UNIVERSIDAD DE OVIEDO

ESCUELA DE INGENIERÍA INFORMÁTICA

MASTER THESIS

"COOPERATIVE AND MULTISENSOR
INDOOR LOCATION SYSTEM"

DIRECTOR: Daniel Fernández Lanvin

CODIRECTOR: Martín González Rodríguez

Vº Bº del Director del Proyecto

AUTHOR: Juan José Marrón Monteserín

Acknowledgements

This Master's thesis was carried out partly at Department of Computer Science and Engineering at University of South Florida (Tampa ,USA) and partly at the Department of Computer Science at University of Oviedo (Asturias, SPAIN).

I would like to give my gratitude to the people who helped me to complete the research presented in this thesis. In particular I wish to thank Dr. Miguel A. Labrador for his insight, guidance and assistance. I appreciate my director Dr. Daniel Fernandez Lanvin and codirector Dr. Martin Gonzalez Rodriguez for their comments and suggestions, which have enabled me to make significant improvements to the quality of this work. Similarly, I thank Adrian Menendez Valle who has contributed with his time and knowledge.

Finally I wish to thank my parents, family and important friends for their love and support during last months.

Resumen

El seguimiento de personas en situaciones de movimiento en interiores es a día de hoy uno de los retos más importantes para dispositivos de navegación personal y en aplicaciones para sistemas globales de navegación.

La capacidad de seguir y reconocer de forma exacta el comportamiento y las secuencias de actividades que llevan a cabo las personas, supondría un crecimiento significante en las posibilidades y utilidad del conjunto de aplicaciones basada en localización, especialmente en entornos de interiores.

El objetivo de este trabajo de investigación es implementar y evaluar un conjunto de algoritmos que permitan el seguimiento ubicuo de los movimientos que los humanos realizamos en el interior de edificaciones utilizando únicamente sensores integrados en los dispositivos móviles de última generación como smartphones o tablets.

El diseño de estos algoritmos está basado en dos fuentes de información: por una lado el concomimiento biomecánico y de las secuencias de movimiento que el cuerpo humano produce en la realización desplazamientos cotidianos como andar y subir o bajar escaleras. Por otro lado el reconocimiento de patrones en los datos recogidos por los sensores; ya que ciertos eventos que los usuarios realizan en interiores como giros en esquinas o el uso de ascensores, producen huellas identificables en el conjunto de valores adquiridos.

La evaluación de nuestro sistema de seguimiento muestra buenos niveles de exactitud en los algoritmos que detallan las actividades una vez identificadas y resultados aceptables en la compleja tarea de clasificar correctamente entre las posibles actividades.

Nuestras propuestas en el tratamiento de las señales recogidas a partir de los múltiples sensores integrados en gran variedad de dispositivos portables, permiten aumentar su funcionalidad siendo capaces de reconocer automáticamente entre un conjunto de actividades humanas y de crear rutas de seguimiento de los movimientos de los humanos en escenarios de interiores.

Palabras Clave

Localización en interiores, dispositivos móviles, Android, reconocimiento de actividades, adquisición de datos, múltiples sensores, acelerómetro, giroscopio, magnetómetro, computación móvil.

Abstract

Tracking pedestrians in indoor environments is one of the most challenging areas of application for Global Navigation Systems (GNSS) in personal navigation devices. The capacity of following and recognize pedestrian behavior accurately can lead to significant growth in location-based applications, especially in indoor scenarios.

The goal of this research is to present and evaluate a group of algorithms for accurate and ubiquitous tracking indoor pedestrian activity using exclusively wearable sensors embedded in smartphones. The knowledge in biomechanical patterns of the human body while accomplishing basic activities (such as walking, or climbing up and down stairs) , with the identifiable signatures that certain indoor locations (as turns or elevators) introduce on sensing data, are the base for our set of algorithms.

The experiment results show a good level of accuracy in the motion tracking algorithms and an adequate classification of indoor activities.

Our approach of signal processing leverages the smart phones sensors functionality to recognize automatically human activity and to create motion traces for following pedestrian movements in indoor locations.

Keywords

Indoor location and positioning, mobile devices, Android, activity recognition, data acquisition, sensor fusion, accelerometer, gyroscope, magnetometer, pervasive computing.

Contents

List of tables

List of figures

List of equations

Chapter 1. Introduction

1.1 Motivation

Impact of mobile computing devices such as smartphones or tablets is nowadays overtaking the popularity of traditional desktop computers. The paradigm of computing has evolved in recent years reducing the prizes of the devices and extending its features, processing power and mobility capabilities. Thus, the interests in location-aware services and new modalities of context interaction have been spurred as well as the development of a platform for a whole range of mobile applications that can make use of location, particularly in wellness sector [1] and in the entertainment business.

This rapid growth in people-centric mobile computing applications has led to improvements in localization technologies, not only in terms of localization accuracy, but also across multiple and specific dimensions as power consumption, ubiquity or universality of access to the positioning. Outdoor localization is successfully solved by Traditional GNSSs, such as Global Positioning System (GPS)-based scheme or cell tower localization. However, these technologies are unable of tracking the user's position in indoor scenarios where most of them spend the majority of their daily lives (offices, schools, universities, malls...). In contrast to outdoor services, most of indoor location systems need some form of additional infrastructure and/or require a higher level of accuracy that is not provided by the state of the art of the technology.

Several solutions have been proposed that partially solve some of the problems related with indoor location like Hybrid Indoor Navigation Platforms [3] [6]. However, most of them are based on strong requirements of the environment or on the use of supplementary hardware devices that makes them unpractical for most of the sceneries.

On the other hand, modern portable devices are equipped with Micro electro Mechanical Systems (MEMS) that can avoid the dependency on an external structure. Despite the low-performance and noisy sensors data, applying proper signal processing and algorithms over the raw data received form the array of inertial sensors (triaxial orthogonal accelerometer, gyroscope and magnetic field detector), acceptable location accuracy results can be achieved. The use of internal sensors embedded in the last generation of mobile devices also introduces advantages in terms of power consumption, simplicity of management and reduction of cost of calibration versus external data sources or infrastructures.

1.2 Aims and objectives

In this paper we present a smartphone based system for pedestrian activity recognition and indoor tracking, working without any additional infrastructure or extra sensors. Our approach models human indoor behavior in terms of turns (T), stationary times (Sy), use of elevators (E) walking (W) and stairs (St). The algorithms proposed to recognize each of these activities are combined with track-splitting and land marking strategies that help to reduce the accumulated error usually inherent to in every inertial sensor based system.

1.3 Challenges

Indoor location introduces a series of challenges that the current systems are not able to cope completely successful:

- **Complexity of indoor environment** [3]: Machines, walls, corridors, open areas, humans movements and the irregularity of the scenario, introduce noise that may affect the quality of the signal reception as well as the accuracy of the inertial sensors measurement.
- **High level of location accuracy required** [4]: Because the indoor context vary at fine spatial granularity, indoor required fairly location accuracy. A minor indoor location error can be easily translated in a wrong or useless position.
- **Optimization of energy consumption** [2]: In most of the cases the indoor location is aimed for battery-dependent devices with constraints regarding the energy consumption.
- **Poor or absence of floor plans database** [5]: While maps for outdoor are traditionally provided by several services, such as Google Maps[1], Bing Maps[2], Apple Maps[3], OpenStreetMaps[4] , the indoor equivalent floor plans are currently very limited, affecting the representation and spread of the concerned applications.

1.4 Possible uses and applications

Providing location services within an indoor environment may be used in several scenarios such as [2]:

- **Safety**: location systems could provide emergency services with an immediate knowledge of where users are inside a building at any time.
- **Security**: location-awareness could permit automatic locking of sensitive resources if the owner is not present.
- **Resource-efficiency**: smart buildings can use the information of where its users are to optimize air conditioned, heating lighting and other resources.
- **Social networking**: allowing users to efficiently find colleagues.
- **Automatic resource routing**: follow-me applications allowing users with visual impairments to be routed to their goals.

1 http://www.google.com/maps
2 http://www.bing.com/maps/
3 http://www.apple.com/ios/maps/
4 http://www.openstreetmap.org/

- **Leisure**: reproduce automatic explanations in museums and gallery arts.
- **Navigation**: visitors unfamiliar with a building can quickly navigate to rooms of interest.
- **Advertisements**: adapting offers and advertising in shopping centers depending on the client position.

1.5 Structure

This report begins with a short introduction (this first chapter) where mainly the motivation, objectives, challenges and possible applications are presented.

Chapter 2 details the previous research done in the area and the related works.

Next Chapter describes the theory considered in this thesis: the built-in inertial sensors, their performance and error characteristics. Also the development platform and the equipment used for the project are described.

The groundwork of this thesis is found in chapter 4 were we describe the key concepts and the system approach.

Chapter 5 describes the modules and algorithms which take part in the system design.

The implementation of the prototype software is defined in Chapter 6.

In chapter 7 we include the evaluation methodology and we discuss final the results of the system performances.

Finally chapter 8 and chapter 9 with conclusions and future works, concludes this thesis.

Chapter 2. Related work

2.1 Background

Several techniques have been proposed to enable indoor location and there is a large body of literature regarding to this topic. A comprehensive coverage about this field is provided by these two work surveys related with inertial systems [2] and wireless positioning systems [7].

Perhaps the most successful indoor systems to date are those based on radio fingerprinting. Here, signal properties such as received strength are compared to a radio map previously collected at a variety of location. The closest match is returned as the estimated position. WiFi is a common choice due to its ubiquity [8] [9]. Accuracies of a few meters are typically reported. However the expense and time required to install, configure and maintain these systems has so far prohibited general deployment.

In contrast with providing coordinate location, a common technique for human tracking is to employ inertial sensors. These systems typically compute their own positions and their key advantage is that they require very little, if any, physical infrastructure to work. These sensors systems are based on user's estimated relative location and they need to be capable of handling noise that would normally lead to cumulative locating errors. They also usually offer an optional degree of location privacy since the user can choose not to share the information with any third party.

2.2 Current alternatives and systems

Previous work on inertial sensor-based user tracking have employed bare functional locations for mounting the sensor or provided low accuracy. Some of them limit the usage of sensors, use additional infrastructures to handle this inherent problem, or require previous knowledge of the indoor map:

Robertson et al. [10] proposed the use of accurate foot-mounted inertial sensors to track pedestrians in indoor environments. These systems provide the heading and the moving distance of the pedestrian but the sensor placement is limited to an unnatural position in order to obtain a correct sensor reading. Displacement can be obtained from the acceleration signal by double integration with respect to time. However, due to the low accuracy of the accelerometer, the presence of noise and the component of acceleration due to the gravity, error accumulates rapidly with time [11] [12].

In [3], Jin et al. propose the use of a digital compass and an accelerometer in a smartphone to track pedestrian location and corrects the location in sparse indoor environments. This scheme relies on an additional ultrasonic sensor, which is sparsely distributed, to adjust the current location directed by the smartphone. Building a radio map [13] is another approach solving the issues of a foot-mounted inertial sensor-based pedestrian tracking system. The scheme employs a particle filter to track the pedestrian using radio map information as one of the parameters for the filter.

