smartphone will typically wobble while inside the pocket once the user starts walking, so it is difficult for a user to align the smartphone’s orientation with the user’s orientation manually. Also, the usage of the sensors’ data only at the inference point is not practical because this can neglect detailed actual data, especially during turns when the reference foot is not on the floor.

In [35], Hong et al. presented an approach for indoor pedestrian tracking using the accelerometer and gyroscope. During the measurements, it is assumed that the smartphone orientation inside a trouser pocket remains the same as the orientation of the user. The approach solves the particular problem of the heading estimation, e.g., a pedestrian carries a smartphone in a trouser pocket, which swings during walking and results in an inaccurate heading. The gyroscope data is integrated to obtain the heading of the user’s movement. However, Hong et al. show that the integration of gyroscope data also increases the error. Therefore, they used the Heuristic Drift Reduction (HDR) [36] technique to minimise for the drift accumulates during the integration of the gyroscope data.

In [38], another approach was presented using two or more pedestrian dead reckoning tracking system carried by a pedestrian. The system is tested in an indoor environment where the user carried phones in both the left and right pockets as well as one phone in hand. The two phones in the pockets were used for dead reckoning, and the phone in hand calculates the ground heading because it placed in a hand similar to a user orientation. However, we see that the orientation of the phones in both pockets was the same as the user's orientation while determining the movement direction of a user.

To the best of our knowledge, there is no approach available, which solves the specific problem of the influence of different smartphone orientations inside a trouser pocket while determining the movement direction.

2.1.2.3 Velocity vector approaches based on smartphone sensors

In [39] and [40] Ayub et al. presented a method that fuses the sensors data (i.e., accelerometer magnetometer and gyroscope) of a smartphone. To determine the orientation of a smartphone in a global coordinate system, they used a "complimentary filter" to fuse the sensors data. However, the smartphone's orientation does not represent the user's movement direction until and unless the smartphone, and the user orientation is the same. Therefore, Ayub et al. used the accelerometer data to determine the movement direction of a user. Firstly, x, y, and z accelerometer data are filtered using the peak detection removal technique. Secondly, x, y, and z accelerometer data of the device's coordinates system are transformed to the global coordinates system and then integrated with respect to time to obtain velocities along the X, Y, and Z-axes. In the global coordinates system, the X, Y, and Z-axes represent the east, north and upward directions, respectively. Moreover, the movement direction of the user carrying a smartphone is determined using the angle of direction equation ($movement\ direction = aTan \times \frac{x}{y}$) , where x is velocity component along the x-axis and y is velocity component along the y-axis).

In a similar approach to [39] and [40], a "WalkCompass" was presented in [41] and [42] by Nirupam et al. where the inertial sensors (accelerometer and gyroscope) estimate the walking direction independent of the position or orientation of the smartphone. Nirupam et al. first transform the accelerometer data into global coordinates, then integrate the accelerometer data to determine the velocity component, and finally determine the movement direction. However, the inertial sensors are noisy and drift over time resulting in the integration of errors along with the accelerometer data. To minimise the noise in the accelerometer and the drift in the gyroscope data, Nirupam et al. [41] and [42] used a low-pass filter whereas Ayub et al. [39] and [40] used a complimentary filter (comprised of a low-pass and high-pass filters) and peak detection removal technique. The disadvantage with the low pass and high pass filters is that the process also removes small accelerations and increases the time delay in detecting the turns.

An approach similar to our method is presented in [43] such that both use the sensors to determine the movement direction independent of the smartphone orientation. The difference is in the way of solving the problem. In [43], Ali et al. estimate the heading misalignment angle between the user movement direction and the smartphone orientation, which is calculated using the accelerometer and compass. The accelerometer data is then integrated over time to determine the velocity component, and the movement direction is calculated from the velocity components. The compass azimuth is the angle between north and the smartphone orientation when it is held in a default position, i.e., parallel to the ground. The misalignment angle is calculated from the movement direction obtained from the accelerometer and compass azimuth. However, as we have seen in previous approaches that integrate the accelerometer and the gyroscope sensor data, an error is also increased, and therefore the sensors' data must be pre-processed to minimise the error.

2.1.3 Sensor Fusion: Smartphone sensors and map-based approaches

In [49] and [50], Alberto et al. proposed an indoor pedestrian navigation system based on inertial sensors of a smartphone and a digital floor map of a building. Their algorithm calculates the user steps, travel distance from the accelerometer data, and movement direction using the compass sensor data of a smartphone. During the experiments, users carried a smartphone in hand and considered that the smartphone and the user orientation was same. However, we observe that it is difficult to align the smartphone orientation to the user's orientation manually. Such approaches are also not practical because of the digital floor maps, which are often not readily available.

In [51], Shin et al. presented an indoor navigation system based on the pedestrian dead reckoning algorithm, which combines the accelerometer, magnetometer, gyroscope, and barometer sensors. To improve the accuracy of the step length estimation in pedestrian dead reckoning, they used a walking recognition algorithm based on an Artificial Neuron Network (ANN) famously known as Artificial Neuron Network (ANN), which is a “mathematical model used for pattern recognition of pedestrians such as stop, walking and running” [51]. Based on the walking status, the ANN calculates the pedestrian’s steps and step length. The primary advantage of this approach is when the walking status is recognised as a stop, and then any other unknown motions detected during this time are not considered, so the error in distance is not increased. Beomju et al. used the magnetometer to determine the movement direction and then corrects the user’s position on a digital map. This approach only works for the determined prerequisite such as the orientation of the smartphone is initially known and remains fixed during the measurements.

In [52], Pai et al. presented a pedestrian dead reckoning system, which is independent of the smartphone position and orientation, where they used the particle filter to determine the user’s motion direction. First, they determined the user’s steps using the accelerometer of a smartphone and estimate the length of the user’s steps. Secondly, only at the starting location, they used the orientation sensor to determine the direction of the user and afterwards use the rotational sensors of a smartphone to determine any changes in the user’s steps direction and compensate for the difference. They assumed the rotational sensors of a smartphone is not very accurate, therefore, computed the average value of orientation for each step. Moreover, Pai et al. use the particle filtering approach to improves the accuracy of motion direction by correcting the orientation data on a floor map. The particle filter uses the term particle instead of samples. The same score is assigned to each particle and weight (i.e., probability) is calculated for a group of particles, particles with lower weight (i.e., below a particular threshold value) are eliminated because they will collide with the obstacles or sideways. The remaining particles are assigned a new position on a map. The process of particle filtering repeats for each step.

In [53], Ruiz et al. determined the location of a user in an indoor environment based on the combination of the Inertial Measurement Unit and RFID. The Inertial Measurement Unit is installed at the foot of a user providing a known and fixed orientation for estimating the user position, step length, and movement direction. Moreover, RFID tags were installed in the building at known locations before measurements. The user position estimation is combined with the RFID’s Received Signal Strength (RSS) to improve the overall position estimation. The Inertial sensors' data integrate over time to determine the movement direction of the user. However, with the integration of sensors data, errors also integrates, which contributes to increased errors in the results. Therefore, this approach uses techniques such as ZUPT (zero velocity update) and ZARU (zero angular rate update) to reduce the drift of the foot-mounted Inertial Measurement Unit. Ruiz et al. use the Inertial Measurement Unit velocity and angular data when the reference foot is on the floor, i.e., stance phase. During stance phase, it is analysed that the user has zero velocity and zero angular update respectively.

In contrast to [53], in [54] Nilsson et al. presented an analysis that stance phase (i.e., stationary) detected while the user is walking contains a non-zero velocity data, which may be caused by a rotation of the foot during the stance phase. Furthermore, Nilsson et al. suggest using ZUPT as a standalone solution for determining the user's direction or position estimation is not sufficient. However, this is in agreement with the findings in [54], techniques such as ZUPT and ZARU will not work if the user is walking very fast or simply running where the stance phase is very short and is difficult to recognise. Moreover, the fixing the position of the Inertial Measurement Unit on foot and the pre-installation of RFID tags is neither practical nor cost-effective.

In [55], Ren et al. presented an approach to determine the movement direction of a user and a wheelchair by integrating the GPS, compass, and accelerometer of a smartphone. To further improve the position accuracy, they matched the movement direction and position on a digital map. However, previous approaches [56] and [57] showed that the GPS receiver of a smartphone is useful for navigation purpose, but it does not yet offer quick updates in determining the movement direction and sufficient accuracy in densely constructed urban environments. In this dissertation, we investigated that the GPS of a smartphone provides a maximum sampling frequency of 1Hz. Therefore, in this case, the GPS with a sampling update of 1Hz may not detect sharp or quick turns (e.g., turns which are taken less than a second) within real-time or when the user or wheelchair only turns at one point. To detect the sharp turns (such as left turn, right turn or U-turn) of a user movement or a wheelchair movement, they also used the compass of a smartphone in conjunction with the GPS. However, the smartphone was placed in a predetermined orientation throughout the experiments, i.e., in line with the user or wheelchair orientation. Moreover, the GPS itself requires significant power, so this type of approach is not practically feasible in a smartphone with limited battery capacity.

In [58], Radu et al. described the "HiMLoc" approach, which combines the accelerometer, compass, and WiFi to determine the user's position, activities and movement direction carrying a smartphone in hand or pocket with similar orientation to the user. They rely on a predetermined orientation of the smartphone and uses WiFi fingerprinting location to improve the positioning accuracy.

2.1.4 Comparison of Approaches

Each approach presented in [19]-[29] and [53] similarly determine the movement direction entirely through inertial navigation sensors, including motion and rotation sensors, installed either on an Inertial Measurement Unit or a self-designed sensor board attached to the user's body. Throughout these demonstrations, the orientation of the device was initially known, or placed in a similar orientation to that of the user as well as remained fixed to the user's body. These inertial sensors data integrate over time to determine the movement direction of the user. However, with the integration of sensors data, errors also integrates, which contributes to increased errors in the results. Therefore, various drift elimination methods were presented in [21], [22], [23], [24], and [25]. Similarly, approaches described in [32] and [37] used smartphone sensors to determine the movement direction of a user assuming the smartphone orientation is in line with the user orientation or the smartphone orientation or an offset is already known. Alternatively, in [33] the movement direction of a user was determined independently of the smartphone orientation by integrating the accelerometer and gyroscope data.

In contrast to [19]-[22], [23]-[33], [37] and [45], our approach determines the movement direction of a user/pedestrian using smartphone sensors, including the accelerometer, gyroscope, and compass, and is independent of the smartphone orientation. We use inertial sensors of the smartphone, such as an accelerometer, to determine its pitch and roll angle with respect to the ground. The peak detection from the accelerometer data also represents when the user/pedestrian takes a step. Based on the pitch, roll and peak detection information, our approach first automatically aligns the smartphone orientation to the user's orientation independent of the physical orientation of the smartphone inside the user's trouser pocket and then determines the movement direction. To avoid error in the inertial sensors, we do not integrate the accelerometer and gyroscope to determine the movement direction, and, instead, use the internal compass. Our proposed approach is also independent of the smartphone's orientation, and the smartphone does not remain in a fixed position (e.g., the smartphone is free to wobble inside the user's trouser pocket).

In [45], Khandelwal et al. used the GPS of a smartphone to determine the movement direction of the user, which supports a maximum sampling frequency of 1Hz. As shown in previous studies [47] and [48], the GPS of a smartphone requires a significant amount of power compared to other smartphone components. Moreover, the GPS of a smartphone provides a maximum sampling frequency of 1Hz, which is useful mainly for navigation purpose but not sufficient to update the movement history in real time for the sharp movement turns that last less than a second. Alternatively, our approach incorporates the accelerometer, and compass to determine the movement direction, which offers a faster direction detection rate compared to the GPS. Furthermore, we demonstrate that the energy consumption of smartphone sensors (including an accelerometer, Magnetometer, gyroscope and compass) is minimal such that they may be readily used to determine the movement direction of a user.

In [34] Li et al., Hong et al. [35] and Jin et al. [38] used a fixed orientation of a smartphone aligned to the user/pedestrian orientation inside the trouser pocket to determine the movement direction. Each of these methods achieved several similar goals, such as minimising the error in the sensors data when the smartphone wobble inside a trouser pocket when the user/pedestrian is walking. To avoid this problem altogether, when the smartphone placed inside the trouser pocket while determining the movement direction, Li et al. [34] used the data only at the inference point, and Hong et al. [35] used the Heuristic drift reduction method. In [38], the sensors data of the smartphone carried in the pocket was compensated with the data of the smartphone carried in hand. Whereas in [49] and [50] users carried the smartphone in hand with a predetermined orientation.

In comparison to these previous approaches, we do not use the limited data available only at the inference point; the Heuristic drift reduction method or the requirement to carry the smartphone in a similar orientation to that of the user. Instead, we use the complete data, so that small turns are not neglected, and the smartphone orientation is also not fixed. We achieve this by first compensating for the deviations caused by the swinging motion of the smartphone and then align the smartphone orientation to the user's orientation. Finally, we determine the movement direction using the compass of a smartphone. To the best of our knowledge, the specific problem of the "influence of smartphone orientations inside a trouser pocket while determining the movement direction” is still not solved. In this dissertation, we addressed this problem in subsequent chapters.

All the approaches in [39], [40], [41], [42], [43], and [51] determined the movement direction of a user independent of the smartphone orientation by integrating data from the accelerometer and gyroscope. However, we observed that when integrating the accelerometer data, error in the results also increases. These approaches used a low pass filter [39] and [40], a complimentary filter [41] and [42] or Artificial Neuron Network [51] to minimise integration error. We analyse that using these filters contributes to the time lag in determining the turns of users. Therefore, to avoid the error, our approach does not integrate the accelerometer and gyroscope, and we instead use the raw data from the accelerometer, gyroscope, and compass to determine the orientation of the smartphone, align it to the user/pedestrian orientation, and determine the movement direction of the user. We achieve this by determining the pitch, roll angle, and peak detection during each step taken by the user. Based on this information, the proposed algorithm aligns the smartphone orientation to the user/pedestrian orientation. In this dissertation, we further investigate the threshold values when determining the peak detection independent of the orientation of a smartphone is inside the pocket.

In [52], a particle filter was used to determine the user movement direction and required a digital floor map for error correction, which is often not readily available. However, this approach only detects the user's steps independent of the smartphone orientation and position but do not detects the users steps direction independent of the smartphone orientation. This is because both orientation and rotation sensors only determine the smartphone orientation with respect to the global coordinates system and the rotational sensor measure the rotation with respect to the device coordinates system. In [55] and [58], both approaches determined the movement direction using smartphone sensors, which were dependent on the smartphone orientation. Moreover, [55] required support from the GPS and digital floor maps, and [58] required support from a WiFi fingerprinting database to correct for positioning errors.

To the best of our knowledge, we have not found any publications, which study the movement direction (using compass) independent of the smartphone orientation. In contrasts to previous comparisons, our presented approach is capable of determining the movement direction entirely independent of the smartphone orientation, the GPS data, floor maps, or WiFi fingerprinting.

2.2 Existing approaches in activity recognition

An activity sometimes referred to as a motion event, is defined as a physical action performed by a user [59] and [60]. Motions sensors, such as an accelerometer, and other information sources, such as video and the calendar are used to discover the activities autonomously [59] and [61]. Context is any information that can be used to characterise the situation of an entity, which is a person, place, or object considered relevant to the interaction between the user and an application and may include the user and the application itself [59] and [62]. For example, a context may contain information about time, location, or activity. The pedestrian safety system utilises this context information of user activities, such as accelerating, decelerating, and slowing down to a complete stop, to avoid collision between cars and pedestrians. The following investigations demonstrate the potential for an accelerometer-based activity recognition system. However, most of the state of the art approaches were only laboratory-based investigations. To the best of our knowledge, there are no activity recognition systems studied in the area of pedestrian safety and accident avoidance systems.

Multiple accelerometers are used for designated the movement recognition of a user. For example, Bao and Intille [63] obtained accuracies up to 95% using five bi-axial accelerometers to detect 20 different movements. In [64], Kern et al. investigated activity recognition using 12 body-worn triaxial accelerometers. Both investigations showed that accelerometer-based activity recognition could achieve more than 90% accuracy. However, to enable the recognition of basic activities, these approaches suggest the use of quite a few sensors placed at fixed strategic positions depending on the targeted activities [64].

In [65], Ravi et al. used meta-level classifiers to recognise eight activities with a dedicated accelerometer coupled to an HP iPAQ. These investigations demonstrated the potential for using a single accelerometer sensor along with an algorithm to provide movement recognition with accuracies up to 99.82%.

Recently, the three-dimensional accelerometer integrated into smartphones was investigated as a potential sensor for movement recognition [66], [67] and [70]. The DiaTrace project [68] used a smartphone with accelerometers for physical activity monitoring. This prototype obtained accuracies of greater than 95% for the activity types of resting, walking, running, cycling, and driving a car. Brezmes et al. also used the acceleration data collected from a Nokia N95 with a K-nearest neighbor algorithm to detect common movements [69]. In [70], Lau et al. demonstrated that a smartphone is suitable for recognition of basic activities, such as standing, sitting, or walking, and to improve the recognition accuracy, they proposed the combination of base- and meta-level classifiers. In this dissertation, we use a similar approach as presented in [70], but we additionally recognise the different activities of users, such as accelerating, decelerating, and slowing down to a complete stop.

3 Pedestrian Movement Recognition

3.1 Introduction

Parts of the contents of this chapter are published in [13].

The car manufacturers and various research groups are currently working to improve pedestrian's safety by developing passive and active pedestrian protection systems. Passive pedestrian protection aims to reduce the impact of a collision while the active approach tries to avoid collisions by using car-mounted sensors (such as radars and cameras) that require a direct line-of-sight to detect pedestrians. In [72], Gandhi et al presented an overview of different approaches, which are based on visible light [73], [74], infrared [75], [76] as well as the time of flight sensors such as radar [77], [78] and laser distance measurement [79]. These approaches consist of different sensors and include several challenges such as a pedestrian's detection, movement recognition and movement direction analysis. However, in these approaches the view from the car point of view is limited. These approaches do not work correctly or fail if there is no line of sight between a car and pedestrians.

Cooperative-based approaches use different wireless technologies (such as Radio frequency communication [2], [3] or GPS [4]) to detect pedestrians without the limitation of line-of-sight communication. However, wireless based radio frequency communication [2], [3] offers a limited range of communication and the GPS sampling frequency of only 1Hz is not timely provide real-time updates in the movement history of pedestrians. In [80], Fackelmeier et al. presented a Tag-based approach, in which the transponder is attached to the vulnerable road user, and a transceiver is mounted on a car. The system works when the transceiver broadcasts the signals, all the transponder receives the signals and responds to the transceiver. The system then detects the location of the pedestrian and tracks the next possible location. The drawback of this approach is that it only detects the location of pedestrians in a specific radius and it does not give the information about the walking direction. An improved system such as in [2] used a transceiver and antenna mounted in the front of a car. This approach calculates the time of arrival of the signal and the direction of the received signal from the transponder and determines the movement direction. This information is used to calculate predictions of the next possible movements and tracks the position of the pedestrians over a short distance. Both tag-based approaches mentioned above works without the line of sight between cars and pedestrians, but they have a limited communication radius. The other drawback of such approaches is the transponder has a battery, and the pedestrian has to take care of the battery.

In [81], another cooperative-based approach is presented which works without the line of sight between a car and pedestrians. This approach exchanges the GPS based positioning of pedestrians and a car to calculate the risk of a collision. These days the GPS is widely used for navigation. However, the accuracy of the GPS of a smartphone is still not sufficient. As investigated in this chapter, that even if the GPS position of a smartphone were accurate enough, the actual GPS sampling frequency of the only 1Hz is not timely provide the real-time updates in the movement history of pedestrians.

In this chapter, we focus on the context information, i.e., pedestrian's movements using accelerometer of a smartphone. Furthermore, in chapter 3 we discuss the exchange of additional context information, such as movement recognition and direction detection of pedestrians using the accelerometer and compass of a smartphone. In a typical accident scenario between cars and pedestrians, the speed of the pedestrian is essential information, which is used to determine the risk of a collision. Quick detection of speed changes of a pedestrian helps to avoid the collision. Measurements conducted within the scope of this chapter are presented in the latter sections. The context information presented in this chapter is the speed determination using the GPS based positioning and with the acceleration sensor.

The chapter is organised as follows: In section 3.2.1, we presented the experiments and data collection process followed by the feature extraction process in section 3.2.2. In section 3.2.3, we mentioned which classification algorithms are used for movement recognition. In section 3.2.4, we discussed the results about the movement recognition using the accelerometer and different classification algorithms. In section 3.2.5, we showed the movement recognition using the GPS of a smartphone. In Section 3.3, we discussed the influence of the speed determination of pedestrians using an accelerometer and the GPS in an accident scenario. Finally, in Section 3.4 we give the conclusion.

3.2 Movement Recognition using Accelerometer respectively GPS

The experiments were designed to investigate the pedestrian’s movement recognition using an accelerometer and the GPS of a smartphone. Furthermore, in this section, we present the accuracy of the pedestrian’s movement recognition using the accelerometer and also with GPS based positioning technique.

3.2.1 Data collection

During the experiments, we used the Nokia N95 8GB smartphone. The Nokia N95 8GB smartphone built in with LIS 302 DL 3D accelerometer chipset, manufactured by ST Microelectronics and a Navilink GPS chipset manufactured by Texas Instruments. We designed the software in m-shell environment for Symbian S60 operating system, which records the acceleration of a user/pedestrian at a maximum sampling rate of 32Hz (i.e., 32 samples per second). We categorised measurements in two different cases; we named it as Case 1 and Case 2.

Case 1: A test user/pedestrian carried a smartphone in his trouser pocket and completed a series of movements on a straight empty road and sidewalk (footpaths) in an urban area. These movements are categorised as running, walking, and stationary (standing). During this experiment, the test user/pedestrian changed between the movements in different combinations, for example, from stationary (standing) to walk and from walking to running and vice versa. In this chapter, we recognise the movements such as slowing down to a stop, accelerating, and decelerating.

Case 2: Case 2 is similar to case 1 except the test user/pedestrian carried the smartphone in hand.

3.2.2 Feature extraction of accelerometer based movement recognition of a pedestrian

In general, feature extraction is a technique, which reduces the number of resources in describing an extensive data set without losing any essential or related information. Feature extraction provides the reduction of dimensionality of the data and discovers the meaningful patterns. The “reduction of dimensionality” is a process to reduce the set of raw variable to more manageable groups for processing without compromising the integrity of the original data.

The raw data such as the time series data cannot be directly used for the processing of classification and clustering and therefore require prior feature extraction [83]. In our case, we used the accelerometer data to recognise the pedestrian movements, which is a time series data and requires the prior feature extraction before classification of the movement recognition. In section 3.2.2.1, we described the sliding window technique with and without overlapping, which is the first step for the feature calculation. In section 3.2.2.2, we presented the feature extraction methods and selected the features for movement recognition within the scope of this dissertation.

3.2.2.1 Sliding window data preparation

In [70], Lau et al. described a method to pre-process the accelerometer data to recognise pedestrian's activities. Similarly, we pre-process the accelerometer data collected during the measurements for case 1 and case 2. The sliding window is the first step for the feature extraction process of the collected data. The sliding window algorithm is an algorithm, which separates the data in a fixed interval of segments [82] and [83]. "The sliding window algorithm is useful if one wants to compare the segments to discover the recurring patterns *[82]*and *[83]*". "In a sliding window, a data segment is grown until it exceeds the given interval to form a so-called window. If no overlapping is desired the following window starts from where the previous window stops at *[83]*". The start position of the next window is dependent on the set of overlapping windows. Thus, in determining an overlap of 0%, the window (W_{m-1}) starts right after the end of the window (W_{m-2}). In Figure 3-1, we exemplary shows the sliding window of a time series data with 0% of overlapping, where each window contains four samples. In 75% overlapping of sliding window, the following window (W_{m-1}) reuses the 75% of data from the previous window (W_{m-2}) as shown in Figure 3-2. Similarly, an overlap of 25% and 50%, the following window (W_{m-1}) reuses the 25% and 50% of data from the previous window (W_{m-2}) respectively.

Based on the results shown in [70] and [83], we selected an overlapping of 75% during the pre-processing. We used the sliding window algorithm for the feature extraction process. Window lengths selected for the feature extraction were 1s, to achieve recognition under a second for the designated application domain.

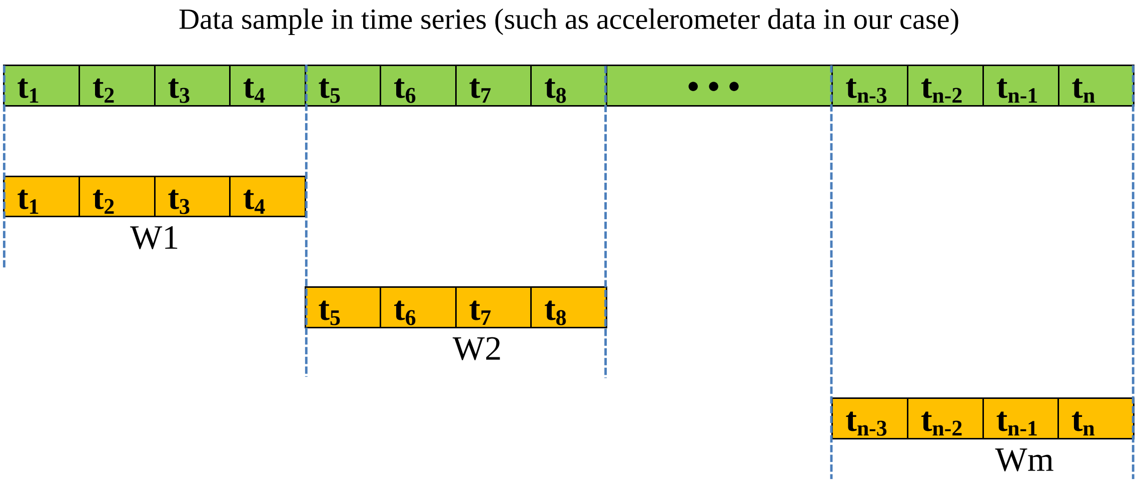

Data sample in time series (such as accelerometer data in our case)
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 • • • t_{n-3} t_{n-2} t_{n-1} t_n
t_1 t_2 t_3 t_4
W1
t_5 t_6 t_7 t_8
W2
t_{n-3} t_{n-2} t_{n-1} t_n
Wm

Figure 3-1 Sliding window algorithm with four segments each window and 0% overlapping

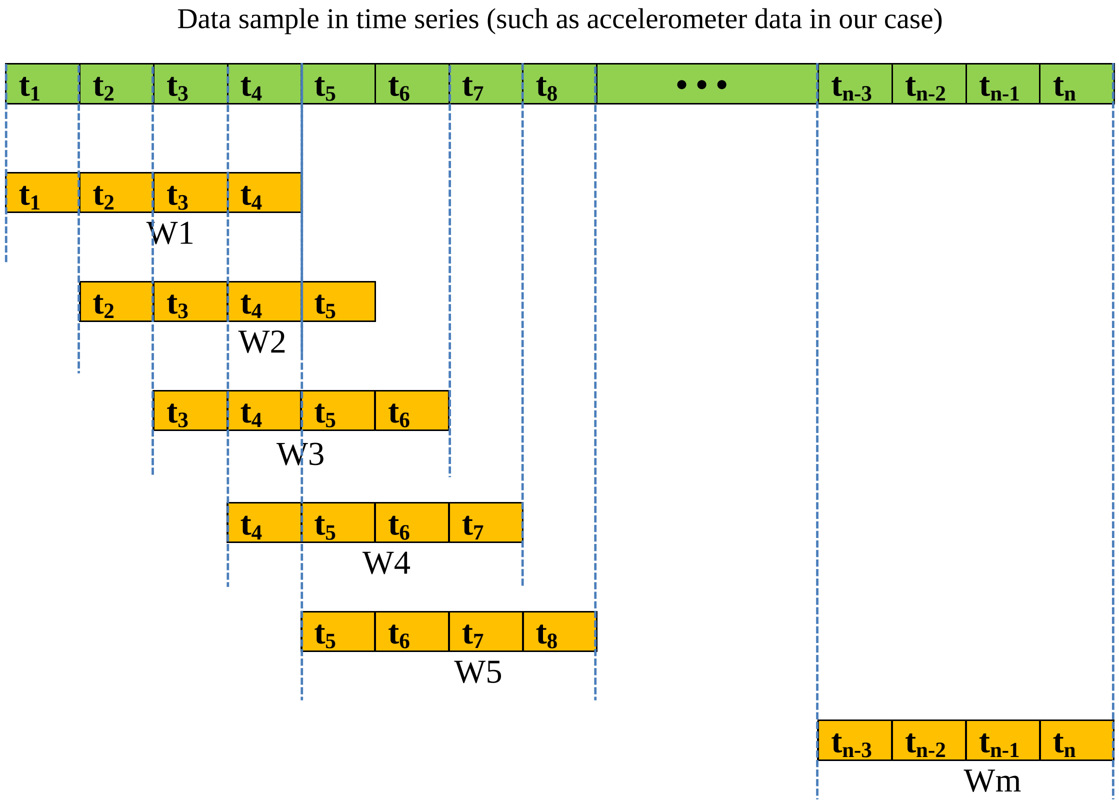

Data sample in time series (such as accelerometer data in our case)
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 • • • t_{n-3} t_{n-2} t_{n-1} t_n
t_1 t_2 t_3 t_4
W1
t_2 t_3 t_4 t_5
W2
t_3 t_4 t_5 t_6
W3
t_4 t_5 t_6 t_7
W4
t_5 t_6 t_7 t_8
W5
t_{n-3} t_{n-2} t_{n-1} t_n
Wm

Figure 3-2 Sliding window algorithm with four segments each window and 75% overlapping

3.2.2.2 Calculation/transformation of feature extraction:

For classification based activity recognition, it is essential to extract the features from the actual data. First actual data is prepared using the sliding window approach, where the data is segmented in windows. Feature extraction process transforms the sliding window data into newer values, for example, the computation of mean values for every window. Based on the mean value of each window and using classification algorithms one can determine the different movements such as the higher mean value of the window represents the fast acceleration and lower mean value of the window represents the slow acceleration and mean value of the window near to zero represents no acceleration. Similarly, it is also possible to determine the transition from fast acceleration to slow acceleration based on the higher and lower mean values of the successive windows.

"Feature extraction is a technique that enables the dimensionality of the data and discovery of useful patterns. A feature is defined as a new attribute generated from the original data. Feature extraction is useful especially when the original data is not directly usable for potential pre-processing using algorithms such as classification or clustering *[83]*". In this dissertation, we used five different features in our investigations as suggested in [70]. These features are the most frequently selected features due to their simplicity and low computational cost. Features used for the classification were:

- Mean and standard deviation
- Variance of the time domain acceleration data and Information entropy
- Energy of the Fast Fourier Transform (FFT) in the frequency domain

3.2.3 Classification algorithms used for movement recognition

We used the 10-fold cross-validation method as suggested in [84], to compare the pedestrian movement recognition accuracies obtained by using different classification algorithms. *"In 10 fold cross-validation method, the training data is randomly divided into 10 subsamples. For every round of evaluation, one subsample is used as the test data while the other nine subsamples are used as training data. The cross-validation process is repeated 10 times, and the produced evaluation results are averaged [83]"*. The weka data mining tool [86] was used to evaluate the results and analyse the recognition accuracies using different classification algorithms. Base-level classification algorithms to be used for evaluation are DT (see [89], [90] and [91]), kNN [83] and jRip

[92]. Based on the state of the art investigations these base-level classification algorithms achieved higher accuracies, i.e., greater than 90% [65], [69] and [70]. In our evaluations we used two meta-level classification algorithms (i.e., Boosting [87] and Bagging [88]) in combination with three base-level classification algorithms (i.e., DT [89], [90] and [91], kNN [83], jRip [92]). As suggested in [65] and [70], combined use of meta- and base- level classification algorithms can improve the classification accuracies. In this section, we describe the working principles of the classification algorithms used within the scope of this dissertation. Moreover, we analyse and discuss the performance accuracies (i.e., movement recognition) obtained using individual or combined classification algorithms.

3.2.3.1 Base-level classification Algorithms

Decision Tree: A decision Tree (DT) is a simple and commonly used classification algorithm [90]. The DT model has a hierarchical or tree-like structure and the output is known as the classification. The structure of the DT starts with a root node followed by an intermediate node and the final node. Final node is also known as a leaf node. The root node and the subsequent node (i.e., an intermediate node) helps to decide the leaf nodes. Finally, the leaf nodes represent the classified objects. A typical example of such a tree structure is shown in Figure 3-3, where the simple decision tree is constructed to classify the user's activity into the walk, run and stationary. The root node “Activity frame” separates the activity into the intermediated node, i.e., “Mobile” and a leaf node (i.e., Stationary). If there is no acceleration, the decision tree classified the activity as stationary. If there is an acceleration in the sensor data, an intermediate node with feature “Mobile” is created to differentiate between activities walk and run. Moreover, when the acceleration recorded is average, the decision tree decides the leaf node and classifies the activity as walking. Similarly, when the acceleration recorded is high, decision tree decides the leaf node and classifies the activity as run.

In this dissertation, we used the Weka implementation of Quinlan’s C4.5 revision 8 in Java, i.e., a decision tree (J48) building algorithm. It is a Java re-implementation of a Decision tree algorithm by Ross Quinlan (C4.5) [91]. In Figure 3-4, we show the pseudo-code of building a decision tree using J48. Firstly, the J48 decision tree algorithm examines the possible base cases from the training data set. A base case is basically define as the stop of recursion (no more calling any other recursion). Secondly, when the base cases are identified, the algorithm starts searching for features that best divide the training data. Thirdly, the normalised information gain is used to select the best attributes. Finally, the algorithm recurs on each best attributes identified. The process is repeated until the stopping base cases are met.

Rule-based Classifier (JRip): The rule-based classification uses specific rules to classify the given data set such as “**IF conditions THEN action**”. The “IF” portion of the rule is known as the “rule antecedent/pre-condition” or also known as Left Hand Side (LHS) of the rule. The LHS part of the rule consists of the logical conditions such as “IF” and “AND” of one or more attributes. The “THEN” part of the rule is known as the “rule consequent” or also known as the Right Hand Side (RHS) of the rule. The RHS part of the rule consists of actions such as the prediction. A rule-based classifier is a descriptive model, which is easy to interpret as it consists of only rules, which can be easily understood by human readers. From a performance point of view, it is easy to compare rule-based classifier with the Decision tree. The rule-based classifier uses a separate-and-conquer algorithm to construct rules. A rule that explains a portion of the training instances separates them from the remaining instances. The algorithm recursively “conquers” these remaining instances by constructing more rules until no instance are left. Exemplary we show few rules to recognise the activities based on “IF conditions THEN action” rule.

