Mobility Recognition Based On Android Sensors To Improve Indoor Localization

Bachelorarbeit
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Mansour Hamidi
2015

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik und angewandte Mathematik

Contents

List of Figures

List of Tables

Acknowledgment

First I would like to give my gratitude to Prof. Dr. Torsten Braun who steered my thesis back in the right direction as it seemed to stray from the topic. I am grateful for the resources, which were generously provided by the CDS research group.

My particular thank goes to my supervisor Zan Li for his continuous support, for the inputs, for the suggestions, for the feedbacks and for his brilliant idea of combining mobility recognition with indoor localization.

I would also like to thank Desislava Dimitrova for helping me to immerse in the theme of human activity recognition.

Abstract

Various approaches have been researched to realize accurate indoor localization systems. However, the accuracy of these systems is still not satisfactory for indoor environments.

Due to low complexity and low costs, a popular localization approach is to make use of WiFi signals. This kind of localization requires the continuous measurement of RSS in order to estimate the position with an appropriate algorithm. The problem that appears in this approach is the scattering of received signals. This problem can be solved by computing the mean value of an increased amount of RSS samples. However, increasing this amount can only be done taking into account the mobility of the user, who has to be positioned. Therefore, recognizing the user's current performing activity enables to adjust the amount of gathered RSS data according to the mobility. This is the reason why the combination of mobility recognition and indoor localization can improve the accuracy of a localization system. Note that this combination process is not the same as in dead reckoning.

Combining two systems requires the investigation of both. For this we attempted to maximize the recognition accuracy between several most observed activities indoors with different parameters. Likewise, we examined several parameters, which affect the localization accuracy. In order to draw conclusions, we compared the execution of localization without using mobility recognition with the execution of localization combined with mobility recognition.

The results of using a mobility recognition system that is combined with the indoor localization application, obviously show an improved accuracy, and the improvement of the mean error by 0.4m is crucial in indoor environments.

Chapter 1

Introduction

The rapid improvements on mobile phones in processing power, embedded sensors, storage capacities and network data rates, let a today's mobile phone evolve from merely being a phone to a smart device with seemingly limitless possibilities of applications. It is thus hardly surprising that over 5 billion people globally have access to a mobile phone [1].

1.1 Motivation

A huge number of applications require knowledge on the current and past locations of users. While many outdoor solutions are based on Global Positioning Systems (GPS), which receive signals from satellites to determine the location, the resulting position is not accurate enough for localization inside buildings. Roofs, walls, trees and other objects can cause a loss of connectivity to satellites or attenuate the signals. As an alternative to GPS, indoor localization concepts make use of different technologies since there is no overall solution based on a single technology such as that provided outdoors by satellite-based systems. Many indoor positioning applications are waiting for a satisfactory technical solution but we are still far away from a cheap provision for indoor positioning with an accuracy of 1 meter. As the demand for seamless positioning in all environments increases, indoor localization has become a focus of research during the past decade [2].

1.2 Indoor Localization

A popular approach for indoor localization is the use of WiFi signals since WiFi access points are available in many indoor environments. The idea behind this approach is to estimate the position of a mobile device within its network. Due to its low complexity the most common method to estimate the position is to make use of the Received Signal Strength (RSS). In this case a standard mobile device, such as a smartphone, can permanently measure the strength of surrounding WiFi signals and continuously estimate its position in a particular time slot. The problem that appears in that way of estimation is the scattering of received signals. The RSS can vary considerably when objects or persons move in the range between the sender and receiver of signals. Therefore, increasing the number of RSS samples helps to counteract noisy signal results. However, collecting RSS samples in a larger time slot for a moving person certainly provides bad localization results. Thus, improving the localization in the compromise between RSS time slot length and movement velocity necessarily ends in the recognition of the person's current behaviour in terms of mobility since identifying the user's activity can draw conclusions of its movement velocity. This information enables to accordingly adjust the RSS time slot to the movement, for instance, to collect RSS data as long as the user is standing, or to collect only in a short time slot when the user is walking, or even in a shorter time slot when the user is running. Identifying the activity is not only of advantage for time slot adjustments, it can also be helpful to realize a floor switch inside buildings, especially by recognizing climbing stairs or walking an inclination. Thus, a WiFi localization system in combination with mobility recognition may improve the accuracy of indoor localization.

1.3 Mobility Recognition

Recognizing human activities is not a research area of recent times. The first works date back to the late '90s, so it existed long before the invention of smartphones with embedded sensors [3]. The significant change, which is brought by the smartphones, is the simpler implementation of data gathering inasmuch as the sensors are no longer attached to the user's body but are included in the mobile phone.

Various activities have been investigated by many researchers. The most commonly explored activities in previous studies are walking, running, sitting, standing, lying, climbing stairs, cycling, driving, eating, drinking, working, watching TV and brushing teeth. For localization in particular it is of interest to distinguish between movements like standing, walking and running, which have been in general well investigated. However, studies on distinguishing between movements like walking an inclination and climbing stairs are not common, despite the fact that these activities are observed most often indoors, for instance in shopping malls and train stations, where the customer can choose between the moving staircase, the normal staircase or the travelator to reach the floor above. This information causes the localization system to proceed on a floor above or below.