Capelle et al. [6] designed a GNSS–based multisensor system based on the fusion of three different technologies: High Sensitivity GNSS (GPS and the future Galileo), MEMS-based Pedestrian Navigation System and WIFI.

More recently Pu proposed WiSee [14], a system that leverages wireless signals like WiFi to enable whole-home sensing and recognition of human gestures and movements.

All these designs solve at least partially the indoor localization problem. However, all of them require additional infrastructure or offline training in order to build a radio map, a strong requirement that will determine the use of these solutions in real environments.

An also interesting approach is the one used by Constandache et al. in [15]. In this work, they use a digital compass and an accelerometer in a smartphone for user tracking. The system is designed in outdoor environments where map information is provided. The system compares the estimated path with the map information, without requiring any external extra device.

Finally, A. J. Ruiz-Ruiz et al. [16] in present new matching techniques using scale invariant features from the images captured by the smartphone camera, in order to build accurate location-based services.

Chapter 3. Theory

This chapter describes the basic knowledge considered and technologies used during this project, as well as their functionality and error characteristics. Also the development platform and equipment used for the project is described.

3.1 Global Positioning System

3.1.1 Overview

A Global Positioning System, also known as GPS, is a satellite-based navigation system designed to help navigate on the Earth, in the air, and on water[5].

The system was created by the United States Department of Defense. In the beginning, it was only used by the U.S. military, but in 1983 an order allow anyone to use the system. The system was declared fully operative in 199. Today the GPS is used also for civilian purposes, such as surveying, map design, tectonics and obviously navigation.

A GPS receiver shows where it is. It also shows how fast it is moving, which direction it is going, how high it is, and maybe how fast it is going up or down. Many GPS receivers have information about places.

Most GPS receivers record where they have been, and help plan a journey. While traveling a planned journey, it predicts the time to the next destination.

3.1.2 Components

The GPS system consists of three segments; the space segment (satellites), the control segment and the user segment (receivers)

The space segment consists of 30 satellites 20,200 kilometers (12,600 mi) above the Earth orbiting in six orbits which have an inclination of 55° relative to the Equator. The orbits are arranged in such way, that far from the North Pole and South Pole, a GPS unit can receive signals from 6 to 12 satellites at once. GPS satellites send navigation messages continuously at a rate of 50 bits per second; the main information in the message is the time when the message was sent, exact orbital information(“ephemeris”), and the general system health and rough orbits of all satellites (the “almanac”).

The control segment is composed of a number of stations and antennas, they are used to control and monitor the health of the satellites, and do necessary corrections when needed, for instance adjust the satellites’ clocks.

The user segment consists of military users of the GPS Precise Positioning Service and the

[5] http://en.wikipedia.org/wiki/Global_Positioning_System

civil users of the Standard Positioning Service. The GPS receiver is mainly composed of an antenna, a very stable internal clock, the software for calculating the user's location and speed, and usually a display for providing the information to the user. Now, more are part of something else such as mobile phones, wrist watches, and cars.

A GPS receiver can calculate its position many times in one second. A GPS receiver calculates its speed and direction by using its change in position and change in time. Many inexpensive consumer receivers are accurate to 20 meters (66 ft) almost anywhere on the Earth.

3.1.3 Position calculation

GPS receivers use geometric trilateration to combine the information from different satellites to predict the correct location. As mentioned above, the GPS message contains information about the time when message was sent, precise orbital information, health of the system and rough information about the orbits of other satellites. The receiver uses the message to calculate the transfer time of each message and computes the distance to the satellite. With the aid of trilateration, the distances to the satellites together with the satellites' locations are used to calculate the position of the receiver.

3.1.4 Limitations and constraints

The advantage of satellite systems is that receivers can determine latitude, longitude, and altitude to a high degree of accuracy. However, line of sight (LOS) is required for the functioning of these systems. This leads to an inability to use these systems for an indoor environment where the LOS is blocked by walls, roofs and other objects.

To track GPS signals indoors typically requires a receiver capable of tracking signals with power levels ranging from -160dBW to -200dBW, however a typical receiver has a noise floor of around -131dBW. Multipath effects are likely to cause degradation in accuracy even if a receiver is able to track signals from a sufficient number of satellites. Unlike in outdoor environments, reflected signals are often stronger than those received via direct line-of-sight when indoors. Hence such systems are not accurate enough for indoor location-aware applications, nearly all of which require at least room-level accuracy.

3.2 Inertial Navigation System

INS (Inertial Navigation System) is based on a self-contained navigation technique in which measurements provided by motion sensors (accelerometers), environment sensors (magnetometer) and rotation sensors (gyroscopes) are used to track the position and orientation of an object relative to a known starting point, orientation and velocity without the need for external references.

3.2.1 Accelerometer

An accelerometer is a device that measures proper acceleration. The proper acceleration measured by an accelerometer is not necessarily the coordinate acceleration (rate of change of velocity). Instead, the accelerometer sees the acceleration associated with the phenomenon of weight experienced by any test mass at rest in the frame of reference of the accelerometer device. For example, an accelerometer at rest on the surface of the earth will measure an acceleration g=9.81 m/s^2 straight upwards, due to its weight. By contrast, accelerometers in free fall or at rest in outer space will measure zero.

Accelerometers have multiple applications in industry and science. Highly sensitive accelerometers are components of inertial navigation systems for aircraft and missiles. Accelerometers are used to detect and monitor vibration in rotating machinery. Accelerometers are used in tablet computers and digital cameras so that images on screens are always displayed upright.

The most important source of error of an accelerometer is the bias. The bias of an accelerometer is the offset of its output signal from the true value, in m/s^2. A constant bias error of ε, when double integrated, causes an error in position which grows quadratically with time. The accumulated error in position is show in Equation 1:

$$S(t) = \varepsilon \frac{t^2}{2}$$

Equation 1: Accelerometer error position

where t is the time of the integration.

It is possible to estimate the bias by measuring the long term average of the accelerometer's output when it is not undergoing any acceleration. Uncorrected bias errors are typically the error sources which limit the performance of the device.

3.2.2 Gyroscope

Generally, a gyroscope is a device for measuring or maintaining orientation, based on the principles of conservation of angular momentum[6].

A conventional (mechanical) gyroscope consists of a spinning wheel mounted on two gimbals which allow it to rotate in all three axes. An effect of the conservation of angular momentum is that the spinning wheel will resist changes in orientation. Hence when a mechanical gyroscope is subjected to a rotation, the wheel will remain at a constant global orientation and the angles between adjacent gimbals will change. A conventional gyroscope measures orientation, in contrast to MEMS (Micro Electro-Mechanical System) types, which measure angular rate, and are therefore called rate-gyros.

MEMS gyroscopes contain vibrating elements to measure the Coriolis effect. A single mass is

[6] http://en.wikipedia.org/wiki/Gyroscope

driven to vibrate along a drive axis, when the gyroscope is rotated a secondary vibration is induced along the perpendicular sense axis due to the Coriolis force. The angular velocity can be calculated by measuring this secondary rotation.

The Coriolis effect states that in a frame of reference rotating at angular velocity ω, a mass moving with velocity v experiences a force:

$$F_c = -2m(\omega * v)$$

Equation 2: Coriolis effect force

An important note to be made is that whereas the accelerometer and the magnetometer measure acceleration and angle relative to the Earth, gyroscope measures angular velocity relative to the body.

The bias of a rate gyro is the average output from the gyroscope when it is not undergoing any rotation (i.e: the offset of the output from the true value), in ˚/h. A constant bias error of when integrated, causes an angular error which grows linearly with time:

$$\theta(t) = \varepsilon * t$$

Equation 3: Angular gyroscope error

The constant bias error of a rate gyro can be estimated by taking a long term average of the Gyro's output whilst it is not undergoing any rotation. Once the bias is known it is trivial to compensate for it by simply subtracting the bias from the output.

Another error arising in gyros is the calibration error, which refers to errors in the scale factors, alignments, and linearity of the gyros. Such accuracy problems tend to produce errors that are only observed whilst the device is turning. These errors lead to the accumulation of additional drift in the integrated signal, the magnitude of which is proportional to the rate and duration of the motions.

3.2.3 Magnetometer

A magnetometer is an instrument used to measure the strength and/or direction of the magnetic field in the surrounding area of the instrument.

Magnetometers can be divided into two basic types: scalar magnetometers that measure the total strength of the magnetic field to which they are subjected, and vector magnetometers (the type used in this project), which have the capability to measure the component of the magnetic field in a particular direction, relative to the spatial orientation of the device.

Magnetometers can be used as metal detectors: they can detect only magnetic (ferrous metals, but can detect such metals at a much larger depth than conventional metal detectors; they are capable of detecting large objects, such as cars, at tens of meters, while a metal detector's range is rarely more than 2 meters.

In recent years magnetometers have been miniaturized to the extent that they can be incorporated in integrated circuits at very low cost and are finding increasing use as compasses in consumer devices such as mobile phones and tablet computers.

The two main sources of measurement errors are magnetic contamination in the sensor, errors in the measurement of the frequency and ferrous (iron containing) material on the operator and in the instruments. If the sensor is rotated as the measurement is made, an additional error is generated

3.3 Mobile plattform

3.3.1 Android Operative System

Android is an open-source platform for mobile devices[7]. It is developed and managed by Google Inc, and it includes operating system, middleware and key applications. Recently it has become the world's most used platform for smartphones. The increasingly high popularity together with the fact of being an open-source project has given the platform a very large community of developers. The mentioned characteristics are the main reasons for choosing Android as the development environment for this thesis rather than any other mobile platform like iOS, Symbian or Windows Phone.

Android is a Linux-based OS software; its stack is divided in four different layers which include five groups as illustrated in the Figure 1.

The layers in the Android architecture are:

- **Application layer**: This layer is the one used by end phone users. Applications can run simultaneously (multitasking) and they are written in Java language.
- **Application framework:** The software framework used to implement the standard structure of an application for the Android OS.
- **Libraries:** The available libraries are written in C/C++ and they are called through a Java interface.
- **Android runtime:** The Android runtime consists of two components. First a set of core libraries which provides most of the functionality in the Java core libraries. And second the virtual machine Dalvik which operates like a translator between the application side and the operating system.
- **The kernel:** This Linux based kernel is used by Android for its device drivers, memory management, process management and networking.