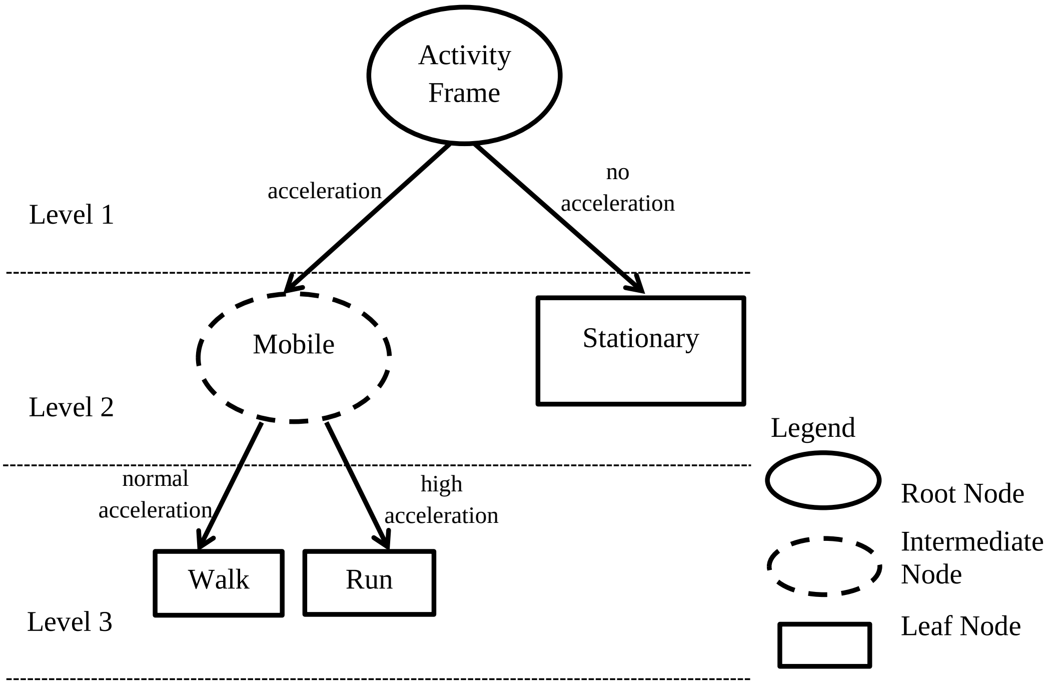


Figure 3-3 An example of a Decision tree

Rule 1:
IF Activity = “acceleration > 0” AND Mobile = normal acceleration
THEN Activity = walk
Rule 2:
IF Activity = “acceleration > 0” AND Mobile = high acceleration
THEN Activity = run
Rule 3:
IF Activity = “acceleration = 0”
THEN Activity = Stationary

In this dissertation, we used the Java implementation of rule-based “Cohen’s RIPPER: Repeated Incremental Pruning to Produce error reduction [92]” provided by the Weka tool. In Figure 3-5, we show the pseudo-code of JRip, which continuously repeat the process of growing and pruning to generate rules from a data set. Cohen’s RIPPER approach uses Description length (DL) [93] which is based on the Minimum Description Length (MDL) [94]. The description length is defined as the sum of the description length of all the rules in the data and the description of instances that are not used by the rule set. The Cohen’s RIPPER algorithm does not add the new rules in the rule set if the Description length of the rule is 64 bits greater than the smallest description length of the rule in the rule set. In [95], Fürnkranz et al. showed that the rule-based learning approaches only evaluate the quality of the set that is covered by the rule. Whereas, the Decision tree learning approaches evaluate the average quality of disjoint sets.

k-Nearest Neighbor (kNN): The K- nearest neighbor (kNN) classifier is known as an instance based classification algorithm. In the k-nearest neighbor algorithm, there is no prior generalisation mechanism to create a model and therefore known as a lazy learner algorithm. The kNN store the training data and only process it when a new instance to be classified. The kNN does not use the whole training data for the classification, but only a selection of instances are used to create a local model for the classification. In Figure 3-6, we showed the pseudo-code of the kNN algorithm. The kNN algorithm calculates the distance between the test data and available training data in order to obtain a list of nearest neighbors and manage a distance metric. For the kNN algorithm, it is essential to know the number of neighbors and the distance metric in order to classify a data set. The kNN algorithm is considered as very fast because it stores the training data in memory without performing any operation.

J48 - implementation of the Decision tree based on C4.5 by Quinlan [91]

```
1. Check for the base cases.
2. For each attribute "a", find the feature that best divides the training data such as
   information gain from splitting on a.
3. Let $a_{best}$ be the attribute with the highest normalised information gain. Create a
   decision node that splits on $a_{best}$
4. Recurs on the sub-lists obtained by splitting on $a_{best}$ and add those nodes as
   children of the node. Stop when the stop condition is met.
```

Figure 3-4 The pseudo-code of the J48 Algorithm in the Weka Tool

JRip – implementation of the Rule based on RIPPER by Cohen [92]

```
1. Initialize an empty rule set "RS = {}"
2. For each class, starting from the less prevalent one to the more frequent one. DO:
   2.1. Building stage:
        2.1.1. Repeat step 2.1.2 to 2.1.4 until the Description Length (DL) of the
               Ruleset {R} and the examples are 64 bits greater than the smallest DL
               met so far.
        2.1.2. Grow one rule by greedily adding antecedents (or conditions) to the rule
               until the rule is perfect (i.e., 100% accurate).
        2.1.3. Incrementally prune each rule and allow the pruning of any final
               sequence of the antecedents (or conditions).
        2.1.4. Add the rule to the rule set {R}
   2.2. Optimisation stage:
        2.2.1. After generating the initial ruleset {$R_i$} in {R}, construct two alternative
               rules. One rule variant is generated from an empty rule while the other
               by greedily adding conditions to the original rule { $R_i$ }. The rule variant
               with the smaller Description Length is selected as the final rule for each
               original rule { $R_i$ }.
        2.2.2. If there are still residual positives, more rules are built using Building
               stage (step 2.1) with these residual positives.
        2.2.3. Remove rules from the rule set {R} that would increase the DL of the
             rule set.
        2.2.4. Add the resultant rule set to {R}.
  END DO
```

Figure 3-5 The pseudo-code of the JRip Algorithm in the Weka Tool

```
k – Nearest Neighbor (kNN)
1. Select the number (k) of neighbors to be searched and determine the distance metric.
2. For each testing instance:
   2.1. Find the k nearest instances in the training set according to the selected distance
        metric.
   2.2. Return the most frequent class label in the found k nearest instances as the
        classified class.
```

Figure 3-6 The pseudo-code of the K-nearest neighbor (kNN) adapted from [83]

Discussion: The Decision tree algorithm is suitable for data instances that can be described by the attribute values. Decision tree uses the divide and conquers mechanism to determine the relationship between attributes and respective class. Moreover, the tree build by the decision tree algorithm can also be viewed as rules. A decision tree is a commonly used classification algorithm and produces simple models that are easy to analyse and understandable for the domain experts. A decision tree is an unstable classifier especially when there is a change in the training data set. The slight change in the training data causes a significant change in the model as well (for example change in the tree structure). Decision tree tends to perform better with the categorical features, but also works with continuous features. Similarly, Decision tree and rule-based classifier produce models that are easy to understand, and we see this as an advantage over other approaches, which produce complex and incomprehensible models. In contrast to a Decision tree, rule-based classifier consists of only rules that are more compact than the trees built from the same training set. One major drawback of the rule-based classifier over the decision tree is the feature to add new rules in an existing set of rules, whereas, in the decision tree is not possible and reconstruction of tree model is required.

kNN is an instance-based classifier and commonly investigated in applications such as activity recognition based on accelerometer data. Moreover, we have analysed that kNN is an attractive classifier and individually achieved better accuracies in comparison to Decision tree and rule-based classifier. The major drawback with the kNN classifier is the classification time, which is higher than the decision tree and rule-based classifier. Furthermore, we combined the above mentioned base-level classifiers with the Meta level classifier to see any improvement in the recognition accuracies. In the next sections, we describe two meta-level classification algorithms and discuss the evaluations and comparisons.

3.2.3.2 Meta-level classification Algorithms

In this dissertation, we have investigated the use of meta-level classifiers in combination with the base-level classifiers. The Meta-level classifiers are famous for their ensemble feature. Meta-level classifiers use one or more base-level classifiers in combination to determine the designated classification. The goal of using the metal-level classifiers is to enhance the classification accuracy by aggregating the prediction of the multiple classifiers [89]. However, this is not always the case, but we will investigate and analyse the accuracies obtained using meta-level classifier combined with a base-level classifier for the activity recognition used with the scope of this thesis. Theoretically speaking, the accuracies obtained using meta-level classifications are expected to be better in comparison to the use of only base-level classifiers. For example, this can be achieved by giving higher weight to a classifier which performs better in an ensemble and may improve the accuracy. These classifiers were used in the investigations of [65] and gave high accuracies. In this section, we used the following two meta level classification algorithm (Bagging and Boosting).

Bagging: The word **“Bagging”** is derived from the **B**ootstrap **agg**regat**ing**. In 1994, Leo Breiman from the University of California proposes the bagging algorithm [9]. Bagging is a machine learning meta-level classification, which ensemble the training set. For example, the bootstrap process generates a smaller number of sets in a training set. Bagging is popular in enhancing the classification accuracy especially for the unstable base-level classifiers such as the decision trees [88] and [90]. In [96], Raudys et al. have shown that the if the training set is small then the base-level classifier such as the decision tree algorithm used for that small training set is analysed as the unstable due to significant variance in the probability of miscalculation. Moreover, the minor changes in the training set cause a significant change in the tree structure of the decision tree which makes the model unstable. To overcome the instability of the decision tree algorithm, bagging (**b**ootstrap **agg**regat**ing**) comes into practice. It also gives better accuracy in comparison to the use of individual base level classifier, i.e., decision tree.

The two phases (training and classification phase) of the bagging algorithm is shown in Figure 3-7. In the training phase, for a given training set denoted by “TS” generates “n” new bootstraps of “TS”. Based on the new generated bootstrap samples TS_1 to TS_n, the classifier C_1 to C_n are built. During the classification phase, the already build classifier C_1 to C_n are used to classify the input x. The output of all these classifiers is then voted for classification. The meta-level bagging classifier has many advantages such as:

1. It may minimise the classification error which is caused by the misleading data in the training set and also avoids overfitting [97].
2. With the use of bagging, it is possible to train and run the classifier ensemble on different processors and devices.
3. If the training set is relatively small, the use of bagging may be essential for the classification of the activities using accelerometer data.

```
Input:

        Training Set "TS" consists of instances "n." <$(x_1,y_1),\ldots(x_n,y_n)$>
        "T" is the number of iterations
Training phase/model generation:

   1.   For i = 1 to T
            Generate a bootstrap sample "$TS_t$" from training set "TS."
            Build a classifier $C_t$ using "$TS_t$" as a training set
   2. Return C for the designated classification
Classification phase/prediction for a give test instance x:

   1.   Given the test data x, run the classifiers $C_1,\ldots C_T$
   2.   For numerical outputs, an average is computed from the output of the
        classifiers in C.
```

Figure 3-7 The pseudo-code of the Bagging algorithm adapted from [83]

Boosting: Boosting is a meta-level classifier, which is used to enhance the performance of any base- level classification algorithm. Theoretically, boosting is useful to reduce the error in the classification of any weak learning algorithm and converts a weak learning algorithm to robust learner algorithm. The boosting algorithm works by repeatedly running on a given base-level classifier and attempts to enhance the classification accuracies. Initially, the base-level classifiers assigned equal weight to all objects in the training set, to achieve better accuracy, boosting changes the weights in every repetition. Afterwards, a classifier is built using this training set and the output of the first classifier is used to reassign the weights to these objects. Boosting assigns the higher weight to the objects, which are wrongly classified. Finally, boosting combines the result of the weak learner and generates a strong learner, which improves the prediction capabilities of the model. In general, boosting focus on the samples, which are misclassified or have higher errors due to the weak rules.

AdaBoost (**Ada**ptive **Boost**ing) is a popular boosting algorithm, which is introduced by Freund and Schapire [87] and [98]. In this dissertation, we used the AdaBoost.M1 algorithm, which adaptively adjusts the error of the hypothesis returned by the base level classifier. In Figure 3-8, the pseudo-code of the AdaBoost.M1 algorithm is presented. *"Given a training set with x_i as the attributes that lead to the corresponding outcomes y_i, the first weight vector w^1 is initialised. A distribution p is computed by normalising these weights. This distribution is then fed to the learner L to obtain the hypothesis h. The requirement in AdaBoost.M1 is that each hypothesis needs to have a prediction error less than 1/2 with the respective distribution. Based on this hypothesis, the weight vector of w^{t+1} for the subsequent iteration is computed. The same process is repeated until the final iteration T. The final hypothesis h_f is the label y that has the maximum sum of the weights of the hypotheses predicting that label [83]."*

There are some limitations of the AdaBoost such as:

1. It may be slow during the training phase due to the resampling or reassigning of weights to the objects.
2. It is also sensitive to noisy sensor data.

Discussion: To the best of our knowledge, base-level classification algorithms are commonly used in past investigations [63], [65], [99]-[103] to determine the accuracy of an activity recognition using accelerometer data. Moreover, other researches such as [65], [104], [105] have investigated the meta level classifier to recognise the activities using accelerometer data. These approaches have also shown the recognition accuracy comparison of meta-level classification algorithms with the base-level classification algorithms and present the analysis that meta-level classification algorithms perform better than the base-level classification algorithms. Theoretically, both meta-level classification algorithm (Bagging and Boosting) improve the classification accuracy of a single base-level classification algorithm. However, the ensemble approach of the meta-level classification algorithm may increase classification time as well as number of iterations during classification.

AdaBoost.M1

```
Input:

        Training Set "TS" consists of instances "n."
        <(x1,y1),…(xn,yn)>
        "D" is the distribution over the "n" instances and "T" is the number of
    iterations

Training phase/model generation:

    1. Initialise the weight vector: $w_i^1 = D(i)$ for $i=1,…,n$
    2. For t = 1 to T
            (a) Compute the distribution, $p^t = \frac{w^t}{\sum_{i=1}^{n} w_1^t}$
            (b) Call the classification $C_t$ with the distribution $p^t$: Obtain a hypothesis
                $h_t = X \to Y$. Add $C_t$ to the ensemble C.
            (c) Calculate the error of the $h_t : \varepsilon_t = \sum_{i=1}^{n} p_i^t \left[ h_i(x_i) \neq y_i \right]$,
            (d) if $\varepsilon > 1/2$, then set $T = t-1$ and abort loop.
            (e) Set classification error weight $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
            (f) Set the new weights vector to be: $w_i^{t+1} = w_i^t \beta_t^{1-[h_i(x_i) \neq y_i]}$
    3. Return C and $\beta_1,...,\beta_t$.

Classification phase/prediction for a give test instance x:

    1. Given the test data x, run the classifiers C1,…CT
    2. The final hypothesis is computed:
       $h_f(x) = \arg\max_{y\varepsilon Y} \sum_{t=1}^{T} \left( \log \frac{1}{\beta_t} \right) \left[ h(x_i) = y \right]$
       The output class is the label with the maximum sum of the weights of the
       hypotheses predicting that label.
```

Figure 3-8 The pseudo-code of the AdaBoost.M1 algorithm [83] originally adapted from [87] and [98]

In this dissertation, we used the base and meta-level classification to recognise the activities of users using the accelerometer of a smartphone. Therefore, we also want to compare the accuracy performance, whether the accuracy of meta-level classification algorithms is similar or better than the respective base-level classification algorithms. Furthermore, these meta-level classifiers are only helpful when they at least provide equivalent or better recognition accuracies compared to a single base-level classification algorithms. The drawbacks of the meta-level classification are the higher classification time and more resources such as memory and processing power.

Many other metrics are used to evaluate the movement recognition. These metrics are commonly known as accuracy, precision, recall and F-measure. Accuracy is defined as the correctness of model or system and can be calculated as the correctly classified instances divided by total instances. Precision is immediately telling us the quality of being exact/ correct, or we can say that how often the same measurements are repeated. Precision is useful when the cost of false positive (an event is positive classified, and this classification is correct) are high. The recall is helpful when the number of the false negatives (an event is false classified, and this classification is incorrect) are high. The F1 is an overall measure that combines "precision" and "recall" to give the system’s accuracy. F1 measure score gives you a mixture of "precision" and "recall" with the low false positives and false negatives. Different researchers used specific or combination of different metrics depending on the requirements of the model and the applications [106], [107], [108]. To the best of our knowledge, most of the pedestrian movement recognition studies use the only metric, i.e., accuracy [109-117]. However, our goal is to determine the accuracy of recognised movements, precision and recall are the additional features, which gives the quality of measurement with respect to other measurements and we found these features out of the scope of this dissertation.

3.2.4 Results discussion of pedestrian movement recognition using accelerometer of a smartphone

In this section, we present the evaluation results and discusses whether the accelerometer in a smartphone can provide an activity recognition and which classification algorithm is capable of providing high and usable recognition accuracies. In Figure 3-9 and Figure 3-10, we showed the 10 fold cross validation evaluation of pedestrian movements based on accelerometer data for Case 1 (i.e., the smartphone in a trouser pocket) and Case 2 (i.e., the smartphone carried in hand) respectively. We analysed from the results of both cases (see Figure 3-9 and Figure 3-10) that the tested classification algorithms (i.e., individual or combined) with the window length of 1s and overlapping of 75% provides the movement recognition accuracy between 93.39% and

96.98% for Case 1 and 94.5% and 96.58% for Case 2. As shown in Figure 3-9 (Case 1) the Meta-level classifier (i.e., Boosting) combined with a base-level classifier (i.e., DT) provides the highest movement recognition accuracy of 96.98%. On the other hand, the base-level classification algorithm (i.e., kNN) combined with the Meta-level classification algorithm (i.e., Bagging) has achieved the second highest movement recognition accuracies, i.e., 96.7% in comparison to another individual or combine base- and meta-level classifiers. The similar results are obtained for Case 2 (see Figure 3-10), kNN individual or combined with Meta-level classification algorithm (i.e., Bagging) has achieved higher movement recognition accuracy. However, the kNN classification algorithm is an instance-based classifier. In these classification algorithms, the movement recognition time may be longer than the other base-level classification algorithm such as DT and JRip.

During the pre-processing, we set the length of the window to 1s and used the window overlapping of 75%. Window overlapping of 75% provides each movement recognised within a time of 250ms. Therefore, it was possible to obtain four movements recognition within a second for the designated accident avoidance application. Base-level classification algorithms such as DT and JRip are known for their speed. This section presents the average runtime for only one classification. The classification is done on a Pentium III 650 MHz notebook. The results of the average runtime are shown in Figure 3-11. Where DT alone completes a classification in a time frame of 0.81ms with an accuracy of ±0.063.

In contrast, JRip alone completes classification accuracy in a same time frame of 0.81ms with an accuracy of ±0.068 ms. Moreover, when we combined the DT with the meta-level classifier (i.e., Bagging and Boosting), the average run time of classification is increased to 1.19ms with an accuracy of ±0.087. Similarly, when JRip is combined with the meta-level classifier (i.e., Bagging and Boosting), the average runtime of classification, in this case, was 1.17ms and 1.30ms with an accuracy of ±0.073 and ±0.088. As analysed from the results (see Figure 3-11) the kNN-based classifier (alone or combine with meta-level classifier) is slower regarding classification runtime. For example, kNN alone completes classification in a time of 18.74ms with an accuracy of ±1.118.

Moreover, the average run time of classification is increased from 40.73ms to 42.51ms, when we combined JRip with the meta-level classifier (i.e., Bagging and Boosting). From the results, we concluded that the feature extraction process is completed in less than 200ms. Furthermore, all investigated classification algorithms within the scope of this research completed the recognition accuracy within a time frame of 250ms. Therefore, it is possible to obtain recognition under 250ms by adjusting the values of the sampling rate, window length and overlapping percentage.

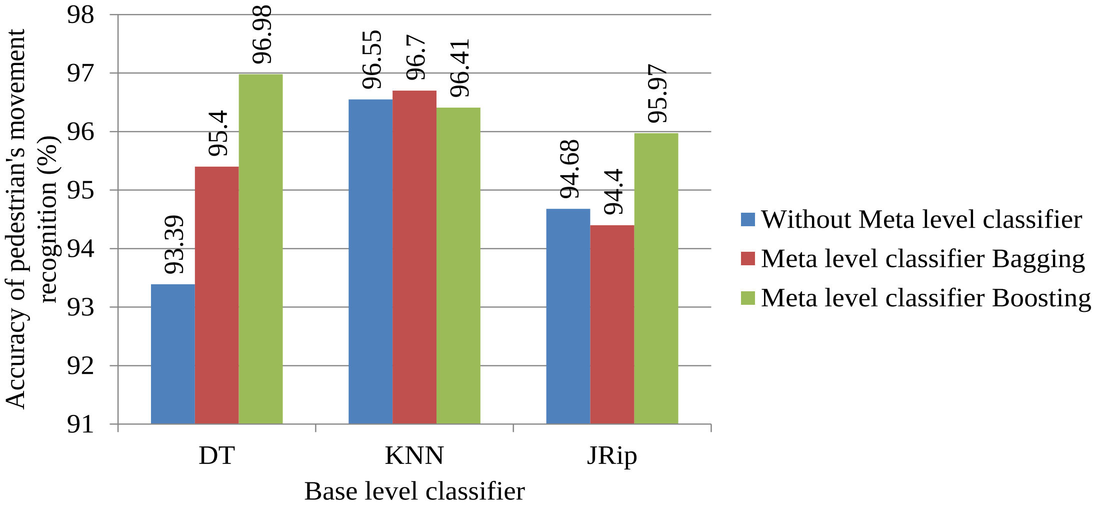


Figure 3-9 Case 1: Accuracy of pedestrian's movement recognition based on the acceleration data [13]

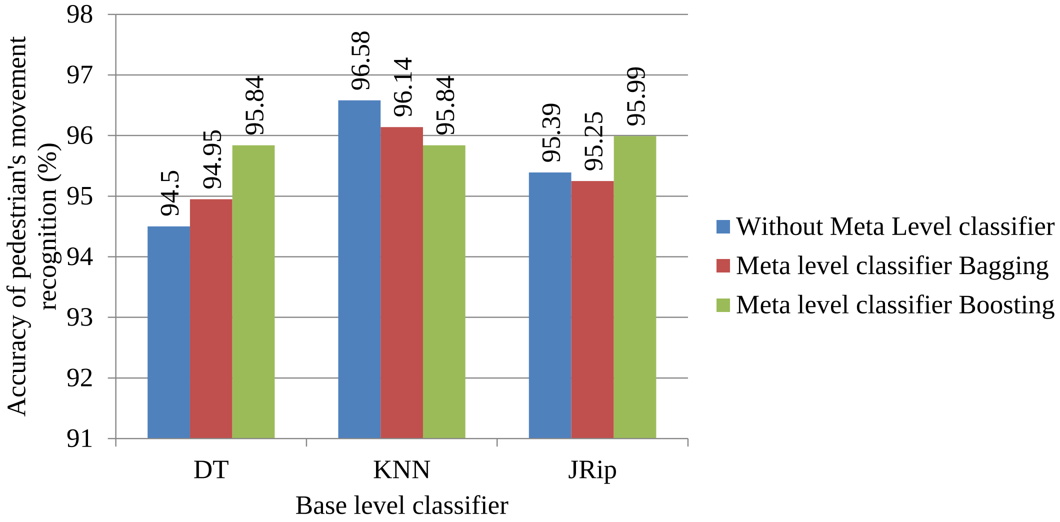


Figure 3-10 Case 2: Accuracy of pedestrian's movement recognition based on the acceleration data [13]

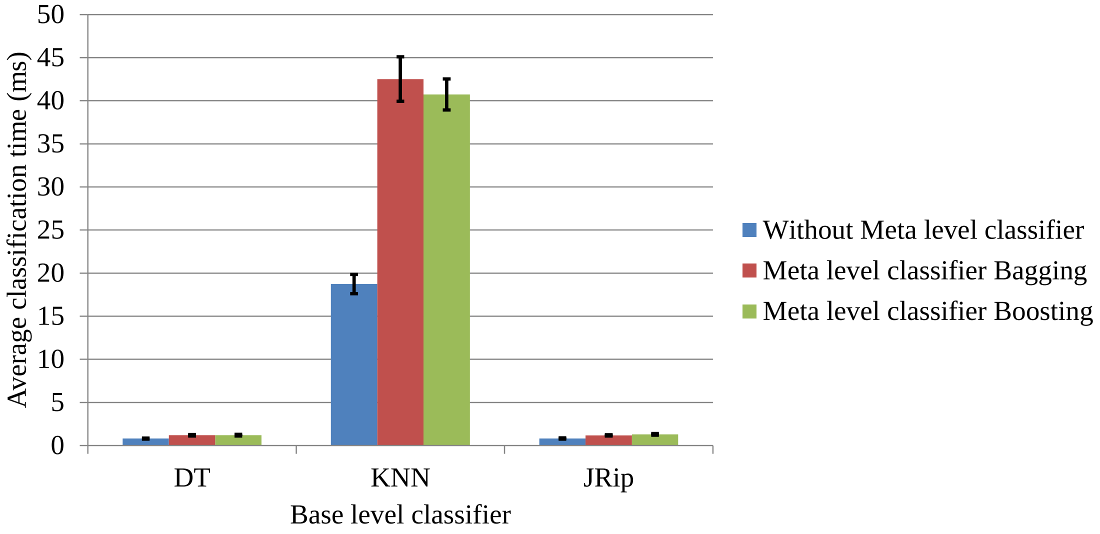


Figure 3-11 Average time needed for a single classification of a pedestrian movement with a confidence level of 95% [13]

3.2.5 Systems based on GPS only

During the measurements (Case 1 and Case 2), the PynetMony software was used to record the GPS data in a smartphone memory parallel to the logging of acceleration data. The GPS based positioning system in a smartphone provides a maximum update frequency of 1Hz (i.e., one sample per seconds). In a positioning based system such as GPS, an average speed of a pedestrian is determined by knowing the distance between two successive points on the ground and time to travel from location one to two as shown in equation (1). Where "S" is the speed in the unit km/h, "d" is the distance in the unit km and "t" is the time in the unit hour.

$$S = d * t \qquad (1)$$

In this section, we present the physical analysis about how much time is required to determine the different movements of a pedestrian based on their speed. Exemplary we show the pedestrian movement, i.e., slowing down to stop in two different scenarios.

Scenario 1: The pedestrian is slowing down to stop exactly between two successive GPS data samples, i.e., "t" and "t+1" as shown in Figure 3-12. Where the X-axis shows the sample update time in seconds, and the Y-axis shows the speed in m/s. In this scenario, the pedestrian is walking normally at an average speed of 1.2m/sec and afterwards slowing down to stop but not suddenly. As shown in Figure 3-12, the pedestrian is walking at an average speed of 1.2m/sec and after the GPS sample update at the time “t” the pedestrian is slowing down the walking speed and stop before the GPS sample update at the time “t+1”. As soon as the data sample “t+1” arrives, GPS determines the average speed based on the travelled distance between time “t” and “t+1” which is not zero but 0.6m/sec and from the GPS data we recognise this as the pedestrian is slowing down but not stop. At the GPS sample update time “t+2”, GPS determines the average speed based on the travelled distance between data samples “t+1” and “t+2”. In this case, the GPS at time “t+2” determine the speed, i.e., zero speed, because the pedestrian has not covered any distance during “t+1” and “t+2” timestamp. To confirm the pedestrian is stopped completely we require two consecutive GPS update with the zero speed. In this scenario, we confirm the stop of the pedestrian at sample “t+3”, because “t+2” and “t+3” show the same speed, i.e., zero. In this scenario, we recognise the pedestrian movement slowing down to stop and confirm the time required to recognise a complete stop takes 3s. We considered the scenario 1 as the best case because it gives minimum delay compared to scenario 2.

Scenario 2: The pedestrian is slowing down to stop after the GPS data sample arrives, for example after “t+1” as shown in Figure 3-13. Where the X-axis shows the sample update time in seconds, and the Y-axis shows the speed in m/s. In this scenario, the pedestrian is walking normally at an average speed of 1.2m/sec as shown at time stamp “t” and afterwards slowing down to stop but not suddenly. As shown in Figure 3-13, the pedestrian is walking at an average speed of 1.2m/sec and after the sample update at time “t” is slowing down the walking speed and stop after sample update at “t+1”. As soon as the data sample at time “t+2” arrives, GPS determines the average speed based on the travelled distance between time “t+1” and “t+2” which is not zero but 0.83m/sec and recognise this as the pedestrian is slowing down but not stop. GPS determines the average speed based on the travelling distance of a pedestrian between data samples “t+2” and “t+3”. In this case, at time “t+3” zero speed is recognised because the pedestrian has not covered any distance during “t+2” and “t+3” timestamp. Similarly, to scenario 1, to confirm the pedestrian is stopped completely we require two consecutive GPS update with the zero speed. In this scenario, we confirm the stop of the pedestrian at sample “t+4”, because “t+3” and “t+4” show the same speed, i.e., zero. In this scenario, we recognise the pedestrian movement, slowing down to stop and confirm the time required to recognise a complete stop takes 4s. Scenario 2 gives 1s higher delay in comparison to Scenario 1 because of the reason that pedestrian changes movement speed between three GPS data samples.

Furthermore, we analysed that the scenarios of a pedestrian accelerating are similar to the scenarios of slowing down to stop (i.e., scenario 1 and scenario 2). Both scenarios of pedestrians acceleration between two and three GPS updates, we analysed similar results as scenario 1 and scenario 2. When the pedestrian accelerates between two GPS updates (similar to scenario 1) from low speed to higher and maintains the speed afterwards, we require 3s to recognise the movement. When the pedestrian accelerates between three GPS updates (similar to scenario 2) from low speed to higher and maintains the speed afterward, we require 4s to recognise the movement. In contrasts to the scenario of slowing down to stop and accelerating, the scenario of a pedestrian moving at a constant speed is different. In the scenario of the pedestrian moving at a constant speed, using the GPS data we determine the walking movement within a time of 2s. The GPS only requires two similar successive speed updates to recognise the pedestrian activity, i.e., walking or stop.

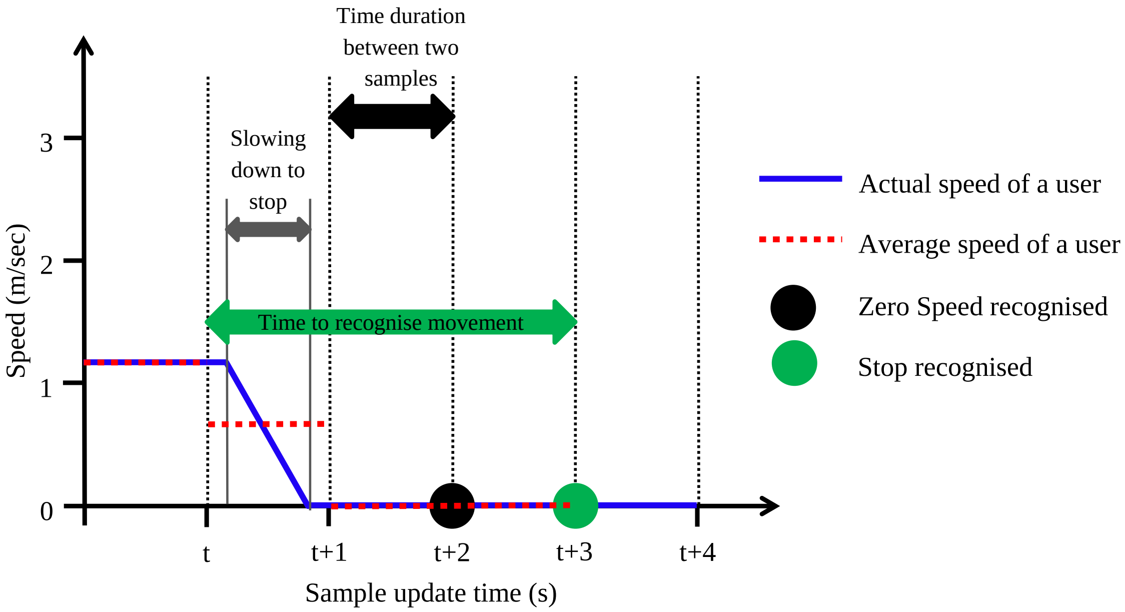


Figure 3-12 Slowing down to stop between two GPS position measurements [13]

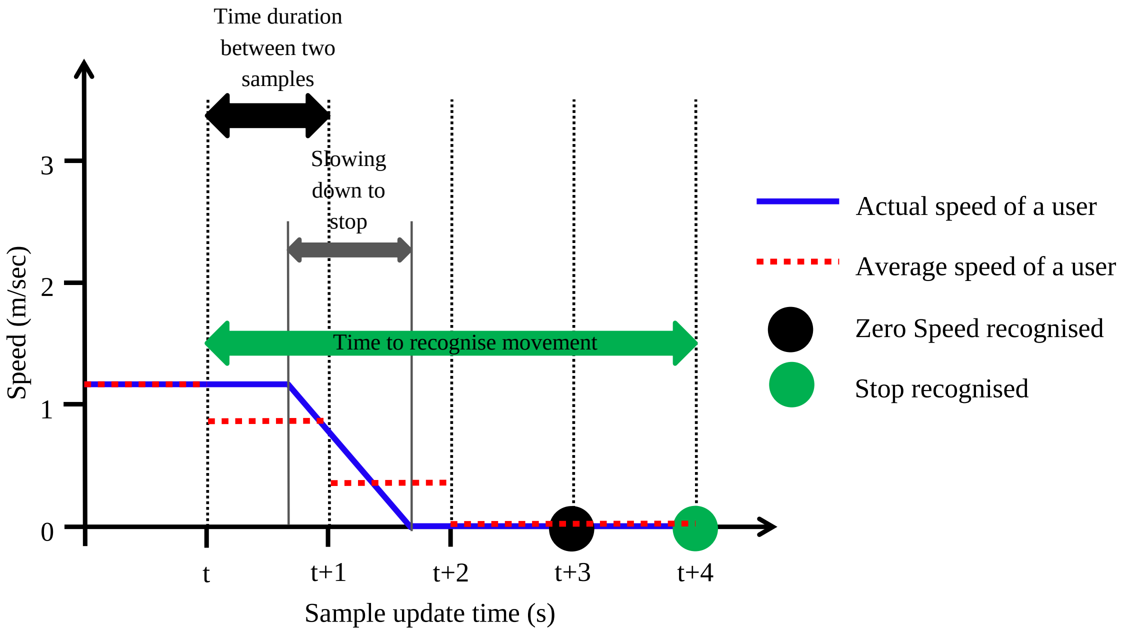


Figure 3-13 Slowing down to stop while a GPS position measurement is taken [13]

3.3 Impact of the measured results on an accident scenario

In this section, we discuss the influence of the GPS and an accelerometer based measurements on an accident scenario. In [12], Flach et al presented an analysis of an accident scenario. In that typical accident scenario, Flach et al described the available time for collision avoidance is 2.1s and that is the time required by the system to transfer the pedestrian movements data to a centralised server, processed the data and sent a warning to a pedestrian and a car in case of an accident occur. We investigated in this chapter that the GPS based on a smartphone provides a maximum sampling frequency of 1Hz, i.e., one position update per second. However, we assumed that the system based on the only GPS might result in misinterpretation of different situations such situations are categorised as false alarms or not warned although the pedestrian is endangered (also known as the missed alarm).

The false alarm is defined as “If the pedestrians are not being endangered and on top of that, the system generates a warning alarm that alarm is known as the false alarm. For example, as shown in a typical accident scenario (see Figure 3-14), the pedestrian is moving towards the road with the constant speed but not with the intention of crossing the road, but to get to the trunk of a parked car and stop. Using the GPS one only estimates the risk of collision using the constant speed and movement direction of the pedestrian and the car. However, as shown in the results, the GPS data might not fast to update the speed of the pedestrian. In such scenario, the pedestrian is stopped at the trunk of the car, but it is already late for the GPS to recognise the new speed and recalculate the risk of collision and therefore fail to avoid generating the alarm, though in actual the pedestrian is not being endangered.