In order to realize a recognition system that can classify activities, different classification algorithms and smartphone sensors need to be considered. These algorithms learn in an initial phase to differentiate the activities based on gathered sensor data. In a subsequent phase the algorithms should be able to classify new unclassified activities. The resulting accuracy of a recognition system is highly dependent on the choice of sensors, algorithms and kind of activities.

1.4 Task Formulation

The goal of this thesis is to develop a WiFi-based indoor localization system, which achieves higher positioning accuracy when it is combined with a mobility recognition system. In order to achieve this goal, this thesis is divided into two parts. While the first part investigates the topic of mobility recognition, the second part focuses on indoor localization:

Part of mobility recognition

In order to generate a mobility recognition system, which is able to distinguish among several activities, we will develop in an initial phase an Android smartphone application that collects sensor data for different activities. In a subsequent phase we will use the gathered sensor data in Matlab to train different classification algorithms, which then should be able to classify new data.

Part of indoor localization

Similar to the first part, we will develop again an Android smartphone application that in addition to sensor data also collects RSS data from several access points. In Matlab we will use the sensor data to predict the activity and the RSS data for positioning. According to the prediction, we can adjust the time slot of the RSS data. In order to be able to show an improvement of localization results, we will compare results of localization including mobility recognition with results of localization excluding mobility recognition.

1.5 Outline

In order to understandable demonstrate the use of mobility recognition for indoor localization, we focus on both systems. Thus, this thesis is organized as follows:
In Chapter 2 we discuss the background of mobility recognition and indoor localization based on related works. In Chapter 3 the theories of classifiers and localization algorithm are explained. Also the combination of these two systems is considered in more detail. Chapter 4 offers an overview of the entire implementation structure with explanations of each processing step. Thereafter, we show and discuss the results in Chapter 5. Finally, in Chapter 6, we offer our conclusions of this thesis and give some ideas for future works.

Chapter 2

Background and Related Works

2.1 Background and Related Works in Mobility Recognition

Human activity understanding comprises two stages. First, it includes the activity pattern discovery, in which is tempted to find unknown patterns in the raw sensor data. Second, the discovered activity pattern can be used to define the activities, which will be recognized [4]. To face these stages, sensor data is needed to be gathered and handled by learning algorithms.

For the sake of simplicity we consider the terms activity recognition and mobility recognition as the same.

2.1.1 Activity Recognition

Human activity recognition is the process where people's behaviour and their situated environment are monitored and analyzed to infer the underlying activities. Accurate recognition of activities is an important goal of pervasive computing [5].

Despite the large number of studies within this field, the accuracy of recognition varies significantly between the studies since each was carried out under different settings. The main differences between the studies concern decisions about activities, sensors and algorithms.

Activities:
Various activity recognition works frequently explored similar activities, such as staying, sitting and driving, or walking, running and climbing stairs. Many evaluations show that climbing stairs mostly has a slight, but nevertheless the biggest confusion with running and walking, as noticed in [6], [7], [8], [9] and in [10].

Human activities, which may be helpful in respect to indoor localization are standing, walking, running, climbing stairs up or down and walking an inclination up or down. The activity 'walking an inclination' may confuse because it can be asked, why this should be important for localization. The answer is given by an example: when a mobile user walks an inclination outdoors, it is not really necessary for the GPS to know that the user is walking an inclination since it only shows the positions on a map, regardless the height. However, if the mobile user walks an inclination indoors, this inclination possibly leads the user to the floor above. This situation is most seen at train stations and in shopping malls, where the customer can choose between the moving staircase, the normal staircase or the travelator (moving walkway) to reach the floor above - and walking on a travelator is the same activity as walking an inclination. Although activities like standing, walking and running have been in general well investigated, studies differing walking from walking an inclination or climbing stairs are not common, despite the fact that these activities are most often observed indoors.

Algorithms:
While many studies analyzed machine learning methods including decision trees, nearest neighbors, Bayesian networks, support vector machines and hidden Markov models, it is still hard to say, which one of these is most effective for human activity recognition since each provided different recognition rates under different settings [11].

Zhao et al. explored in [12] the problem of a cross-people activity recognition. This problem states that a classification model learnt from a specific person often cannot provide accurate recognition results when used on a different person. The classification model only performs well for a person, if the model learnt from this person. This implies that the system should be retrained for each new user, which might be pretty inconvenient for elder people or patients with mental pathologies. However, Lester et al. show in [13] that the system does not need to be customized to each individual for a reliable accuracy of recognition. Their system worked reliably when it was tested on a person whose data was not used as training data for the learning algorithm. Thus, a subject-independent classifier should be capable to recognize an individual's activity even if the participant is a new user, on the condition that the system has been trained by participants with different characteristics.

Recognizing characteristics of activities is commonly achieved by identifying patterns in the raw data. However, raw data from sensors contain many hidden information and noise. Therefore, feature extraction filters relevant information from raw data, which can then be compared. Depending on the activity set, some features might contain redundant or irrelevant information that can negatively affect the recognition accuracy [3]. The features, which have proved to be useful in human activity recognition are minimum, maximum, mean, standard deviation and correlation [7], [14], [15]. Especially, the correlation is useful to discriminate among activities, which differ in the number of dimensions. For example, distinguishing walking and running from stair climbing is possible since walking and running usually involve translation in one dimension whereas climbing stairs involves translation in more than one dimension [8].