[7] http://en.wikipedia.org/wiki/Android_(operating_system)

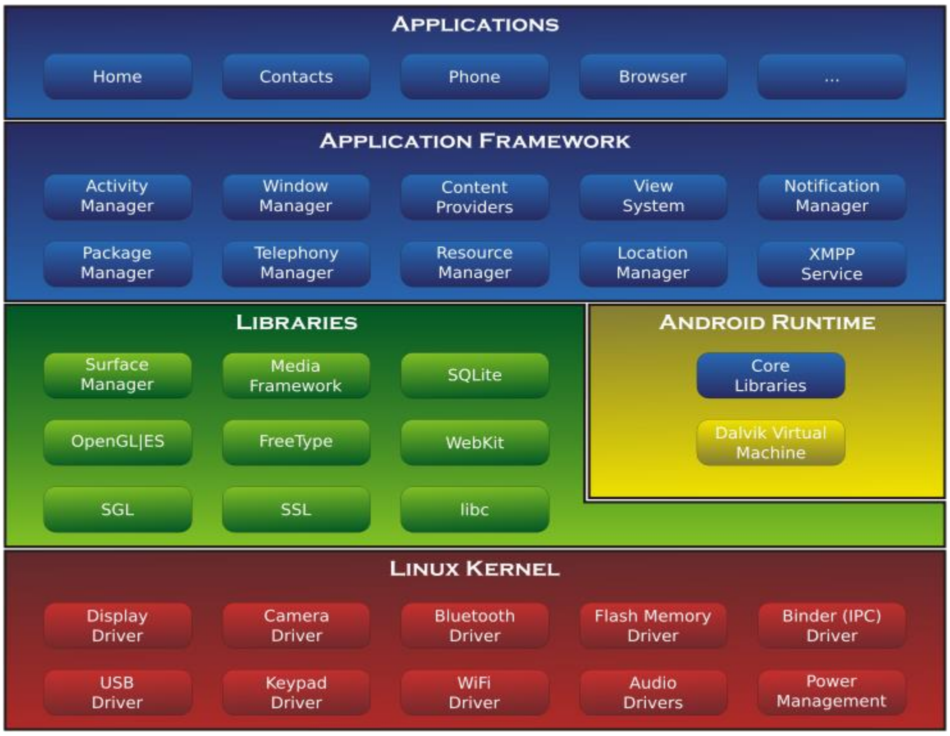


Figure 1: Android Architecture diagram

3.3.2 Andorid Sensor API

Most Android-powered devices have built-in sensors that measure motion, orientation, and various environmental conditions. These sensors are capable of providing raw data with high precision and accuracy, and are useful if developers want to monitor three-dimensional device movement or positioning, or you want to monitor changes in the ambient environment near a device.

The Android platform supports three broad categories of sensors:

- **Motion sensors:** These sensors measure acceleration forces and rotational forces along three axes. This category includes accelerometers, gravity sensors, gyroscopes, and rotational vector sensors.
- **Environmental sensors:** These sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers.
- **Position sensors:** These sensors measure the physical position of a device. This category includes orientation sensors and magnetometers.

The Android sensor framework lets developers access these three types of hardware-based sensors. Hardware-based sensors are physical components built into a handset or tablet device. They derive their data by directly measuring specific environmental properties, such as acceleration, geomagnetic field strength, or angular change.

Users can access these sensors and acquire raw sensor data by using the Android sensor framework. The sensor framework is part of the *android.hardware*[8] package and the layers of the architecture are show in Figure 2.

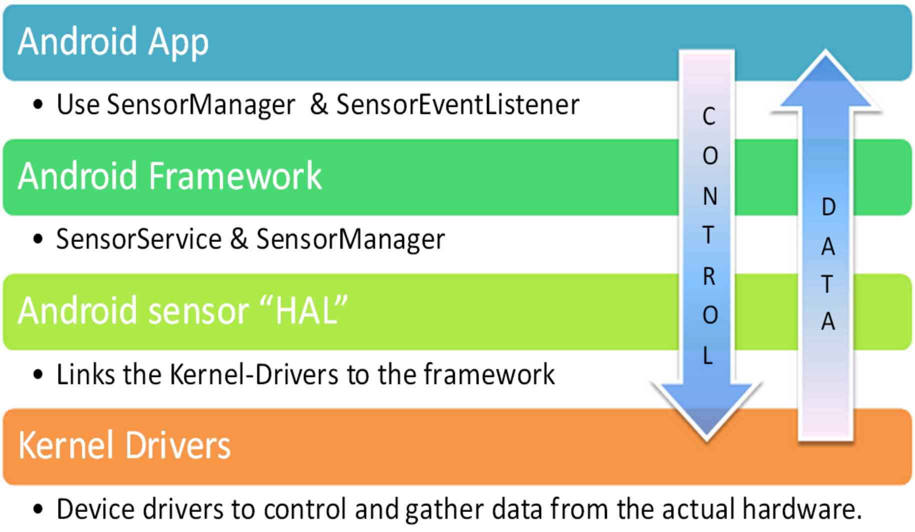


Figure 2: Android Sensor API layers

Appendix C is a session show explaining how to use the Android Sensor API and an example of implementation.

Samsung Galaxy S II SGH-I777 [9]	
Dimensions	126 x 66 x 8.9 mm (4.96 x 2.60 x 0.35 in)
Weight	121.9 g (4.27 oz)
Processor	Exynos C210 Dual-core 1.2 GHz Cortex-A9
Operating System	Android OS, v4.1.2 (Jellybean)
Memory	16GB storage, 1 GB RAM
Display	Super AMOLED Plus capacitive touchscreen 16M colors 480 x 800 pixels, 4.3 inches
Connectivity	GSM 850 / 900 / 1800 / 1900 HSDPA 850 / 1900 / 2100 Wi-Fi 802.11 a/b/g/n Bluetooth v3.0 with A2DP, HS
Battery	Li-Ion 1650 mA battery
Sensors	Accelerometer, gyro, proximity, compass, magnetometer

Table 1: Samsung Galaxy S II SGH-I777 specifications

[8] http://developer.android.com/reference/android/hardware/package-summary.html
[9] http://www.gsmarena.com/samsung_galaxy_s_ii_i777-4130.php

3.3.3 Samsung Galaxy SII

The AT&T Samsung Galaxy S II SGH-I777 (Figure 3) is the initial release for the Galaxy S II line of smartphones on AT&T's network. It keeps the 4.3" display seen in the international version. It uses a dual-core Exynos 4 processor clocked at 1.2 GHz. The AT&T Samsung Galaxy S II SGH-I777 uses an 8.0 MP rear camera, as well as a 1.9 MP front-facing camera. The cell phone is running Android version 4.1.2 (Jellybean).

Previous Table 1 details the Samsung Galaxy S II SGH-I777 specifications.

Figure 3: Samsung Galaxy S II SGH-I777

The main reason of choosing this device is the large number of built-in sensors. Furthermore, at the project start this cell phone was available in the department set of devices and was the only one including and supporting a gyroscope.

3.3.4 Inertial sensors quality tests

In this section an error analysis of the all considered information sources is performed. The inertial sensors are tested for errors and to consider the need for calibration before the outputs is used. The built-in inertial sensors in the Samsung SII are: 3-axes accelerometer, Gyroscope and Magnetometer.

To analyze the accuracy and the behavior of the sensors, stationary tests where the device was laying on a table, were performed. The output of the sensors is relative to the device's orientation, here referred to the device's coordinate system[10].

The device's coordinate system is defined relative to the screen of the phone in its default orientation. The axes are not swapped when the device's screen orientation changes.

The x axis is horizontal and points to the right, the y axis is vertical and points up, and the z axis points towards the outside of the front face of the screen (Figure 4).

[10] http://developer.android.com/guide/topics/sensors/sensors_overview.html

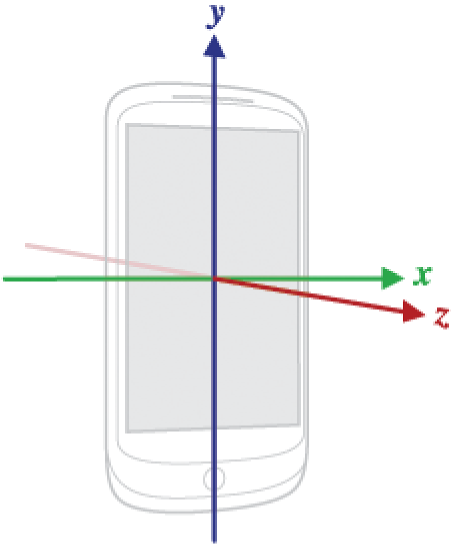


Figure 4: Coordinate system that is used by the Sensor API

In this error analysis test the samples are recorded during a period of approximately 15 seconds with the phone lying flat with its back on the table. The phone is immobile in order to prevent any force other than gravity from affecting the output.

3.3.4.1 Accelerometer

The accelerometer measures the acceleration in three axes in m/s^2. It outputs the acceleration applied to the device by measuring forces applied to the sensor. The measured acceleration is always influenced by the force of the earth's gravity:

$$a_d = -g - \sum \frac{F}{m}$$

Equation 4: Measured acceleration by the accelerometer

where a_d is the acceleration applied to the device, g the force of gravity, F the force acting on the device, and m the mass of the device. The sign Σ represents the sum of the x, y and z axis.

As a result, when the device is put on the table (and obviously not accelerating) the accelerometer output should read 0 m/s^2 for x and y axes, and negative earth's gravity of 9.81 m/s^2 for the z axe.

Figure 5 shows the phone's actual acceleration output for the x, y and the z axes while being stationary on the table.

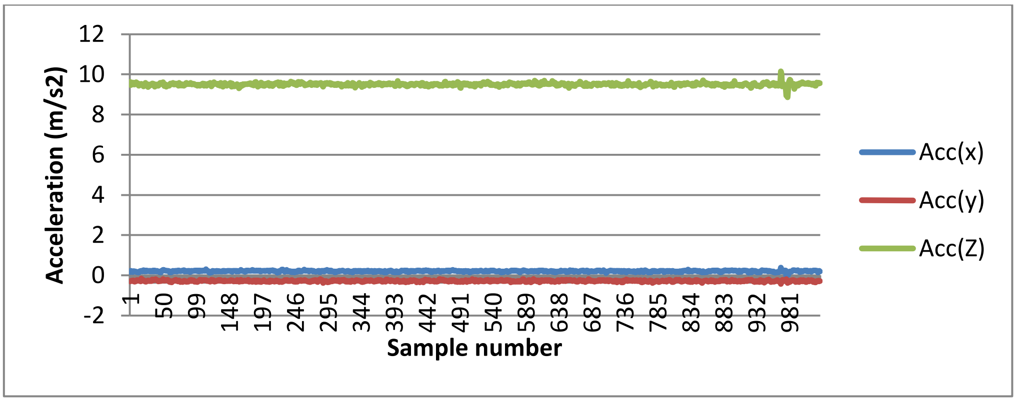


Figure 5: Acceleration output x, y and z axes

The accelerometer test (Table 2) shows that the total acceleration measured at stationary position was on average about 9.642 m/s^2 and not the expected 9 .81 m/s^2. The standard deviation for the total acceleration, 0.13 m/s^2, is equivalent to more than one percent of the total acceleration, which, with time, could potentially generate a large error.

Acceler. test	Average	Max	Min	Std deviation
Acc_x[m/s^2]	0.204663	0.38137	-0.05448	0.029994
Acc_y[m/s^2]	-0.27933	-0.16344	-0.43585	0.03228
Acc_z[m/s^2]	9.50077	10.14716	8.853226	0.070122
Total Acc [m/s^2]	9.426103	10.36508	8.362893	0.132402

Table 2: Accelerometer test results

3.3.4.2 Gyroscope

The gyroscope values are in radians/second and measure the rate of rotation around the x, y and z axis. Rotation is positive in the counter-clockwise direction. When the device is at rest on a table and not moving, the gyroscope values should read a magnitude of 0 radians per second.

Figure 6 shows the measured angular speed around the x, y, and z axis when the device is stationary on the table. (Values of the x axis are made invisible on the graph because of the offset from y and z axis).