Let's take an example of the pedestrian is being endangered but not warned by the system. GPS only estimates the risk of collision based on the information of actual trajectory and speed of the pedestrian and the car. Such a system works only if the speed of the pedestrian remains constant. However, this is not the practical case, pedestrians change their speeds unpredictably, and if this happens, the system based on the GPS only can miss judge the collision risk estimation. For example, the pedestrian is walking towards the road with constant speed, and the system based on the GPS only estimates the risk of collision, i.e., the pedestrian is not being endangered. All of a sudden the pedestrian increases the speed with the intention to cross the road and put himself in endanger. In such case, it is already late for the GPS to recognise the new speed, estimate the risk of collision, and therefore miss the warning alarm, though in actual the pedestrian is being endangered.

As shown in the results, activity recognition based on the accelerometer of a smartphone can determine the stop, deceleration or acceleration of a pedestrian. The single movement recognition can be achieved in a time window of 500ms. This analysis concludes that the system based on accelerometer-based activity recognition can avoid generating false alarms and also gives an opportunity to be 1.5s quicker in comparison to the GPS only when the pedestrian is walking at constant speed.

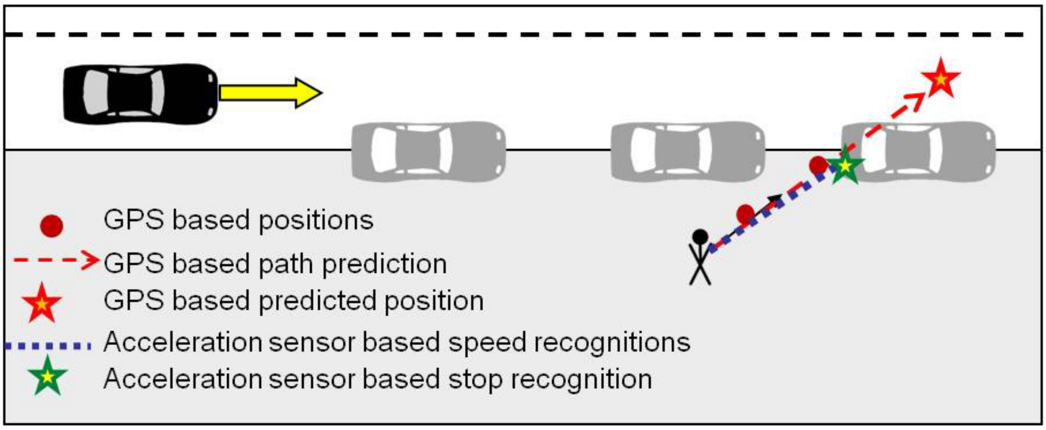

GPS based positions
GPS based path prediction
GPS based predicted position
Acceleration sensor based speed recognitions
Acceleration sensor based stop recognition

Figure 3-14 Reduction of false alarms with the help of acceleration sensor based [13]

3.4 Conclusion

In this chapter, we presented an approach, which uses the built-in accelerometer of a smartphone to recognise the changes in the speed of the pedestrian. From measurements and results, the accuracy and time needed for movement recognition were analysed. The results of these experiments have shown the movement recognition accuracies between 93.39% and 96.98%. Also, the time needed for classification indicated that all classifiers presented were able to complete the feature extraction process and desired recognition within 450ms. Furthermore, accelerometer-based activity recognition has an advantage of being 1.5s quicker than the GPS. This comparison was made with an accelerometer sampling frequency 32Hz and the GPS sampling frequency of 1Hz and only when the pedestrian walking speed is constant.

The ability to determine changes quickly in the speed of a pedestrian is important for many applications such as pedestrian safety and pedestrian dead reckoning. Similarly, the ability to determine changes quickly in the walking direction is equally important. Therefore, the next step is to analyse the use of a compass or gyroscope sensor of a smartphone to determine changes in the walking direction.

4 Pedestrians Movement Direction Detection

Smartphone sensors deliver useful information for applications such as indoor and outdoor navigation. An integral part of such applications is the detection of the orientation and movement direction of a smartphone user. Until now, movement direction detection using smartphones typically relies on the GPS, which is often not available indoors. Alternatively, other approaches use sensors such as accelerometer and compass instead. These approaches rely on carrying a smartphone in a predefined orientation or knowing the orientation of a smartphone in relation to the orientation of the user. In this chapter, we present an approach to detecting the orientation and movement direction of users/pedestrians carrying smartphones inside the trouser pocket. This approach first determines the orientation of the smartphone's top using the compass and the orientation sensor. Second, this approach determines the orientation of the smartphone's screen, and the user's movement direction by observing compass and accelerometer during at least two steps the user/pedestrian takes. After these two steps, the approach is capable of continuously aligning a smartphone orientation and the user/pedestrian orientation. With our approach, the user/pedestrian is free to change direction, movement speed, or to stop moving at all. The smartphone can be placed in the trouser pocket arbitrarily. Moreover, the smartphone is free to wobble in the trouser pocket. How well our approach works, is investigated based on experimental measurements. In this dissertation, we used the term "user" and "pedestrian" interchangeably.

4.1 Introduction

Parts of the contents of this chapter are published in [85].

In our previous paper [13], the approach is presented for Pedestrian Movement Recognition (i.e., slowing down to a stop, accelerating, and decelerating) by using the accelerometer of a smartphone. Besides the Pedestrian Movement Recognition, knowing the pedestrian movement direction is also important. In this chapter, we discuss the possibility of using further sensors and contexts to determine the direction of pedestrians' movement. Therefore, we proposed to add a compass of a smartphone to recognise the movement direction of pedestrians. This movement direction ranges from 0 to 360 degrees. In our previous paper [7], an analysis and a solution are presented to justify why the compass of a smartphone is still seen as feasible for the detection of the movement of pedestrians.

In this chapter, we discuss the detection of the movement of pedestrians using the sensors of a smartphone, especially when the user/pedestrian carries a smartphone in the trouser pocket. Firstly, this section defines, what is the user/pedestrian orientation? Moreover, what is the user/pedestrian movement direction? A User/pedestrian orientation is the direction to which a user's feet point. A user/pedestrian has an orientation even if she is standing still. A User/pedestrian movement direction is the direction in which a user/pedestrian moves. If a user/pedestrian stands still, no movement direction is available. A smartphone orientation is the way a smartphone is situated in the coordinates system. It is described by three angles that express how many degrees the smartphone is turned around the X-, Y-, and Z-axis of the coordinates system. For a smartphone, three axes are defined relative to the screen when it is held in a default position as shown in Figure 4-1 and known as smartphone coordinates system (device coordinates system). A compass is a sensor, which provides three angles (pitch, roll and azimuth) measured in degrees in relation to the smartphone coordinates system and global coordinates system. The term pitch is represents the rotation around the X-axis, the roll represents the rotation around the Y-axis and azimuth is defined as the rotation around Z-axis of the device coordinates system relative to the magnetic north in the global coordinates system as shown in Figure 4-1.

To infer the correct movement direction from the compass azimuth, the smartphone orientation and the pedestrian movement direction must be the same. If this is not the case, the compass azimuth might mislead the movement direction of pedestrians. When users/pedestrians carry a smartphone, e.g., in the trouser pocket, a user/pedestrian orientation (user/pedestrian frame) differs from a smartphone orientation (device frame) as shown in Figure 4-2. Here, the problem arises that the data collected with the smartphone is not aligned with the orientation of the smartphone user. For example, Data collected from the smartphone accelerometer for the X-axis of the device frame does not mean, the user/pedestrian accelerated along the X-axis of the user/pedestrian frame. Figure 4-2 shows the two coordinate systems, i.e., "P" describes the users/pedestrians frame in red and "D" describes the device frame in green. The Y_P is the pedestrian movement direction, and Y_D is the device orientation direction. In this case, the phone orientation does not represent the actual pedestrian movement direction.

Moreover, pedestrians might carry their smartphones in different orientations such as possible 16 orientations in a pocket as shown in Figure 4-3. A smartphone orientation is a specific mode of carrying the smartphone in a trouser pocket. Furthermore, it represents most of the smartphone orientations are not the same as the movement direction of pedestrians and causes wrong movement direction. We call this the “smartphone in the pocket problem”. The question is how to align the user/pedestrian frame and device frame? Until now, either the smartphone had to be carried in a predefined way. Alternatively, the orientation of the smartphone was not allowed to change in relation to the orientation of the user.

In this chapter, we present the algorithm called by us Smartphone Orientation and Movement Direction Alignment (SOMDA). The SOMDA is an approach to align users/pedestrians orientation and a smartphone orientation while the user/pedestrian is free to change direction, movement speed, or to stop moving at all. The smartphone can be placed in the trouser pocket arbitrarily. Moreover, the smartphone is free to wobble in the trouser pocket. To do this, SOMDA used both accelerometer and compass of a smartphone. The SOMDA determines the exact orientation of a smartphone inside the pocket, based on the pitch, and roll angle in conjunction with steps peak detection obtained from the accelerometer. Once the orientation of a smartphone in a pocket is identified, the algorithm aligns the smartphone orientation to the movement direction of pedestrians in order to determine the correct direction from compass azimuth. We believe that this work can leverage a wide range of applications that collect sensor data with smartphones.

In this work, we applied the SOMDA to one particular scenario, i.e., pedestrian collision avoidance [6], to show its working principle and applicability: A user/pedestrian – Alice – is carrying a smartphone and walking on the sidewalk. If Alice changes her movement direction towards the road, the information that she approaches the road is communicated to all nearby cars. If a risk exists that a car and Alice may collide, the concerned car driver is alarmed to avoid the collision. In such pedestrian collision avoidance, the movement direction of Alice needs to be known. If she turns, her direction needs to be detected as soon as possible. The time required for detecting movement direction is critical in this scenario. Each millisecond the turn can be detected earlier reduces the reaction time of the car driver and enlarges the possible braking distance. We show that the SOMDA is capable of accurately detecting movement direction and orientation of a user. Moreover, the SOMDA detects turns considerably faster than using A-GPS.

This chapter is organized as follows: In section 4.2, we present the methodology and working of the SOMDA algorithm. In Section 4.3 the design and purpose of measurements are explained, and the corresponding results are discussed. In Section 4.4, we give the conclusion.

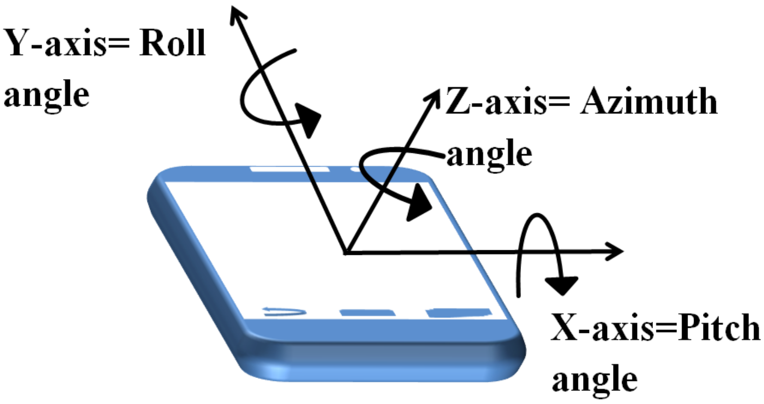


Figure 4-1 Smartphone coordinate system (Device frame) [85]

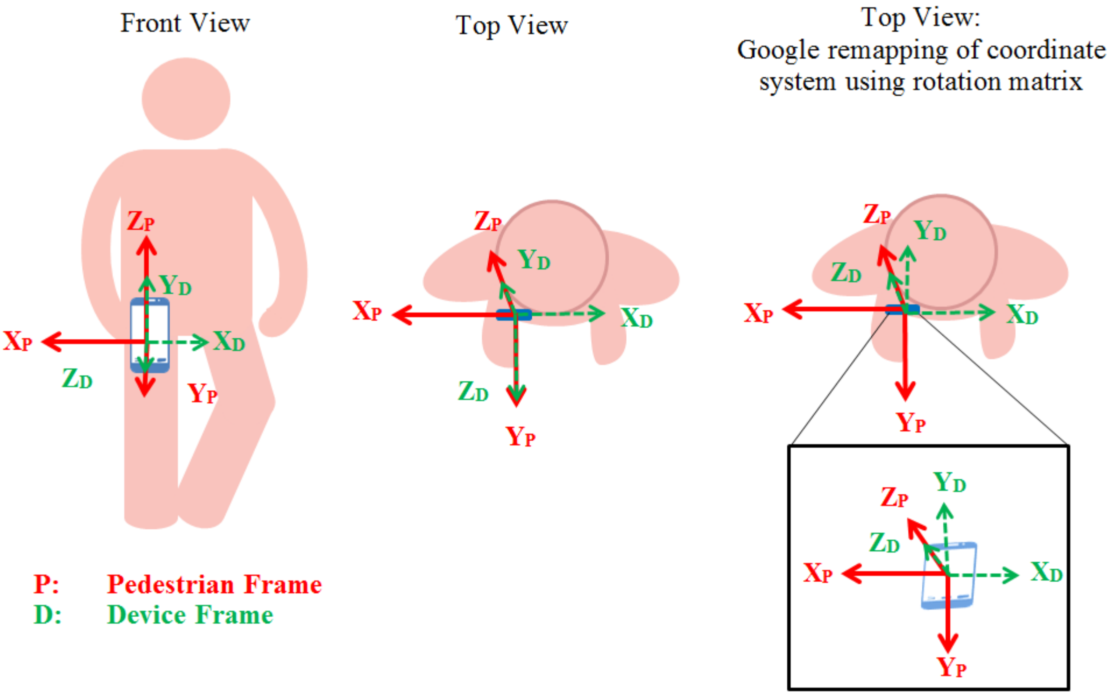

Front View
Top View
Top View: Google remapping of coordinate system using rotation matrix
ZP
YD
XP
XD
ZD
YP
P: Pedestrian Frame
D: Device Frame

Figure 4-2 User/pedestrian frame and device frame

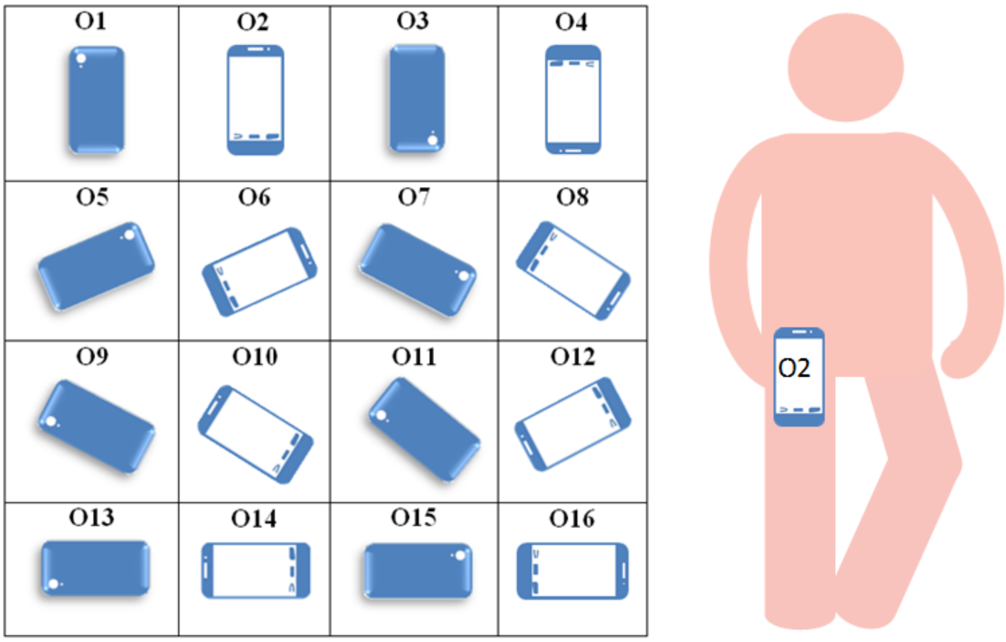


Figure 4-3 Smartphone Orientation in the trouser pocket [85]

4.2 Methodology and Approach

In this section, we first give the introduction of the sensors used in this work and we explain the influence of the smartphone orientation on the user/pedestrian movement direction. Afterwards, we present in detail how to align both the smartphone orientation and the user/pedestrian movement direction using SOMDA. The SOMDA approach works independently of a mobile operating system or a smartphone model. However, we implemented and evaluated the SOMDA on a Google Nexus 5 smartphone, using Android OS version 4.1.2. During the measurements, we recorded sensors data at a sampling rate of 50Hz. Moreover, we have analysed that 50Hz sampling rate provides the stable data rate, i.e., each sample/data point recorded after every 0.02s (20ms).

4.2.1 Hardware specification of built-in sensors of a smartphone

Accelerometer: The Google Nexus 5 has an integrated MPU 6515 6-axis accelerometer chipset manufactured by InvenSense. The application records the accelerometer data in m/sec^2 along X, Y, Z-axes. The X, Y, Z-axes are defined in relation to the screen of the smartphone (device frame) as shown in Figure 4-1.

Magnetometer: The Google Nexus 5 has an integrated AKM 8963C magnetic field sensor chipset manufactured by Asahi Kasei Microdevices (AKM) Pvt Ltd. The application records the magnetic field data in micro-Tesla (uT) along X, Y, Z-axes. The X, Y, Z-axes are defined in relation to the screen of the smartphone (device frame) as shown in Figure 4-1.

Orientation Sensor: An orientation sensor is a virtual sensor, providing the pitch, roll and azimuth angle measured in degrees in relation to the smartphone coordinates system (device frame). The device frame is defined relative to the screen when the smartphone is placed in its default orientation as shown in Figure 4-1. The X-axis is the horizontal axis, points to the right and left when a smartphone held in a default position. The Y-axis is the vertical axis, points to the top and bottom when a smartphone held in a default position. The Z-axis points towards the outside of the front face and back face of the screen. The term pitch is defined as the rotation (in degrees) around the X-axis of smartphones coordinates system (device frame), value ranges -180 to 0 to +180 degrees. The term roll is defined as the rotation (in degrees) around the Y-axis of the smartphone coordinates system (device frame), value ranges -90 to 0 to +90 degrees. Azimuth is defined as the rotation around the Z-axis of the device coordinates system in relation to the global coordinates system; value ranges 0 to 359 degrees.

Compass: A compass is a virtual sensor in a smartphone, which provides the rotation around the Z-axis of the device coordinate system in degrees.

Global Positioning System (GPS) receiver: The GPS is a satellite-based navigation system. The GPS receiver receives geographical location information from GPS satellites. It calculates the GPS bearing and distance between two geographical locations in degrees providing a maximum sampling rate of 1Hz. Also provides the position of the receiver on the ground.

Assisted-Global Positioning System (A-GPS) receiver: A-GPS use the mobile network or any data connection to improve the accuracy, the reception and the “time to first fix”.

4.2.2 Effect of smartphone orientation on pedestrians movement direction

Pedestrians might carry their smartphones in a trouser pocket with different orientations. In this section, we investigate the influence of the smartphone orientation on the compass azimuth while detecting the pedestrian’s movement direction. Therefore, we design and collect measurements, where the smartphone is placed inside the trouser pocket of a user. For each measurement, we put the smartphone in a trouser pocket with different orientations including O1-O16 as shown in Figure 4-3. As shown in Figure 4-3, most of the smartphone orientations are not the same as the movement direction of pedestrians and causes wrong movement direction. The test path and the movement direction of users/pedestrians remained the same for all measurements. The measurement with orientation (O1) is considered as the reference measurement; this is because the orientation of O1 measurement is similar to the movement direction of the user. The user/pedestrian is walking along a specific test path, and the sensor logging application records the sensors raw data.

Figure 4-4 exemplarily shows the influence of the smartphone orientations on the compass azimuth. We record the compass sensor data at a sampling rate of 50Hz, i.e., each sample recorded at an interval of 20ms. The X-axis shows the number of samples taken from the compass. The Y-axis shows the compass azimuth in degrees (0-359°).

Here we introduce the term “misalignment” that represents the difference in degrees between the measured and reference/actual data. Figure 4-5 exemplarily shows the misalignment of the compass azimuth and the user’s movement direction (in degrees) for the smartphone orientations O1 to O4. On the X-axis, it shows the number of samples taken from the compass. Before applying the SOMDA, we observed that the compass azimuth is different for O2 and O3 with respect to the reference measurement, i.e., O1. From this analysis, we can trust the compass azimuth and say this is the movement direction when the user carries the smartphone with orientations O1 and O4, but this does not apply to orientation O2 and O3.

To analyse how the measured data look like we used the most frequently selected features are mean, median, and standard deviation. We choose these simple statistics features due to simplicity and low computational cost. The terms Mean and Median are used to represent the central tendency of the data set. Central tendency is defined as a single value by identifying the central position/location of any data set. In our case, the data set consists of the movement direction data of the user/pedestrian (ranging 0-360 degree), while the user/pedestrian walks on a defined test path. In this case, mean is defined as the single and average value (sum of all data points available in the data set divided by the number of data points) of the whole data set. Standard deviation represents the dispersion in the measured data with respect to the mean value. A lower standard deviation shows that the data points tend to be closer to the mean value of the data set, while higher standard deviation represents the data points in the set are spread out in comparison to mean value. Median is defined as the middle value when the data set is arranged in numerical order (i.e., from least to high). In this dissertation, we limit the data of the movement direction of a user/pedestrian to three decimal digits in order not to lose any micro-movement.

In Table 4-1, we show features such as mean, standard deviation, median of measured data set with different orientations of a smartphone, i.e., O1-O4. Furthermore, we show that the O2 has an average misalignment of 171° and O3 has an average difference of 176° before alignment. This is because O2 and O3 are different orientations than the movement direction of pedestrians. i.e., a rotation of 180° of the smartphone for O2 and O3 compared to O1 and O4. This means, as long as we do not know the orientation of the smartphone we cannot detect the actual movement direction of the smartphone user. In such cases, the smartphone orientation influences the compass azimuth detecting the wrong movement direction of pedestrians. In contrast, O4 has an average misalignment of 4.8°; this is because O4 has a similar orientation to the movement direction of pedestrians.

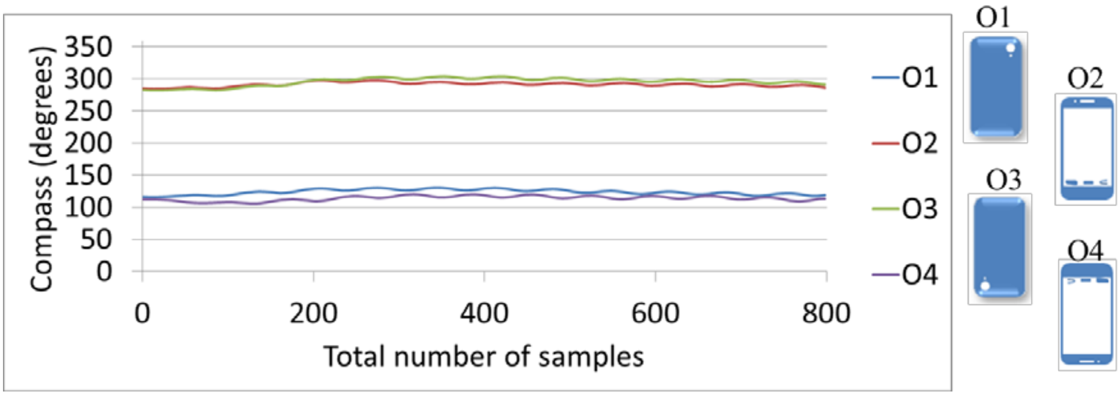


Figure 4-4 Compass sensor data before aligning smartphone orientation and user/pedestrian orientation by SOMDA [85]

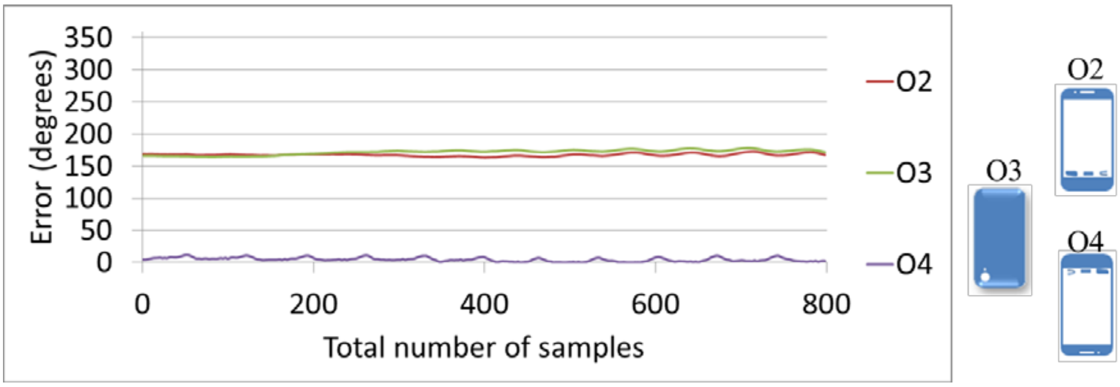


Figure 4-5 Misalignment of compass sensor data and user/pedestrian movement direction [85]

Table 4-1 Compass Azimuth Features (Degrees) before SOMDA Alignment [85]

Features	Measurements			
	O1	O2	O3	O4
Mean	118.743	290.741	295.686	113.918
Standard Deviation	3.769	3.115	5.367	3.466
Median	118.643	290.905	296.375	114.427
Average Misalignment	-	171.998	176.944	4.824

4.2.3 Smartphone Orientation and Movement Direction Alignment (SOMDA) algorithm

In this section, we explain how smartphone orientations on the one hand, and the user/pedestrian orientation, as well as the user/pedestrian movement direction, on the other hand, is aligned using the SOMDA. The user/pedestrian is free to change direction, movement speed, or to stop moving at all (once the user/pedestrian took two steps). The smartphone can be placed in the trouser pocket arbitrarily. Moreover, the smartphone is free to wobble in the trouser pocket. The compass azimuth is the angle of rotation in degrees (0-359°) around the Z-axis of the device coordinates system, relative to the magnetic north. To detect the movement direction of a user/pedestrian from the compass azimuth, the smartphone orientation and the user/pedestrian orientation (and in addition to that, the users/pedestrians movement direction) have to be identical. If this is not the case, the compass azimuth might mislead the movement direction of users. Therefore, it is important to align the smartphone orientation to the user orientation while determining the movement direction. In Figure 4-6, we show the detailed flowchart about how to align the smartphone and the user/pedestrian orientation using the SOMDA algorithm. The SOMDA algorithm works independent of smartphone orientations, GPS, and other state of the art approaches. The SOMDA algorithm uses the built-in accelerometer and compass of a smartphone. We divided the SOMDA into two steps, which repeats continuously:

1. Determine the top of the smartphone:

In this step, the SOMDA checks the pitch angle θt of the smartphone, which is derived from the orientation sensor. If θt is below zero (i.e., a negative value), the roll angle ϕt is added to the compass azimuth Ψt. If θt is above zero (i.e., a positive value), the roll angle ϕt is subtracted from Ψt (see (1) and Figure 4-6).

$$\psi_{RC} = \begin{cases} \psi_t + \phi_t & \left| \theta_t < 0 \right. \\ \psi_t - \phi_t & \left| \theta_t > 0 \right. \end{cases} \tag{1}$$

We call this step as roll correction ΨRC. After roll correction, the SOMDA assumes that the smartphone is perpendicular to the ground inside the user/pedestrian pocket. The roll correction also eliminates the problem that a wobbling smartphone in the trouser pocket causes, e.g., while the user/pedestrian is walking.

2. Determine the screen of the smartphone and the movement direction of the user:

In this step, using SOMDA, we determine if the smartphone orientation is the same as (screen points towards the user) or opposite to (screen points away from the user) the movement direction of the user. To infer this information, the SOMDA uses the accelerometer to detect the peaks of two steps a user/pedestrian takes. Peak detection is the process of finding the minimum and maximum peak values in accelerometer data, which occurs during the pedestrian walk cycle. The maximum or minimum step peak detection occurs when pedestrians is at high point stage. Based on the positive, negative peaks and Pitch angles, SOMDA identifies the smartphone orientation in the pocket. From the physical analysis, we observed that pedestrians averagely take 1.6s to complete one cycle and in each cycle, accelerometer observes one peak. A peak is a maximum value in the accelerometer data during one-step.

To detect a peak, the SOMDA segments the accelerometer data using the windowed peak detection algorithm. In [120], Brajdic et al. showed that the error rate of the windowed peak detection algorithm is 1.3%. They evaluated the data based on 27 adults, 130 walks and 6 different smartphones. Based on the Brajdic analysis, we decided to use the windowed peak detection algorithm. We set the window size to 1.6s (mean duration of one-step) and a threshold for the acceleration az_TV = ±2.3 m/sec^2. This threshold value is based on the mean acceleration of one-step in the data we measured (see Figure 4-7). Accelerometer inaccuracy and drift can be neglected because the accelerometer is only used to detect the exceedance of a threshold (az_TV) over a short period of time (two steps). Equation (2) is used to detect a peak while taking steps. *PD*ws_a$_z$ represents the result of the peak detection in the accelerometer data within the given windows size. Where a$_z$, is the acceleration value on the Z-axis of the accelerometer. *PD*ws_a$_z$ equals a$_z$ if a peak is detected inside the sliding window.

$$PD_{ws\_a_z} = \begin{cases} a_z & |a_z > a_{z\_Tv} \;\; or \;\; a_z < -a_{z\_Tv} \\ 0 & |else \end{cases} \tag{2}$$

If *PD*ws_a$_z$ is equal to 0, it means the SOMDA detects no peak. If *PD*ws_a$_z$ is smaller than zero, the SOMDA detects a negative peak. If *PD*ws_a$_z$ is above zero, the SOMDA detects a positive peak. A peak can be positive or negative depends on the screen orientation of the smartphone in the pocket. Figure 4-7 shows an example of positive and negative peak detection using the threshold az_TV = 2.3 m/s^2. The X-axis shows the time in seconds, and the Y-axis shows the acceleration in m/s^2 for two different orientations (O1 and O2, see Figure 4-3) of a smartphone carried inside the pocket. When the smartphone is placed in a trouser pocket with O1, the SOMDA detects the negative, and for O2, the SOMDA detects a positive peak.

Next, the information about a positive or negative peak is combined with the pitch angle, which again is determined using the orientation sensor. As shown in Figure 4-6, if the pitch angle is negative and a negative peak was detected, the SOMDA assumes the smartphone screen points towards the user. Similarly, if the pitch angle is positive and a positive peak was detected, the SOMDA assumes the smartphone screen also points towards the user. In both cases, the smartphone orientation, the user/pedestrian movement direction, and the user/pedestrian orientation are already aligned.

Moreover, if the pitch angle is negative and a positive peak was detected, the SOMDA assumes the smartphone screen points away from the user. Similarly, if the pitch angle is positive and a negative peak was detected, the SOMDA assumes the smartphone screen also points away from the user. In these two cases, the SOMDA analyses that the smartphone and the user orientations are not the same. Therefore, the SOMDA aligns the smartphone orientation, and the user/pedestrian movement direction and the user/pedestrian orientation by adding 180° to the compass azimuth. All four cases are shown in (3) and in Figure 4-6. *Ψ*AA is the compass azimuth after the smartphone orientation, the user/pedestrian movement direction, and the user/pedestrian orientation was aligned by SOMDA.

$$\psi_{AA} = \begin{cases} \psi_{RC} & \left| \theta_t < 0 \ and \ PD_{ws\_a_z} < 0 \right. \\ \psi_{RC} + 180 & \left| \theta_t < 0 \ and \ PD_{ws\_a_z} > 0 \right. \\ \psi_{RC} + 180 & \left| \theta_t > 0 \ and \ PD_{ws\_a_z} < 0 \right. \\ \psi_{RC} & \left| \theta_t > 0 \ and \ PD_{ws\_a_z} > 0 \right. \\ 0 & \left| else \right. \end{cases} \tag{3}$$

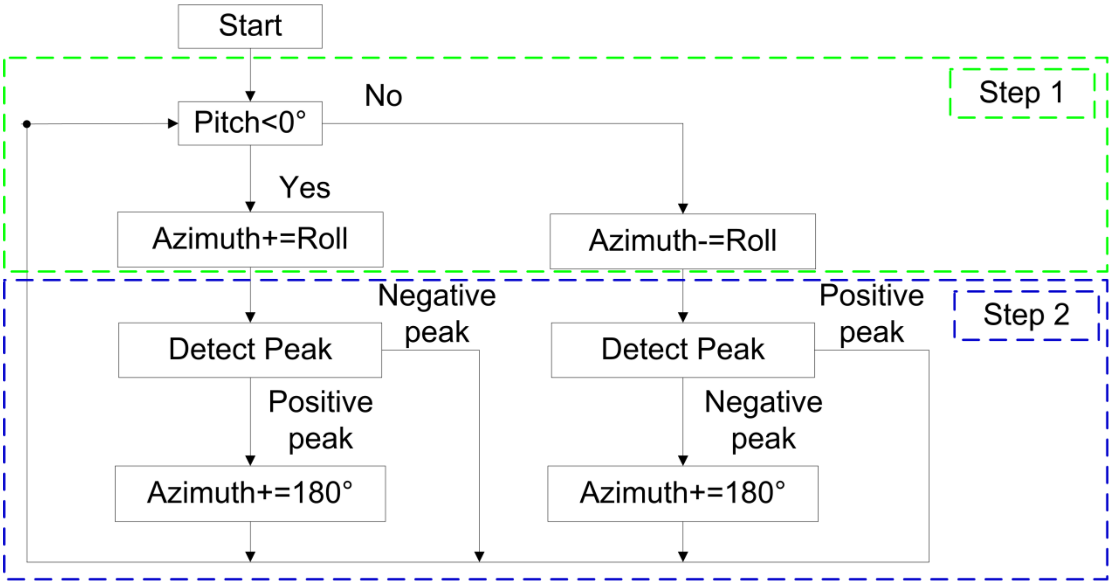


Figure 4-6 Flowchart of SOMDA [85]

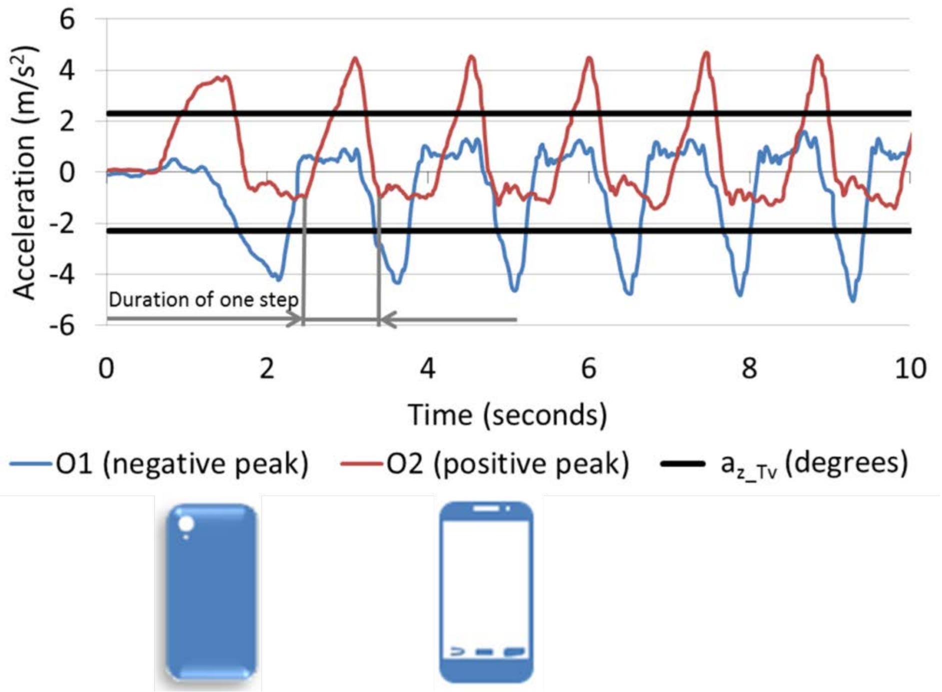


Figure 4-7 Accelerometer pattern and Thresholds during step detection [85]

4.3 Results and Discussion

In this section, we present the results and the performance analysis of the SOMDA. The accuracy with which the SOMDA aligns the smartphone orientation and the user/pedestrian orientation is presented in Section 4.3.1. In Section 4.3.2, we show how fast the SOMDA detects changes in the user/pedestrian orientation and movement direction compared to A-GPS.