Sensors:
Despite the great variety of available sensors, as we can see in Table 2.1, the majority of previous studies have focused on using a single sensor type placed on several body locations, notably accelerometers, because they gather motion data. However, Wilde explains in her overview [5] that simultaneously using several accelerometers improves the accuracy of recognition only marginally. Additionally, the loss of accuracy using a single sensing location can be compensated by the use of additional sensor types. Wilde's comments are based on the study of Bao et al. in [14], whose system was able to recognize 20 activities including ambulation and daily activities such as scrubbing, vacuuming, watching TV and working at the PC. Five biaxial accelerometers were placed simultaneously on the participant's ankle, leg, waist, arm and wrist. They concluded that the recognition accuracy was diminished only by 5% when they only used

two accelerometers, placed on the waist and wrist, instead of five. Besides, ambulation activities were recognized very accurately whereas activities such as stretching, scrubbing, riding escalator and riding elevator were often confused.

Important findings with respect to sensors were also made by Milker. He investigated in [16] the activity recognition of the four simple situations going, walking, standing and sitting with data of accelerometer, linear accelerometer, orientation and magnetic field gathered by a smartphone. Used algorithms with accuracy were the decision tree with 98%, naive Bayes with 74% and support vector machines with 53%. He came to the conclusion that activity recognition is possible by using only a single accelerometer, but not by using data of the magnetic field. However, he used every sensor solely. A combination of sensors would have yielded better results. To the same conclusion came Khan in [17], who only used an accelerometer on the participant's backbone to recognize lying, sitting, walking, running, standing, cycling, ascending and descending stairs with the decision tree algorithm. His results have shown that one triaxial accelerometer is enough for identifying these physical activities, though with confusion errors.

Regarding the smartphone position and the number of different sensors, Lester et al. attempted to answer in [13] valuable questions like: 'Does it matter where the sensors are placed?' or 'How many sensors are really needed to recognize a significant set of basic activities?' To present answers they collected data from 12 individuals performing activities such as sitting, standing, brushing teeth, walking, walking stairs up/down and riding elevator up/down, by carrying sensors like microphone, accelerometer, compass, barometer, humidity, ambient light and visible light phototransistor, located on wrist, waist and shoulder. They concluded that it does not matter where the users place the sensors as long as the algorithms and features have been chosen carefully. In their work the accuracy of recognition between learning and testing at one specific sensor location is not even by 1% higher than learning and testing at three locations. Anyway, sensors are best placed at locations where the intrinsic characteristics of the target activities can be well captured. Thus, trying to recognize brushing teeth with a foot-mounted accelerometer might probably be more confused with similar activities like scratching the head than a wrist-mounted accelerometer. Among the selected sensors, the microphone, barometer and accelerometer yielded the most discriminative information because these three modalities provide complementary information about the environment and the wearer. The microphone captures the sounds produced during various activities, whereas accelerometer data is sensitive to the movement of the body, and the barometric pressure provides important height changes, such as detecting the activity of riding in an elevator or moving up and down stairs. To properly infer underlying activities from microphone data, the microphone should not be worn on a covered place such as the pocket. Likewise, the barometric pressure is useful to discriminate between climbing stairs up and down under the condition that the staircase is high enough since this sensor is insensitive to little changes. Besides, the barometer is deployed in only few smartphones.

Another point to be considered is the sampling rate of the sensors. Although Maurer et al. [18] found that no significant gain in accuracy is achieved by frequencies above 20 Hz, many studies record with a sampling rate of 50 Hz. Particularly in situations with only one sensor, higher recognition rates can be achieved by increasing the sampling rate. Especially in short activities, sampling with low frequency may miss crucial information [5].

Sensor	Type	Description
Magnetic Field	Hardware	Measures the ambient geomagnetic field for all three physical axes (x, y, z) in μT. Commonly used in creating a compass.
Ambient Temperature	Hardware	Measures the ambient room temperature in C°. Commonly used in monitoring air temperatures.
Relative Humidity	Hardware	Measures the relative ambient humidity in $\%$. Commonly used in monitoring dewpoint, absolute, and relative humidity.
Accelerometer	Hardware	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), including the force of gravity. Commonly used for motion detection (shake, tilt, etc.).
Light	Hardware	Measures the ambient light level in lx. Commonly used in controlling screen brightness.
Gyroscope	Hardware	Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y, and z). Commonly used for rotation detection (spin, turn, etc.).
Proximity	Hardware	Measures the proximity of an object in cm relative to the view screen of a device. Commonly used to determine whether a handset is being held up to a person's ear.
Pressure	Hardware	Measures the ambient air pressure in hPa or $mbar$. Commonly used in monitoring air pressure changes.
Rotation Vector	Software	Measures the orientation of a device by providing the three elements of the device's rotation vector. Commonly used for motion detection and rotation detection.
Linear Acceleration	Software	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z) like above, but excluding the force of gravity. Commonly used in monitoring acceleration along a single axis.
Gravity	Software	Measures the force of gravity in m/s^2 that is applied to a device on all three physical axes (x, y, z). Commonly used for motion detection (shake, tilt, etc.).

Table 2.1: Android supported sensor types. Note that some of the listed sensors are physical sensors, whereas others are virtual sensors. Listed software sensor types are also supported when used as hardware sensor [20].