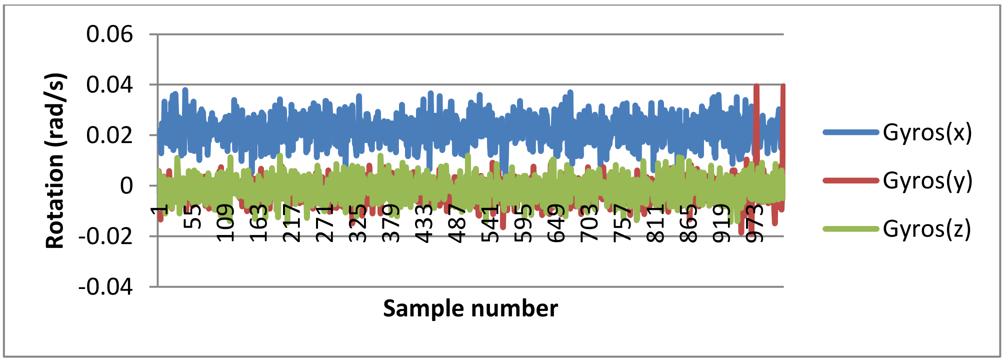


Figure 6: Rotation output x, y and z axes

Gyroscope test	Average	Max	Min	Std deviation
ω_x [rad/s]	0.022092	0.037874	0.003054	0.005496
ω_y [rad/s]	-0.00231	0.041844	-0.08247	0.005628
ω_z [rad/s]	-0.00119	0.011912	-0.01466	0.004645
Total [rad/s]	0.018586	0.09163	-0.09407	0.015769

Table 3: Gyroscope test results

Observing the results in Table 3 it can be seen that there exists an offset (the bias) on all of the three axes. To calculate angle α, the output of the gyroscope ω (angular speed), is integrated over time t :

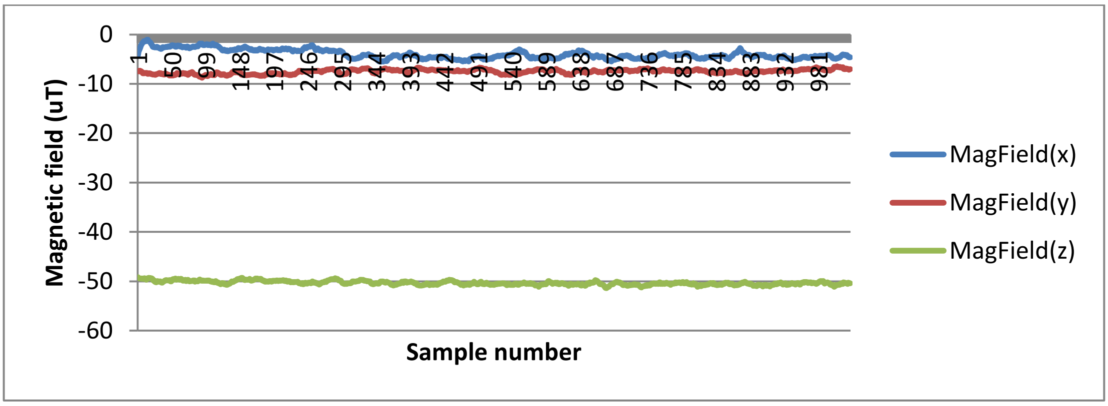
$$\alpha = \sum(\omega * \Delta t)$$

Equation 5: Error angle estimation from angular speed

During 15 seconds we get an error angle of 0.27 rad. Since the expected error average is zero, the calculated average of the measured values is approximately equal to the offset error.

3.3.4.3 Magnetometer

The magnetometer measures the strength of the ambient magnetic field in micro-Tesla (uT), in the x, y and z axes. The magnetometer output together with accelerometer values can be passed to a function to get the orientation.



shows the measured magnetic field in the x, y and z axis respectively.

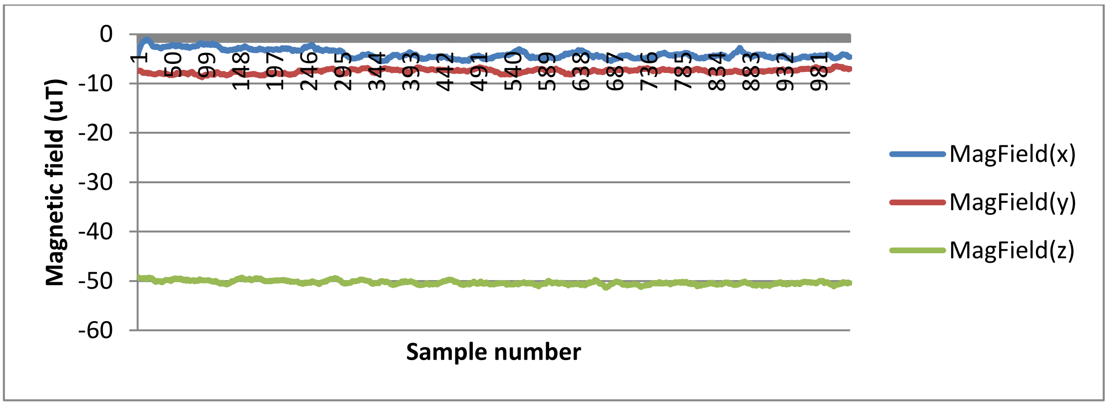


Figure 7: Magnetic field output x, y and z axes

Statistics from the test are in**Error! Reference source not found.**

Magnetom. test	Average	Max	Min	Std deviation
Mag_x [uT]	-3.99611	-1.08	-5.52	0.916693
Mag_y [uT]	-7.52068	-6.42	-8.76	0.413968
Mag_z [uT]	-50.3266	-49.2	-51.36	0.39095

Table 4: Magnetometer test results

One interesting observation is the sensitivity of the sensor, which results in a rather high standard deviation with an average of 0.57 uT for each axis.

Chapter 4. System approach

This section contains all the requirements specification and the analysis of the application, from which the design will be developed later.

4.1 System definition

Requiring infrastructure beyond the common mobile phone can make a solution unpractical for several kinds of scenarios. On the other hand, recent developments in Pedestrian Dead-Reckoning (PDR) systems have demonstrated the ability of these systems to urban sensing and activity [15] [17] recognition. For instance, several inertial sensors worn simultaneously on different parts of the body can detect when a user is walking, turning into a corridor, or climbing up the stairs; microphones and magnetometers can detect ambient sounds and magnetic fluctuations [18][19]. While these signatures have been primarily used for various forms of context awareness, they allow themselves to localization as well. The signatures can be treated as landmarks, and be useful for indoor dead-reckoning systems to combine sensor information in order to recognize ambiences and user behavior.

Figure 8: Floor plan example. First floor of fine Arts Hall Building at University of South Florida

Our work is based on the tracking of pedestrians in indoor environments by automatically detecting landmarks and pedestrian motion traces where map data are not provided a priori. Based on the accelerometer readings of the mobile phone, it is possible to record the number of steps/stairs a person has walked/climbed [17][20][21] and therefrom derive the displacement of the person. By using the compass, the direction of each heading change can be tracked [22] and finally, using magnetometer readings anomaly context behavior can be detected [23]. Additionally the proposed technique depends on resetting the accumulations of error by splitting the complete trajectory in independent motion traces.

4.2 Indoor actions

In order to model indoor human behavior, we reduce the possible movements or activities to five states or actions: turns (T) applied when the pedestrian changes the heading in her/his route, stationary (Sy) where the person remains in the same location. Elevator (E) when the user makes use of an elevator to move to a different floor. Walking (W) moving across the same floor and finally stairs (St) where the user goes up and down to change the floor.

Consider for example the floor plan of the first floor map of fine Arts Hall Building at University of South Florida shown in Figure 8. This real location is basically formed by: (i) corridors where user can walk straight (W); (ii) corners were pedestrians make heading changes (T); (iii) stairs where user climb up/down (St) steps or (iv) elevators where user go up/downs floors automatically (E). Also, (v) we consider stationary time (Sy) where the user remains in the same position for a defined period of time. This set of predictable activities can be translated to identifiable context signatures or landmarks and detailed using the data gathered by sensors integrated in a modern mobile phone. For instance, elevators exhibit a remarkable variation in the magnetic field magnitude, human gait can be identified by a repetitive pattern in the accelerometer raw data, and heading changes may be obtained from the gyroscope measures. We take advantage of this approach to simultaneously harness sensor-based dead-reckoning and environment sensing.

4.3 Key concepts

Our proposal defines a set or rules based on two key concepts that automatically allow the system to detect the activity among the possible states defined above. These two fundamentals are:

- **Indoor points of interest (POI):** our tests showed that certain locations in indoor scenarios present identifiable signatures on one or more sensing dimensions. Processing the raw data gathered by the sensors these signatures can be detected and translated into the real indoor events. This principle is used in our design to detect elevator and heading changes.

- **Human body behavior patterns:** pedestrian actions, like walking, generate repetitive and identifiable patterns that are ubiquitously detectable by the inertial sensors. This idea is used to detect human gait and climbing stairs. For instance, human gait is defined as bipedal, biphasic forward propulsion of center of gravity of human body, in which there is alternate sinuous movements of different segments of the body with least expenditure of energy. The gait cycle begins when one foot contacts the ground and ends when that foot contacts the ground again. Thus, it can be classified in two phases: stance and swing (Figure 9).Each cycle begins at initial contact with a stance phase and proceeds through a swing phase until the cycle ends with the limb's next initial contact. Stance phase accounts for approximately 60 percent, and swing phase for approximately 40 percent, of a single gait cycle.

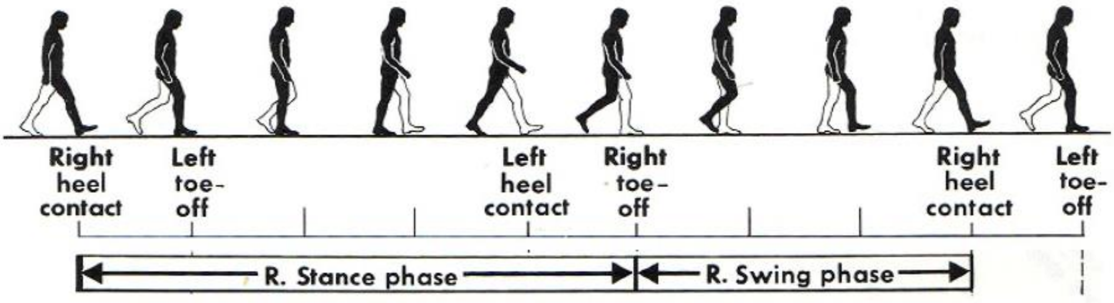


Figure 9: Human gait cycle [11]

The approach of navigation based on landmarks and split by activity frames reduce accumulative error indoor constraints. Furthermore processing signal algorithms are applied over each one of the frames including additional information like: number of steps/stairs, time in the detected action, distance walked/climbed, elevator direction and turn direction. Thanks to this extra information our system is able to rebuild detailed traces of user's motion useful for pedestrian tracking, indoor positioning or participatory floor plan construction.

[11] http://www2.warwick.ac.uk/fac/sci/eng/meng/nongps/rnd/gait/

Chapter 5. System design

Our architecture consists of four main part or modules: the data collection module; the traces segmentation module; the activity recognition and the activity detailer module that apply the specific algorithms (Step detection and Counting, Stairs detection and counting and elevator frame classification).