4.3.1 Accuracy evaluation

In Figure 4-8, we exemplarily show for O1 to O4 that the SOMDA successfully aligned the smartphone orientation to the user/pedestrian orientation. The X-axis shows the number of samples taken from the compass, and the Y-axis shows the compass azimuth in degrees. In Figure 4-9, we exemplarily show for O1 to O4 the misalignment of the compass azimuth and the user's movement direction (in degrees). On the X-axis, it shows the number of samples taken from the compass, and the Y-axis shows the difference (misalignment) between compensated/aligned data to the actual/referenced data. The average misalignment after applying the SOMDA is 2° to 6°. In contrast, the average misalignment is 117° without applying the SOMDA (see Figure 4-5). The results show that the compass azimuth was successfully aligned to the reference measurement with orientation O1 (see Section 4.2.3). We discuss in detail the accuracy of the SOMDA later in this paragraph.

Table 4-2 exemplarily shows for O1 to O4, different features of the compass azimuth after applying the SOMDA. Mean represents the average value of the measured data. Standard Deviation represents the dispersion in the measured data with respect to the mean. Median represents the middle value in a list of data points ordered by value. Average Misalignment (MA) represents the average difference in degrees between the measured and reference/actual data. Table 4-2, shows that the O2-O3 has an average misalignment of less than 6° and O4 has an average misalignment of 1°. In contrast, before the use of the SOMDA algorithm, the average misalignment for O2 and O3 was 171° and 176° respectively.

The accuracy of the SOMDA for all measurements with different orientations O1-O16 conducted within the scope of this chapter is shown in Figure 4-10. The X-axis shows the threshold value and the Y-axis shows the accuracy of the SOMDA algorithm. The accuracy is defined as the correctness of the algorithm (measured in percent) considering a certain threshold. The threshold represents how far the measured data is allowed to differ from the reference measurement. The 100% accuracy represents the measured data coincide with the reference measurement. Using the SOMDA, we achieve an accuracy of 96% at a threshold value of 15° using a sampling rate of 50Hz. The accuracy of the SOMDA algorithm is calculated using equation (4).

$$Accuracy = \left(\frac{TC_{dp}}{T_{dp}} \right) * 100 \tag{4}$$

Where "C_{dp}" is used as a symbol for the correctly fixed data point in the measurement, "TC_{dp}" is used as a symbol for the total correctly fixed data point in the measurement and "T_{dp}" is used as a symbol for total data points in the measurement. The data points that are under the threshold value for the whole measurement are considered as the correctly fixed data points.

Furthermore, the original data measured at 50Hz is downsampled to 25Hz, 12.5Hz, and 6.25Hz. Downsampling from 50Hz to 6.25Hz reduces the accuracy by 6% at a threshold value of 20° and 15° respectively as shown in Figure 4-10.

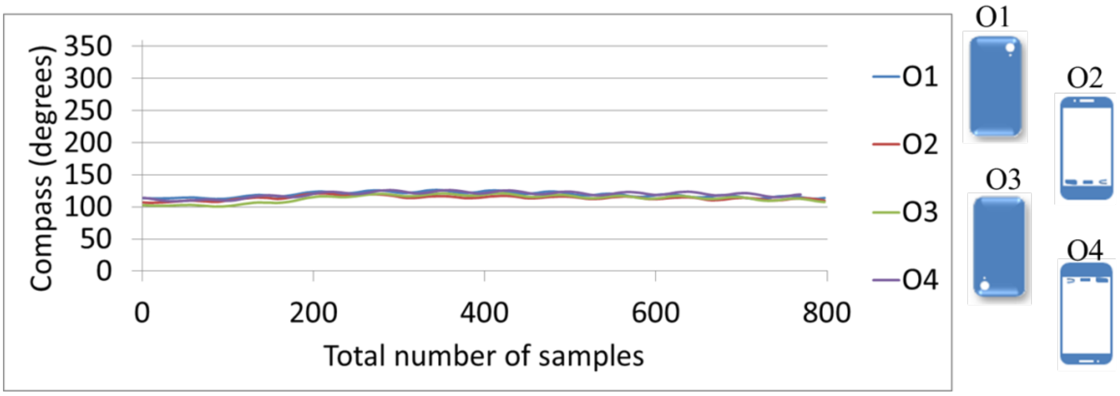


Figure 4-8 Compass sensor data after aligning smartphone orientation and user/pedestrian orientation by SOMDA [85]

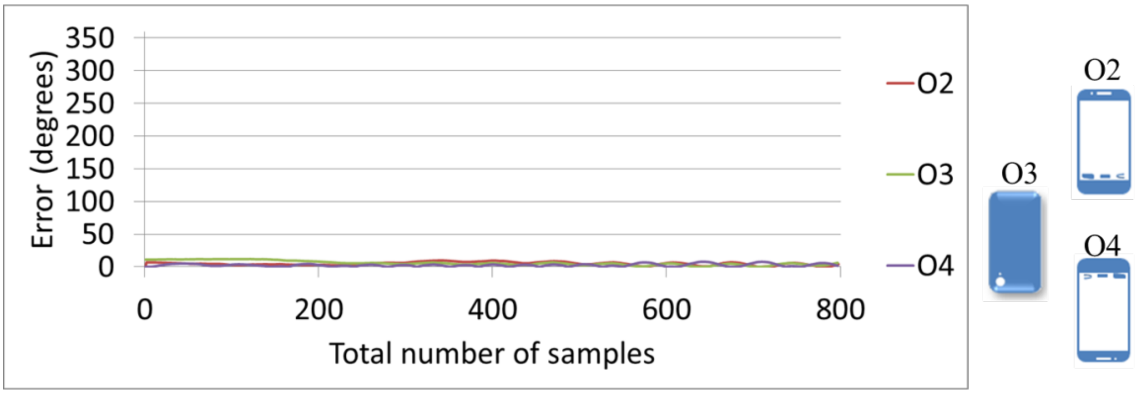


Figure 4-9 Misalignment of compass sensor data after aligning smartphone orientation and user/pedestrian movement direction by SOMDA [85]

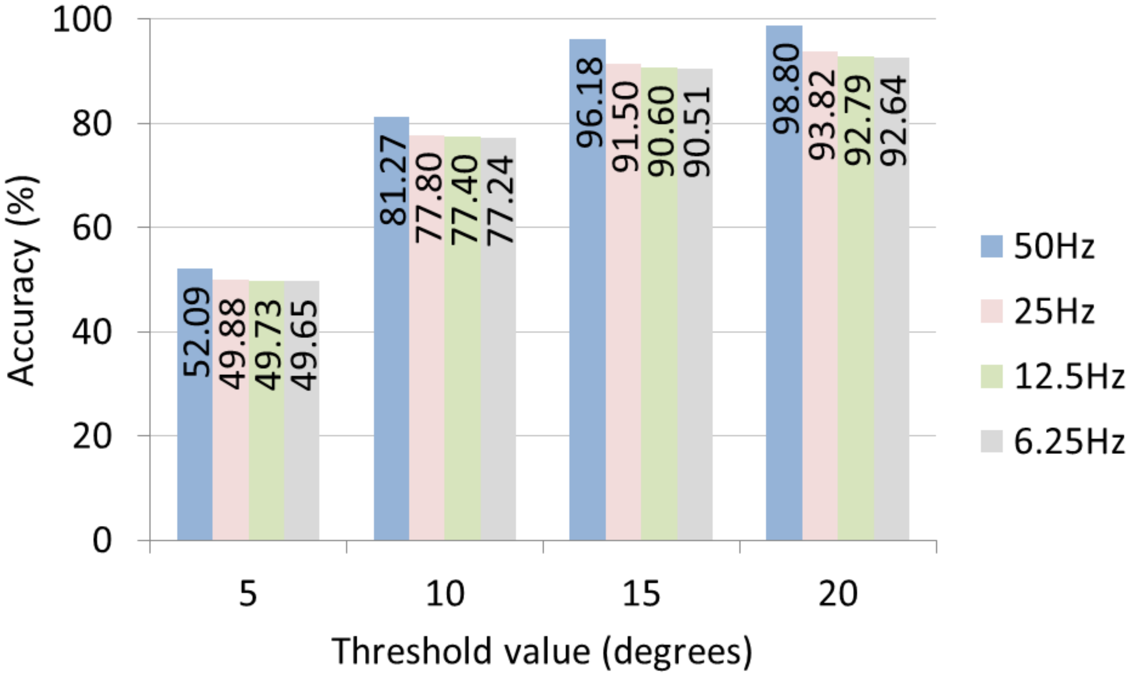


Figure 4-10 Accuracy of SOMDA for different sampling rates [85]

Table 4-2 Compass Azimuth features (Degrees) after applying SOMDA [85]

	Before applying SOMDA				After applying SOMDA			
	O1	O2	O3	O4	O1	O2	O3	O4
Mean	118.743	290.741	295.686	113.918	118.743	113.471	112.828	119.632
Standard Deviation	3.769	3.115	5.367	3.466	3.769	3.186	5.574	3.826
Median	118.643	290.905	296.375	114.427	118.643	113.466	113.368	119.912
Average Misalignment	-	171.998	176.944	4.824	-	5.271	5.915	0.889

4.3.2 Comparison of SOMDA and A-GPS

In this section, we present the comparison of our proposed SOMDA algorithm with A-GPS while detecting the movement direction of a user/pedestrian. The experiments were conducted to determine how much time the SOMDA and A-GPS need to detect the movement direction and different turns of a user. Five different test paths were analysed that include walking straight and turning as shown in Figure 4-11.

1. Test path 1 turning 90° (TS1_90 degrees turn)
2. Test path 2 turning 45° (TS2_45 degrees turn)
3. Test path 3 turning 135° (TS3_135 degrees turn)
4. Test path 4 turning 180° (TS4_180 degrees turn)
5. Test path 5 turning 0-360° while standing on one point.

During the measurements, two test persons carried a smartphone in the pocket with arbitrary orientations. In addition, the test persons carried one smartphone in the default orientation, i.e., the user/pedestrian frame and device frame were aligned. We trained the test persons were trained to walk along these test paths with different speeds (slow walking with an average speed of 3.24km/h, normal walking with an average speed of 4.58km/h and fast walking with an average speed of 5.8km/h). Test paths 1 to 4 were measured applying two types of turns, i.e., short turn and long turn. A short turn is a turn with a radius of 1 m, and a long turn is a turn with a radius of 2.5m. All test persons measured a total of 240 walks corresponding to a distance of approximately 5.6km. This section compares how fast the SOMDA and A-GPS can detect the movement direction and turns of a user.

The results of all walks show that the SOMDA detects complete turns on average 3.4s earlier than A-GPS. As an example Figure 4-12 shows the SOMDA and A-GPS data while the user/pedestrian is walking on a specific test path (Test path 1 and turning 90°). Based on our analysis, A-GPS showed the time delay in the movement direction data. The time delay is defined as the difference in timestamps between A-GPS and Compass using the SOMDA while determining changes in the movement direction data. Furthermore, it also observed that if the user/pedestrian starts walking after standing still, using A-GPS it takes 3s to 4s to detect the actual movement direction.

In contrast, using the SOMDA, it detects the orientation and movement direction of a user/pedestrian after an average time of 490 ms, if the user/pedestrian took two steps at any time before. By using A-GPS, it is not possible to detect a change in the user/pedestrian orientation when the user/pedestrian does not move forward but turns while standing at one point (test path 5). Figure 4-13 shows that when the user/pedestrian is standing at one point and turning clockwise, there is no change observed in the A-GPS data, this is due to the fact that the A-GPS works on the successive location points and, in this case, the location of the user/pedestrian does not change. In contrast, using the SOMDA, it detects such changes after an average time of 240 ms. The time to detect the user/pedestrian orientation and movement direction depends on how fast the user/pedestrian moves and on the type of turn (short turn or long turn).

Furthermore, we analysed the accuracy of a smartphone compass using the SOMDA compared to GPS reference points. We obtained GPS reference points for test paths 1 to 4 from the land registry Hessen, Germany. These ground GPS reference points are marked on the ground and can be seen in Figure 4-11. The average accuracy of the smartphone compass is shown in Figure 4-14. The X-axis shows the threshold value (in degrees) and the Y-axis shows the accuracy of the smartphone compass. Using the compass, the SOMDA achieves an accuracy of 98.8% at a threshold of 10°. From these results, we conclude that using the SOMDA we detect the movement direction of pedestrians 3.4s earlier than Assisted-GPS with an accuracy of upto 98.8%.

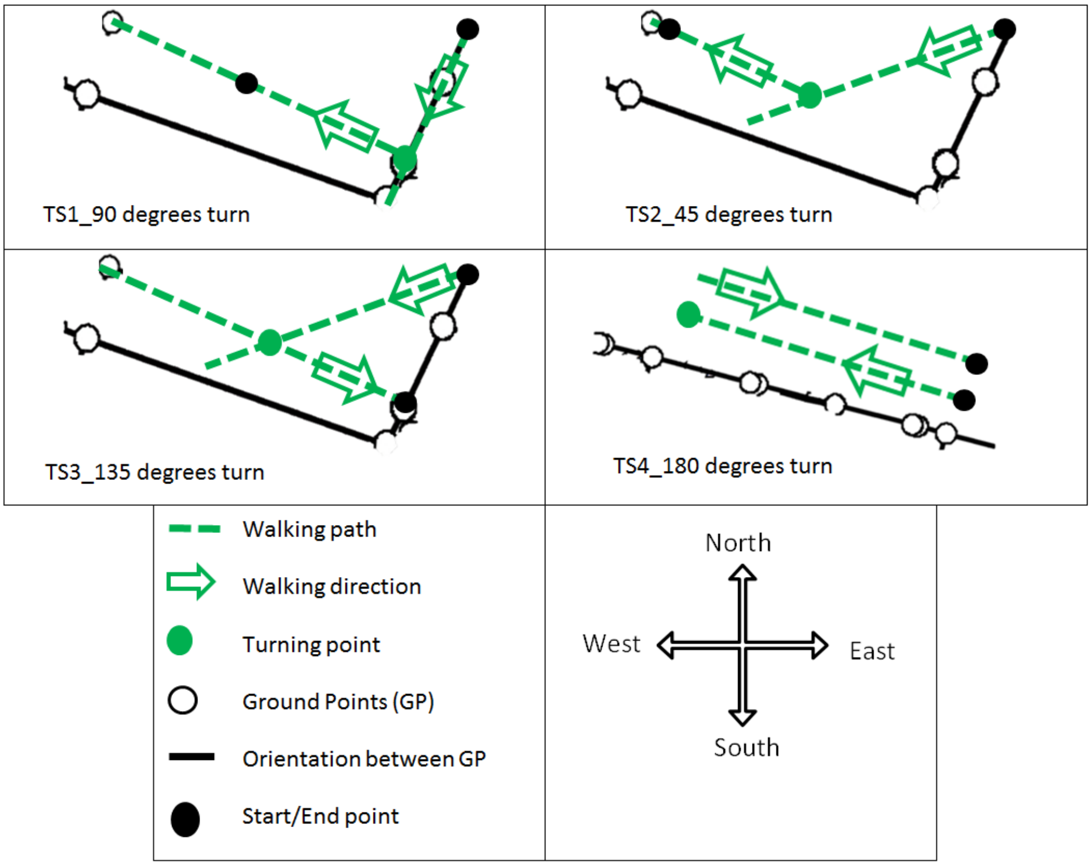


Figure 4-11 Test path 1-4

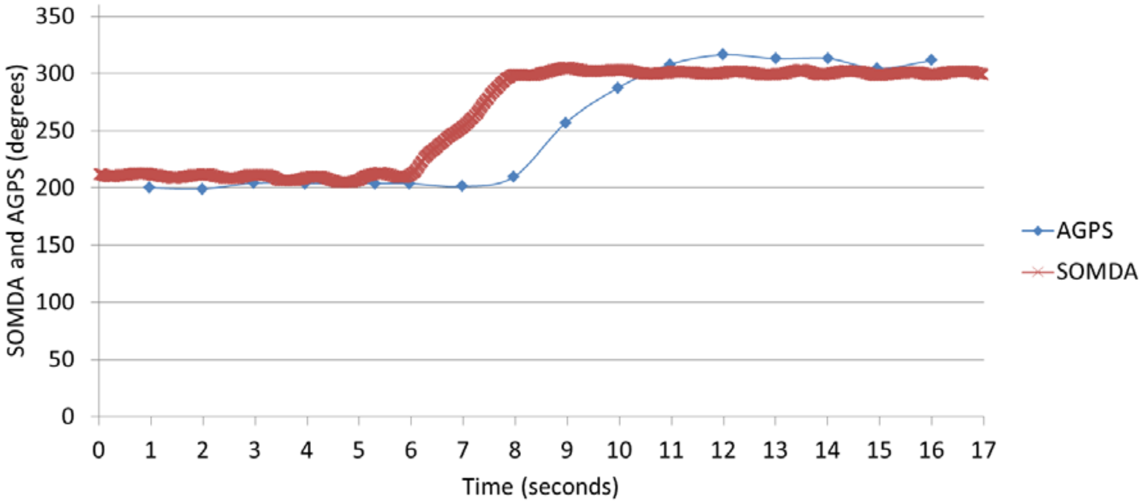


Figure 4-12 SOMDA and Assisted-GPS comparison while walking

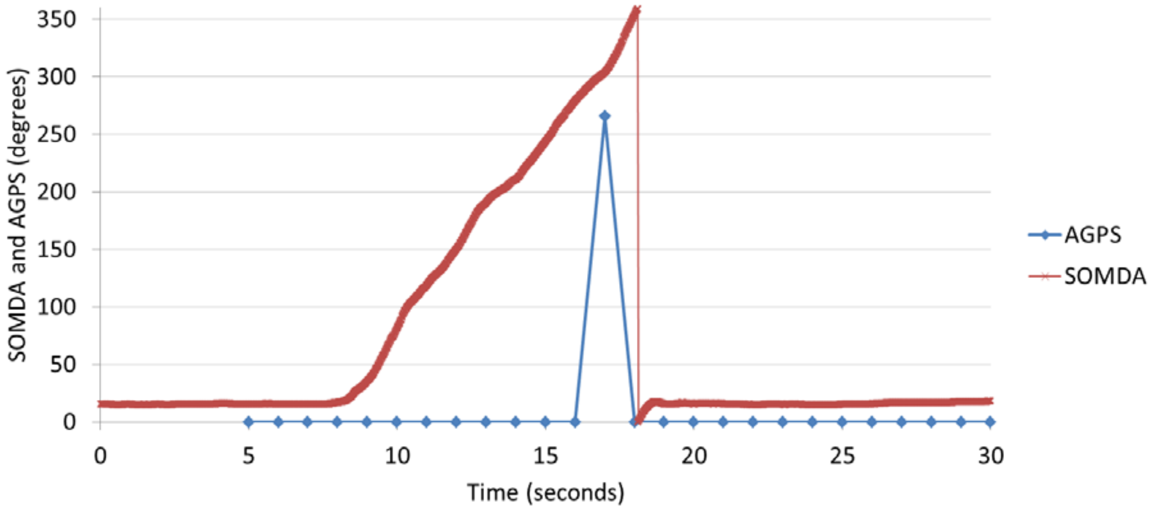


Figure 4-13 SOMDA and Assisted-GPS comparison, standing on one point and turning 0-360 degrees

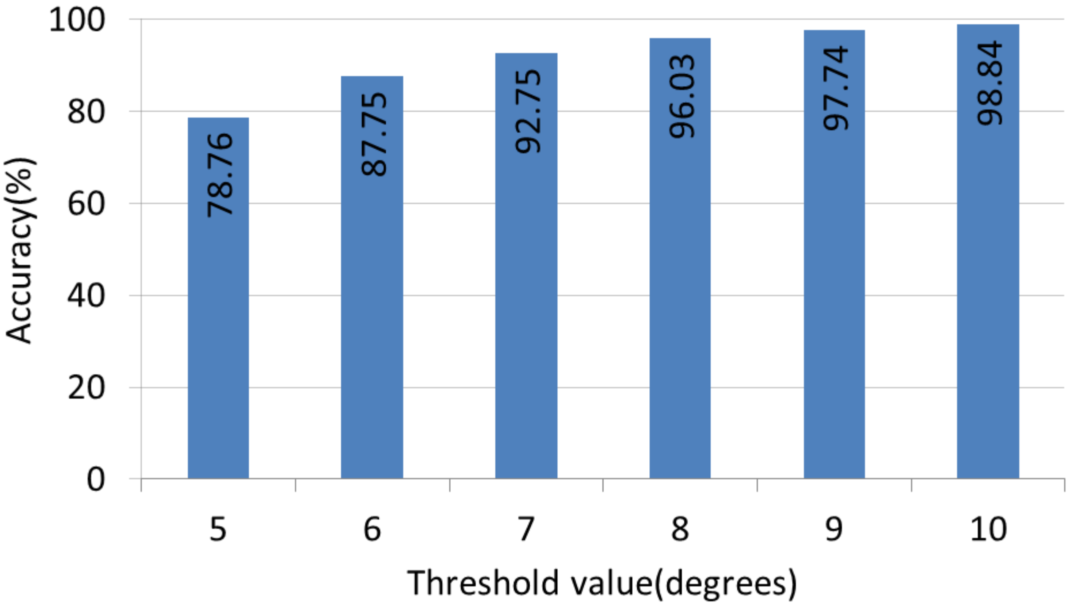


Figure 4-14 Accuracy of compass with GPS ground heading

4.4 Conclusion

In this chapter, we investigated the influence of the smartphone orientation measuring the compass azimuth on the movement direction of pedestrians. The results showed that to determine the correct movement direction the smartphone orientation must be the same as the movement direction. If this is not the case, the compass azimuth misleads the movement direction of pedestrians. To address this problem we present the SOMDA ("smartphone orientation and movement direction alignment") algorithm. In this approach, we detect the orientation and movement direction of users/pedestrians carrying smartphones inside their trouser pocket. First, the SOMDA detects the orientation of the smartphone's top using the compass and the orientation sensor. Second, the SOMDA detects the orientation of the smartphone's screen, and the user's movement direction by observing the compass and accelerometer during two steps the user/pedestrian takes. Thereafter, the smartphone orientation and the user/pedestrian orientation are continuously aligned. The SOMDA solves an issue we call the "smartphone in the pocket problem." The data collected with the smartphone is not necessarily aligned with the orientation of the smartphone user. Using the SOMDA, the user/pedestrian is free to change direction, movement speed, or to stop moving at all. The smartphone can be placed in the trouser pocket arbitrarily. In addition, the smartphone is free to wobble in the trouser pocket. Because SOMDA aligns the smartphone orientation and the user/pedestrian orientation, it leverages a wide range of applications that collect sensor data with smartphones. We exemplarily showed that the performance of SOMDA is feasible for a pedestrian's collision avoidance. The results show that the SOMDA detected the movement direction of a user/pedestrian with an accuracy of up to 96%. Moreover, the SOMDA detects complete turn on an average 3.4s earlier than A-GPS.

5 Investigation and Compensation of the Magnetic Deviation

The magnetometer of a smartphone is an attractive sensor suitable for the pedestrian movement direction detection. The movement direction detection can be used in many applications such as pedestrian safety, pedestrian dead reckoning and navigation. However, surrounding metallic materials have undesired effects, and this influences the accuracy of the magnetometer. In this chapter, we investigate the amount of magnetic deviation caused by surrounding metallic material, such as a car when the magnetometer sensor of a smartphone is used to determine the orientation. A set of experiments were conducted to investigate the effects of magnetic deviation. The experimental setup includes both parked and moving cars. Based on the results of the experiments, we show that the magnetic deviation can influence the efficiency of distinguishing the endangered or safe pedestrians in a car to pedestrian collision avoidance scenario. If we intend to use the magnetometer to detect the pedestrians' movement directions, the magnetic deviation needs to be compensated in order to detect the accurate movement direction of pedestrians. Therefore, in this chapter we present the compensation algorithm, which successfully compensates the magnetic deviation in the magnetometer of a smartphone.

5.1 Introduction:

Parts of the contents of this chapter are published in [7].

For centuries, the magnetic compass is used for navigation. In smartphones, the MEMS compass is used to provide functions similar to a conventional magnetic compass. The compass sensor is found in almost all smartphones. Today the compass is commonly used for pedestrians navigation [17] and [49], direction detection [19], and augmented reality [15] and [16]. The pedestrian movement direction detection is a context information which can be used in many applications such as the “pedestrians safety: by knowing the movement direction of pedestrians, collision between a car and pedestrians can be avoided”, and “indoor navigation, where the GPS reception is not available”. The compass sensor of a smartphone is used to determine the movement direction of pedestrians, as it is commonly available in almost all smartphones nowadays. One drawback of the compass of the smartphone is the existence of magnetic interference due to metallic or electric objects close-by. The magnetic interference causes the deviation in a compass of a smartphone, which influences the accuracy of any approach or system using compass. The magnetic deviation is an error, which deflects the actual magnetic north. In [121] a list of sources is presented that can cause magnetic deviation in a compass.

There are investigations that utilize different techniques to predict the direction of moving pedestrians using external or dedicated compass sensors [29] and [122]. In [123], external magnetometer sensors are installed on the roadside for the detection of vehicles. Vehicles are detected through the disturbances in the earth’s magnetic fields, which are caused by a vehicle itself. The installation of magnetometer sensors along roads is neither practical nor cost-effective.

There are many approaches available in the state of the art, which use the compass to provide the movement direction of pedestrians. To the best of our knowledge, there is no investigation publically available, which studies the influence of the surrounding metallic materials on a compass while detecting the movement direction of pedestrians.

The objective of this chapter is to investigate the amount of magnetic deviation on a compass of a smartphone, caused by surrounding metallic materials. The focus of this investigation is to study the influence of nearby cars on a compass. The filter is designed that is able to recognise the safe and endangered pedestrians based on their movement direction. It considers the pedestrians moving on the sidewalks as safe and “filtered out” if they are not moving towards the road [12]. This chapter have shown that the magnetic deviation on a compass can influence the movement direction detection of pedestrians and the efficiency of distinguishing the endangered or safe pedestrians. Therefore, the algorithm is proposed in order to compensate for the magnetic deviation. The compensation algorithm is based on the compass and gyroscope of a smartphone.

The chapter is organized as follows: In Section 5.2, the design and purpose of the experiments is presented. In Section 5.3, a physical analysis is presented about how magnetic deviation influences the efficiency of the filtering pedestrians. Section 5.4 presents the algorithm for compensating magnetic deviation on a compass. In section 5.5, the respective results are discussed. Finally, conclusion is presented in Section 5.6.

5.2 Investigation of the magnetic deviation:

An open source “phone compass application”, which is developed by Nokia, is modified to record the data from the compass of a smartphone type Nokia N97. The software records the data at a maximum sampling rate of 9Hz. A Nokia N97 has a built-in magnetometer with the chipset AK8974 from Asahi Kasei Microdevices (AKM) Corporation. According to the manufacturer, the accuracy of magnetometer chipset AK8974 in an ideal case is ±1º. All the experiments are taken with the best calibration level of a compass as suggested by the Nokia Sensor API. A set of experiments (E1-E4) is designed to observe the amount of magnetic deviation in a magnetometer, caused by surrounding metallic materials such as a car in our case. All experiments are conducted at an empty parking lot area. During the experiments, two different cars are used. The Volkswagen Golf III (VW) with the weight of 1360 kg and the BMW-530d Touring (BMW) with the weight of 1835 kg are used to cause the magnetic deviation in the compass.

During the experiments, we parked or moved the car at different distances from the smartphone. Distances are marked on the ground, and the driver is instructed to drive the car along these markings. For the conducted experiments presented in this chapter, the smartphone is placed on a cardboard box at trouser pocket height, i.e., 1m above the ground. All the experiments are repeated thrice in order to obtain the average values for the intended evaluations. The results from measurements slightly vary for each repetition. The results of experiments (E1-E4) are accompanied by an error bar, which shows the maximum and minimum magnetic deviation observed during the measurements. Below are the descriptions and the corresponding results of each experiment carried out within the scope of this chapter.

5.2.1 Experiment 1 (E1): Magnetic deviation due to the car at different distances

In E1, the smartphone is placed parallel on a box. The car is parked at different distances: 10, 8, 6, 4, 2, 1, and 0.5m away from the smartphone, as shown in Figure 5-1. E1 intends to identify what amount of magnetic deviation is observed when each car is parked singly at different distances. The smartphone and the car both are stationary, and the engine of the car is kept running.

The results of E1 are shown in Figure 5-2. The results show that different type of cars cause different amount of magnetic deviation to the compass. Furthermore, the BMW caused slightly higher amounts of magnetic deviation, i.e., 1° compared to the VW, when both cars are located separately at a distance of 4 meters or less. This higher magnetic deviation of 1° is because the more metallic material is used in the BMW model in comparison to the VW model. As the distance between the cars and the smartphone is increased from 4m to 10m, the deviation almost remains identical for both cars.

The VW causes a magnetic deviation of 9.5°; the BMW causes 11° deviation on the compass of the smartphone when both cars are parked separately at a minimum distance of 0.5m from the smartphone. When the car is parked at a distance of 10m away from the smartphone, a magnetic deviation of 2° is observed. If the distance between the compass of a smartphone and the parked cars is decreased below the distance of 0.5m, the performance and accuracy of the sensor is affected up to 30°. From the results of E1, it is observed that if the distance between the compass sensor and the car is decreased, the magnetic deviation will increase and vice versa. At a distance of less than 2 m, the increment of the magnetic deviation is higher (=> 2° per meter).

5.2.2 Experiment 2 (E2): Magnetic deviation due to the orientation of a car

In E2, the smartphone is placed parallel on a box at pocket height. The smartphone and the car are static during the experimental setup. Cars are parked singly at a distance of 2m. The engine of the car is kept running. The purpose of conducting this experiment is to observe whether or not the parallel or perpendicular orientation of the car is causing the same or a different amount of magnetic deviation on the compass. We further subdivided the E2 into two cases. In Case 1 the car is parked parallel to the smartphone at a distance of 2m as shown in Figure 5-3(a) and in Case 2 perpendicular as shown in Figure 5-3(b).

The results of the E2 are presented in Figure 5-4. The magnetic deviation caused by the VW is 6° while the BMW causes 7° magnetic deviation. From E2, we observed that there is no much difference between case 1 and case 2. This means the orientation of the car to the smartphone does not influence the amount of magnetic deviation. E2 shows again that different cars cause different magnetic deviations.

5.2.3 Experiment 3 (E3): Magnetic deviation due to the orientation of the smartphone

In experiment E3, we investigated the amount of magnetic deviation caused by a moving car. The experimental setup addresses possible differences on the magnetic deviation caused by the orientation of a smartphone. Two orientations are considered: parallel and perpendicular to the surface of the earth. The smartphone is placed on a box, and the car passes at a distance of 2 m, as shown in Figure 5-5. The results are presented in Figure 5-6. A difference of 1° deviation is observed when the device is parallel to the surface of earth as compared to the device placed perpendicular to the surface of the earth.

Also, we have repeated E3 with the smartphone placed on the ground. This experimental setup is used to investigate the influence of the magnetic deviation on a magnetometer at different heights. The results of this part of the experiment are similar to E3. Therefore, we concluded that the observed amount of magnetic deviation is same regardless of the height while the orientation changes of a smartphone, i.e., parallel or perpendicular can cause a difference of 1°.

5.2.4 Experiment 4 (E4): Magnetic deviation caused by a moving vehicle

In E4, we placed the smartphone parallel on a box at pocket height. The car passes the smartphone at a distance of 1, 2 and 3m respectively; the similar setup is shown in Figure 5-5. The speed of the car is 15-20km/h. The purpose of E4 is to investigate the amount of magnetic deviation caused by a moving car on the static compass sensor of the Nokia N97. The results of E4 are presented in Figure 5-7, which show that the VW causes the constant magnetic deviation of 2° for different distances. The BMW causes almost the same amount of deviation while passing at different distances greater than 1m. The amount of magnetic deviation shown in Figure 5-7 depends upon how fast the sensor is updating its values when the car is moving closer to the compass of a smartphone.