2.1.2 Sensor Technology

As micro-electro-mechanical sensors are becoming increasingly popular due to their small size, decreased price, light weight and less power requirements, their incorporation in smartphones becomes a standard feature for various applications. For instance, the camera can be used as an image sensor, the microphone as an acoustic sensor and the proximity sensor turns off the smartphone's screen during a call in order to keep the battery consumption low. Further use of inertial sensors, especially in smartphones, is in compasses, in cameras for measuring and counteracting hand jitter in order to take high resolution images, in drop protection mechanisms, which prevent damages of a hard drive due to a fall, and in gaming applications where the actual hand motion is replacing the game controller [25].

Despite the wide choice of embedded sensors, a considerably large number of mobile sensing applications only make use of the inertial measurement unit including accelerometer, gyroscope, magnetometer and barometer, which can collectively be used to detect motion.

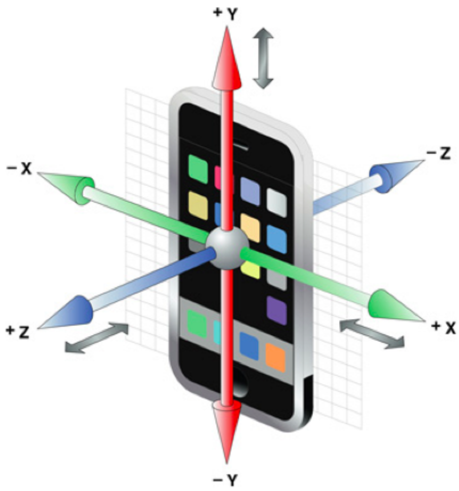


Figure 2.1: Embedded inertial sensors in smartphones measure a defined unit along three orthogonal axes, each splitted into a positive and negative part [27].

Accelerometer:
Accelerometers react to many types of movement, including linear and centripetal acceleration, gravity and vibration. The measured unit is expressed in m/s^2. Different algorithms can be used to extract measurements of tilt, position, vibration, shock and free-fall. Generally, accelerometers measure acceleration by sensing how much a mass presses on something when a force acts on it. That means that a smartphone lying on a table with the face upwards, indicates at its z-axis approximately a force of +1g and not zero as assumed, because any point on the Earth's surface is accelerating upwards relative to the sensor. In this case, the accelerometer is stationary, but it is producing an output indistinguishable from the actual dynamic motion. To obtain the linear acceleration due to motion, the gravity constant must be subtracted [25, 28].

Gyroscope:
Generally, a gyroscope is a device for measuring or maintaining orientation, based on how quickly an object rotates. This rate of rotation can be measured along any of the three axes expressed in rad/s. The rotation around the x, y and z-axis is also referred to as roll, pitch and yaw. For example, taking the lying smartphone on the table again, its axes indicate a angular velocity of zero. A conventional, mechanical gyroscope consists of a spinning wheel mounted on two gimbals, which allow it to rotate in all three axes. In contrast, a microelectromechanical gyroscope measures orientation computed from the angular rate, and is therefore also called rate-gyro. An important note to be made is that the gyroscope measures the force relative to the body as opposed to accelerometer and magnetometer, which measure the force relative to the Earth [25], [28], [27].

Magnetometer:
A magnetometer is an instrument used to measure the strength of the magnetic field in the surrounding area of the instrument. Its measured force is expressed in $tesla$. To get a perception of it, the Earth's magnetic field on the equator measures $31\mu T$ (0.00031T) and a typical fridge magnet 5mT (0.005T). Taking again the lying smartphone, its pointing head direction can be calculated by using those two axes, which are showing to the horizontal plane, in this case the x and y-axis. The computed direction then is indicated in degrees like $0°$ North, $90°$ East, $180°$ South and $270°$ West. The problem by magnetometer measurings are mainly the errors caused by iron containing material in the environment. A prolonged exposure of the smartphone to a fridge magnet decalibrates the magnetometer by at least a week [28], [27].

Barometer:
The barometer measures the ambient air pressure expressed in hPa or $mbar$. In contrast to the accelerometer, gyroscope and magnetometer, the barometer does not measure in three different axes because its measurement is not aligned to a particular direction. The use of a barometer for identifying height change information is motivated by the fundamental property that atmospheric pressure drops with an increase in altitude. It is well-known that the relation between barometric pressure and altitude is affected not just by the temperature, but also by various environmental phenomena such as weather patterns and humidity. For example, during hurricanes or temperate depressions, the pressure readings will obviously drop [29].

2.1.3 Applications

The demands for understanding human activities have grown in security and health care domain, especially in elder care support, rehabilitation assistance, diabetes and cognitive disorders. Huge amounts of resources can be saved if sensors can help caretakers to record and monitor patients with dementia or other mental pathologies all the time and report automatically when any abnormal behavior is detected [6]. Particularly, elder people could be monitored to recognize unpredictable incidents, such as falling from stairs or being hemmed in the toilet.

Another possible usage might be an automatic and personal reminder, which monitors one's physical daily activities and their lasting duration in order to estimate the calories burnt each day. Based on these calories, the system persuades participating in physical exercises if previous activities weren't enough to stay healthy. To realize such an application, the reminder has to recognize the physical activities and then to estimate how much calories these activities burn. The reason for applications of this kind is the prevailing sedentary lifestyle in modern society that leads to various physical and mental diseases like obesity, coronary heart diseases, type 2 diabetes and depression [17], [10].