5.1 Data collection module

This module is responsible for collecting measurements from the various sensors embedded in the users' mobile devices. The measurements gathered can be treated locally in the device or buffered and then sent opportunistically to the server for instantly or later processing.

Data collected are measurements from inertial sensors including: accelerometers, gyroscopes and magnetometers. These sensors have the advantage of being ubiquitously installed on a large class of smart phones, having a low-energy footprint, and being always on during the phone operation to detect changes in the orientation of the phone. Our approach does not require calibration and the main challenge that needs to be addressed is handling the noise that low cost quality these inertial sensors introduce.

In our test we developed the data collection module for an Android OS cell phone. The raw data has been collected every 15ms (66.6 Hz). This duty cycle is good enough to detect user's activity and motion details.

5.2 Traces segmentation module

Since corners are common occurrences in indoor scenarios, an important event to be detected in indoor traces is the change on heading directions.

Turns can be recognized based on the gyros sensor measurements applying the algorithm explained below.

5.2.1 Turns detection algorithm

The algorithm implemented for turnings detection is based on the finding of the significant changes in the gyroscope readings. Turns are identified when the compass headings change more significantly than due to random oscillations. The algorithm based on thresholds is based on the following steps:

1) Calibration Routine

During a segment of movement, samples of compass [g(x), g(y), g(z)] are collected to estimate the mean in the compass values (see Equation 6). The energy of the average is calculated and considered as bias to compensate (Equation 7).

$$[\overline{Gyro}(x), \overline{Gyro}(y), \overline{Gyro}(z)] = \frac{1}{N}\left[\sum_{i=0}^{i=N} gx_i\, , \sum_{i=0}^{i=N} gy_i\, , \sum_{i=0}^{i=N} gz_i\right]$$

Equation 6: Mean of three axis compass values

$$\overline{Gyro}_{calib} = \sqrt{\overline{Gyro}(x)^2 + \overline{Gyro}(y)^2 + \overline{Gyro}(z)^2}$$

Equation 7: Energy of compass averages

Where N is the number of samples used in the calibration routine for the movement frame

2) Moving average filter

Following Equation 8 we compute the energy of the compass samples, gi, for every sample I.

$$gi = \sqrt{{gx_i}^2 + {gx_i}^2 + {gz_i}^2}$$

Equation 8: Energy of a simple compass sample

Estimate the average of energy in a set window of size ω and compensate the bias (see Equation 9)

$$\overline{g_\iota} = \frac{1}{2\omega + 1} \sum_{j=i-\omega}^{j=i+\omega} (gj - \overline{Gyro}_{calib})^2$$

Equation 9: Average of energy for a compass sample in a set window

3) Thresholding

The threshold defined in Equation 10 is applied to detect the heading changes.

$$G_T = \left\{ \begin{array}{ll} T_1 & if \quad \overline{g_\iota} > T_1 \\ 0 & otherwise \end{array} \right\}$$

Equation 10: Threshold to detect heading changes

With the threshold T_1 fixed to the value 1.2

4) Turn detection

A turn is detected when the following condition is satisfied: a transition from low to high in the new signal of compass values $T_{1_{i-1}} < T_{1_i}$ and, samples next, a transition from high to low $T_{1_{i-1}} > T_{1_i}$). The turn sample is considered in the average value of the T1 high level step.

5) Turn direction

Once a turn is detected for a sample i, to determine its direction is necessary to study the sign of the compass reading with the biggest magnitude for the set of three axes.

$$MainSignal\ (t_i) = max\{\ \overline{gx_i}\ , \overline{gy_i}\ , \overline{gz_i}\} \qquad \text{with} \qquad \overline{g_i} = \frac{1}{2\omega + 1} \sum_{i=j-\omega}^{i=j+\omega} abs(g_i)$$

Equation 11: MainSignal function definition

If the *MainSignal* (Equation 11) value for the detect turn sample is positive, a turn left has been detected. Otherwise, if the sign is negative a turn to the right has been performed.

Figure 10 represents the value of the important signals in the algorithm applied in a basic example with the following turn's sequence: 72 - Right, 247 - Left, 568 - Right and 608-Right.

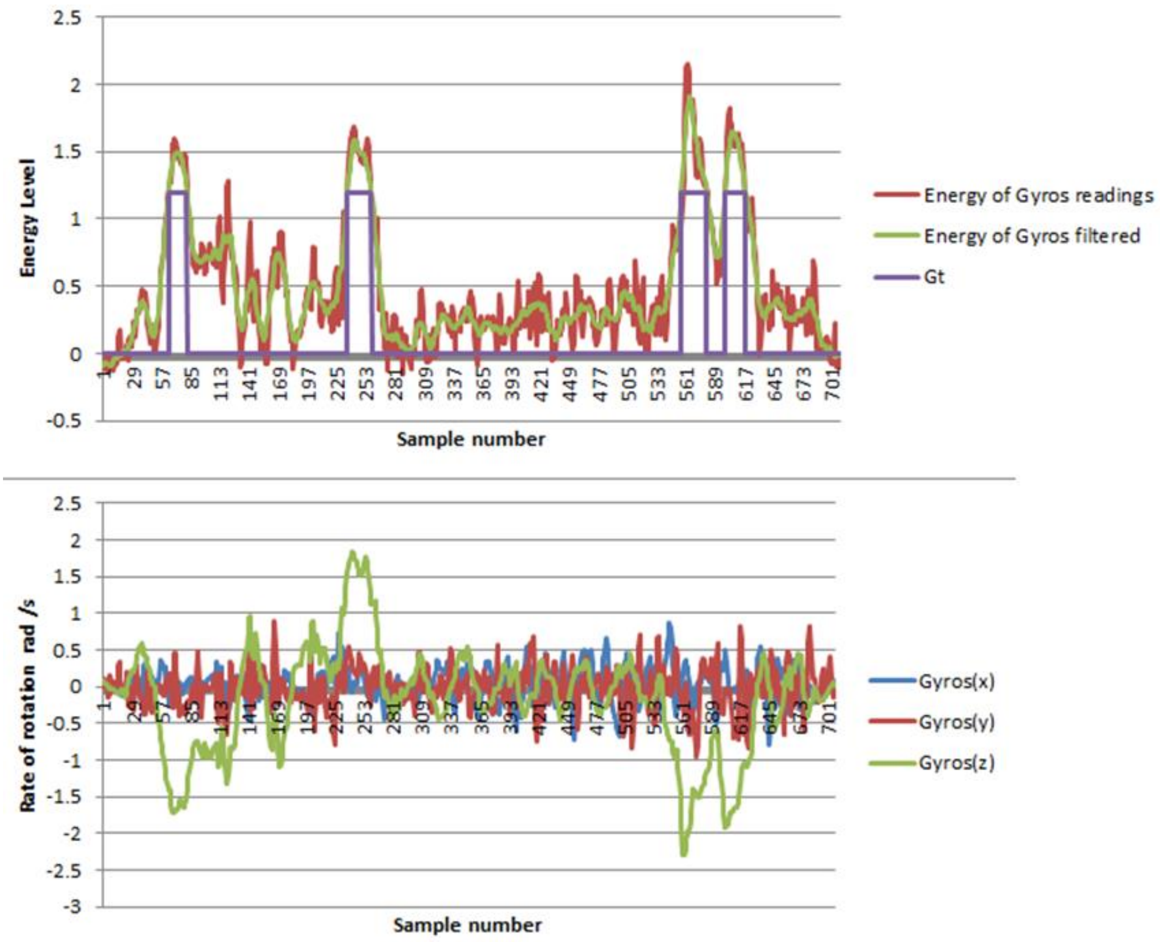


Figure 10: Turns detection algorithm signals

These *turn* landmarks (T) are used in our solution to split the continuous motion sequence in independent traces. Figure 11 shows an example of an indoor motion traced split by turns. These approach contrasts with the classical relative navigation techniques as dead reckoning, where the new location of a user is estimated with the help of the start location the distance traveled and direction of motion. Traces segmentation avoids the error accumulation introduced by the inertial sensors using relative techniques.

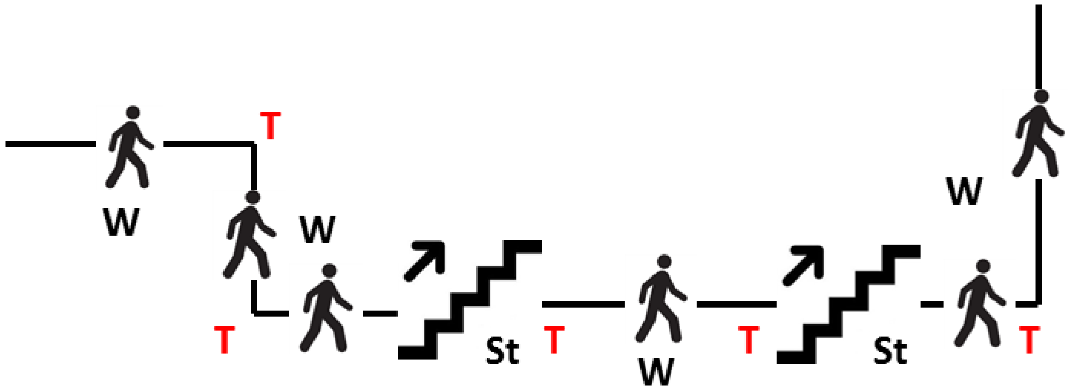


Figure 11: Example of pedestrians states in indoor trace split by turns

5.3 Activity recognition module

The goal in this module is on defining a set or rules that automatically allows the system to detect the activity among the possible states defined above. It means being able to automatically identify elevator or stairs conditions and to separate them from other patterns such as regular walking or being stationary.

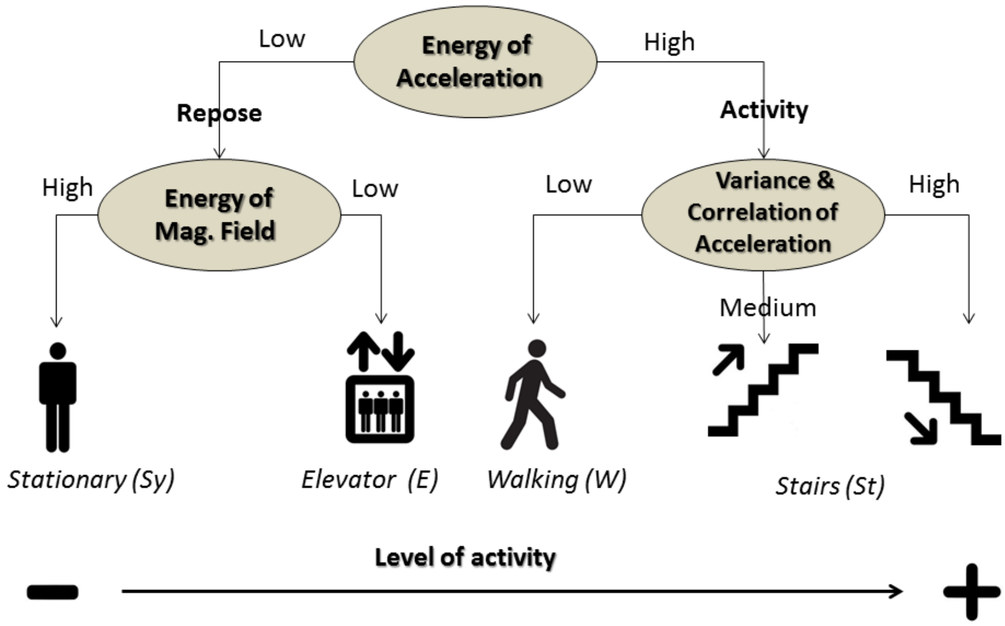


Figure 12: Decision tree to classify the possible indoor states

Figure 12 shows the decision tree that has been defined to classify the possible states.