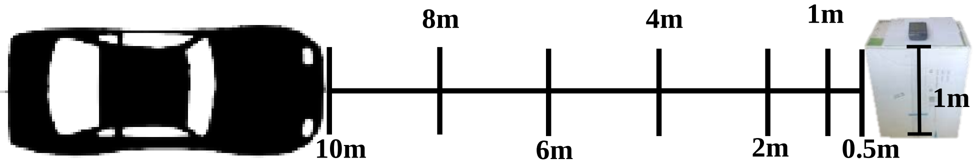


Figure 5-1 E1: Car is parked at different distances from the stationary smartphone [7]

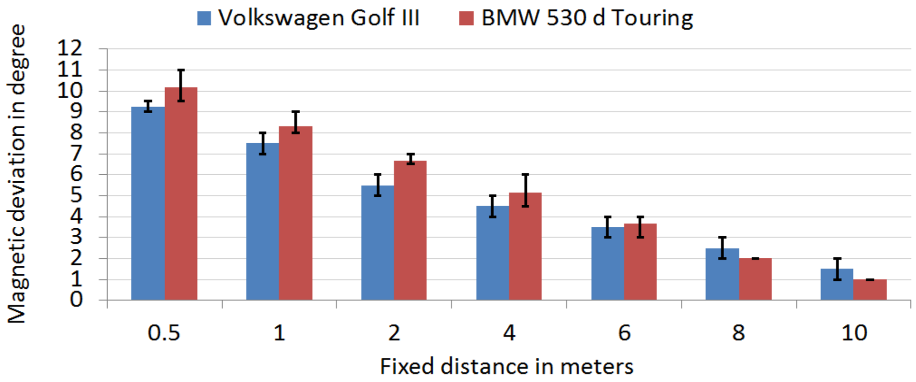


Figure 5-2 E1: Magnetic deviation caused by different cars at different distances [7]

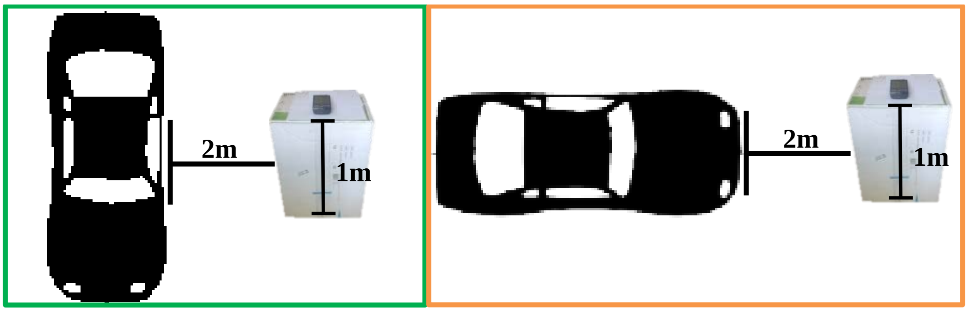


Figure 5-3 E2: Car is parked parallel & perpendicular to the smartphone [7]

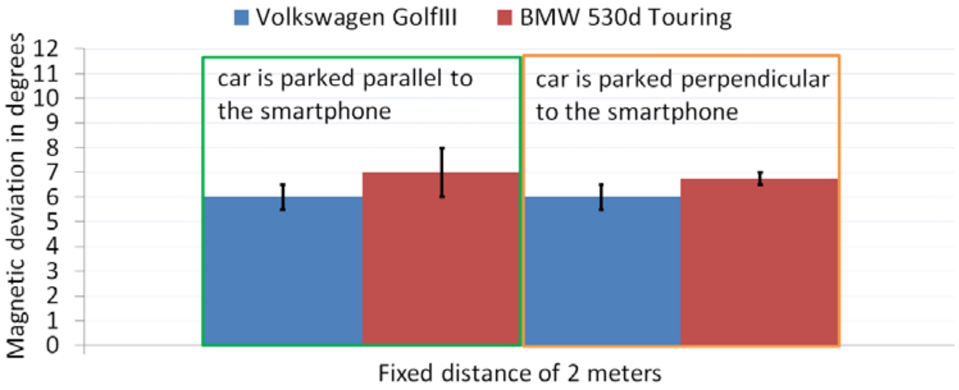


Figure 5-4 E2: Magnetic deviation caused by parallel and perpendicular position [7]

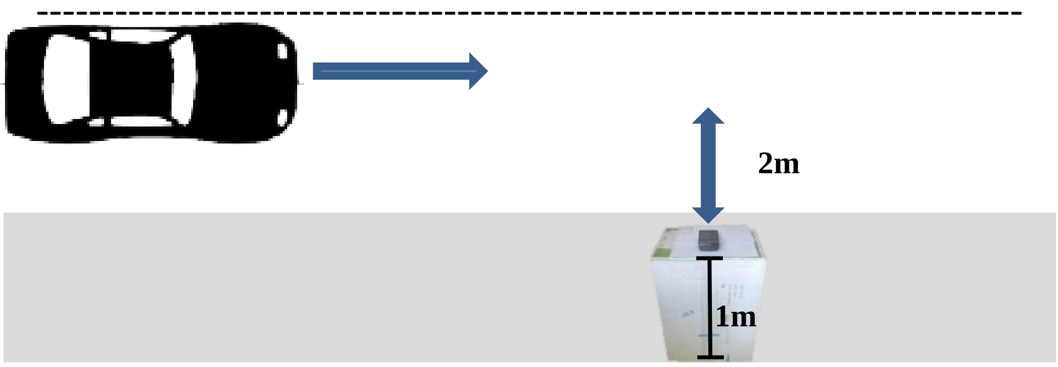


Figure 5-5 E3: Car passes the smartphone at a distance of 2m during the different orientation of the smartphone [7]

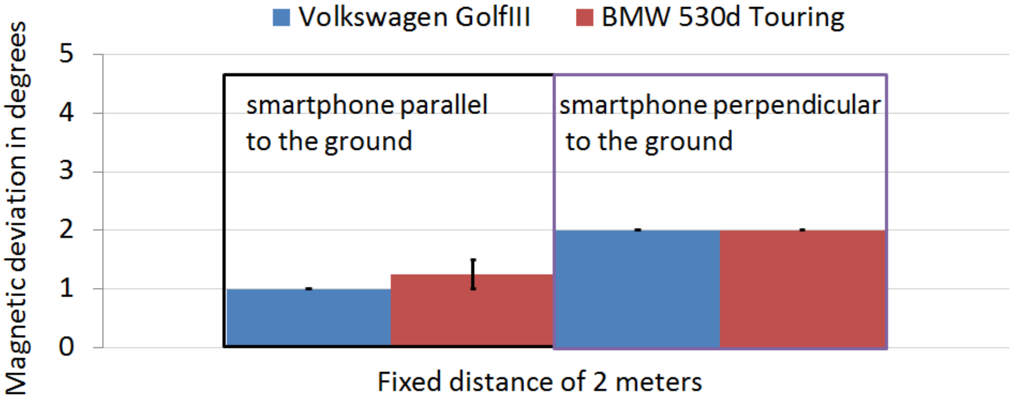


Figure 5-6 E3: Magnetic deviation caused by a moving car at different orientation of the smartphone [7]

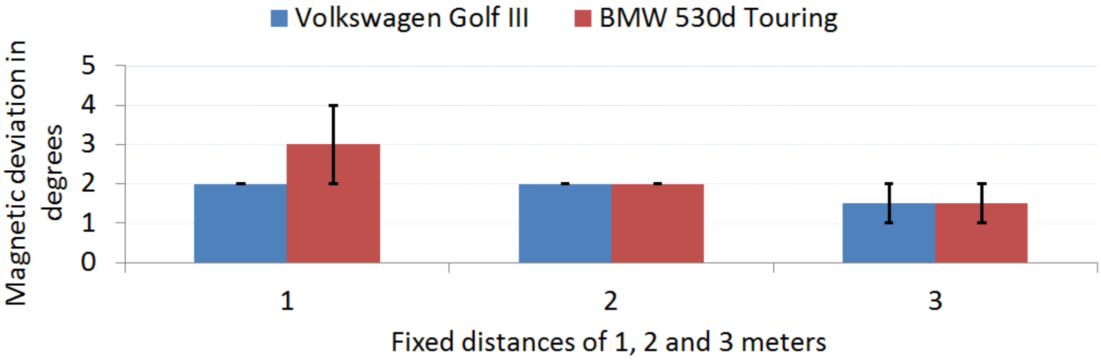


Figure 5-7 E4: Magnetic deviation caused by a moving car at different distances [7]

5.3 Efficiency of filtering pedestrians

In the experiments, we showed that when the car is close to the smartphone, compass observed the deviation due to the metallic components of the car, which influence the performance of compass while detecting the pedestrian direction. At a distance of 10m and 0.5m, a smartphone compass observed the deviation of 2° and 10° respectively. In these cases, the existence of magnetic deviation of 2° or 10° deflects the compass azimuth by 2° and 10° while determining the movement direction of pedestrians. Therefore, the presence of magnetic deviation in the compass of a smartphone reflects that the pedestrian is not moving in the direction which compass is showing.

In this regard, we designed a filter based on the simulated data, which is able to differentiate between the safe and endangered pedestrians based on their movement direction with and without the availability of magnetic deviation in a compass of a smartphone. The filter monitors the walking direction of pedestrians and considers pedestrians are only safe if they are moving parallel, and away from the road, the other pedestrians such as moving towards the road are considered as being endangered. Pedestrians moving parallel and away from the road on sidewalks are being "filtered out" and considered as safe.

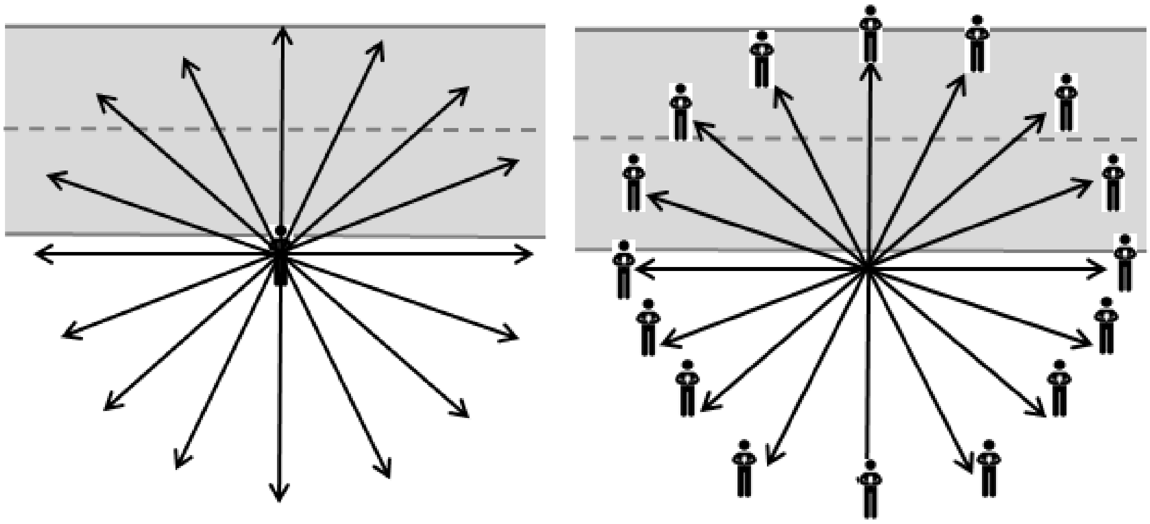
Filter efficiency is calculated in percentage (%), which describes the correctness of the filter, i.e., how many pedestrians are filtered out, e.g., considered as safe. Filter efficiency is calculated using equation (5), where *"Tp"* is the total pedestrians walking on sidewalk and *"TCp"* is the correctly identified pedestrians moving parallel and away from the road.

$$Accuracy = \left(\frac{TC_p}{T_p} \right) * 100 \tag{5}$$

Let us assume that the movement of pedestrians on the sidewalk is equally distributed in all directions in angle (0-360°) as shown in Figure 5-8 (a) and Figure 5-8 (b). Let us consider if the observed magnetic deviation is 0°, the filter considers 50% of all pedestrians as safe due to their movement direction, i.e., walking parallel and away from the road as shown in marked black half circle (see Figure 5-9 (a)). Furthermore, 50% pedestrians are considered as being endangered due to their movement direction, i.e., walking towards the road as shown in red half circle (see Figure 5-9 (b)).

Based on the experiments (E1-E4), the magnetic deviation in a compass can influence the efficiency of distinguishing the endangered or safe pedestrians. In Figure 5-11, we showed the efficiency of the filtering pedestrians, where the X-axis represents the magnetic deviation in degrees and the Y-axis represents the efficiency of filter, i.e., safe pedestrians. Let us assume that 360 pedestrians moving on sidewalk and equally distributed in all direction ranges 0-360° (each pedestrians movement direction is separated by 1°) and compass azimuth is correctly represents the pedestrian's movement direction. As shown in Figure 5-11, when there is no magnetic deviation, i.e., 0°, filter determines that 180 pedestrians out of 360 pedestrians i.e., 50% are moving away from the road and considered as safe. Similarly, when the compass observed the magnetic deviation, i.e., even 1°, this time filter determines that 178 pedestrians out of 360 pedestrians i.e., 49.44% are moving away from the road and considered as safe. In this case, the efficiency of filtering pedestrians decreases because pedestrians moving 0-1° parallel or away from the road are considered as moving towards or parallel to the road due to the magnetic deviation and deflection of 1° in the compass while determining the pedestrian direction. In Figure 5-11, we analysed the similar trend for all magnetic deviation, i.e., if the observed amount of magnetic deviation increases, the efficiency of filtering pedestrians (safe pedestrians) decreases. Moreover, when the compass observed the magnetic deviation of 11°, this time filter determines that 178 pedestrians out of 360 pedestrians i.e., 43.89% are moving away from the road and considered as safe. In this case, the efficiency of filtering pedestrians decreases because pedestrians moving 0-11° away from the road are considered as moving towards the road due to the magnetic deviation and deflection of 11° in the compass while determining the pedestrian direction. In contrasts to Figure 5-9, in Figure 5-10 we graphically showed that with the presence of magnetic deviation in a compass, the pedestrians moving 0-11° parallel or away from the road are considered as moving towards the road and as being endangered. In result, the percentage of filtering out the safe pedestrians reduces and endangered pedestrians increase due to the observed magnetic deviation of 11°.

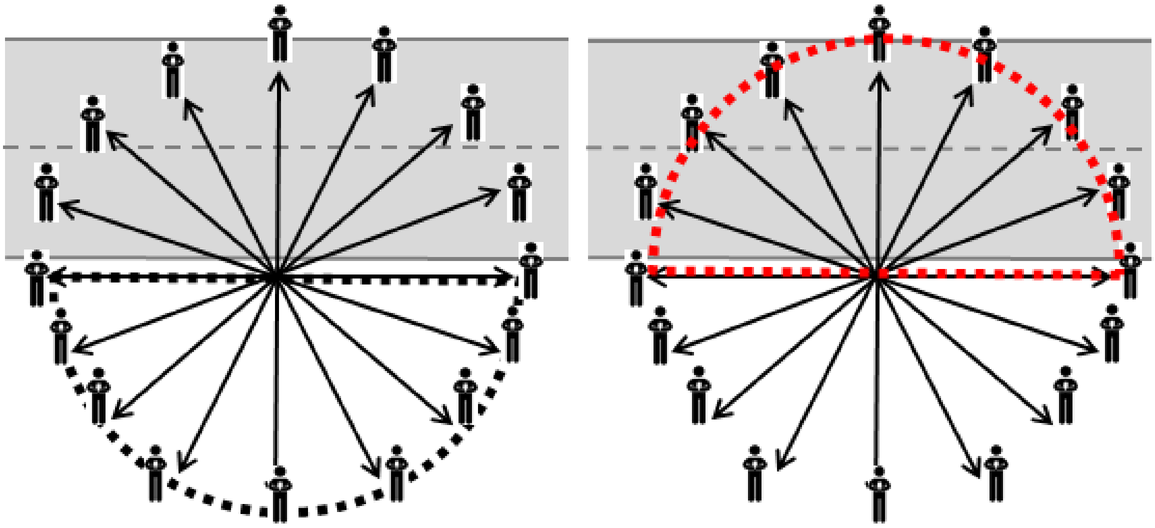
In actual pedestrians are moving away from the road but due to the magnetic deviation influence the performance of the filter and consider pedestrians as not safe due to the incorrect movement direction obtained using compass of a smartphone. Therefore, if one intends to use the magnetometer sensor in order to detect the movement direction of a pedestrian, it would be helpful if the magnetic deviation could be compensated to identify the accurate detection of moving pedestrians.



(a) Pedestrians on sidewalk

(b) Pedestrians equally distributed in all directions

Figure 5-8 Movement direction of pedestrians (equally distributed in all directions)



(a) 50% pedestrians moving parallel or away from the road are considered as safe

(b) 50% pedestrians moving towards the road are considered as endangered

Figure 5-9 Safe and endangered pedestrians without magnetic deviation

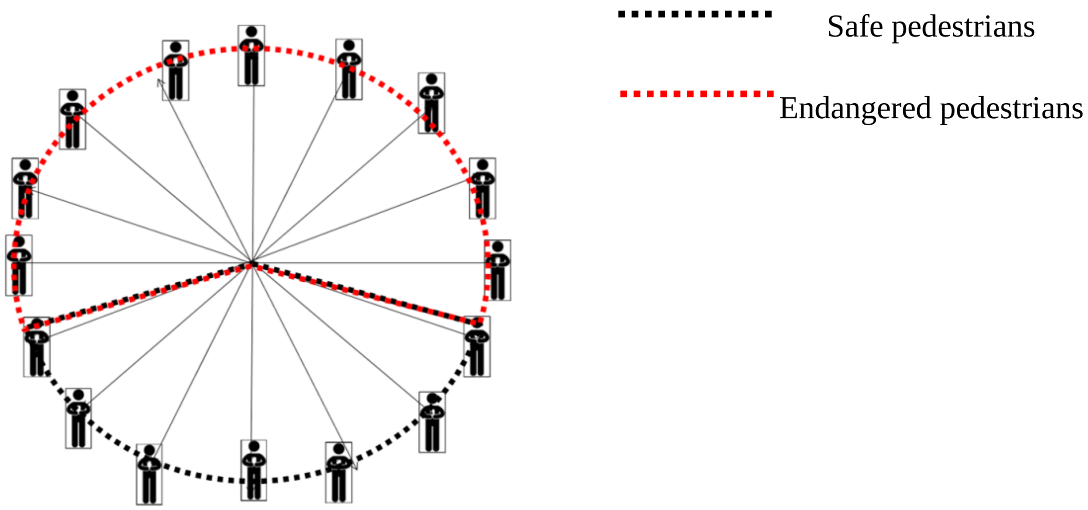


Figure 5-10 Safe and endangered pedestrians with magnetic deviation

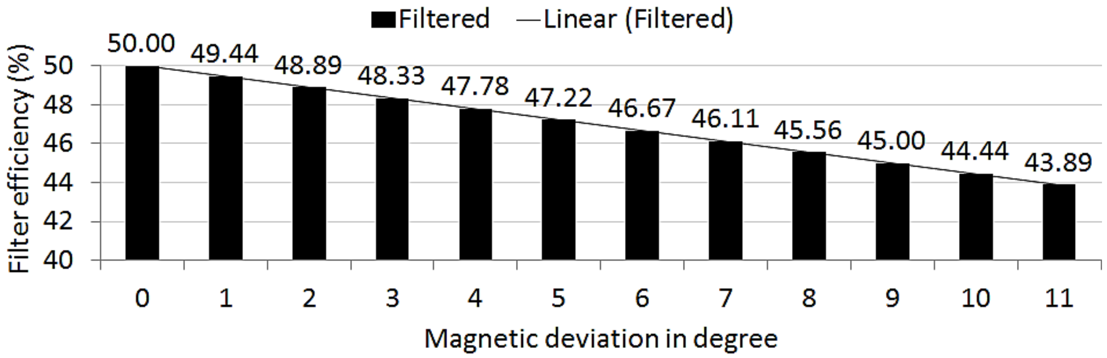


Figure 5-11 Efficiency of all filtering pedestrians [7]

5.4 Compensation algorithm for magnetic deviation

This section presents the Magnetic Deviation Compensation Algorithm (MDCA) in the compass of a smartphone. In order to compensate for the magnetic deviation, one option is to use the gyroscope sensor of a smartphone. The purpose of using the gyroscope sensor is to differentiate between the rate of turn and the magnetic deviation. The rate of turn is the angle of rotation of an object in rad/s. We used the gyroscope of a smartphone to determine the rate of turn. Moreover, the magnetic deviation is an error in measured compass data, which is caused by any magnetic and electric material to deflect the actual magnetic north.

During the static case if the change in the compass heading is beyond the threshold value, i.e., 0.5° and simultaneously there is no turn observed from the gyroscope data, the change in the compass heading is considered as the magnetic deviation and compensated accordingly as shown in the Figure 5-12. MDCA is the process of removing the magnetic deviation (error) from measured compass data. During the experiments (E1-E4) a Nokia N97 is used in order to investigate the amount of magnetic deviation. The proposed MDCA in a compass of a smartphone is based on the gyroscope sensor. Due to the unavailability of a gyroscope in a Nokia N97, we used the Samsung Galaxy S2 and S3 for magnetic deviation compensation algorithm. All these devices, Nokia N97, S2 and S3, have the same series of embedded magnetometer chipset, i.e., AK897x. Therefore, it is assumed that a similar magnetic deviation is observed in the aforementioned models of the smartphone.

A new experiment is designed and conducted to support the compensation algorithm, which is shown in Figure 5-12. A Samsung Galaxy S2 is placed on a table. The compass and the Gyroscope sensor data are recorded. From the results, it is observed that there is no change in the compass and the gyroscope sensor of a smartphone for some time because of the smartphone is placed static on the table. To cause the magnetic deviation in the compass of a smartphone, a magnet is used artificially. As soon as the magnet gets closer to the smartphone, a change is observed in the compass of the smartphone. At the same time, no change is observed in the gyroscope data as shown in Figure 5-13. This change in the data of the compass sensor of a smartphone is considered as the magnetic deviation.

In order to understand the working method of the MDCA, we divided the MDCA in 7 steps, which are described below. Furthermore, the flow of the data in the MDCA is shown in Figure 5-12.

1. For the first time, MDCA initialises the compass azimuth data (ψ_t), set the fixed window size of the data to 1 second to compute the average values of compass azimuth.

2. As soon as new compass azimuth data sample arrives (ψ_{t+n}, where n= 1,2,3…), Compute the difference (Δ_{t+1}) of the compass heading at timestamps t and t+n as shown in equation (6).

$$\Delta_{t+n} = \psi_{t+n} - \psi_t \tag{6}$$

3. If the difference (Δ_{t+n}) of compass heading at timestamp t and t+n is under the threshold value (ψ_{Tv}), i.e., 0.03°. In that case, compute step 3 and compass heading at timestamp t+1 (ψ_{t+n}) require no compensation and already corrected as shown in equation (7) and jump to step 6. else compute step 4

$$\psi_{t+n\_compensated} = \psi_{t+n} \qquad \left| -\psi_{Tv} < \Delta_{t+n} < +\psi_{Tv} \right. \tag{7}$$

4. Check for the gyroscope value (ω_{t+n}). If the gyroscope value is under the gyroscope threshold value (ω_{Tv}), it means there is no turn observed in gyroscope data, which shows the change in the compass azimuth in step 3 is the magnetic deviation caused by surrounding metallic/electric materials. In order to compensate for the magnetic deviation compute step 4 and after that jump to step 6. Else, compute step 5 if the gyroscope data is not under the gyroscope threshold value. For magnetic deviation compensation, the already calculated difference between successive compass heading is subtracted from the current compass heading to remove the observed magnetic deviation as shown in the equation (8).

$$\psi_{t+n\_compensated} = \psi_{t+n} - \Delta_{t+n} \qquad \left| -\omega_{Tv} < \omega_{t+n} < +\omega_{Tv} \right. \tag{8}$$

5. If the gyroscope data is not under the threshold value, it means a turn is observed in the gyroscope data. Moreover, the change in the compass heading is considered as the turn and the compass heading at time t+n do not require corrections and MDCA consider the compass heading is already correct as shown in equation (9).

6

$$\psi_{t+n\_compensated} = \psi_{t+n} \qquad \left| -\omega_{Tv} > \omega_{t+n} > +\omega_{Tv} \right. \tag{9}$$

7. Compass heading at timestamp t+n is compensated.

8. Repeat step 2-6.

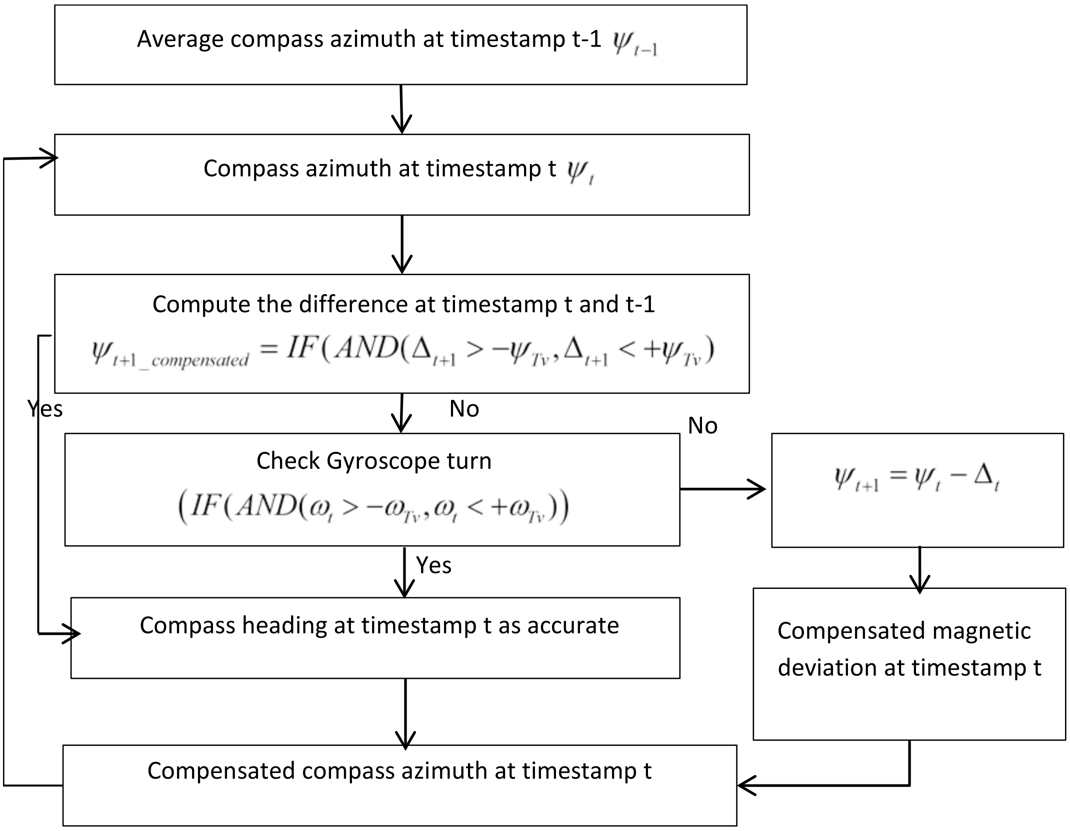

Average compass azimuth at timestamp t-1 ψ_{t-1}
Compass azimuth at timestamp t ψ_t
Compute the difference at timestamp t and t-1
$\psi_{t+1\_compensated} = IF(AND(\Delta_{t+1} > -\psi_{Tv}, \Delta_{t+1} < +\psi_{Tv})$
Yes
No
Check Gyroscope turn
$(IF(AND(\omega_t > -\omega_{Tv}, \omega_t < +\omega_{Tv}))$
No
$\psi_{t+1} = \psi_t - \Delta_t$
Yes
Compass heading at timestamp t as accurate
Compensated magnetic deviation at timestamp t
Compensated compass azimuth at timestamp t

Figure 5-12 Compensation algorithm for magnetic deviation [7]

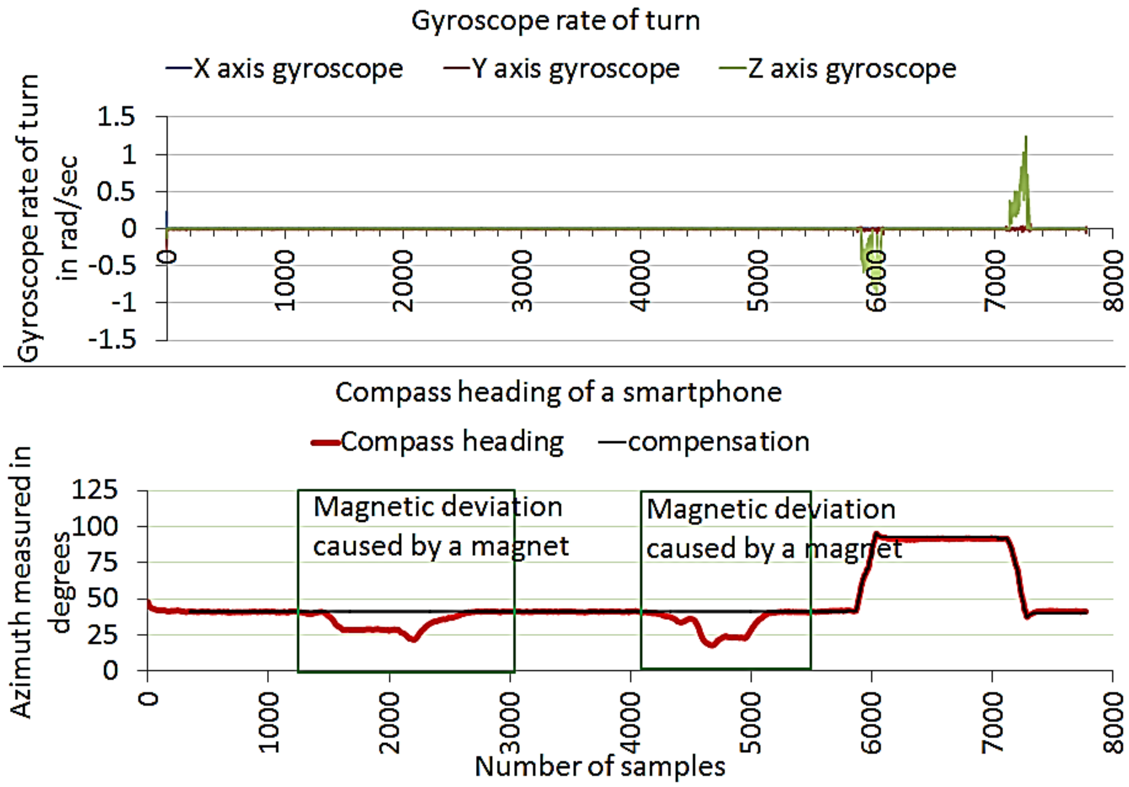


Figure 5-13 Compensation of magnetic deviation [7]

5.5 Results and Discussion

Let us have a look at the compass data regarding of Probability Distribution. Probability Distribution provides the probability of each sample of the data set. There are many distributions available, which tells how the data is distributed and what is the probability of each outcome. There are two ways to specify the Probability Distribution.

1. Probability density function (pdf)
2. Cumulative distribution function (cdf)

If the probability distribution of the data set is known, one can calculate the “pdf” and “cdf”. Statisticians defined pdf as the individual probability of all measurable subsets or ranges of subsets and cdf is an integral part of the pdf and the summation of all the successive probabilities. In this dissertation, we used the “Matlab distribution fitting tool” to analyse which probability distribution fits our data set. We analysed that “t-location-scale” probability distribution is closely fit our collected data set. The selection of probability distribution is based on the calculated standard error between measured and particular selected probability distribution. The lower standard error shows the probability distribution is closely fit the measured data. We analysed that t-location-scale" probability distribution closely fit our measured and compensated data. Based on the t-location-scale probability distribution we calculated the pdf and cdf of the measured and compensated data. The purpose is to show how the measured and compensated data is distributed.

In Figure 5-14, we showed the probability density function (pdf) of the collected data set, i.e., compass azimuth measured in degrees before the compensation of the magnetic deviation (we called it as noisy sensor data). The horizontal axis, i.e., the X-axis represents the compass azimuth in degrees and vertical axis, i.e., the Y-axis shows the probability density function of the X-axis. Each value of noisy sensor data on the X-axis has a specific probability density function on the Y-axis. Higher pdf on the Y-axis for a specific range of data on the X-axis represents the frequent occurrence of the measured data values. In this case, the data range of 41-43° has higher pdf, i.e., range from 0.2 to 0.35 for each data. We confirmed from the ground truth that the data range of 40-43° with higher pdf of 0.2 to 0.35 is matching the ground truth. Moreover, from our experiments, we know that the smartphone was placed statically on the table in a specific direction and the artificially magnetic deviation was produced using a magnet.

The other method of representing the data set is using cumulative distribution function (cdf). In Figure 5-15, we showed the cdf of the noisy sensor data. The horizontal axis, i.e., the X-axis represents the compass azimuth in degrees and vertical axis, i.e., the Y-axis shows the cdf of the compass azimuth. In contrasts to pdf, cdf provides the cumulative probabilities of the data set, i.e., the summation of all the consecutive probabilities of the data in the set. In Figure 5-15, we showed the whole data set including actual and noisy sensor data have a cdf of 1. Moreover, from the ground truth, we know that the actual data lies in the range of 41-43°, which we can analyse that this range of data has the cdf of approximately 0.70 or 70%.

From our experiments, we know that the smartphone was placed statically on the table in a specific direction and the artificially magnetic deviation was produced using a magnet. From measurements, we also know the ground truth, i.e., ranges between 41-43°. Moreover, we calculated the error between the measured data (noisy sensor data) and the reference measurement. In Figure 5-16, we showed the pdf of the error in the measured data (noisy sensor data), where the horizontal axis, i.e., the X-axis represents the error in degrees and the vertical axis, i.e., the Y-axis shows the pdf of the error. In Figure 5-16, we observed the error from value 1-25°, which we produced artificially using a magnet in a compass of a smartphone placed statically on a table. Each error value has the pdf value that represents their likelihood. Similarly, in Figure 5-17 we showed the cdf of the calculated error, which showed the 55% of the error, lies in the data range of 0-1° and 45% of the error lies in the data range from 1-25°.

Therefore, we proposed the algorithm, which compensates for the magnetic deviation in order to detect the accurate movement direction of pedestrians carrying a smartphone. We applied our proposed magnetic deviation compensation algorithm on the same data set, i.e. noisy sensor data. We presented the compensated results in Figure 5-13. The compensation algorithm subtracts the possible amount of the magnetic deviation from the compass sensor to have more precise direction detection of pedestrians.

Let us have a look at the compensated compass data regarding probability distribution. In Figure 5-18, we showed the pdf of the compensated compass data. The horizontal axis, i.e., the X-axis represents the compass azimuth in degrees for the data set (we called it as compensated compass data) and the vertical axis, i.e., the Y-axis shows the pdf of the compass azimuth (i.e., compensated compass data). Each value of compensated compass data on the X-axis has a specific pdf on the Y-axis. Higher pdf for a specific range of data on the X-axis represents the frequent occurrence of the measured data values. As the compensation algorithm correctly removed the observed magnetic deviation and the compensated data accurately matches with the reference data, i.e., all the compensated data lies in the range of 41-43°.

The other method of representing the compensated compass data is using cdf. In Figure 5-19, we showed the cdf of the compass azimuth (compensated compass data). The horizontal axis, i.e., the X-axis represents the compass azimuth in degrees after compensation and the vertical axis, i.e., the Y-axis shows the cdf of the compass azimuth (compensated data). In contrasts to pdf, cdf provides the cumulative probabilities of the data set, i.e., the summation of all the consecutive probabilities of the data in the set. In Figure 5-19, we showed the cdf of the compensated compass data, which is aligned with the reference measurement and shows the cdf of 0.99 or 99% of data ranges from 41-43°.

Moreover, we calculated the error between the compensated compass data and the reference measurement. In Figure 5-20, we showed the error of the compensated compass sensor data; where the X-axis shows the calculated error in degrees and the Y-axis show the pdf of the calculated error. In contrast to Figure 5-16, Figure 5-20 observed there is no error from value ranges 1-25°, which we compensate using the magnetic deviation compensation algorithm. Furthermore, in Figure 5-21 we presented the cdf of the calculated error, which shows 99% cumulative probability of 0-2° of error in the compensated compass data compared to 99% cumulative probability of 0-25° of error in the noisy compass data (see Figure 5-17).

In Figure 5-22 to Figure 5-23, we presented the comparison of 2 data set, i.e., noisy compass sensor data and compensated compass data regarding the measured and error data. In Figure 5-22, we showed the cdf comparison of the noisy compass sensor data and compensated compass sensor data. The horizontal axis, i.e., the X-axis represents the compass azimuth in degrees and the vertical axis, i.e., the Y-axis shows the cdf of the compass azimuth. As we know, the reference data lies in the range of 41-43°. We analysed that the noisy compass sensor data contains the magnetic deviation from 22-40° with cdf of around 0.35 or 35%, which is successfully compensated using the magnetic deviation compensation algorithm. In the compensated compass the data ranges from 41-43° contains the 99% of the data.

Furthermore, we compared the error of the noisy sensor data and compensated compass sensor data. In Figure 5-23, we showed the calculated error in the noisy and compensated compass sensor data on the X-axis while the Y-axis shows the cdf of the calculated error. We observed that in the noisy sensor data has an error of 0-24° with the cumulative probability of 99% while in the compensated compass sensor data there is only 0-2° error available with a cumulative probability of 99%.