In security domains, activity recognition may avoid accidents by automatic forwarding incoming calls to the mailbox when the current activity of the user is recognized as cycling or driving a car. Likewise, the mobile navigation may switch its mode automatically from pedestrian mode to driving mode when driving activity is recognized. This causes the map to scale down the focused section and show important information like the current speed [16].

2.1.4 Approaches

The recognition of human activities has been approached in two different ways, namely using external or wearable sensors. In the case of external sensors, devices such as cameras are fixed in predetermined points of interest, so the recognition is no longer possible when the user is out of the reach of the sensors. Although these systems are able to recognize complex activities, such as eating, taking a shower or washing dishes, their deployment entails high costs and its implementation is complicated and impractical [3]. In the case of using wearable sensors, the devices are attached to the user. Since these devices are becoming so small that they even fit into a watch, they can be worn during the whole day. However, the use of any body-worn sensor for gathering sensor data, e.g. accelerometer and gyroscope mounted on the foot, wrist or waist, is uncomfortable because it has also to be carried around for everyday activities. Furthermore, obtrusive devices distract the attention of the user, which is disturbing while concentrating. Using watches with embedded sensors might be a good idea for everyday activities since people wear their watches all the time. However, the drawback of this solution is that the movement of arms does not necessarily have a direct relationship with ongoing activities, even if body sensors may provide accurate results. Also a GPS is not suitable for activity recognition. Although it could detect and interpret one's movement as a displacement of location, it cannot recognize a specific activity. In particular, GPS does not work inside buildings where people spend most of their time in. Therefore, using sensor based activity recognition with a smartphone becomes the primary choice among all the solutions [10]. On the other hand, data gathering with a smartphone has to deal with other challenges since the smartphone has not a fixed position like a body-worn sensor.

The most challenging point in the activity recognition research is the nature of human activities. For instance, people can do several activities at the same time, e.g. watching TV and talking to a friend, or certain real life activities may be interleaved, e.g. stop cooking to do a phone call and then come back cooking. And depending on the situation, the interpretation of similar activities may be different, e.g. drinking coffee or drinking tea. Those concurrent and interleaved activities and the ambiguity of interpretation, let the recognition of complex activities remain a challenging and active area of research [4].

Valuable overviews of activity recognition systems can be seen in [3], [19] and in [15].

2.2 Background and Related Works in Indoor Localization

Indoor positioning deployment comprises in a first step finding the suitable system regarding requirements. Depending on requirements, the next step concerns the implementation of technologies and techniques, which entail different properties.

For the sake of simplicity we consider the terms indoor positioning and indoor localization as the same.

2.2.1 Indoor Positioning

The reason for positioning in indoor environments arises when outdoor positioning systems, e.g. GPS, do not provide sufficiently accurate results inside buildings. Therefore, alternative systems have to be considered to meet the requirements of indoor positioning.

Technologies:
At the highest level, all indoor localization technologies can be divided into categories employing three different physical principles: inertial navigation, mechanical waves and electromagnetic waves [2].

Inertial Navigation Systems (INS) make use of the Inertial Measurement Unit (IMU) consisting of sensors like accelerometers, gyroscopes, magnetometers and barometers. These sensors either sense information about the sensing device itself or information about the environment. Since these sensors have a significant drift, an INS is usually fused with other positioning technologies.

Positioning systems using the mechanical wave technology, such as audible or ultra sound, use the air and building material as propagation media. The measurements for this technology are based on sound pulses, which are sent from a sound emitter to a sound receiver. The difference between audible and ultra sound is the wavelength.

Most positioning technologies rely on electromagnetic waves. This category can be divided into three subcategories employing waves in 1. visible spectrum such as cameras, 2. infrared spectrum applying technologies such as beacons, and 3. microwave and radio spectrum. Systems performing in the 3. spectrum are inter alia bluetooth, radars, zigbee, cellular networks, ultra-wideband, radio frequency identification and WLAN.

Among localization technologies, the WiFi draws special attention. WLAN or WiFi (Wireless Local Area Networks; WiFi is used interchangeably or as a superset of IEEE 802.11) can be used to estimate the location of a mobile device within its network. Since WiFi access points are available in almost every indoor environment, the use of WiFi signals for indoor localization is a tempting approach. The advantage over other technologies is that the covered range outreaches that of many systems, such as that of Bluetooth or RFID. The property to use WiFi positioning systems with a standard mobile device, let this technology become the most widespread approach for indoor positioning [2].

Techniques:
Positioning techniques, also called measuring principle or positioning method, denote principal techniques for handling a particular technology for localization purposes. It is worth knowing that every technology can use many different techniques and every technique can be used by many different technologies. According to [21], positioning techniques can be divided into four general categories: dead reckoning, proximity, scene analysis and triangulation.

Dead reckoning refers to the usage of sensors, namely inertial navigation technology, which work without the need for external reference positions. If the initial position and orientation are known, subsequent positions and orientations of the moving platform can be updated continuously using information about a previously-estimated location. Therefore, this technique makes localization possible in environments, where the installation and maintenance of external infrastructure is not affordable. The disadvantage of this technique appears due to sensor drifts. Inaccuracies for estimated positions grow with time since the calculated position is based on the previous position. Hence, the success of this method is limited. [2], [21]. Most used positioning techniques in this category are step detection algorithms, which count the number of steps multiplied by an average human step length, and velocity approach techniques, which attempt to estimate the position by calculating the direction and the movement velocity.