The simplest way to produce useful data out of the inertial sensor is to take the magnitude of the three components data vector. For example, in order to estimate the magnitude for the acceleration, we need to calculate the energy represented in each sample provided by the accelerometer. After filtering the signal to make it smoother, the first decision to detect movement vs. repose is based on this level of acceleration. Fixing a threshold over the acceleration energy leads to estimate periods of activity and complementary the stationary ones.

Other different inertial sensor signals can be used to detect the activity subclasses:

5.3.1 Elevator (E):

Similar to cars or planes, elevators behaves like a Faraday shield and they have a unique magnetic field pattern that makes distinguishable with accuracy. Since a typical elevator is an enclosure formed by conducting material, it blocks external static and non-static electric fields. The different values for the Magnetic Field Energy coming from the outside and the inside of elevator show a notable difference, thus this transaction of states is easily identifiable (Figure 13).

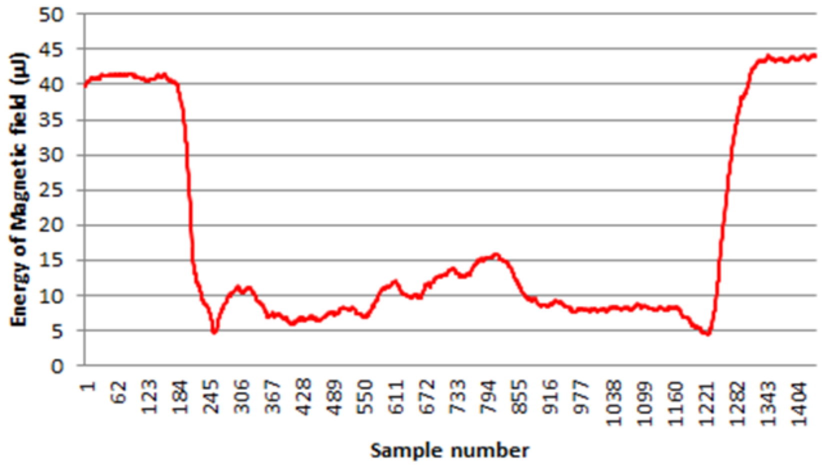


Figure 13: Transition of Magnetic Field energy outside and inside an elevator

There are two additional details that have to be detected in the elevator motion segments: the direction of motion and the number of floors traveled. The identification of the elevator direction can be estimated based on the energy of the acceleration measurements when the elevator starts and stops its travel. These events produce a pattern of acceleration peaks in the elevator segment (Figure 14) and studying the order of appearance the elevator movement can be classified.

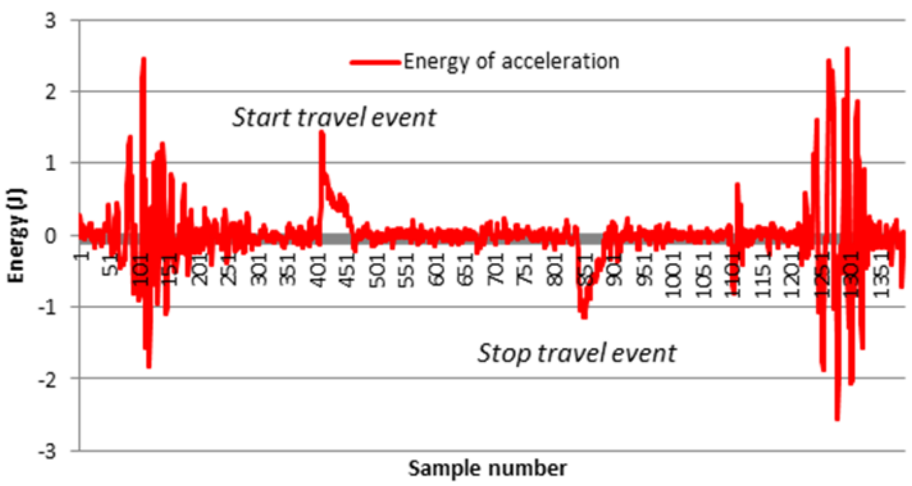


Figure 14: Energy of acceleration values for an up direction elevator travel

The number of floors traveled can be estimated based on the time duration of the displacement during the inside elevator travel.

5.3.2 Stationary (Sy):

Based on the previous observation, and after a frame is detected like repose time the stationary periods are complementary of the elevator periods.

5.3.3 Walking (W) and Stairs (St):

Once the repose states have been separated using the energy of the acceleration to detect active scenarios, is necessary to differentiate between stairs and walking cases. The observation is that when the pedestrian are using stairs the variance of acceleration is broader than in the walking case. Figure 15 represents this variance of acceleration's energy in a Walking –Stationary–Stairs sequence. The correlation between the acceleration in the direction of motion (Y axis) and the direction of gravity (Z axis) can be a good clue to separate stairs scenarios compared with walking. Furthermore, the measurements show that climbing stairs down (helped by the gravity acceleration) shows higher motion intensity than climbing up.

Combining these two processed signals, the three final states can be detected also including the direction of the user in the stairs frame.

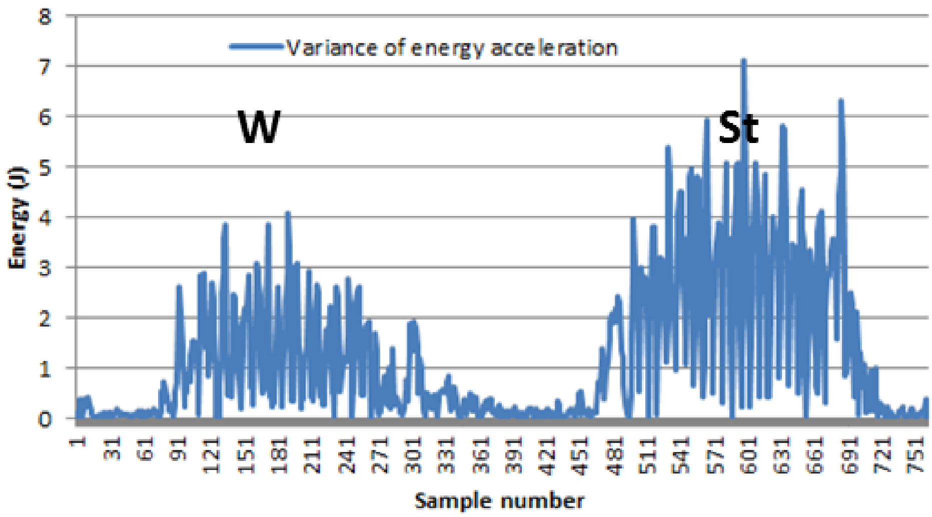


Figure 15: Variance of energy acceleration in walking and stairs states

5.4 Traces detailer module

Once the previous module has identified each segment and classified its activity, there are some of them, like walking or stairs states, that require more details to enable a complete tracking of the user. Processing the original signal, based on the accelerometer readings of the mobile phone, by specific algorithms is possible to derive the displacement of the person while walking or climbing stairs.

As was mentioned in the related work, theoretically the distance traveled can be calculated by integrating acceleration twice with respect to the time. However due to the presence of noise in the accelerometer output, error accumulates rapidly with the time. Another source of error is the presence of a component of acceleration due to the gravity of the Earth. These factors lead to errors in displacement that will grow constantly with time. Figure 16 shows the linear acceleration component on the y-axis (forward acceleration), the velocity resulted from integrating the acceleration and the walked distance calculated by integrating the velocity over time. Despite the form of the curves seems correct, and that it is possible to identify the steps while walking (13) and also the increment/decrement of speed with the steps, the results obtained are not similar to the real walked distance. They are approximately two times smaller than the total distance 10 meters.

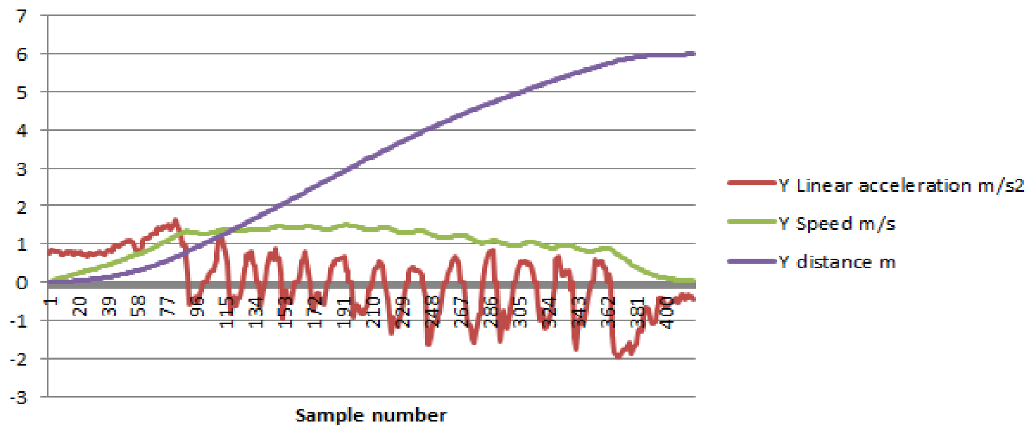


Figure 16: Calculated speed and distance by integrating acceleration

To reduce the accumulation of errors, we extend the previous approach estimating the distance traveled as the sum of the individual step sizes. We apply the step detection algorithm to detect the pattern that the magnitude of acceleration goes through when a step is made.

5.4.1 Step detection and distance estimation algorithm

Thanks to the detection of cycles in the sensor data (swing and stance phases), caused by the repeating patterns or events in motion of walking, it is possible to count the number of steps a person has walked, and therefrom derive the displacement of the person.