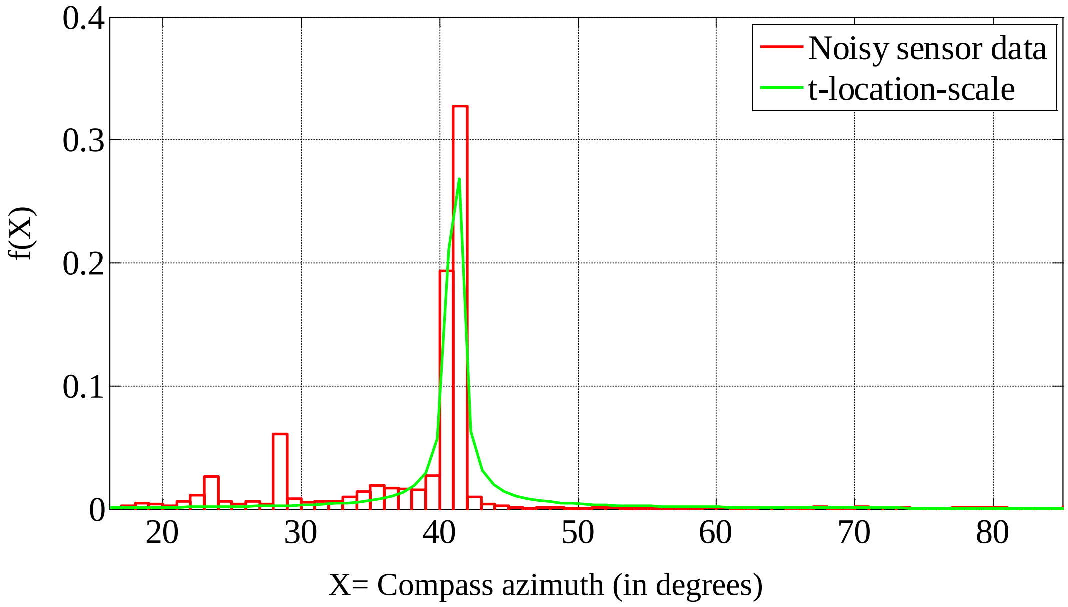


Figure 5-14 pdf of noisy sensor data and t-location-scale distribution

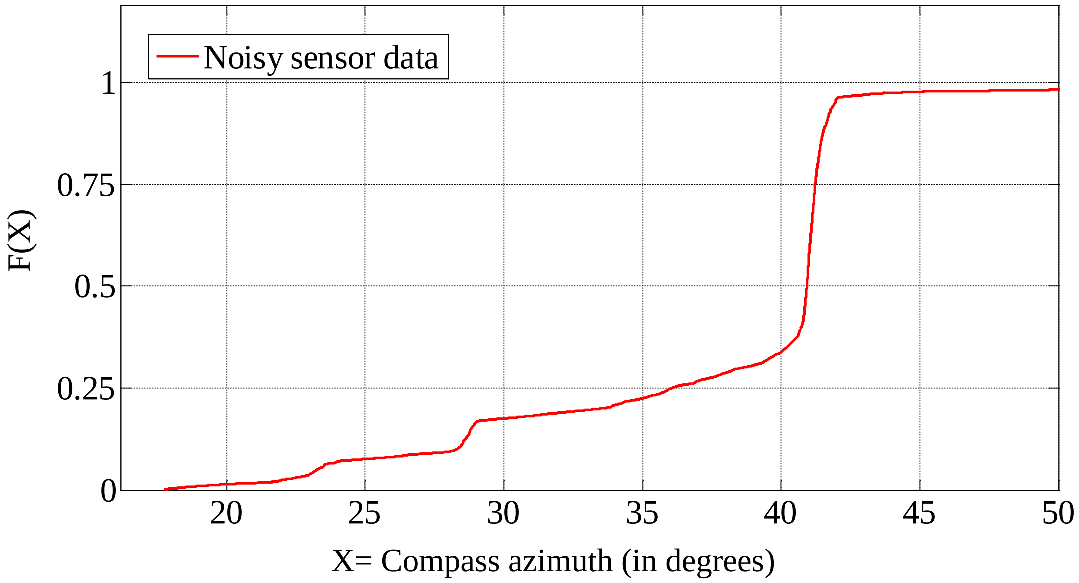


Figure 5-15 cdf of noisy sensor data

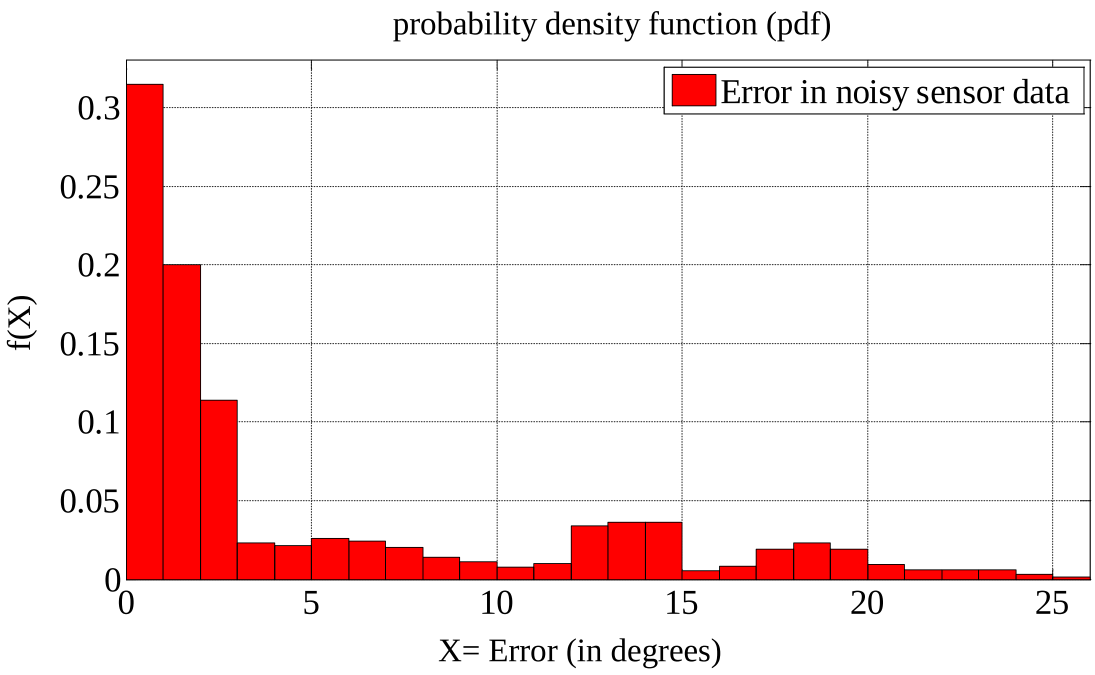


Figure 5-16 pdf of error calculated in noisy sensor data

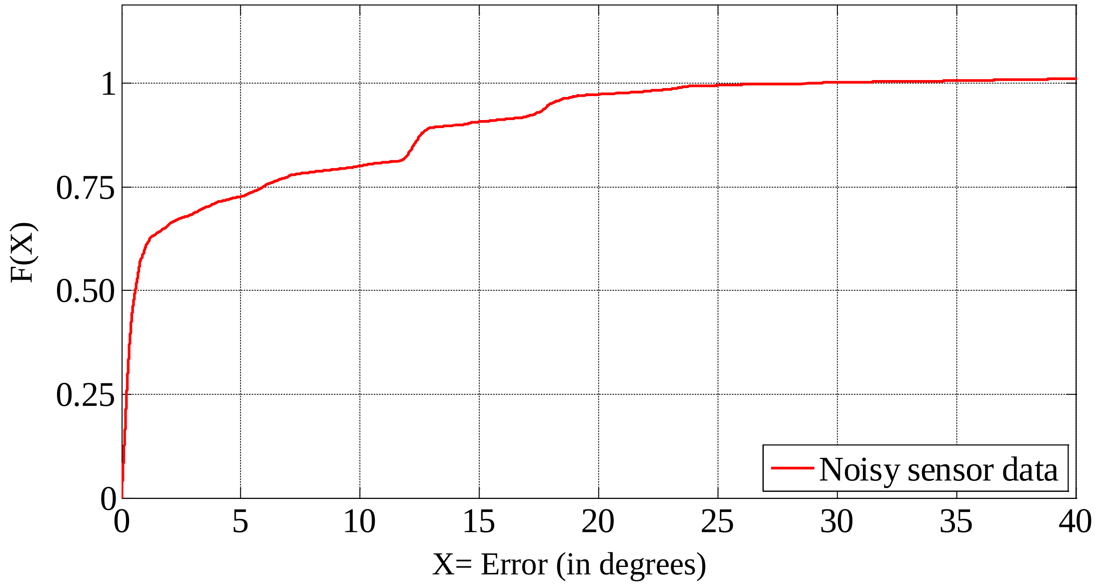


Figure 5-17 cdf of error calculated in noisy sensor data

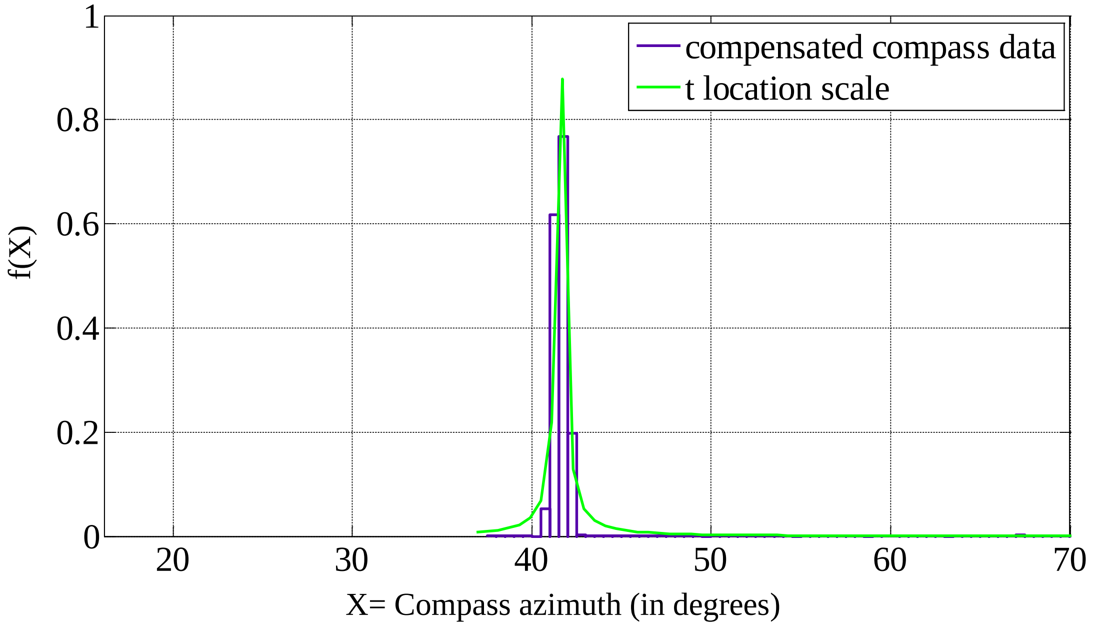


Figure 5-18 pdf of compensated compass data and t-location-scale distribution

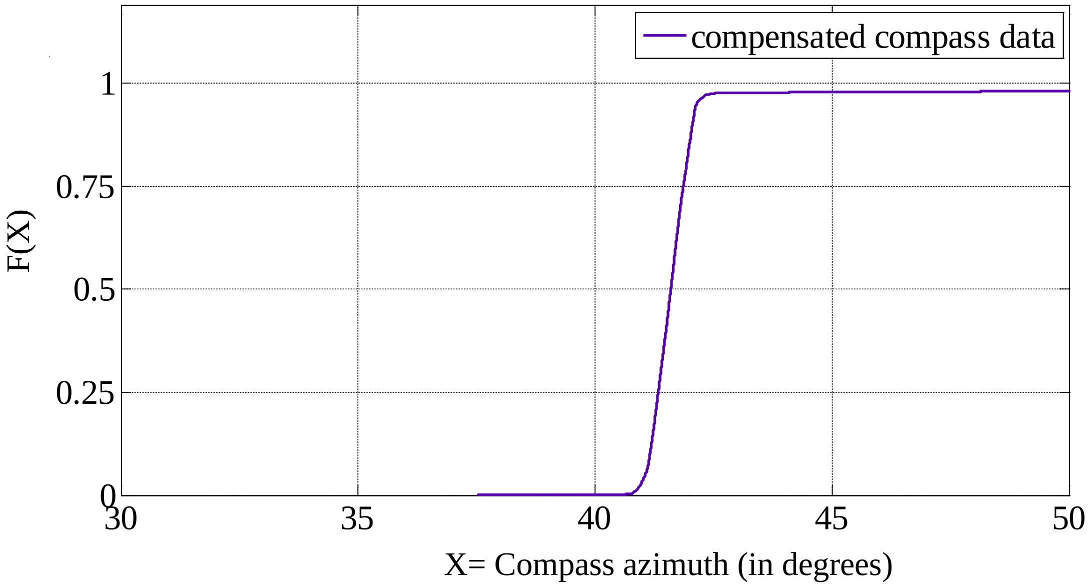


Figure 5-19 cdf of compensated compass data

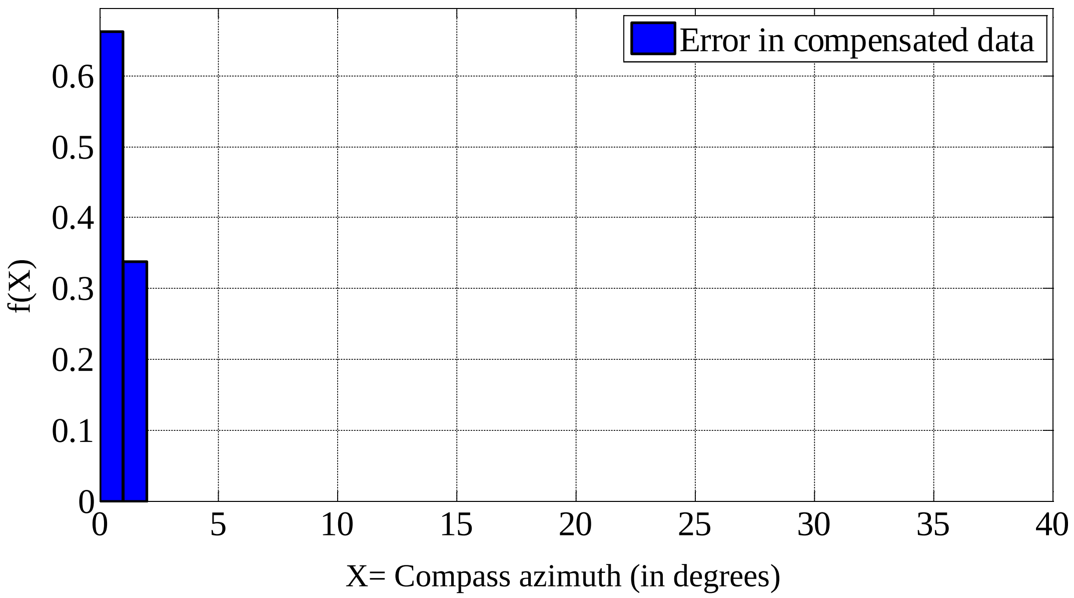


Figure 5-20 pdf of error calculated in compensated compass data

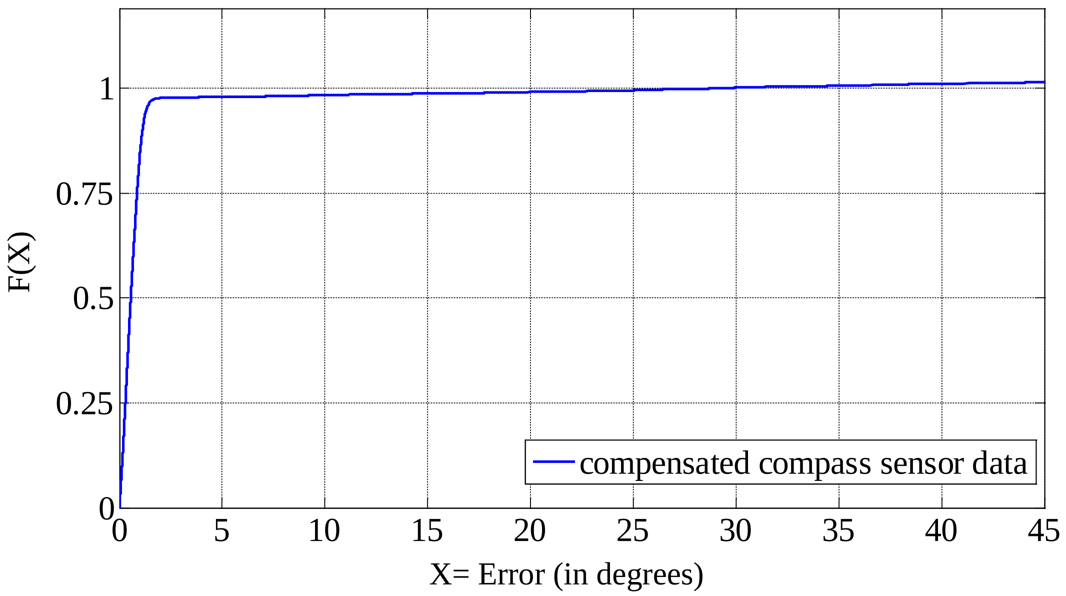


Figure 5-21 cdf of error calculated in compensated compass data

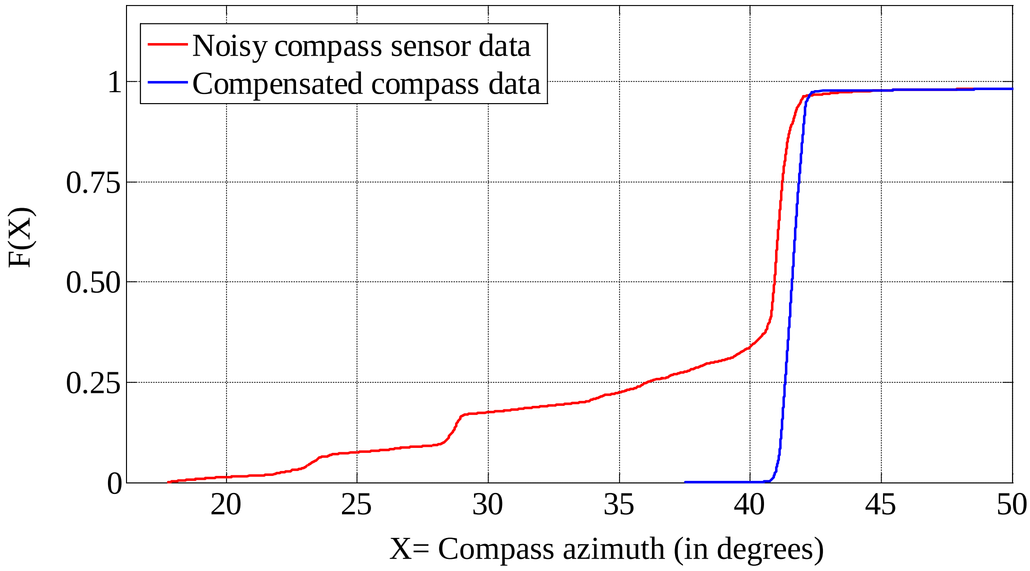


Figure 5-22 cdf comparison of noisy sensor data and compensated compass data

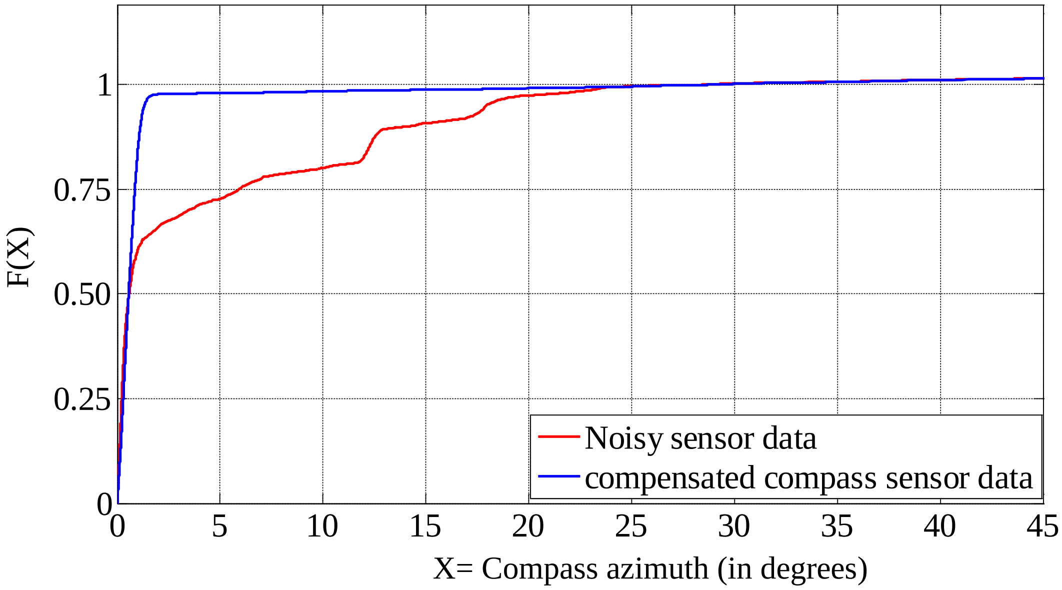


Figure 5-23 cdf comparison of error calculation of noisy and compensated compass sensor data

5.6 Conclusion

This chapter presents an investigation of the amount of magnetic deviation in a magnetometer of a smartphone caused by a car. From the results of the experiments, we analysed that the selected smartphone has measured up to 9.5° and 11° of magnetic deviation when different cars are parked at a distance of 0.5m. Furthermore, at a distance of 10 m, a magnetic deviation of 2° is still observed. A moving car that passes the smartphone at a distance of 1, 2 and 3m can cause a magnetic deviation of up to 3°. The amount of magnetic deviation caused by a car depends on the type of a car. Due to the occurrence of the magnetic deviation in a magnetometer of a smartphone, the car to pedestrian collision avoidance system suffers from reduced efficiency of filtering pedestrians. One option to compensate for the magnetic deviation is to use the gyroscope sensor of a smartphone. The gyroscope can differentiate between the rate of turn and occurrence of magnetic deviation in a magnetometer. As shown, the proposed algorithm “magnetic deviation compensation algorithm” successfully compensates the magnetic deviation.

This algorithm can be used to improve the accuracy of recognising the movement direction of a pedestrian in order to improve the performance of distinguishing the endangered or safe pedestrians in a car to pedestrian collision avoidance system. The proposed compensation algorithm can be useful in this scenario of collision avoidance between cars and pedestrians. The correction proposed in this chapter can also be used for many other applications. One of many further examples is indoor "navigation", based on the compass data and deviations caused by metal inside the building or due to machines and metallic furniture can be compensated.

6 Energy consumption of sensors of smartphones

6.1 Introduction

Parts of the contents of this chapter are published in [18].

In this dissertation, the smartphone sensors are used to provide pedestrians movement recognition and direction detection. The increased in the capabilities of the smartphone result in increased energy consumption. Therefore, it is essential to investigate the energy consumption of smartphone sensors individually and as well as combined. This analysis will answer the three questions, firstly, “How long will be the duration of a smartphone battery when it is used for the pedestrian movement recognition and movement direction detection? In addition, is the battery runtime of investigated smartphones is optimal when the selected sensors are ON while detecting the user movements and direction? Secondly, How precise are the software API based measurements compared to the device based measurements? Thirdly, what is the influence of the different sensors sampling rate on the battery runtime?

We used the different smartphones during the measurements such as Samsung Galaxy SII (SII), Samsung Galaxy SIII (SIII) and iPhone4. In section 6.2.1, we describe the reason that why we selected these smartphone models for energy consumption measurements. Furthermore, we present two different techniques to investigate the energy consumption of the sensors of these smartphones. Firstly, we used a voltmeter and an ammeter (named as “device based measurements”), which connects with the battery and a smartphone in a series connection. Secondly, the software is used (named as the “software based measurements”), which is installed on a smartphone and measures the energy consumption from the internal smartphone API. The dual approach of investigating the energy consumption provides the accuracy of the software based measurements.

The remainder of this chapter is organised as follows: In section 6.2, we present the methodological approach of two different techniques of measuring energy consumption. In section 6.3, we present the results. In section 6.4, we discuss the results in detail. Finally, we give the conclusion in section 6.5.

6.2 Methodology and Approach

In this section, we describe the methodical approach in detail. In section 6.2.1 and section 6.2.2, we describe the specification of used smartphones and limitations respectively. The measurement setup is explained in section 6.2.3. Afterwards, in section 6.2.4 and section 6.2.5, we explain two different techniques of measuring the energy consumption of the smartphone sensors, i.e., device-based measurements and software-based measurements.

6.2.1 Specification of used Smartphones

In this dissertation, we used four different smartphones in our investigations. We selected different smartphones based on their popularity and designed by different vendors, i.e., Samsung Galaxy and Apple. These vendors work on different platforms, i.e., Android and iOS respectively. These four selected smartphones are installed with different versions of the operating system, such as Android 2.3.6, 4.0.4, 4.1.2 and Apple iOS 4.3.3.

1. Two different versions of Samsung Galaxy SII
    a. SII GT-I9100 (SII): Android 4.0.4 version of operating system was installed on this model. The total battery capacity of this model is 1650 mAh.
    b. SII GT-I9100G (SIIG): Android 2.3.6 version operating system was installed on this model. The total battery capacity of this model is 1650 mAh.

2. Samsung Galaxy SIII GT-I9300 (SIII): Android 4.1.2 version of operating system was installed on this model. The total battery capacity of this model is 2100 mAh.

3. iPhone4 (MC603DN): Apple iOS 4.3.3 version of operating system was installed on this model. The total battery capacity of this model is 1420 mAh.

In general, the purpose of selecting the different vendor, types, and versions of operating system and different models of the smartphone was to evaluate the effectiveness of energy consumption of the selected sensors on the different platform. In specific, the purpose is to find the answer to justify our three research questions:

1. How long will be the duration of different smartphone batteries when we used it for the pedestrian movement recognition and movement direction detection? In

addition, is the investigated smartphones battery runtime is optimal when the selected sensors are ON while detecting the user movements and direction?

2. How precise are the software API based measurements compared to the device based measurements?

3. What is the influence of the different sensors sampling rate on the battery runtime?

Previously we used Nokia N95 and N97 in our measurements for activity recognition and detection of magnetic deviation compensation. However, with the time we have seen the popularity of these Nokia smartphones and the Symbian platform decreased. Besides, we see limitations in Symbian platform such as the sensors maximum sampling rate is small (for accelerometer maximum sampling rate was 32Hz and for compass maximum sampling rate was 9Hz). However, the accelerometer maximum sampling rate of 32Hz in Nokia N95 is sufficient, but the compass maximum sampling rate of 9Hz will not serve our purpose in determining the movement direction as early as possible. To compete with the other approaches such as A-GPS regarding of early recognition of movements and direction detection, we need the faster sampling rate of the selected sensors. Therefore, we decided to move on to the Android and iOS platform, where we can get quick sampling rate such as 100Hz in comparison to the Nokia Symbian platform.

On the other hand, we need an accelerometer, magnetometer and compass on one platform to determine the movements and direction. There are two reasons that we have not considered the Nokia N95 and N97 in our sensors energy consumption measurements. Firstly, N95 and N97 do not provide us with a complete solution to our magnetic deviation compensation and the detection of the movement direction approach. N95 have no magnetometer, compass, and gyroscope, and similarly, N97 does not have a gyroscope. However, our magnetic deviation compensation approach relies on the gyroscope of a smartphone, which is an essential sensor to differentiate between magnetic interference and the rate of turn. Secondly, Nokia N95 and N97 use the Symbian platform, which restricts in providing the faster sampling rate of the accelerometer (32Hz) and compass (9Hz).

Therefore, we proposed to add more measurement devices that provide the required sensors in one platform. Initially, we measured the energy consumption of iPhone4 and Samsung Galaxy SII (GT-I9100). Both these smartphones provides a faster sampling rate (up to 100Hz) compared to our previous chosen smartphones, i.e., N95 and N97. In addition, N97, three models of Samsung Galaxy (SII, SIIG, SIII) and iPhone4 incorporated with the same magnetometer chipsets, i.e., AKM 897X. However, we assume that all these smartphones observe the same magnetic interference due to the same magnetometer chipset. Therefore, we provide a compensation solution on a device, which is integrated with the gyroscope. Therefore, we selected these four models Samsung galaxy (SII, SIIG, SIII) and iPhone4 to investigate the energy consumption of the required sensors in one platform.

Moreover, with the time, new smartphones arrived in the market with better sensor specifications, power efficient and higher battery capacity. Therefore, for our direction detection approach, we used Google Nexus 5. The Google Nexus 5 has an integrated MPU 6515 6-axis accelerometer chipset manufactured by InvenSense and AKM 8963C magnetic field sensor chipset manufactured by Asahi Kasei Microdevices (AKM) Pvt Ltd. However, based on two reasons we have not considered the sensors energy consumption of Google Nexus 5 in our analysis. Firstly, based on the datasheet of the sensors chipset, we analysed that both these chipsets are advanced and power efficient in contrasts to their successive models. Therefore, we assume that the energy consumption of the selected sensors of the Google Nexus 5 consumes less energy as compared to the smartphone used in our investigations. Secondly, due to the higher battery capacity of Google Nexus 5 model (i.e., 2300mA/h), we assume that the battery runtime of the Google Nexus 5 either with the device or API based measurements will also be more or like the same in comparison to the selected smartphones.

Initially, we used two models for energy consumptions, i.e., Samsung Galaxy SII (GT-I9100) and iPhone4. However, as the results varied widely and we wanted to validate our measurements, we decided to add another hardware revision of the Samsung Galaxy SII (GT-I9100G) and Samsung Galaxy SIII (GT-I9300). Still, we do not know the exact reasons about why SII (GT-I9100) consumes much power when the sensors are ON. However, we have noticed that when the sensors are ON, the smartphone gets warm and consumes more energy. We analysed that there is the hardware failure in SII (GT-I9100) as the Pitch angle showed the wrong readings (i.e., 35 degrees instead of near to 0 degrees) even if the SII (GT-I9100) placed flat on the ground or the installed operating system may have the bug. Therefore, we decided to add two another Samsung Galaxy models to validate the results, and these are SIIG (GT-I9100G) and SIII (GT-I9300).

6.2.2 Limitations of used Smartphones

The investigated iPhone4 showed some limitations during the measurements. In the Samsung Galaxy smartphones, the sensors can be switched on and off separately. For the iPhone4, this is not possible for all combinations of the sensors. The standby mode stops some of the processes of the running applications in the iPhone4 (e.g. sensing with the accelerometer). Therefore, the standby mode had to be avoided, by switching off the idle timer of the iPhone4. The idle timer also switches off the display of the iPhone4.

Nevertheless, the display had to be switched off because this adjustment is more realistic for a context-aware application, which most of the time will probably work in the background. We decided to use the proximity sensor instead to turn the display off. Thus, the proximity sensor had to be switched on and was covered to turn the display off. In iPhone4, the accelerometer cannot be switched off using the API. This sensor is always sensing in the background, even when the rotation lock is deactivated.

Another limitation regarding iPhone4 measurements using device-based method is that we were not able to activate the cellular connection while the back cover was removed. The back cover had to be removed and stay unattached during the device-based measurements. Therefore, we could not measure the energy consumption (as shown in Figure 6-3 in measurement 5 “Gyroscope Compass and 3G”) with activated cellular connection on iPhone4.

6.2.3 Measurement setup

We chose the same combination of sensors for selected smartphones, to be able to compare measurements. Because of the described limitations, we activated the proximity sensor and the accelerometer for all measurements. The gyroscope and the compass were switched on and off separately and in combination, measuring the energy consumption with both methods. One method was to measure the energy consumption via a voltmeter and an ammeter (device-based measurements). The other method to measure the energy consumption was using the internal software API of smartphones (i.e., software-based measurements). We designed five different combinations of measurements (which are mentioned below) to investigate the energy consumption of the smartphone sensors. The energy consumption measurements were repeated three times. In order to evaluate the results, we averaged the same combinations of measurements.

1. The Proximity and the accelerometer sensors are switched ON. This measurement is named as “None” as shown in Figure 6-3 to Figure 6-6.

2. The proximity sensor, accelerometer, and gyroscope sensors are switched ON. This measurement named as “Gyroscope” as shown in Figure 6-3 to Figure 6-6.

3. The proximity sensor, accelerometer, and compass sensors are switched ON. This measurement is named as “Compass” as shown in Figure 6-3 to Figure 6-6.

4. The proximity sensor, accelerometer, gyroscope, and compass sensors are switched ON. This measurement is named as “Gyroscope and Compass” as shown in Figure 6-3 to Figure 6-6.

5. The proximity sensor, accelerometer, gyroscope, and compass with cellular connection switched ON, excluding iPhone4. We named this measurement as “Gyroscope Compass and 3G” as shown in Figure 6-3 to Figure 6-6.

To measure the energy consumption of sensors as accurately as possible, we reduced side effects on battery runtime as much as possible. Before the measurement was taken, a test person fully charged the selected smartphones. Wi-Fi and Bluetooth were deactivated. The cellular connection was only switched ON for measurements number five. All applications were closed, and none was visible in the multitasking toolbar or the task manager except our measurement application.

For the software-based measurement, we activated one specific sensor in the application. Afterwards, we immediately removed the charger, the software application records the data, and we covered the proximity sensor to turn off the display. For the iPhone4 and Samsung Galaxy models, the application logged the battery status every 5 minutes. While the software application records the battery status, a test person checked the battery status once per day (in a 24 hours interval). Therefore, the test person removes the cover from the proximity sensor, this turns the display on, and the battery state is displayed. After a second or two, the proximity sensor is covered again. The selected smartphones stay unmoved on the desk during the whole measurement process.

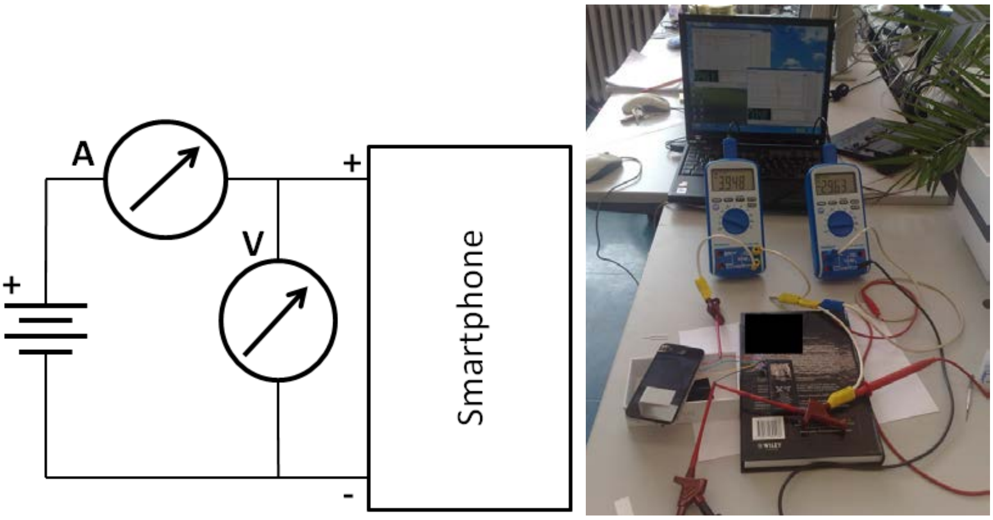
For the device-based measurement, the same procedure was applied as the software-based measurement, accept the smartphone battery, and multimeter and an ammeter was connected in a series connection to record actual current and volts which is logged in a computer, connected using the USB data cable when a specific sensor was activated.

6.2.4 Voltmeter and Ammeter (device based) measurements

During the measurements, we used two PeakTech 3415 USB digital multimeters to measure the energy consumption of the smartphone sensors. We record measurements when both multimeters were connected to the computer using the USB cable, also with the smartphone, and its battery in a series connection as shown in Figure 6-1. PC-Link data software is used to log the data in a computer at a sampling rate of 2Hz. According to the datasheet, PeakTech 3415 multimeter measures the current and voltage with an accuracy of ±1.5% ±0.5%.