Proximity techniques are positioning methods using technologies such as infrared, bluetooth or radio frequency identification to provide relative location information. These technologies need the proximity of a mobile device to detect it either through physical contact or through signal recognition. In order to be able to locate a mobile client, several sensors need to be deployed in the location area such that their sensing range has a minimal overlapping. Due to the knowledge of every sensor position, the position of a mobile client is located as the closest sensor position, which receives the strongest signal. [21], [22]. The most popular technique for this category is the cell of origin method, in which the sensing area consists of cells. The accuracy of this method relates to the density of sensor positions and signal range.

Scene analysis techniques first collect features in different points of a scene and then estimate the location by recognizing these features. More precisely, radio frequency technologies are deployed in a wide area of interest in order to provide radio signals in a certain position. A mobile device that receives these signals is used to collect the received signal strength of every sensor at various positions in the area of interest. These information are then saved in a table. Each position may have a characteristic pattern of sensor values that makes the positions distinctive. Because of that, a mobile client can be located by recognizing a particular pattern [23]. A commonly used technique according to this category is called fingerprinting because each position information can be unique like a fingerprint. The main challenge for this method is the recognition of previously collected signal patterns since received signals may be affected by many reasons.

Triangulation methods use geometric properties of triangles to estimate the target location, mainly utilizing radio frequency technologies, especially the WLAN. This technique category has two derivations: lateration and angulation.

Lateration measuring principles estimate the position of a mobile device by measuring its distance from at least three reference points with overlapping signal ranges. The distance can either be derived by measuring the Received Signal Strength (RSS) and computing the attenuation of the emitted signal strength, or by measuring the Time Of Arrival (TOA), Time Difference Of Arrival (TDOA) or Roundtrip Time Of Flight (RTOF) and computing the radio signal velocity and travel time [23].

Angulation measuring principles locate a mobile device by computing angles relative to at least two reference points. A common angulation technique is Angle Of Arrival (AOA), in which the location of the mobile device can be computed by two angle measurements and the knowledge of the distance between the two reference points [23].

Comparing localization accuracies of different works turns out to be very difficult due to the large variety of various technologies. Additionally, for each technology there are several measuring principles, which entail different results in terms of accuracy, coverage and costs. For instance, WiFi-based localization technology can be used with measuring principles like Cell of Origin, Fingerprinting or Triangulation. There is no best positioning system since each technology has unique advantages and disadvantages in performing location information. Depending on the environment and localization requirements the appropriate technology may be different. For instance, the WiFi-based localization does not need new infrastructure because it can reuse the devices with WLAN technology, which are widely deployed in public environments. However, they also have shortcomings such as limitations due to their properties as apparent in Figure 2.2, which shows that WLAN positioning systems achieve an accuracy of few meters covering an area under 100m. Thus, finding the best tailored solution for a certain request requires a careful analysis of parameters over all available technologies. [24], [2].

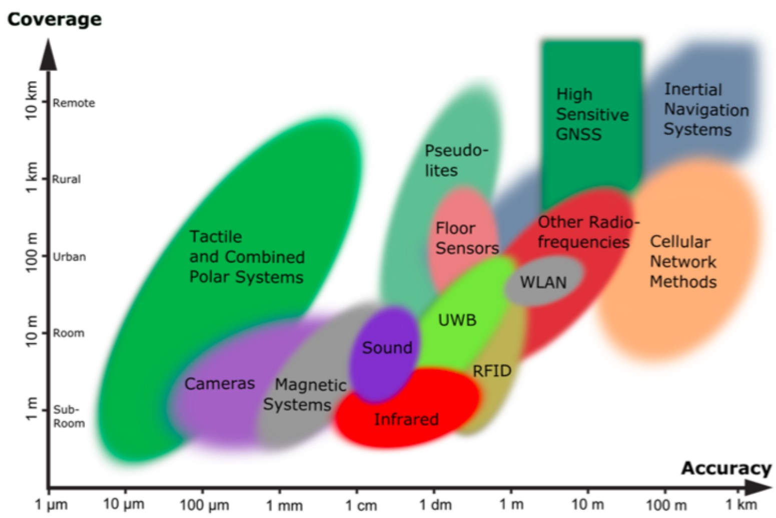


Figure 2.2: Overview of indoor positioning technologies in dependence on accuracy and coverage [2].

In order to gain an overview for indoor localization systems, following studies may be useful:

- Gu et al. give in [24] a comprehensive survey of numerous indoor positioning systems, which include both commercial products and research-oriented solutions. They assess different systems with criteria such as security and privacy, cost, performance, robustness, complexity, user preferences, commercial availability and limitations.
- Liu et al. provide in [23] an overview of existing wireless indoor positioning solutions and attempt to classify different techniques and systems.
- Farid et al. compare in [22] different indoor positioning methods and give comprehensive descriptions of common positioning technologies, techniques and algorithms.
- Torres-Solis et al. give in [21] a comprehensive overview of dominant indoor localization technologies. They identified radio frequency, photonic, sonic and inertial localization technologies as leading solutions in this field. They recommend to combine different technologies in order to benefit of their advantages and mitigate the disadvantages.
- Mautz provides with his Habilitation Thesis [2] a comprehensive and extensive overview of indoor positioning technologies, measuring principles, user requirements and definitions. For each introduced technology he characterizes some representative system implementations and concludes in a short summary the findings.