Based on our experiments the effect of walking on the magnitude of the acceleration vector is independent from the phone orientation and tilt. Therefore, our step counting algorithm is designed based on the magnitude of the acceleration, making this approach for displacement estimation independent from the placement of the mobile phone (messaging in hands, calling in user's ear or swinging in the pocket)

The algorithm implemented for step detection and counting consists of the following steps:

1) Calibration Routine

During a frame of movement, samples of linear acceleration [Acc(x), Acc(y), Acc(z)] are collected to estimate the mean in the acceleration (see Equation 12Equation 12). The energy of the average is estimated and considered as bias to compensate (Equation 13).

$$[\overline{Acc}(x), \overline{Acc}(y), \overline{Acc}(z)] = \frac{1}{N}\left[\sum_{i=0}^{i=N} ax_i\,, \sum_{i=0}^{i=N} ay_i\,, \sum_{i=0}^{i=N} az_i\right]$$

Equation 12: Mean of three axis acceleration values

$$\overline{Acc}_{calib} = \sqrt{\overline{Acc}(x)^2 + \overline{Acc}(y)^2 + \overline{Acc}(z)^2}$$

Equation 13: Energy of acceleration averages

Where N is the number of samples used in the calibration as much as possible in the frame

2) Mean acceleration

Compute the energy of the acceleration, a_i, for every sample i as shown in Equation 14.

$$ai = \sqrt{ax_i^2 + ax_i^2 + az_i^2}$$

Equation 14: Energy of a simple acceleration sample

Estimate the average of energy in a set window of size ω and compensate de bias (see Equation 15).

$$\bar{a_l} = \frac{1}{2\omega + 1} \sum_{j=i-\omega}^{j=i+\omega} (aj - \overline{Acc}_{calib})^2$$

Equation 15: Average of energy for an acceleration sample in a set window

3) Thresholding

A first threshold is applied following the rule Equation 16 in to detect the swing phase with high accelerations.

$$B_{1_i} = \left\{ \begin{array}{cc} T_1 & if \quad \bar{a_l} > T_1 \\ 0 & otherwise \end{array} \right\}$$

Equation 16: Threshold to detect swing step phase

A second threshold T_2, as defined in Equation 17**Error! Reference source not found.** is used to detect the stance phase

$$B_{2_i} = \left\{ \begin{array}{cc} T_2 & if \quad \bar{a_l} < T_2 \\ 0 & otherwise \end{array} \right\}$$

Equation 17: Threshold to detect stance step phase

T_1 and T_2 are symmetric values respect to 0 fixed heuristically to +0.5 and -0.5.

4) Step detection

A step is detected in sample i when a swing phase ends and stance phase starts. These two conditions must be satisfied:

1) A transition from high to low acceleration ($B_{1_{i-1}} > B_{1_i}$), and

2) There should be at least one low acceleration detection in a window of size ω ahead

of current sample i, i.e.: $min\ (B_{2\ i}: i + w) = \ T_2$

5) Finally the steps vector is iterated to obtain the average time between steps

Figure 17 shows the main signals taking part in the step detection and counting algorithm applied in a basic walking example.

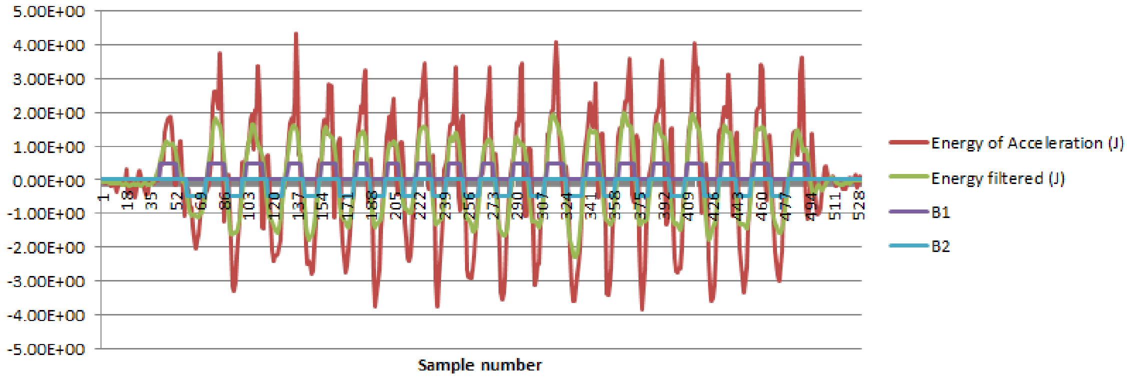


Figure 17: Step counting algorithm signals

Obtained the number of steps, the total distance walked can be directly estimated considering the stride length of each step to be constant and with a value of 0.74 meters [21].

5.4.2 Stairs detection and counting algorithm

The algorithm implemented for stairs detection and counting based on peak detection consists of the following stairs:

1) Calibration routine

During a frame of movement, samples of linear acceleration [Acc(x), Acc(y), Acc(z)] are collected to estimate the mean in the acceleration. The energy of the average is estimated and considered as bias to compensate similar to the step detection algorithm in Equation 12 and Equation 13.

2) Energy of acceleration

The simplest way to produce useful data out of the three components of the sensor is to take magnitude of the acceleration vector. Compute the energy of the acceleration, ai, for every sample similar to Equation 14.

3) Low pas filter and bias compensation

Low-pass filters provide a smoother form of the signal, removing the short-term fluctuations, and leaving the longer-term trend. LPF [ai] is the discrete low pass filter signal of the acceleration energy readings. It has been applied using a discrete-time implementation of a simple RC low-pass filter as show below in Equation 18, with a smoothing factor of α=0.9.

$$LPF[ai] = \propto a_{i-1} + (1 - \propto)\, a_i$$

Equation 18: Discrete implementation of a low pass filter

To compensate the bias the value $\overline{\mathrm{Acc}}_{\mathrm{calib}}$ is removed for all the energy filtered samples (Equation 19)

$$Energy[a_i] = LPF[a_i] - \overline{Acc}_{calib}$$

Equation 19: Bias compensation for the energy filtered signals

4) Peaks detection

A peak is detected if during PREVIOUS_SAMPLES (value = 5) the *backwardSlope* of the current sample is positive and in the next sample the *forwardSlope* becomes negative being these two functions as detailed in Equation 20.

$$forwardSlope = a[k+1] - a[k] \;\; and \quad backwardSlope = a[k] - a[k-1]$$

Equation 20: forwardSlope and backwardSlope functions definition

5) Step detection

A threshold is applied to detect those peaks considers stairs.

The energy signal if is traversed by a buffer of a fixed number of samples. In this implementation, the buffer length is 100 samples.

For every set of samples in the buffer *peakMean* (Equation 21) is calculated and multiple by a guard factor (G=0.6). This new value C is the threshold for each set of samples. The peaks detected have effect on the threshold and on the responsiveness of the algorithm changing the threshold value.

$$peakMean = \frac{1}{\#Peaks} \sum PeakValues$$

Equation 21: peakMean function definition

The final part of the algorithm is to iterate again over the Energy [a_i] and detect which peaks are above the threshold.

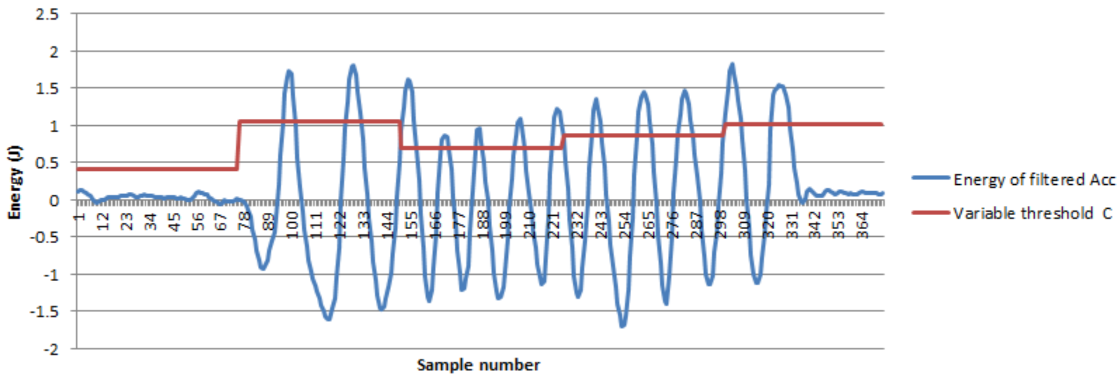


Figure 18: Stair counting algorithm signals

Figure 18 shows the main signals taking part in the stairs detection and counting algorithm applied in a basic climbing down stairs example.

The total height climbed can be estimated considering the maximum stair height regulated by the international code council and fixed to a value of 0.178 millimeters.

Chapter 6. System implementation

The system implementation can be divided in two main parts: the mobile application to gather the data from the user and the server side to apply the signal processing algorithms and the activity recognition decisions.

6.1 Mobile application

The data acquisition process from the sensors embedded in the smartphone has been implemented through a basic mobile Android application.

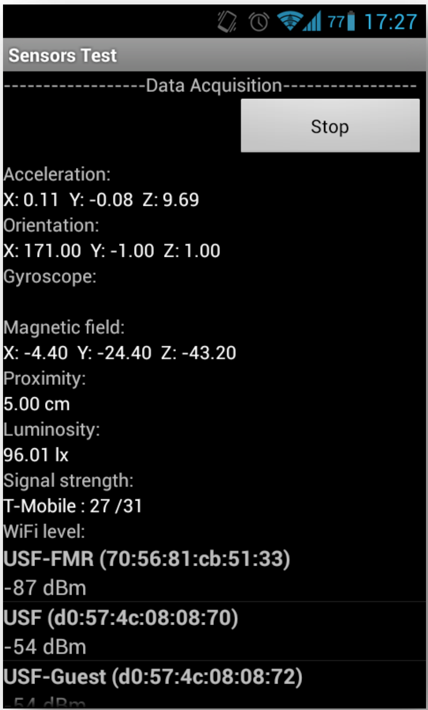


Figure 19: Mobile application screenshot

As we can see in the screenshot (Figure 19), the application show the raw data collected from the inertial sensor available in the device: accelerometer, gyroscope and magnetometer and other built-in sensors. Several text views are placed on the only activity screen to display the information collected. Additionally the mobile application is in charge of send the information packet to the server when the motion test finishes.

6.2 Server side

The server receives the raw data sent by the mobile device at the end of every single test via a web service. The different modules detailed in the system design are accomplished by several synchronized threads. Finally the server display the activity decision for the traces sequences and the details estimated for every single trace.

6.3 Programming languages

Java EE has been de programming language used to develop the whole Project.

6.4 Development tools

Eclipse has been the integrated development environment (IDE) used to implement both sides: server and mobile application.

For the Android mobile application development we have used the Android SDK [12]. This environment provides the API libraries and developer tools necessary to build, test, and debug apps for Android. It includes the Eclipse IDE with built-in ADT (Android Developer Tools) to streamline Android apps development.

Additionally the ADT Bundle includes everything needed to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

The server code has been developed using Eclipse Juno (4.2 platform version)

[12] http://developer.android.com/sdk/index.html

Chapter 7. Evaluation

In this section, we evaluate the performance of our system with several basic tests and a complete motion trace. We start by describing the methodology applied, followed by the detail of the result charts obtained and it concludes with final discussion of the tests results.

7.1 Methodology

We implemented the system on a Samsung Galaxy SII running the Android OS, and evaluate across different indoor spaces and activities. In our experiments different users carried phones in texting position.

The first set of tests is used to evaluate the accuracy of the basic algorithms while the second testbeds analyze the activity recognition tree.

The results are sent to the server, processed and the results are obtained immediately after the test.

7.2 Results

Table 5 shows the results of applying step detection and counting algorithm. The test consists on repeating five times, three walking trips in corridors with different distances of: 5, 10 and 15 steps

Step counting tests	Test 1	Test 2	Test 3	Test 4	Test 5	Average	# Errors
Distance 1: 5 steps	5	5	5	5	5	5	0
Distance 2: 10 steps	10	10	10	10	10	10	0
Distance 3: 15 steps	15	15	15	15	15	15	0

Table 5: Step counting tests results

Table 6 summarizes the scores of heading changes detection algorithm for four different sequences of three heading changes: Left-Rigth-Left (L–R–L), R–L–R, R–L–L and R–R–L. Each experiment was repeated five times.