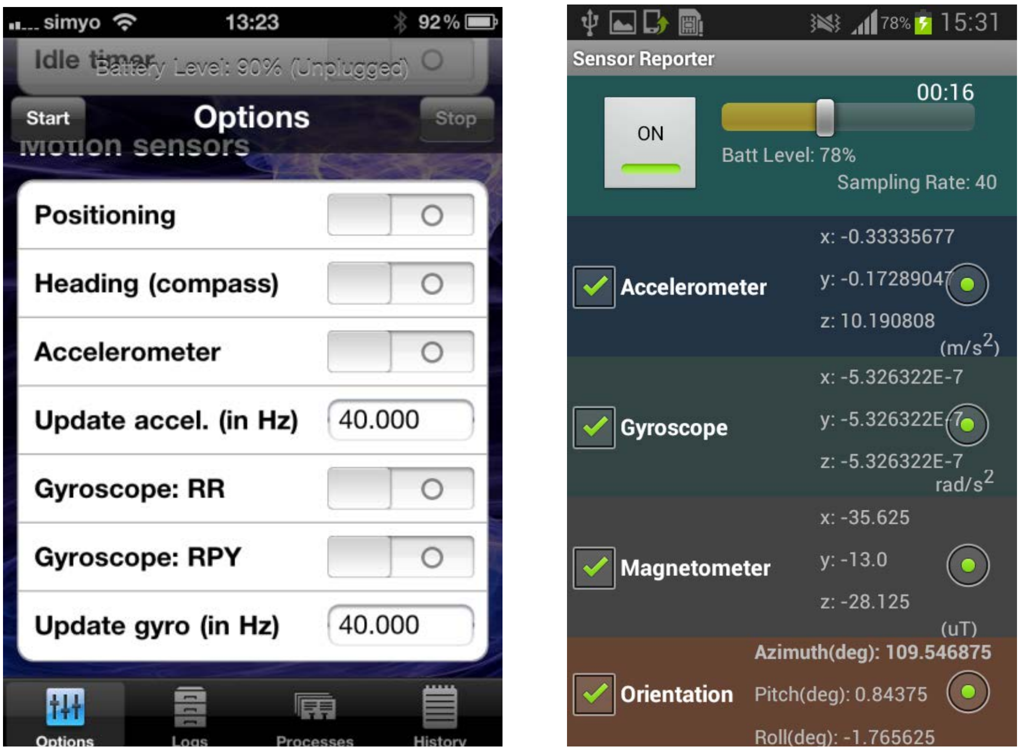
6.2.5 Software-based measurements

We designed two software applications, one in Android and one in iOS platform to collect the API based measurements of the smartphone sensors. Both software records the battery status and calculates the energy consumptions of the smartphone sensors. Also, provides the features such as to enable or disable any sensor and increase or decrease the sampling rate of a sensor. A screenshot of the applications can be seen in Figure 6-2.



[redacted]

Figure 6-1 Device based measurement setup [18]



Idle timer

Battery Level: 90% (Unplugged) ○

Motion sensors

Positioning ○

Heading (compass) ○

Accelerometer ○

Update accel. (in Hz) 40.000

Gyroscope: RR ○

Gyroscope: RPY ○

Update gyro (in Hz) 40.000

Sensor Reporter

00:16

Batt Level: 78%

Sampling Rate: 40

✓ Accelerometer x: -0.33335677 y: -0.17289047 z: 10.190808 (m/s2)

✓ Gyroscope x: -5.326322E-7 y: -5.326322E-7 z: -5.326322E-7 rad/s2

✓ Magnetometer x: -35.625 y: -13.0 z: -28.125 (uT)

✓ Orientation Azimuth(deg): 109.546875 Pitch(deg): 0.84375 Roll(deg): -1.765625

Figure 6-2 Software based measurement applications (left: iPhone, right: Android) [18]

6.3 Results

In this section, we presented the measurement results. In each of the Figures (Figure 6-3 to Figure 6-6), we present the results for one investigated smartphone. In Figure 6-3 to Figure 6-6, we showed the energy consumption of individual combination of sensors using both techniques, i.e., software and device-based measurement. Where the X-axis shows the individual combination of sensors, i.e., as shown in measurements setup (measurement 1 to measurement 5) and the Y-axis shows the energy consumption in milliWatt (mW). Note that the proximity sensor and the accelerometer (40Hz) were activated for all measurements and also for the one entitled with “None”.

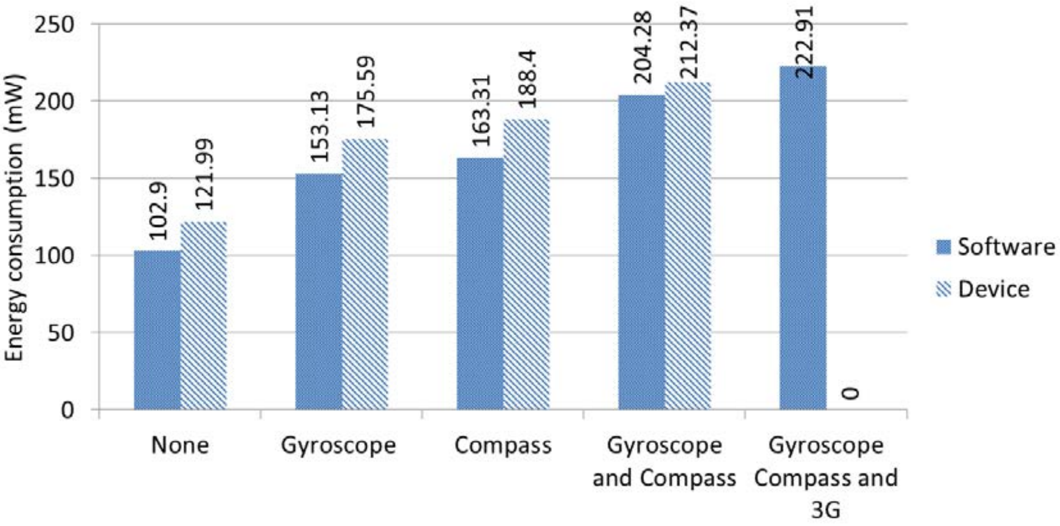


Figure 6-3 Energy consumption of iPhone4 sensors

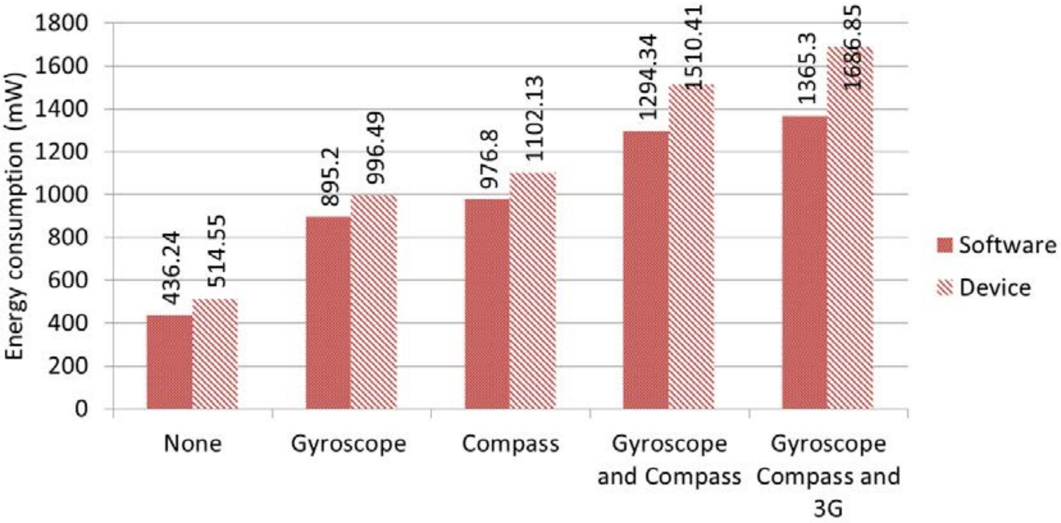


Figure 6-4 Energy consumption of Samsung Galaxy SII GT I9100 sensors

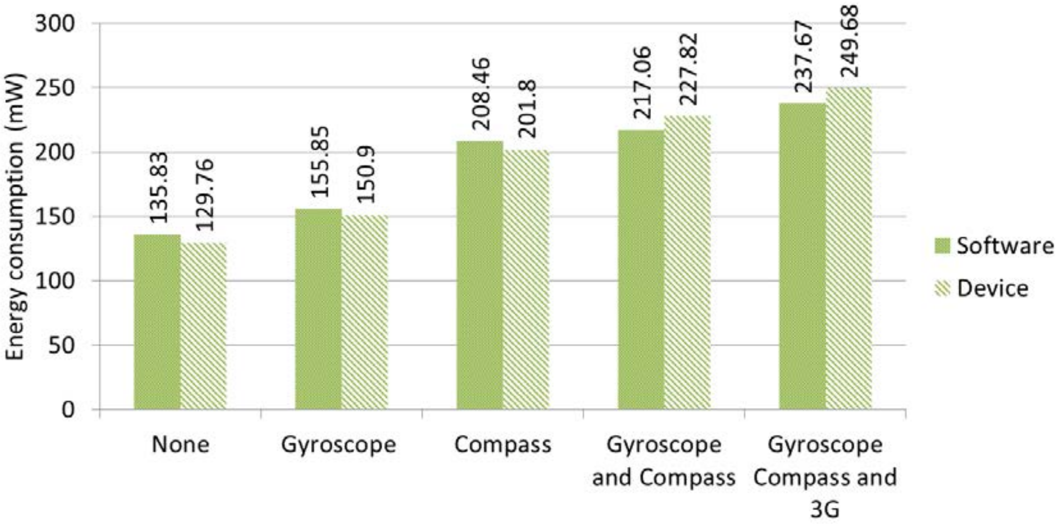


Figure 6-5 Energy consumption of Samsung Galaxy SII GT I9100G sensors

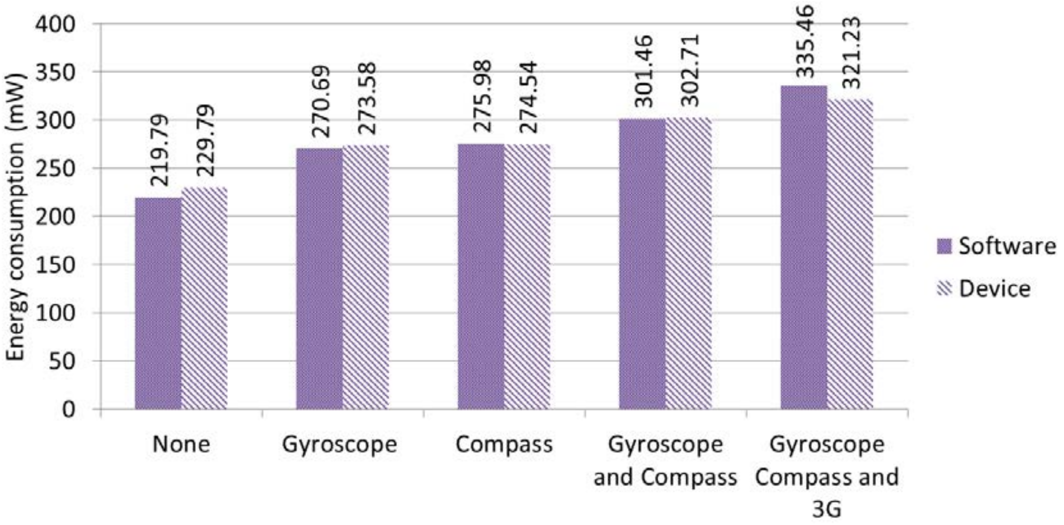


Figure 6-6 Energy consumption of Samsung Galaxy SIII GT I9300

6.4 Discussion of the results

In this section, we discuss the measurement results concerning three aspects:

1. How long is the duration of the battery charge when a specific sensor or combination of sensors is ON using software and device based measurement? In addition, is the investigated smartphone battery runtime is optimal when the selected sensors are ON while detecting the user movements and direction?

2. How precise are the software API based measurements compared to the device based measurements?

3. What is the influence of the different sampling rate of the sensors on the battery runtime?

6.4.1 Duration of a battery charge using API and device based measurements

In Figure 6-3 to Figure 6-6, we showed the energy consumption the sensors using software and device-based methods of the selected smartphones, i.e., iPhone4, Samsung Galaxy SII, Samsung Galaxy SIIG and Samsung Galaxy SIII. In Figure 6-3 to Figure 6-6, we analysed that for an individual combination of sensors there is always a difference in energy consumption between the software and device-based technique.

Furthermore, we analysed from the Figure 6-3 to Figure 6-6 that the Software and Device-based measurements have an average variation of 18mW for iPhone4, 8mW for SIIG (GT I9100G) and 5.8mW for SIII smartphone. Comparing iPhone4 and SII (GT I9100) it is unclear why the average energy consumption of SII (GT I9100) is so high in both software and device based techniques. Not all investigated Samsung phones showed the same reaction, as it turned out. The newer Samsung phones showed much lower energy consumption.

By comparing the values for the gyroscope and compass for example with an iPhone4 and SII (GT I9100), we find a considerable variation. The iPhone4 consumes 212.37mW for the measurement “gyroscope and compass” (see Figure 6-3). The same sensors cause an energy usage of 1510.41mW for the SII (GT I9100) (see Figure 6-4). Still, we do not know the exact reasons about why SII (GT-I9100) energy consumption is so high. We cannot tell which part of the smartphone consumes the energy. Nevertheless, most probably the energy is not consumed by the sensor ICs, but by the central processor of the smartphone, as the back of the phone was getting warm (not only where the battery is located). We can only assume that maybe there is the hardware failure, or the installed operating system may have the bug.

From our measurements, we can derive the duration time of one battery charge while the smartphone senses. In Figure 6-7, we showed the battery runtime (in hours) using software-based measurements of four different smartphones and different measurements. Similarly, in Figure 6-8 we showed the battery runtime (in hours) using device-based measurements. Almost all investigated smartphones manage to cover a sensing time of about 24 hours using both techniques excluding one smartphone model (SII GT-I9100); still, we do not know the exact reasons why (SII GT-I9100) energy consumption is so high. Therefore, we can conclude that most of the selected smartphones can be used for our daily purpose and also to recognise pedestrians movement and direction detection. The battery duration of 24 hours would still leave the phone with enough energy reserves for the standard activities of telephony and browsing. Moreover, we see that the newer models of the smartphone come up with higher battery capacity and with improved energy consumption; exemplary we showed that Google Nexus 5 has higher battery capacity from its successive models.

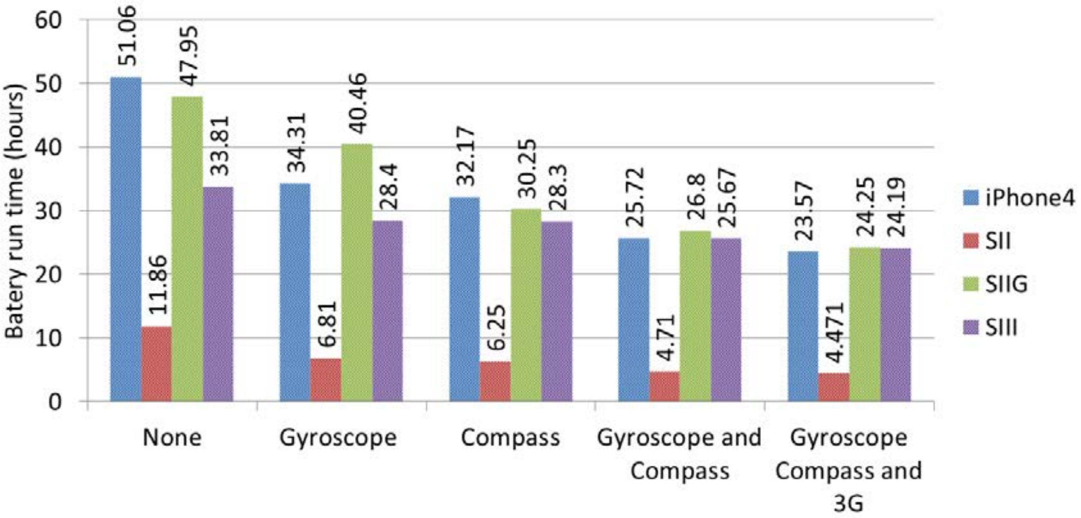


Figure 6-7 Battery run time (hours) using API based measurements

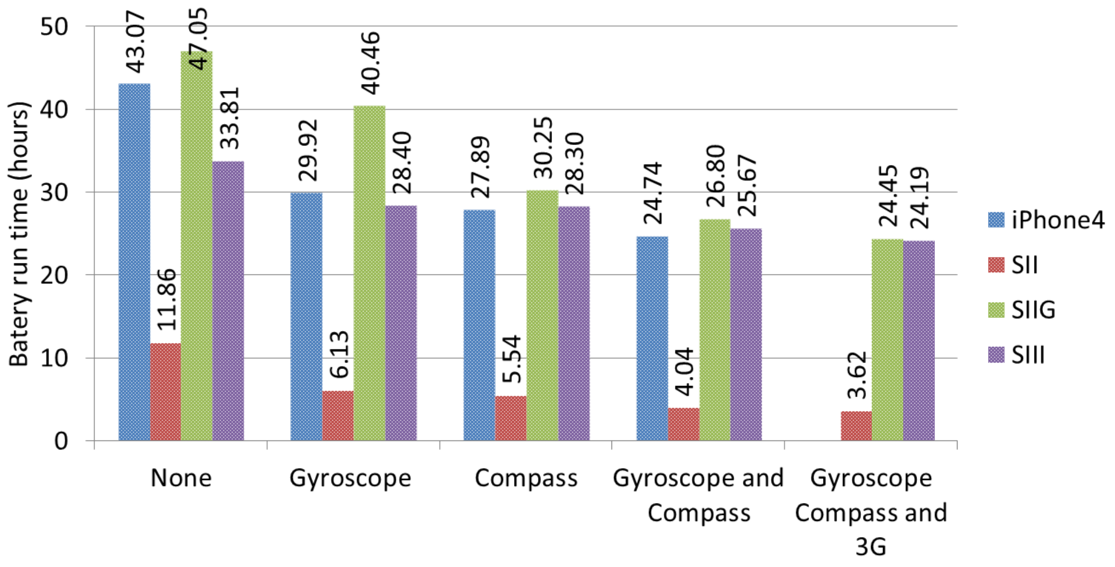


Figure 6-8 Battery run time (hours) using device based measurements

6.4.2 Comparison of API and device based measurements

We chose a dual approach for our investigation concerning the energy consumption of the smartphone sensors. First, we started in utilizing a software-based approach to measure the energy consumption. To verify the results we also measured the energy consumption of the sensors with a voltmeter and an ammeter (device-based measurement). Our dual approach of measuring the energy consumptions enables us to compare how accurate a software-based measurement as compared to a device-based measurement. In Table 6-1, we show the difference of energy consumption (in %) between software and device-based measurements. For iPhone4, the software values vary a maximum of 18.55% compared to the device-based measurement. For the SII (GT I9100), the software values vary a maximum of 23.55% compared to the device-based measurement. For the SIIG (GT I9100G) and the SIII (GT I9300), the software values differ less than 5.05% and 4.54% respectively compared to the device-based values.

Smartphones do not have a voltmeter and an ammeter to measure the actual volt and current. Smartphones can log the battery status in percentage. Based on the battery status in percentage, smartphone API calculates the battery runtime using total battery capacity divided by the percentage of battery consumed hourly. Moreover, smartphone API calculates the power consumption by assuming the voltage is constant, i.e., 3.7V. In contrasts, device-based technique measures the actual current and volt using an ammeter and a voltmeter. However, we have analysed during device-based measurements that the voltage of the battery is not constant it varies from 4v to 3.5v as the battery discharges. For example, when the battery is fully charged, the voltmeter measures the 4V. In contrasts, software-based method assumes 3.7V, and this could be the one reason of higher power consumption of device-based measurements when the voltmeter measures 4V.

Moreover, we assume that the other possible reason for higher energy consumption using device-based method could be due to some unknown loose connection between the battery of the smartphone and designed prototype. Alternatively, the accuracy of the multimeter is not good as it is shown in the datasheet.

Table 6-1 Comparison of energy consumption (in %) between software based and device based measurements

Sensor	Investigated smartphones			
	iPhone4	SII	SIIG	SIII
None	18.55	17.95	4.46	4.54
Gyroscope	14.66	11.31	3.17	1.06
Compass	15.36	12.83	3.19	0.52
Gyroscope and Compass	3.96	16.69	4.95	0.41
Gyroscope Compass and 3G	-----	23.55	5.05	4.24

6.4.3 Comparison of energy consumption and battery runtime using different sampling rate

In this section, we presented the comparison of energy consumption (current consumption) and battery runtime of the different sampling rates of the sensors. However, we assumed that the different models of the smartphone, different sensors and their sampling rate consume different energy consumption and therefore, influence the battery runtime. Based on this assumption, the battery runtime will be different. Moreover, we justified the assumption after analysing the energy consumption of different models of the smartphone, different sensors and their sampling rate. In Table 6-2 and Table 6-3, we exemplary showed the energy consumption (current consumption) and battery runtime of iPhone4 for the measurement “None” (Proximity and accelerometer sensors are switched ON) using the software and device-based technique respectively.

In Table 6-2, we analysed that when the measurement “None” is activated at a sampling rate of 8Hz, 32Hz, 40Hz, 64Hz, the current measures 26.14mA, 27.77mA, 27.81mA, 30.37mA respectively and the battery provides the runtime of 54.33 hours, 51.33 hours, 51.06 hours, and 46.75 hours respectively. Similarly, in Table 6-3, we showed the results using device-based measurements. As we analysed from Table 6-2 and Table 6-3, higher sampling rate consumes more power and therefore reduces the battery runtime. However, almost all the investigated smartphones (accept SIIG) models and the combination of different sensors consume a reasonable amount of energy and provide the battery runtime of a minimum of 24 hours when all the selected sensors are ON. The battery duration of 24 hours would still leave the phone with enough energy reserves for the standard activities of telephony and browsing.

Table 6-2 Energy consumption and Battery runtime of iPhone4 using the software based technique

Energy consumption and Battery runtime using the software based technique				
Sampling rate	Current (mA)	Power (mW)	Battery runtime (hours)	Delta T
8Hz	26.14	96.71	54.33	-
32Hz	27.77	102.76	51.13	3.20
40Hz	27.81	102.89	51.06	3.27
64Hz	30.37	112.39	46.75	7.58

Table 6-3 Energy consumption and Battery runtime of iPhone4 using the device based technique

Energy consumption and Battery runtime using the device based technique				
Sampling rate	Current (mA)	Power (mW)	Battery runtime (hours)	Delta T
8Hz	30.53	117.99	46.49	-
32Hz	31.44	121.04	45.16	1.33
40Hz	31.59	121.99	44.93	1.55
64Hz	32.73	126.03	43.37	3.12

6.5 Conclusion

This chapter presented the investigations on how much the different smartphones model and different sensors used the energy. Based on the results and analysis presented in this chapter, we concluded that the data collection of the smartphone sensors at a recommended sampling rate of 30-50Hz would be optimal. At 30-50Hz, the sampling rate battery provides the runtime of minimum 24 hours, which served our purpose of recognising the movements and direction detection. This enables the smartphone to last the battery over 24 hours, and it is also practical for the user to recharge the battery at night. Furthermore, we presented the comparison of energy consumption using software and device-based approach. This dual approach of investigating the energy consumption helps to compare how accurate the software-based measurements are. Exemplary, for iPhone4, the software-based measurements varies at maximum 18.55% in comparison to the device-based measurements. Similarly, for SII measurements vary a maximum of 23% compared to the device-based measurements. For the SII (GT I9100G) and the SIII (GT I9300), the software-based measurements vary 5% and 4% respectively in comparison to the device-based measurements. However, the difference in the energy consumption of using software and device-based methods is due to the software-based measurement assumes the constant voltage, i.e., 3.7V and device based measurements measures the current voltage that can be 4V.

7 Conclusion and Outlook

We divide this chapter into two sections. In Section 7.1, we present the summary and conclusion of this Ph.D. dissertation. Finally, in Section 7.2 we provide an outlook on open issues and future work.

7.1 Summary and Conclusion

The recognition of pedestrian movements, such as accelerating, decelerating, slowing to a complete stop, and direction detection is the basis of many pedestrian safety and pedestrian dead reckoning applications. As motivation for our work, we reviewed pedestrian safety applications. Every year, thousands of pedestrians are killed or injured in traffic accidents. The car manufacturers and various research groups address the challenge of improving pedestrian safety through passive and active pedestrian protection. In Section 1 of Chapter 1, we summarised few existing approaches, which have many limitations, such as the need for direct line-of-sight between cars and pedestrians to work correctly. In the above-mentioned related work, the GPS is utilised to overcome this limitation. However, we presented observations suggesting that the GPS functions at a maximum sampling frequency of 1Hz, which is useful mainly for navigation purpose but not sufficient to update the movement history in real time for the detection of sharp movement turns that last less than a second. Moreover, the GPS also does not detect the turns especially when the pedestrian is standing on one point and turning (see Figure 4-13). Furthermore, every time a pedestrian changes his speed or direction, the GPS response in updating the movement and direction information takes 3 to 4s (see section 3.2.5).

To improve the recognition time of a pedestrian movement and direction, we presented an alternative to the GPS that uses an accelerometer and a compass. We investigated how our approach can recognise the pedestrian movement (using the accelerometer) and direction (using the compass) within a meaningful timescale. A survey showed that a smartphone is the most widely used device as nearly one-third of the world’s population utilises a smartphone. Most smartphones are now equipped with different types of sensors, including an accelerometer, magnetometer, gyroscope, compass, and GPS. Therefore, we propose the use of smartphones sensors as a matter of convenience because smartphones are so common device people carry with them along with tablets and smartwatches. For this research, we used data acquired from the smartphone accelerometer to identify the pedestrian movement and the smartphone compass to determine the pedestrian direction. In particular, we addressed several challenges for our alternative approach compared to that of using GPS including:

1. We determined the pedestrian movement direction independent of the smartphone's orientation when it is placed inside a trouser pocket.
2. We used the compass of a smartphone to determine the movement direction. To manage magnetic interference within the smartphone compass, we proposed the algorithm to compensate for any detected magnetic deviations.
3. Finally, we investigated the energy consumption of the smartphone sensors.

In Chapter 2, we summarised the state of the art in the following categories:

- Movement direction of pedestrians using the following methods:
  - pedestrian dead reckoning approaches using the shoe-mounted inertial measurement unit
  - smartphone-based approaches:
    - handheld
    - body attached (fixed)
    - velocity vector based on smartphone sensors
  - sensor fusion based approaches
- Movement detection (activity recognition) of pedestrians using an accelerometer

In Chapter 3, we presented an approach using the accelerometer of a smartphone to determine the pedestrian's movements, including accelerating, decelerating, and slowing down to a complete stop. We conducted a variety of experiments to investigate the suitability of our approach where the user carried a smartphone in the trouser pocket and performed a series of movements. The accelerometer data was recorded at 32Hz for the intended classification process. As suggested by Lau et al. [70], a sliding window technique was used to segment the accelerometer time series data to perform the feature extraction process. The window length for the feature extraction was set at 1s to achieve recognition in less than one second for the designated application. An overlap of 75% was selected and features used for the classifications included mean, standard deviation, variance, information entropy, and the energy of the FFT in the frequency domain.

For evaluation, the Weka data-mining tool was used along with different base-level classification algorithms, including DT, kNN, and jRip, to determine the accuracy of pedestrian movement recognition. Moreover, these base-level classifiers were used in conjunction with the meta-level classification algorithm, i.e., bagging and boosting, to improve the movement recognition accuracy. The results provided recognition accuracies between 93.39% and 96.98%. Using a window length of 1s and overlapping of 75%, each movement recognition was estimated to be 250ms apart. Therefore, it was possible to obtain four movements within one second. In addition, the time needed for classification indicated that all classifiers presented were able to complete the desired recognition within 250ms. Furthermore, this approach offers the advantage of being 1.5s quicker than the GPS according to our comparison with the GPS sampling frequency of 1Hz only when the pedestrian is walking at a constant speed. The ability to determine quick changes in the pedestrian speeds and walking direction are both important information sets in the context of collision avoidance.

In Chapter 4, we presented an analysis to infer the movement direction of pedestrians from the compass of a smartphone. Typically, pedestrians carry a smartphone in different orientations, and until now, movement direction detection in pedestrian safety applications using the smartphone relies on the GPS. As stated previously, the GPS functioning at a maximum sampling frequency of 1Hz, which is useful mainly for navigation purpose but not sufficient to update the movement history in real time for the sharp movement turns that last less than a second. Moreover, the GPS also does not detect the turns especially when the pedestrian is standing on one point and turning (see Figure 4-13). Furthermore, every time a pedestrian changes his speed or direction, the GPS response in updating the movement and direction information takes 3 to 4s (see section 3.2.5). Alternate approaches use sensors, such as the accelerometer and compass, which relies on carrying the smartphone in a predefined orientation or knowing the orientation of the smartphone in relation to that of the user/pedestrian orientation. The experiments presented in this chapter demonstrated that different orientations of the smartphone inside the trouser pocket gave different information for the compass, while the test path, the pedestrian’s orientation and movement direction of the pedestrian remained same. This suggests that most of the smartphone orientations are not the same as the movement direction of pedestrians, which results in an incorrect movement direction. Therefore, this chapter presented the algorithm, SOMDA, which detects the smartphone orientation and aligns it to the movement direction of the pedestrian carrying the smartphone inside the trouser pocket. This method first determines the orientation of the smartphone’s top using the compass and an orientation sensor. Next, it determines the orientation of the smartphone’s screen and then the user’s movement direction by observing the compass and accelerometer during at least two steps taken by the user. After these two steps, the algorithm continuously aligns the smartphone and the user/pedestrian orientation. With our proposed approach, the user/pedestrian is free to change direction, speed, or stop moving. The smartphone may be placed in the trouser pocket arbitrarily, and the smartphone is free to wobble inside the trouser pocket.

Experimental measurements provided an investigation into the performance of the proposed approach. The SOMDA achieved the direction detection accuracy of 96% at a threshold value of 15° using a sampling rate of 50Hz. Downsampling from 50Hz to 6.25Hz reduced the accuracy of the SOMDA by 6%. Moreover, Assisted-GPS was observed to take 3 to 4s (see section 3.2.5) to detect the actual movement direction, while SOMDA detected the orientation and movement direction of a user/pedestrian after an average time of 490ms. This demonstrates that the SOMDA has a 3.5s advantage of detecting the movement direction when the pedestrian walking speed is not constant. Furthermore, by using A-GPS, it is not possible to detect a change in the user/pedestrian orientation when the user/pedestrian stands at one point and turns clockwise or anticlockwise and does not move forward. This is because A-GPS works on the successive location point. In contrast, the SOMDA detects such changes after an average of 240ms. The time to detect the user/pedestrian orientation and movement direction depends on how fast the user/pedestrian moves and on the type of turn, i.e., short turn or long turn. In Chapter 4, we presented the detailed performance of the SOMDA algorithm, which is efficiently determined the movement direction 3.5s earlier in comparison to the use of A-GPS when the walking speed of the pedestrian is not constant.

In Chapter 5, we showed a weakness in the compass of a smartphone was identified as the presence of magnetic deviation. A filter was designed to differentiate between safe and endangered pedestrians, and, as shown in the results, the magnetic deviation in the smartphone compass influenced the performance of the filtering process. For example, if a group of pedestrians are equally distributed in all directions, then half of pedestrians are moving away from the road and are considered safe, and the other half of pedestrians are moving towards or parallel to the road and are considered endangered. When there is no magnetic deviation, in this case, the efficiency of filtering pedestrians is 50%. If the observed magnetic deviation is 11°, then the efficiency of filtering pedestrians drops to 43.89%, which means pedestrians moving 0° to 11° away from the road are also considered as moving towards or parallel to the road due to the influence of the magnetic deviation. Therefore, this magnetic deviation must be compensated to determine the accurate movement direction of pedestrians. In this chapter, we presented the MDCA algorithm, which makes use of the gyroscope sensor of a smartphone. The gyroscope can differentiate between the rate of turn and the magnetic deviation in a compass. As shown in Chapter 5, the MDCA successfully compensated for the magnetic deviation and so can be used to improve the accuracy of the direction detection of a pedestrian and improve the performance of distinguishing safe and endangered pedestrians in a pedestrian safety application.

In Chapter 6, we investigated the energy consumption of the smartphone sensors. This is another important aspect in our goal to use the smartphone to provide a solution to the movement recognition and direction detection of pedestrians. Two methods were proposed to investigate the energy consumption; first to connect a voltmeter and ammeter with the battery of the smartphone while the sensors are active. Second is to use software to extract the energy consumption from the internal smartphone's API. This analysis enabled us to conclude the smartphone may be used for pedestrian safety application because, almost all tested smartphones provide the battery runtime up to 24 hours excluding one smartphone model (SII GT-I9100); still, we do not know the exact reasons why (SII GT-I9100) energy consumption is so high. We cannot tell which part of the (SII GT-I9100) consumes the energy. We can only assume that maybe there is the hardware failure, or the installed operating system may have the bug.

In Chapter 7, we summarised the main contribution of this Ph.D. dissertation based on the conceptual introduction and the performance evaluation of movement recognition and direction detection of pedestrians. A smartphone can be used for daily sensing and monitoring context information, such as movement recognition and direction detection of pedestrians. The use of this context information mainly benefits the pedestrian safety application with the timely determination of the movement history of pedestrians to avoid collision with cars. Our findings could be applied quite reliably in other applications, such as indoor localisation, navigation, and PDR without a significant degradation in performance.

7.2 Outlook

In this dissertation, we investigated the movement recognition and direction detection of users/pedestrians carrying a smartphone in the front trouser pocket. Our proposed approach uses the built-in sensors of a smartphone such as acceleration, magnetometer, gyroscope and compass and solves the issue of determining the movement direction independent of a smartphone orientation or without the help of any other source such as using the GPS. There are research questions that are out of scope for this dissertation, but we would like to address them in future. To complete this dissertation, here we will summarise the interesting research questions:

We determined the movement direction of users/pedestrians using smartphones sensors while carrying in a trouser pocket. However, in [124] a survey shows that 61% of women and 10% of men carry their smartphones in the bag. Therefore, we think other positions where users/pedestrians could carry a smartphone such as in a bag is still need to be investigated to determine the movement direction regardless of the smartphone orientation. For example, in [125], a survey shows that users can carry a smartphone in different bags at different positions. These bags and positions are categorised as follows:

- Backpack (carried over both shoulders and positioned on the back centre of the body)
- Handbag (carried in the hand and positioned on the side of the body)
- Messenger bag (carried over one shoulder with a strap that goes across the chest and positioned the bag on the back of the body or sideward)
- Shoulder bag (carried on the same side of the shoulder as the bag and positioned at the side of the body)

Other positions where users/pedestrians could carry a smartphone also needs to be investigated to determine the pedestrian’s movements and direction regardless of the smartphone orientation. For example, including jacket pocket, shirt front pocket, trouser back pocket and in hand while walking and hand is swinging.

One of our future works will be the use of Kalman filter to fuse the smartphone sensors data to improve the direction accuracy and recognition time. To achieve this goal, first, we have to develop the update and predict model of the system.