2.2.2 Applications

A positioning system is an application that enables a mobile device to determine its position and makes the position of the device available for position-based services, such as navigating, monitoring or tracking [24].

Applications of position-based navigating systems are, for example, touristic applications, which may lead tourists to find a museum or a historical attraction in a big city. Inside the museum, a navigation system could lead visitors to chosen expositions in different rooms. Other examples of navigating systems may be applications, which navigate people to a special shop in the shopping mall, or lead people to the correct train track at the train station.

Position-based monitoring systems may be useful particularly for medical purposes. Elder people with dementia could be monitored while going out, in order to find them if they forget the way back to home. In hospitals indoor monitoring systems can be used to prevent theft of expensive equipment, or to monitor patients with psychotic or mental diseases.

Applications in position-based tracking systems may be useful for policemen, firefighters and rescue services, for example, to detect the location of police dogs, which are trained to find explosives in a building, or tracking a firemen while rescuing people in a building on fire.

2.2.3 Challenges

The mentioned applications, which are thought for outdoor environments, work well by using GPS. However, GPS can not be deployed for indoor use, because the line-of-sight transmission between receivers and satellites is not possible in indoor environments. Comparing with outdoor, indoor environments are more complex. There are various obstacles, for example, walls, equipment or human beings influencing the propagation of electromagnetic waves, which lead to multipath effects. Some interference and noise sources from other wired and wireless networks degrade the accuracy of positioning [24].

According to [2] indoor environments are particularly challenging for position finding for reasons, such as no line of sight conditions, severe multipath due to signal reflection from walls and furniture, high attenuation and signal scattering due to greater density of obstacles, fast temporal changes due to the presence of people and opening doors, and high demand for precision and accuracy. On the other hand, indoor environments may also facilitate indoor positioning for reasons, such as small coverage areas, low weather influences, and infrastructure availability, such as electricity, internet access and walls suitable for target mounting.

Chapter 3

Theory of Mobility Recognition and Indoor Localization

3.1 Mobility Recognition

In order to focus on a recognition system that distinguishes fairly accurately between the indoor activities standing, walking, running, climbing stairs up/down and walking an inclination up/down, we need to make some decisions about the sensor, feature and classifier selection.

For our requirements we need sensors, which enable to differ similar activities. Therefore, we will employ more than one sensor for our system. The sensors should provide raw data, which allow a proper preprocessing of measurements to extract features; hence, no virtual sensors providing data obtained by fusing with other sensors. The position of the smartphone can be at one specific location since there is no reason to test several positions [13]. According to Ichikawa et al. [26], 57% of men put their mobile phones in their trouser pockets. Therefore, it is well established to gather data from a fix positioned smartphone in the front trouser pocket that records measurements from the accelerometer, gyroscope magnetometer and barometer with a sampling rate of 20 Hz.

3.1.1 Features

Calculations of classification algorithms do not rely on raw sensor data, rather they make their predictions based on extracted features from raw sensor data. Useful features for our selected sensors are the mean value 3.1, standard deviation 3.2, axes correlation 3.3 and linear regression 3.4, where x, y are samples of a sensor in one axis and t is the sample number:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.1}$$

$$\sigma_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2} \tag{3.2}$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y} \tag{3.3}$$

$$\hat{\beta} = r_{xt}\frac{\sigma_x}{\sigma_t} \tag{3.4}$$

3.1.2 Classifiers

According to [30], activity recognition algorithms can be broadly divided into two major categories. The first one uses machine learning techniques based on probabilistic and statistical reasoning. This category includes both supervised and unsupervised learning methods. The second category of algorithms is based on logical modelling and reasoning. The idea of logical approaches is to use logical knowledge representation for activity modelling. However, the vast majority of recognition systems use the first approach since its implementation is capable of handling noisy, uncertain and incomplete sensor data.

Supervised learning uses labeled data to train an algorithm, which then becomes able to classify unlabeled data. Unsupervised learning on the other hand tries to directly construct recognition models from unlabeled data. The idea behind it is to manually assign a probability to each activity and to predefine a stochastic model. The main difference between supervised and unsupervised probabilistic techniques is that supervised learning keeps a trace of its previous observed experiences and use it to dynamically learn the parameters of the stochastic activity models as opposed to unsupervised learning, which is using a predefined model to update the activity probability.

Supervised learning is further subdivided into online and offline. The question here is, whether the recognition task should be done in the device itself, that is carried to measure signals, or externally in a server, which receives the signals from the device through communication. On one hand, online processing provides immediate feedback but has the drawback of limited battery life and low computational capabilities. On the other hand, offline processing on a server is expected to have huge processing, storage and energy capabilities but it can only be used in systems where the user does not need to receive immediate feedback [3]. Since supervised learning algorithms need intensive computation to generate models from training data, the most widespread implementation is being done in servers [15].

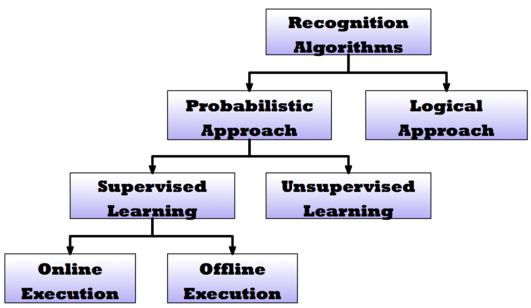


Figure 3.1: Categories of machine learning algorithms.