Tuns detection tests	Test 1	Test 2	Test 3	Test 4	Test 5	# Errors
Sequence 1: L-R-L	LRL	LRL	LRL	LRL	LRL	0
Sequence 2: R-L-R	RLR	RLR	RLR	RLR	RLR	0
Sequence 3: R-L-L	RLL	RLL	RLL	RRL	RLL	1
Sequence 4: R-R-L	RRL	RRL	RRL	RRL	RRL	0

Table 6: Turns detection tests results

Table 7 shows the results for our stairs detection and counting algorithm repeated five times for two motion traces of 5 and 10 stairs respectively.

Stairs detection tests	Test 1	Test 2	Test 3	Test 4	Test 5	Average	#Errors
Motion 1: 5 stairs	5	5	5	5	5	5	0
Motion 2: 10 stairs	10	10	10	10	10	10	0

Table 7: Stairs counting test results

Table 8 shows the confusion matrix for the activities classification.

Activity /Detected	Stationary	Elevator	Walking	Stairs	#Errors
Stationary	10	0	0	0	0
Elevator	1	9	0	0	1
Walking	0	0	9	1	1
Stairs	0	0	2	8	2

Table 8: Activity classification tests results

7.3 Discussion

The previous tables show good results for the specific algorithms. One the activity is identified, the individual processing signal algorithms are able to detect with high accuracy walking steps, stair steps and turns. According with the previous Tables (Table 5, Table 6 and Table 7) our pattern recognition techniques are able to detect and to track successfully the number of walking steps and stairs steps performed by the testers.

In contrast, the activity recognition task results more indeterminate and errors are found in the decision tree results between walking/stairs traces and elevator/stationary periods (Table 8). This confusion in activity classification leads to false positives. The main reason in the activity detection fails is that indoor signatures are not stable and they can fluctuate over time and depending on the location. For instance the Faraday shield phenomenon does not affect in the same way to the set of elevators tested, showing that the reduction in the level of magnetic field energy is not constant. Also, stairs in indoor environments can be built with a difference inclination grade (relation axis Y and Z) and to fix a threshold for the YZcorrelation value is not evident. However, even with this difficulty, the activity recognition module can still achieves good accuracy, with an overall of 90% of hit rate.

While more rigorous experimentation is necessary (across more buildings, people, phone and models), we believe the results from these small scale tests are promising to justify moving forward. Thus, we are working on new techniques to improve the current results, such as the use of new sensors to estimate height variations or a participatory sensing approach to combine activity decision results from multiple users. These ideas, complemented with some other future works, are detailed in the following section.

Chapter 8. Conclusions

In the global scenario of Navigation Systems the ability to track people in indoor environments has particular interest for many ubiquitous computing systems.

Our work provides a technique to create automatically accurate indoor pedestrian traces based on the noisy inertial sensors in today's commodity smart phones. The system proposed tracks pedestrians in indoor scenarios where maps data are not provided a priori and in a transparent manner to the users. Corners (using a gyroscope), stairs, steps, and corridors (making use of an accelerometer), and elevators (considering a magnetometer readings) are detected and detailed in constrained indoor environments.

Currently, we are expanding our system working on different directions including: independence of phone's location and multisensory fusion adding new collectors of date to the system. Performance results suggest strong motivation to pursue the goal of real deployment.

Chapter 9. Future Works

The presented evaluation has been performed under a unique position of the smartphone and the analysis shows the results in the messaging position. A complete solution should consider the performance of the system in several usual positions of the device, like: calling, swing (refers to the position in which the user holds the device in hand while walking) and pocket (the device sits in the user's pocket/bag, which is the most common position for the device when the user is walking). Since most of the algorithms use the magnitude of the signal considering the source of data the 3D sensor measurements they should potentially work in any of the described cell phone's location.

In the current version of the system, the collecting data module is initiated manually by the user when entering in an indoor location. A useful improvement would be the automatic change from outdoor location to indoor location. For instance, a smart implementation could activate the indoor location part taking into account the fact that building entrances are characterized by a visible drop in the GPS confidence when the user moves from outdoors to indoors.

Recently modern smartphones and tablets are equipped with new sensors such as Barometric pressure sensor. Based on previous works and techniques for detecting floor changes [24] [25], the use of these algorithms could be integrated in our indoor localization system to ameliorate the detection of stair or elevator states and the estimation of height climbed/traveled.

So far just individual human motion traces have been considered, but the system could multiply its performance using participatory sensing and taking motion traces from multiple users. Location of landmarks can be made more accurate by combining the rough estimates form multiple users and additional functionalities could be developed like floor plans combined creation, group security plans, resource-efficiency politics or group activity recognition [26].

To include concrete additional functionalities to the system, some relative landmarks should be identified as absolute points. For example, in the case of floor plan creation based on participatory sensing in a building with several entrances, multi-modal fusion sensing would be required to associate the multiple locations and identify those that are the same as unique.

Chapter 10. References

[1] U. Varshney, “Pervasive Healthcare and Wireless Health Monitoring,” *Mob. Networks Appl.*, vol. 12, no. 2–3, pp. 113–127, Jul. 2007.

[2] R. Harle, “A Survey of Indoor Inertial Positioning Systems for Pedestrians,” *Ieee Commun. Surv. Tutorials*, pp. 1–13, 2013.

[3] Y. Jin, M. Motani, W.-S. Soh, and J. Zhang, “SparseTrack: Enhancing indoor pedestrian tracking with sparse infrastructure support,” in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1–9.

[4] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, “No need to war-drive: Unsupervised indoor localization,” in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 197–210.

[5] M. Alzantot and M. Youssef, “CrowdInside: automatic construction of indoor floorplans,” in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2012, pp. 99–108.

[6] Y. Capelle, M. T. GMV, D. Kubrak, M. Monnerat, A. M. GMV, and D. J. ESA, “DINGPOS: A Hybrid Indoor Navigation Platform for GPS and GALILEO.”

[7] Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *Ieee Commun. Surv. Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[8] K.Pahlavan, F.Akgul nd Y.Ye, “Taking Positioning Indoors. Wi-Fi Localization and GNSS” *Insidegnss, Tecnhical article May 2010*

[9] J. A. Besada, A. M. Bernardos, P. Tarrío, and J. R. Casar, “Analysis of tracking methods for wireless indoor localization,” in *Wireless Pervasive Computing, 2007. ISWPC’07. 2nd International Symposium on*, 2007.

[10] P. Robertson, M. G. Puyol, and M. Angermann, “Collaborative pedestrian mapping of buildings using inertial sensors and footslam,” in *ION GNSS*, 2011.

[11] C. Fischer, P. Talkad Sukumar, and M. Hazas, “Tutorial: implementation of a pedestrian tracker using foot-mounted inertial sensors,” *Ieee Pervasive Comput.*, 2012.

[12] O. Woodman and R. Harle, “Pedestrian localisation for indoor environments,” in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 114–123.

[13] O. Woodman and R. Harle, “RF-based initialisation for inertial pedestrian tracking,” in *Pervasive Computing*, Springer, 2009, pp. 238–255.

[14] “Whole-Home Gesture Recognition UsingWireless Signals.pdf.” .

[15] I. Constandache, R. R. Choudhury, and I. Rhee, “Towards mobile phone localization without war-driving,” in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1–9.

[16] A. J. Ruiz-Ruiz, O. Canovas, and P. E. Lopez-de-Teruel, “A Multisensor Architecture Providing Location-based Services for Smartphones,” *Mob. Networks Appl.*, vol. 18, no. 3, pp. 310–325, Nov. 2012.

[17] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, “A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU,” in *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, 2009, pp. 37–42.

[18] M. Azizyan, I. Constandache, and R. Roy Choudhury, “SurroundSense: mobile phone localization via ambience fingerprinting,” in *Proceedings of the 15th annual international conference on Mobile computing and networking*, 2009, pp. 261–272.

[19] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, “Indoor location sensing using geo-magnetism,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 141–154.

[20] M. Mladenov and M. Mock, “A step counter service for Java-enabled devices using a built-in accelerometer,” in *Proceedings of the 1st international workshop on context-aware middleware and services: affiliated with the 4th international conference on communication system software and middleware (COMSWARE 2009)*, 2009, pp. 1–5.

[21] M. Alzantot and M. Youssef, “Uptime: Ubiquitous pedestrian tracking using mobile phones,” in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, 2012, pp. 3204–3209.

[22] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, “Did you see Bob?: human localization using mobile phones,” in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, 2010, pp. 149–160.

[23] M. Alzantot and M. Youssef, “CrowdInside: automatic construction of indoor floorplans,” in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2012, pp. 99–108.

[24] M. Tanigawa, H. Luinge, L. Schipper, and P. Slycke, “Drift-free dynamic height sensor using MEMS IMU aided by MEMS pressure sensor,” in *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, 2008, pp. 191–196.

[25] S. Vanini and S. Giordano, “Adaptive context-agnostic floor transition detection on smart mobile devices". *CoMoRea Workshop - March 18th, 2013*

[26] D. Gordon, J.-H. Hanne, M. Berchtold, A. A. N. Shirehjini, and M. Beigl, “Towards Collaborative Group Activity Recognition Using Mobile Devices,” *Mob. Networks Appl.*, vol. 18, no. 3, pp. 326–340, Oct. 2012.

Chapter 11. Appendices

Appendix A: Content in the CD-ROM Submitted

This appendix includes a description of the contents submitted in the CDROM. The documentation, source code and additional material included in the presentation are explained in the following Table 9.

Directory	Content
./readme.txt	File explaining basic info about the project
./Documentation/Report.doc	.doc file including all the information about the project.
./Documentation/Report.pdf	.pdf created form Report.doc
./Documentation/AndroidSensorAPI.pdf	Presentation created during the development of the thesis, to explain the use of Android Sensor API. It includes a detailed example of use to collect raw data from the Android Sensors.
./Documentation/Paper.pdf	.pdf showing the paper created from the research work and submitted to the Springer journal Mobile Networks and Applications.
./SourceCode	This folder contains all the source code necessary to run the location system.
./SourceCode/SensorApp	Java Project developed in Eclipse for the Android mobile device

Table 9: Contents in the CDROM

Appendix B: Journal Paper

A journal paper titled

" Multi sensor algorithms for pedestrian activity recognition in indoor scenarios",

has been created thanks to the research work developed in this master thesis.

The current version of the paper has been included in the CDROM detailed above, and the 3rd July 2013 the paper has been submitted to the JCR Springer **Journal Mobile Networks and Applications**. The submission is being processed and the current status is pending of editor assignment.

Appendix C: Android sensor API seminar

During the development of this thesis I was asked to create a presentation about the use of the Android Sensor API. This presentation was explained in a seminar at Department of Computer Science and Engineering at University of South Florida (Tampa – USA)

The presentation includes basic information about the sensor frameworks and details an example of use to collect raw data from the Android Sensors. Similar to the journal paper, a .pdf file with the slides is available in the documentation folder included in the CDROM submitted.