Finally, we will address another interesting research question in future is to design and implement a software-based application which demonstrates the collision avoidance between cars and pedestrians based on their movement information. To achieve the goal we have to design and develop a dead reckoning method for both the car and the pedestrian. For a pedestrian, the dead reckoning system must be independent of the smartphone orientation and position. Also, develop a geometric collision avoidance algorithm and a centralised communication system to process the necessary information, the system makes the decision and communicates the result to the participants.

List of Figures

Figure 1-1 Typical accident scenario [12] ... 3

Figure 3-1 Sliding window algorithm with four segments each window and 0% overlapping ... 31

Figure 3-2 Sliding window algorithm with four segments each window and 75% overlapping ... 31

Figure 3-3 An example of a Decision tree ... 34

Figure 3-4 The pseudo-code of the J48 Algorithm in the Weka Tool ... 36

Figure 3-5 The pseudo-code of the JRip Algorithm in the Weka Tool ... 36

Figure 3-6 The pseudo-code of the K-nearest neighbour (kNN) adapted from [83] ... 37

Figure 3-7 The pseudo-code of the Bagging algorithm adapted from [83] ... 39

Figure 3-8 The pseudo-code of the AdaBoost.M1 algorithm [83] originally adapted from [87] and [98] ... 41

Figure 3-9 Case 1: Accuracy of pedestrian’s movement recognition based on the acceleration data [13] ... 44

Figure 3-10 Case 2: Accuracy of pedestrian’s movement recognition based on the acceleration data [13] ... 44

Figure 3-11 Average time needed for a single classification of a pedestrian movement with a confidence level of 95% [13] ... 45

Figure 3-12 Slowing down to stop between two GPS position measurements [13] ... 47

Figure 3-13 Slowing down to stop while a GPS position measurement is taken [13] ... 48

Figure 3-14 Reduction of false alarms with the help of acceleration sensor based [13] ... 49

Figure 4-1 Smartphone coordinate system (Device frame) [85] ... 54

Figure 4-2 User/pedestrian frame and device frame ... 54

Figure 4-3 Smartphone Orientation in the trouser pocket [85] ... 55

Figure 4-4 Compass sensor data before aligning smartphone orientation and user/pedestrian orientation by SOMDA [85] ... 58

Figure 4-5 Misalignment of compass sensor data and user/pedestrian movement direction [85] 59

Figure 4-6 Flowchart of SOMDA [85]........ 63

Figure 4-7 Accelerometer pattern and Thresholds during step detection [85] 63

Figure 4-8 Compass sensor data after aligning smartphone orientation and user/pedestrian orientation by SOMDA [85] 65

Figure 4-9 Misalignment of compass sensor data after aligning smartphone orientation and user/pedestrian movement direction by SOMDA [85] 66

Figure 4-10 Accuracy of SOMDA for different sampling rates [85]........ 66

Figure 4-11 Test path 1-4 69

Figure 4-12 SOMDA and Assisted-GPS comparison while walking 69

Figure 4-13 SOMDA and Assisted-GPS comparison, standing on one point and turning 0-360 degrees 70

Figure 4-14 Accuracy of compass with GPS ground heading 70

Figure 5-1 E1: Car is parked at different distances from the stationary smartphone [7]........ 77

Figure 5-2 E1: Magnetic deviation caused by different cars at different distances [7]........ 77

Figure 5-3 E2: Car is parked parallel & perpendicular to the smartphone [7]........ 77

Figure 5-4 E2: Magnetic deviation caused by parallel and perpendicular position [7]........ 77

Figure 5-5 E3: Car passes the smartphone at a distance of 2m during the different orientation of the smartphone [7] 78

Figure 5-6 E3: Magnetic deviation caused by a moving car at different orientation of the smartphone [7] 78

Figure 5-7 E4: Magnetic deviation caused by a moving car at different distances [7]........ 78

Figure 5-8 Movement direction of pedestrians (equally distributed in all directions) 81

Figure 5-9 Safe and endangered pedestrians without magnetic deviation 81

Figure 5-10 Safe and endangered pedestrians with magnetic deviation 82

Figure 5-11 Efficiency of all filtering pedestrians [7]... 82

Figure 5-12 Compensation algorithm for magnetic deviation [7].............................. 85

Figure 5-13 Compensation of magnetic deviation [7]... 86

Figure 5-14 pdf of noisy sensor data and t-location-scale distribution 90

Figure 5-15 cdf of noisy sensor data 90

Figure 5-16 pdf of error calculated in noisy sensor data 91

Figure 5-17 cdf of error calculated in noisy sensor data 91

Figure 5-18 pdf of compensated compass data and t-location-scale distribution.. 92

Figure 5-19 cdf of compensated compass data 92

Figure 5-20 pdf of error calculated in compensated compass data 93

Figure 5-21 cdf of error calculated in compensated compass data 93

Figure 5-22 cdf comparison of noisy sensor data and compensated compass data 94

Figure 5-23 cdf comparison of error calculation of noisy and compensated compass sensor data 94

Figure 6-1 Device based measurement setup [18] 102

Figure 6-2 Software based measurement applications (left: iPhone, right: Android) [18]... 103

Figure 6-3 Energy consumption of iPhone4 sensors... 104

Figure 6-4 Energy consumption of Samsung Galaxy SII GT I9100 sensors 104

Figure 6-5 Energy consumption of Samsung Galaxy SII GT I9100G sensors 105

Figure 6-6 Energy consumption of Samsung Galaxy SIII GT I9300..................... 105

Figure 6-7 Battery run time (hours) using API based measurements 107

Figure 6-8 Battery run time (hours) using device based measurements 108

List of Tables

Table 4-1 Compass Azimuth Features (Degrees) before SOMDA Alignment [85] 59

Table 4-2 Compass Azimuth features (Degrees) after applying SOMDA [85] 67

Table 6-1 Comparison of energy consumption (in %) between software based and device based measurements 109

Table 6-2 Energy consumption and Battery runtime of iPhone4 using software based technique 111

Table 6-3 Energy consumption and Battery runtime of iPhone4 using device based technique 111

Bibliography

[1] Federal Highway Research Institute, “IRTAD: International Road Traffic & Accident database”, Bergisch Gladbach, Germany, Online available: https://www.bast.de/BASt_2017/EN/Statistics/staistics_node.html, last checked: 25.03.2019.

[2] R. H. Rasshofer, D. Schwarz, E. Biebl, C. Morhart, O. Scherf, S. Zecha, R. Grünert, and R. Frühauf, “Pedestrian Protection Systems using Cooperative Sensor Technology”, in Advanced Microsystems for Automotive Applications AMMA, Vol. 2, New York, USA, Springer-Verlag, pp. 135–145.

[3] D. Schwarz, "Pedestrian Protection Systems using Cooperative Sensor Technology", presentation, BMW research group and technology, May 2007.

[4] C. Sugimoto, Y. Nakamura, and T. Hashimoto, “Prototype of pedestrian-to-vehicle communication system for the prevention of pedestrian accidents using both 3G wireless and WLAN communication”, in Proc. of the International Symposium on Wireless Pervasive Computing, Santorini, Greece, May 7-9, 2008, pp. 764–767.

[5] A. F. L. Flach, “Kontextbasiertes System von Kollisionen zwischen Autos und Fußgängern”, Ph.D. Dissertation, University of Kassel, Germany, 2012.

[6] K. David and A. Flach, “CAR-2-X AND PEDESTRIAN SAFETY Innovative Collision Avoidance System”, in IEEE Vehicular Technology Magazine, March 2010, pp. 70-76.

[7] A. Memon, Sian Lun Lau, and K. David, “Investigation and Compensation of the Magnetic Deviation on a Magnetometer of a Smartphone Caused by a Vehicle”, in IEEE 78th Vehicular Technology Conference (VTC Fall), Las Vegas, USA, 2013, pp. 1–5.

[8] M. Kühn, and V. Schindler, “Assessment of vehicle related pedestrian safety”, in the 19th International Technical Conference on the Enhanced Safety of Vehicles, Washington DC, USA, 2005, pp. 1-9.

[9] J. R. Crandall, K. S. Bhalla, and N. J. Madeley, “Designing road vehicles for pedestrian protection”, in the British Medical Journal (BMJ), vol. 324, no. 7346, May 2002, pp. 1145–1148.

[10] W. Koch and M. Howard, “Comprehensive approach to increased pedestrian safety in pedestrian-to-car accidents”, in Proc. of the Institute of Mechanical Engineering, Part D, vol. 217, no. 7, February 2003, pp. 513–519.

[11] Volvo car UK, “Volvo's Safety World First Set To Help Further Reduce the 1.3 Million People Killed On Roads: The Pedestrian Airbag Technology”, press release, online available: https://www.media.volvocars.com/uk/en-gb/media/pressreleases/43844, last checked: 25.03.2019.

[12] A. Flach and K. David, “A physical analysis of an accident scenario between cars and pedestrians”, in the IEEE 70th Vehicular Technology Conference (VTC 2009), Anchorage, Alaska, USA, September 20-23, 2009, pp. 1-5

[13] A. Flach, A. Q. Memon, S. L. Lau, and K. David, “Pedestrian movement recognition for radio based collision avoidance: A performance analysis”, in the IEEE 73rd Vehicular Technology Conference (VTC Spring), Budapest, Hungary, May 15-18, 2011, pp. 1 –5

[14] eMarketer, "Smartphone Users and Penetration Worldwide, 2014-2020 (billions, % of mobile phone users and % change)”, Online available: https://www.emarketer.com/Chart/Smartphone-Users-Penetration-Worldwide-2014-2020-billions-of-mobile-phone-users-change/188679, last checked: 25.03.2019.

[15] R. Paucher, M. Turk, "Location based augmented reality on mobile phones", in the twenty third IEEE conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, USA, June 13-18, 2010, pp. 1-8.

[16] X. Hu, Y. Liu, Y. Wang, Y. Hu, "Auto calibration of electronic compass for augmented reality", in Proc. of the International Symposium on Mixed and Augmented Reality (ISMAR’05), Santa Barbara, USA, October 5-8, 2005, pp. 1-2.

[17] S. W. Lee and K. Mase, “Incremental motion-based location recognition”, in the Fifth International Symposium on Wearable Computers, ETH Zurich, Switzerland, October 7-9, 2001, pp. 123 –130.

[18] I. König, A. Q. Memon, K. David, "Energy consumption of the sensors of Smartphones", in the Tenth International Symposium on Wireless Communication Systems ISWCS 2013, TU Ilmenau, Germany, August 27-30, 2013, pp. 1-5.

[19] R. Stirling, K. Fyfe, and G. Lachapelle, “Evaluation of a New Method of Heading Estimation for Pedestrian Dead Reckoning Using Shoe Mounted Sensors”, Journal of Navigation, vol. 58, no. 01, pp. 31-45, 2004.

[20] M. Khider, P. Robertson, M. Frassl, M. Angermann, L. Bruno, M. G. Puyol, E. M. Diaz and O. Heirich, "Characterization of Planar-Intensity Based Heading Likelihood Functions in Magnetically Disturbed Indoor Environments", in the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Montbeliard-Belfort, France, October 2013, pp. 1-10.

[21] A. R. Jimenez, F. Seco, J.C. Prieto and J. Guevara, “Indoor Pedestrian Navigation using an INS/EKF framework for Yaw Drift Reduction and a Foot-mounted IMU”, in the 7th workshop on Positioning, Navigation and Communication, Dresden, Germany, March 2010, pp. 135-143.

[22] M. Romanovas, V. Goridko, A. Al-Jawad, M. Schwab, M. Traechtler, L. Klingbeil, Y. Manoli, “A Study on Indoor Pedestrian Localization Algorithms with Foot-Mounted Sensors”, in the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2012), Sydney, Australia, November 2012, pp. 1-10,

[23] N. Castaneda and S. Lamy-Perbal, “An improved shoe-mounted inertial navigation system”, in the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2010), Zurich, Switzerland, September 2010, pp. 1-6.

[24] S. Godha, G. Lachapelle, and M. E. Cannon, “Integrated GPS/INS System for Pedestrian Navigation in a Signal Degraded Environment”, in Proc. of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2006), Fort Worth TX, USA, September 2006.

[25] T. Gadeke, J. Schmid, M. Zahnlecker, W. Stork, K.D. M¨uller-Glaser, “Smartphone Pedestrian Navigation by Foot-IMU Sensor Fusion”, in the IEEE Conference on Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS 2012), Helsinki, Finland, October 2012, pp. 1-8.

[26] E. Foxlin, “Pedestrian tracking with shoe-mounted inertial sensors”, in the IEEE Computer Graphics Applications, vol. 25, 2005, pp. 38-46.

[27] X. Yun, E. R. Bachmann, J. Calusidan, “Self-contained position tracking of human movement using small inertial/magnetic sensor modules”, in Proc. of the IEEE International Conference on Robotics and Automation, Rome, Italy, April 10–14, 2007, pp. 2526-2533.

[28] F. Woyano, S. Lee and S. Park, "Evaluation and Comparison of Performance Analysis of Indoor Inertial Navigation system Based on Foot Mounted IMU", in the 18th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, South Korea, 2016, pp. 792-798.

[29] L. Fang, P. J. Antsaklis, L. Montestruque, M. B. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie, “Design of a wireless assisted pedestrian dead reckoning system - the NavMote experience”, in the IEEE Transactions on Instrumentation and Measurement, vol. 54, no. 6, 2005, pp. 2342-2358.

[30] K. Kunze, P. lukowicz, K. Partridge, and B. Begole, "Which way am I facing: Inferring horizontal device orientation from an accelerometer signal", in Proc. of the International Symposium on Wearable Computers-ISWC, Linz, Austria, 2009, pp. 149-150.

[31] Z. A. Deng, J. Yu, Z. Na, Y. Zhou and D. Wu, "Heading estimation indoors using a smartphone in the pocket", in the 10th International Conference on Communications and Networking in China (ChinaCom), Shanghai, China, 2015, pp. 361-364.

[32] J. Qian, J. Ma, R. Ying, P. Liu, L. Pei, “An Improved Indoor Localization Method Using Smartphone Inertial Sensors”, in the 4th International Conference on Indoor Positioning and Indoor Navigation (IPIN 2013), Montbeliard-Belfort, France, October 2013, pp. 1-7.

[33] X. Zhu, Q. Li, and G. Chen, “APT: Accurate outdoor pedestrian tracking with smartphones”, in the IEEE Annual Joint Conference: INFOCOM, IEEE Conference on Computer Communication (INFOCOM 2013), Turin, Italy, April 2013, pp. 2508–2516.

[34] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, “A Reliable and Accurate Indoor Localization Method Using Phone Inertial Sensors”, in Proc. of the 2012 ACM Conference on Ubiquitous Computing, New York, NY, USA, 2012, pp. 421–430.

[35] F. Hong, H. Chu, L. Wang, Y. Feng, Z. Guo, “Pocket Mattering: Indoor Pedestrian Tracking with Commercial Smartphone”, in the IEEE 2012 International Conference on

Indoor Positioning and Indoor Navigation (IPIN 2012), New South Wales, Sydney, Australia, Vol. 13, November 2012, pp. 1-7.

[36] J. Borenstein, L. Ojeda, and S. Kwanmuang, "Heuristic Reduction of Gyro Drift For Personnel Tracking Systems", in the Journal of Navigation, Vol 62, No 1, January 2009, pp. 41-58.

[37] W. Kang, S. Nam, Y. Han, S-J. Lee, “Improved Heading Estimation for Smartphone-Based Indoor Positioning Systems”, in the IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2012), New South Wales, Sydney, Australia, September 2012, pp. 2429-2453.

[38] Y. Jin, H. S. Toh, W. S. Soh, W. C. Wong, “A Robust Dead-Reckoning Pedestrian Tracking System with Low Cost Sensors”, in IEEE International Conference on Pervasive Computing and Communications (PerCom 2011), Seattle, USA, March 2001, pp. 222-230.

[39] S. Ayub, A. Bahraminasab, and B. Honary, “A Sensor Fusion Method for Smartphone Orientation Estimation”, in the 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool, UK, June 2012, pp. 1-6.

[40] S. Ayub, B.M. Heravi, A. Bahraminasab, and B. Honary, “Pedestrian Direction of Movement Determination Using Smartphone”, in the 6th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2012), Paris, France, September 2012, pp. 64–69.

[41] N. Roy, H. Wang, and R. Roy Choudhury, “I Am a Smartphone and I Can Tell My User’s Walking Direction,” in Proc. of the 12th ACM Annual International Conference on Mobile Systems, Applications, and Services, New York, NY, USA, 2014, pp. 329–342.

[42] N. Roy, “Walkcompass: Finding Walking Direction Leveraging Smartphones Inertial sensors”, Master thesis, School of Computer Science and Engineering, University of South California, USA, January 2013.

[43] A. Ali, H. W. Chang, J. Georgy, Z. Syed, C. Goodall, “Heading Misalignment Estimation between Portable Devices and Pedestrians”, in the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation, Nashville, USA, September 2013, pp. 1626-1633.

[44] U. Steinhoff and B. Schiele, “Dead reckoning from the pocket - An experimental study,” in IEEE International Conference on Pervasive Computing and Communications (PerCom 2010), Manheim, Germany, March-April 2010, pp. 162–170.

[45] S. Khandelwal, S. Sharma, P. Kumar, “Tracking location in the absence of GPS using smartphones,” in International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 4, April 2012, pp. 400–403.

[46] Paonni M., Anghileri M., Ávila-Rodríguez J.A., Wallner S., Eissfeller B., “Performance Assessment of GNSS Signals in Terms of Time to First Fix for Cold, Warm and Hot

Start", in Proc. of the International Technical Meeting of the Institute of Navigation: ION ITM 2010, 25-27 January, San Diego, CA, USA, pp. 1-16.

[47] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone", in Proc, of the 19th USENIX Annual Technical Conference (USENIX ATC 2010), 2010, Boston, MA, USA, pp. 1-16.

[48] L. Zhang, B. Tiwana, R. P. Dick, Z. Qian, Z. M. Mao, Z. Wang and L. Yang , "Accurate online power estimation and automatic battery behavior based power model generation for smartphones", in the IEEE/ACM/IFIP International Conference on Hardware/Software Code sign and System Synthesis (CODES+ISSS), 2010, Scottsdale, AZ, USA, pp. 105-114.

[49] A. Serra, T. Dessi, D, Carboni, V. Popescu, L. Atzori, "Inertial Navigation Systems for User-Centric Indoor Applications," in Proc. of the NEM Summit- Towards Future Media Internet, Barcelona, Spain, October 2010, pp. 1-5.

[50] A. Serra, D. Carboni, V. Marotto, "Indoor Pedestrian Navigation System Using a Modern Smartphone", in Proc. of the 12th international conference on Human computer interaction with mobile devices and services (Mobile HCI 2010), Lisbon, Portugal, September 7-10, 2010, pp. 397-398.

[51] B. Shin, "Indoor 3D Pedestrian Tracking Algorithm Based on PDR using Smartphone", in the 12th International Conference on Control, Automation and Systems (ICCAS 2012), Jeju Island, South Korea, October 2012, pp. 1442,1445.

[52] D. Pai, M. Malpani, I. Sasi, N. Aggarwal, and P. S. Mantripragada, "Padati: A Robust Pedestrian Dead Reckoning System on Smartphones", in the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2012), Liverpool, UK, 2012, pp. 2000-2007.

[53] J. Ruiz, A. Ramon, F. Seco Granja, P. Honorato, J. Carlos, and G. Rosas, "Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID", in the IEEE Transactions on Instrumentation and Measurement, Volume 61, Issue 12011, pp. 178-189.

[54] J.-O. Nilsson, I. Skog, and P. Händel, "A note on the limitations of ZUPTs and the implications on sensor error modelling", in Proc. of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia, 13-15th November 2012, pp. 1-4.

[55] M. Ren and H. A. Karimi, "Movement Pattern Recognition Assisted Map Matching for Pedestrian/Wheelchair Navigation", in the journal of Navigation, vol. 65, no. 04 2012, pp. 617–633.

[56] L. Wang, P. D. Groves, M. K. Ziebart, "Smartphone Shadow Matching for Better Cross-street GNSS Positioning in Urban Environments", in the journal of the Royal Institute of NavigationVolume 68, Issue 3, 2014, pp. 411-433.

[57] P. D. Groves, Z. Jiang, L. Wang and M. K. Ziebart, "Intelligent Urban Positioning using Multi-Constellation GNSS with 3D Mapping and NLOS Signal Detection", in Proc. of the 25th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2012), Nashville, Tennessee, September 17-21, 2012, pp. 458-472.

[58] V. Radu and M. K. Marina, “HiMLoc: Indoor Smartphone Localization via Activity Aware Pedestrian Dead Reckoning with Selective Crowdsourced WiFi Fingerprinting”, in the IEEE 2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN 2013), Montbeliard-Belfort, France, October 2013, pp. 1-7.

[59] Andreas Jahn, " Benefit from unobtrusive contexts: Towards the recognition of short and non-periodic activities", Ph.D. Dissertation, University of Kassel, Germany, 2017.

[60] L. Rybok, B. Schauerte, Z. Al-Halah, and R. Stiefelhagen, “Important stuff, everywhere Activity recognition with salient proto-objects as context”, in the IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, USA, 2014, pp. 646–651.

[61] B. Begole, “Ubiquitous Computing for Business: Find New Markets, Create Better Businesses, and Reach Customers Around the World 24-7-365”, FT Press, 2011.

[62] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a Better Understanding of Context and Context-Awareness”, in Proc. of the 1st International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany: Springer-Verlag, 1999, pp. 304–307.

[63] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data”, in the International Conference on Pervasive Computing (Pervasive 2004), volume 3001 of Lecture Notes in Computer Science, Springer, April 2004, pp. 1–17.

[64] N. Kern, B. Schiele, and A. Schmidt, “Recognizing context for annotating a live life recording”, in journal of Personal Ubiquitous Computing- Memory and sharing experiments, vol. 11, Issue. 4, 2007, pp. 251–263.

[65] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data”, in Proc. of the 17th conference on Innovative Applications of Artificial Intelligence (IAAI 2005), Pittsburgh, Pennsylvania Volume 3, July 2005, pp. 1541-1546.

[66] Y-S. Lee, S-B. Cho, “Activity recognition using hierarchical hidden markov models on a smartphone with 3D accelerometer”, in Proc. of the 6th International Conference on Hybrid Artificial Intelligence Systeme (HAIS’11), volume part 1, Wroclaw, Poland, 2011, pp. 460–467.

[67] S. Zhang, P. McCullough, C. Nugent, H. Zheng, “Activity Monitoring Using a Smart Phone's Accelerometer with Hierarchical Classification”, in Proc. of the IEEE Sixth International Conference on the Intelligent Environments (IE), Kuala Lumpur, Malaysia, 19–21 July 2010, pp. 158–163.

[68] G. Bieber, J. Voskamp, and B. Urban, “Activity recognition for everyday life on mobile phones”, in Proc. of the 5th International on Conference Universal Access in Human-Computer Interaction (UAHCI 2009), San Diego, USA, 2009, pp. 289–296.

[69] T. Brezmes, J. Gorricho, and J. Cotrina, “Activity recognition from accelerometer data on a mobile phone”, in Proc. of the 10th International Work-Conference on Artificial Neural Networks (IWANN 2009), Salamanca, Spain, 2009, pp. 796–799.

[70] S. L. Lau, I. König, K. David, B. Parandian, C. Carius-Düssel, and M. Schultz, “Supporting Patient Monitoring using Activity Recognition with a Smartphone”, in Proc. of the Seventh International Symposium on Wireless Communication Systems (ISWCS 2010), York, UK, September 2010, pp. 810-814.

[71] R. Perez-Torres, and C. Torres-Huitzil, “A power-aware middleware for location & context aware mobile apps with cloud computing interaction”, in the World Congress on Information and Communication Technologies (WICT 2012), Trivandrum, India, November 2012, pp. 691-696.

[72] T. Gandhi, M. M. Trivedi, “Pedestrian Protection Systems: Issues, Survey and challenges”, in the IEEE Transactions on intelligent Transportation systems, vol. 8, no. 3, September 2007, pp. 413-430.

[73] Y. Abramson and B. Steux, “Hardware-friendly pedestrian detection and impact prediction,” in Proc. of the IEEE Intelligent Vehicle Symposium, Parma, Italy, June 2004, pp. 590–595.

[74] T. Gandhi and M. M. Trivedi, “Vehicle mounted wide FOV stereo for traffic and pedestrian detection,” in Proc. of the International Conference of Image Processing, vol. 2, September 2005, pp. 121–124.

[75] B. Fardi, I. Seifert, G. Wanielik, and J. Gayko, “Motion-based pedestrian recognition from a moving vehicle,” in Proc. of the IEEE Intelligent Vehicle Symposium, Tokyo, Japan, Jun. 2006, pp. 219–224.

[76] M. Bertozzi, A. Broggi, P. Grisleri, T. Graf, M. Meinecke ”Pedestrian Detection in Infrared Images ”, in Proc. of the IEEE International Vehicle Symposium, Columbus, OH, USA, 2003, pp. 662-667.

[77] S. Milch and M. Behrens, “Pedestrian Detection With Radar and Computer Vision”, Braunschweig, Germany, 2001, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.9264, last checked in: 19.03.2022.

[78] M. Töns, R. Doerfler, M.-M. Meinecke, and M. A. Obojski, “Radar sensors and sensor platform used for pedestrian protection in the EU funded project SAVE-U”, in Proc. of the IEEE Intelligent Vehicle Symposium, Parma, Italy, June 2004, pp. 813–818.

[79] K. Fuerstenberg and V. Willhoeft, “Object tracking and classification using laser scanners—Pedestrian recognition in urban environment”, in Proc. of the IEEE Intelligent Transportation System Conference, Oakland, USA, Aug. 2001, pp. 451–453.

[80] A. Fackelmeier, C. Morhart, E. Biebl, "Dual-frequency Methods for Identifying Hidden Targets in Road traffic", in title of the Advanced Microsystems for Automotive Applications, 2008, Springer Verlag, Berlin, Germany, pp. 11-20.

[81] C. Sugimoto, Y. Nakamura, T. Hashimoto, “Prototype of pedestrian-to- vehicle communication system for the prevention of pedestrian accidents using both 3G wireless and WLAN communication”, in Proc. of the International Symposium on Wireless Pervasive Computing (ISWPC), Santorini, Greece, May 2008, pp. 764-767.

[82] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani, “An online algorithm for segmenting time series”, in Proc. of the 2001 IEEE International Conference on Data Mining (ICDM 01). Washington DC, USA, 2001, pp. 289-296.

[83] S. L. Lau, “Towards a user-centric context aware system: empowering users through activity recognition using a smartphone as an unobtrusive”, Ph.D. Dissertation, University of Kassel, Germany, 2011.

[84] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection”, in Proc. of the 14th International Joint Conference on Artificial Intelligence (IJACAI’95), San Francisco, CA, USA, 1995, pp. 1137-1143.

[85] R. Kusber, A. Q. Memon, D. Kroll, and K. David, "Direction Detection of Users Independent of Smartphone Orientations", in the IEEE 82nd Vehicular Technology Conference (VTC2015 Fall), Boston, USA, 6-9 September 2015, pp. 1-5.

[86] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update”, in the ACM Special Interest Group on Knowledge Discovery in Data (SIGKDD), vol. 11, Issue 1, 2009, pp. 10-18.

[87] Y. Freund, and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting”, in Proc. of the Second European Conference on Computational Learning Theory (EuroCOLT'95), Barcelona, Spain, 1995, pp. 23-37.

[88] L. Breiman, “Bagging Predictors”, in the journal of the Machine Learning, vol. 24, Edition 2, Springer Nature, 1996, pp. 123-140.

[89] P.-N. Tan, M. Steinbach, and V. Kumar, “Classification: Basic Concepts, Decision Trees, and Model Evaluations”, in the Introduction to Data Mining, Chapter 4, Boston, MA, USA, Addison Wesley Longman Publishing Co., Inc., 2006, pp. 25-44.

[90] L. I. Kincheva, “Base Classifiers”, in the Combining Pattern Classifiers: Methods and Algorithms, Chapter 2, Section 2, Canada, John Willey & Sons, Inc., 2014, pp. 55-66.

[91] J. R. Quinlan, “C4.5: Programs for machine learning”, in the journal of the Machine Learning, vol. 16, San Francisco, CA, USA, Morgan Kaufman Publishers Inc., 1993, pp. 235-240.

[92] W. W. Cohen, "Fast effective rule induction," in Proc. of the 12th International Conference on Machine Learning (ICML'95), Tahoe city, Californea, USA, 1995, pp. 115-123.

[93] J. Rissanen, “Modeling By Shortest Data Description”, in the journal of Automatica, vol. 14, issue 5, 1978, pp. 465-571.

[94] J. R. Quinlan, “MDL and categorical theories (continued)”, in Proc. of the 12th International Conference on Machine Learning, Tahoe City Californea, USA, 1995 pp. 464-471.

[95] J. Fürnkranz, “Separates and conquer rule learning”, in the journal of the Artificial Intelligence Review, vol. 13, issue 1, Springer, 1999, pp. 3-54.

[96] S. Raudys and A. Jain, "Small sample size effects in statistical pattern recognition: recommendation for practitioners", in the journal of the IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 3, IEEE computer Society, pp. 252-264, March 1991.

[97] L. Breiman, "Random forests," in journal of the Machine learning, vol. 45, issue 1, Springer, 2001, pp. 5-32.

[98] Y. Freund, and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting”, in the journal of Computer and System Sciences, vol 55, issue 1, August 1997, pp. 119-139.

[99] K. van Laerhoven, N. Kern, H.-W. Gellersen, and B.Schiele, "Towards a wearable inertial sensor network", in the IEEE Eurowearable Conference, Birmingham, UK, 2003, pp. 125-130.

[100] K.S. Kunze, O. Luckowicz, H. Junker, and G. Tröster, "Where am i: Recognizing on-body positions of wearable sensors", in the Lecture Notes in Computer Science, vol. 3479, Springer 2005, pp. 264-275.

[101] C. Lombriser, N.B. Bharatula, D. Roggen, and G. Tröster, "On body activity recognition in a dynamic sensor network", in Proc. of the 2nd International Conference on Body area networks (BodyNets 07), Brussels. 2007, pp. 1-6.

[102] J. Yang, "Towards physical activity diary: motion recognition using simple acceleration features with mobile phone", in Proc. of the 1st International Workshop on Interactive multimedia for consumer electronics, Beijing, China 2009, pp. 1-10.

[103] S. L. Lau and K. David, "Movement recognition using the accelerometer in smartphones," in Future Network and Mobile Summit, Florence, Italy, 2010, pp. 1-8.

[104] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data”, in Proc. of the 17th conference on Innovative applications of artificial intelligence (IAAI 2005), Pittsburgh, Pennsylvania Volume 3, July 2005, pp. 1541-1546.

[105] A. Dalton and G. O. Laighin, "Identifying activities of daily living using wireless kinematic sensors and data mining algorithms", in the Sixth International Worksop on Wearable and Implementable Body Sensor Networks (BSN 2009), Berkeley, CA, USA, June 2009, pp. 87-91.

[106] S.J. Preece, J.Y. Goulermas, L.P. Kenney, D. Howard, K. Meijer, R. Crompton, "Activity identification using body-mounted sensors—A review of classification techniques", in the journal of Physiological. Measurement. Vol. 30, 2009, pp. 1-34.

[107] A. Bulling, U. Blanke, B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors" in Proc. of the ACM Computing Surveys , vol. 46, issue 3, Association for computing Machinery, New York, NY, USA, 2014, pp. 1-34.

[108] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga, "A Survey of Online Activity Recognition Using Mobile Phones", in the journal of Sensors, vol. 15, no. 1, pp. 2059–2085, Jan. 2015, pp. 2059–2085,.

[109] Y. Liang, X. Zhou, Z. Yu, B. Guo, Y. Yang, "Energy Efficient Activity Recognition Based on Low Resolution Accelerometer in Smart Phones", in the Lecture Notes Computer Science, vol. 7296, Springer, 2012, pp. 122–136.

[110] E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng, A.T. Campbell, "Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application", In Proc. of the 6th ACM Conference on Embedded Network Sensor Systems, Raleigh, NC, USA, 4–7 November 2008, pp. 337–350.

[111] P.Siirtola, J. Roning, "Ready-to-use activity recognition for smartphones", in Proc. of the 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Singapore, 16–19 April 2013, pp. 59–64.

[112] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, K. Aberer, "Energy-Efficient Continuous Activity Recognition on Mobile Phones: An Activity-Adaptive Approach", in Proc. of the 2012 16th International Symposium on Wearable Computers (ISWC), Newcastle, Australia, 18–22 June 2012, pp. 17–24.

[113] Mustafa. Kose, O.D. Incel, C.Ersoy, "Online Human Activity Recognition on Smart Phones", in Proc. of the Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data, Beijing, China, 16 April 2012, pp. 11–15.

[114] P. Siirtola, "Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data", in the journal of: Special Issue on Distributed Computing and Artificial Intelligence, vol.1, 2012, pp. 38–45.

[115] S. Thiemjarus, A. Henpraserttae, S. Marukatat, "A study on instance-based learning with reduced training prototypes for device-context-independent activity recognition on a mobile phone" in Proc. of the 2013 IEEE International Conference on Body Sensor Networks (BSN),Cambridge, MA, USA, 6–9 May 2013, pp. 1–6.

[116] B. Das, A. Seelye, B. Thomas, D. Cook, L. Holder, M. Schmitter-Edgecombe, "Using smartphones for context-aware prompting in smart environments" in Proc. of the 2012 IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 14–17 January 2012 pp. 399–403.

[117] M. Berchtold, M. Budde, D. Gordon, H. Schmidtke, M. Beigl, "ActiServ: Activity Recognition Service for mobile phones" in Proc. of the 2010 International Symposium on Wearable Computers (ISWC), Seoul, Korea, 10–13 October 2010, pp. 1–8.

[118] United Nations Population Division, Department of Economics and Social Affairs, "World population prospects: the 2017 Revision", Online available: https://population.un.org/wpp/Download/Standard/Population/, last checked 25.03.2019.

[119] Statista- Number of mobilephone users worldwide from 2015-2020 (in billions), online available: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/, last checked 19.11.2018.

[120] A. Brajdic and R. Harle, “Walk detection and step counting on unconstrained smartphones”, in Proc. of the ACM international joint conference on Pervasive and Ubiquitous Computing (UbiComp 2013). Zurich, Switzerland, September 8-12, 2013, pp. 225-234.

[121] Handbook of Magnetic Compass Adjustment, Chapter 10- Transient deviation of the magnetic deviation, Published: Defense Mapping Agency Hydrographic/Topographic Center, Washington DC, 2004, pp: 40-41.

[122] Q. Ladetto, J. van Seters, S. Sokolowski, Z. Sagan, and B. Merminod, “Digital Magnetic Compass and Gyroscope for Dismounted Soldier Position & Navigation”, in Proc. of the NATO-RTO Meetings: Military Capabilities enabled by Advances in Navigation Sensors & Electronics Technology, Istanbul, Turkey, 2002, pp. 1-15.

[123] M. Caruso and L. S. Withamwasam, “Vehicle detection and compass applications using AMR magnetic sensor,” Honeywell International Inc., Plymouth, USA, 1999, pp. 1-13.

[124] Y. Cui, J. Chipchase, and F. Ichikawa, “A cross culture study on phone carrying and physical personalization”, in Proc. of the 12th international conference on human-computer interaction, Beijing, China, 22–27 July 2007, pp. 483–492.

[125] K. Fujinami, "On-Body Smartphone Localization with an Accelerometer", in the journal: Information 2016: Future Information Technology and Intelligent Systems (special issue), volume 7, 2016, pp. 1-23.