As we strive for accurate results, which require high computational ressources, we will implement supervised learning algorithms in an offline execution. Thus, concerns about time, memory and battery consumption will not be taken into consideration, rather a high classification accuracy is of a greater interest. Due to advantages of subject-independent classifiers for cross-people activity recognition, we will implement the plurality voting system with base-level classifiers k-nearest neighbors, naive Bayes, discriminant analysis and classification tree. These classifiers will be trained with the features mean value, standard deviation and axes correlation for the tri-axial sensors accelerometer, gyroscope and magnetometer. For the barometer we will use the feature linear regression.

Naive Bayes:
NB classification uses simple probabilistic methods based on Bayes' theorem to predict unclassified events. Since NB assumes that each single feature is independent of other features, it computes the likelihood based on the number of occurences of a feature f_i (i=number of features) belonging to its class c_j (j=number of classes). Using NB to classify between the activities inclination up and inclination down, and the features linear regression and variance, it would calculate the probability of an unclassified activity with the Formula 3.5 for 2 features and 2 classes. This calculation is done once for the activity inclination up and once for inclination down. The final prediction c for the unclassified activity is the class with the highest probability.

$$c = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_i P(f_i|c_j) \tag{3.5}$$

Classification Tree:
A CT is a decision tree used for classification. This algorithm represents a tree with classification group names in its leaves. To reach a leaf and thus to provide a prediction for unclassified data, the CT algorithm performs feature comparisons at each node. For instance, using the linear regression feature at its node to decide whether an unclassified activity belongs to the class inclination up or inclination down. A possible rule may be to classify the activity as inclination up if its linear regression value is negative, or to classify it as inclination down if its linear regression value is positive, as we can see in Figure 3.2. This rule makes sense since barometric pressure drops in increasing altitude. To set up these rules, the CT algorithm applies recursive partitioning on the training data set in order to obtain different subsets differing from each other by their feature values.

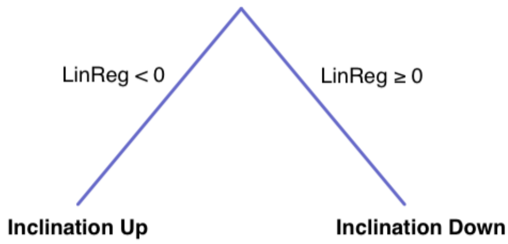


Figure 3.2: Process of CT classification algorithm.

K-Nearest Neighbor:
The idea behind the KNN algorithm is based on the comparison of features in a certain data set. Suppose this algorithm is used to classify an activity, which is either walking an inclination up or walking an inclination down. Before this classification can take place, the algorithm was been trained with a number of features in several data sets for both activities, for instance, the features linear regression and variance extracted from barometric sensor in 10 data sets of the activity inclination up and in 10 data sets of the activity inclination down. These 20 values are then stored in a trained file. Now, to classify a new activity, the KNN algorithm compares the feature values linear regression and variance of the new activity with 20 saved values and searches the nearest neighbor, which is the value with the greatest similarity to the unclassified features. Finding the nearest neighbor is provided with Euclidean distances. The unclassified activity is then classified as the same class as the nearest neighbor's class since their features are the most similar.

In Figure 3.3 the red and blue points represent saved feature values with known classes, i.e. linear regression value on x-axis and variance value on y-axis. Saving these values for a classification algorithm is called classification training because we have to let the classificator know which features can have what values. With this knowledge the classificator can classify unclassified values, i.e. the unclassified yellow point has a variance value and a linear regression value but contains no class information. To classify the yellow point, KNN assigns it to that class that is the closest to it. In this case, the red point is closer to the yellow point than the blue one, therefore, the yellow point is classified as the same class as the red point's class, which is inclination down.

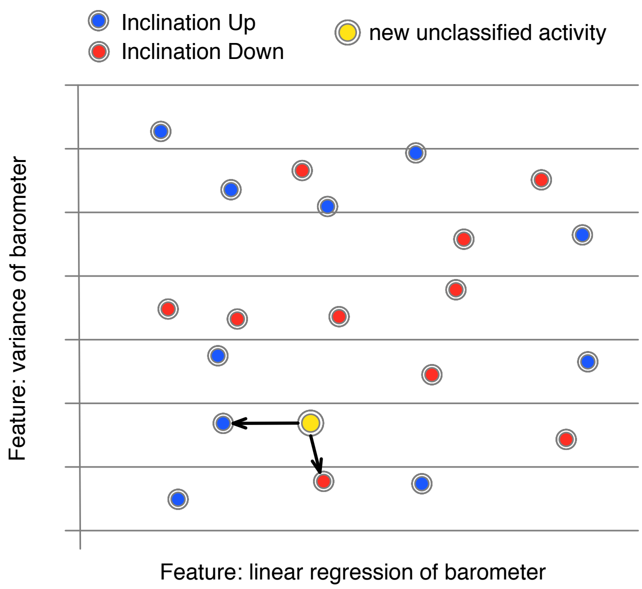


Figure 3.3: Process of KNN classification algorithm.

Discriminant Analysis:
The DA algorithm is a statistical method to analyze simultaneously several features in order to figure out to which class they belong to. Among other models, the model of linear discriminant analysis attempts to summarize multiple variables to a single variable by using linear combination. Similar to the method of regression analysis, DA defines a straight line separating two classes based on that single variable. For the previous example this means that the discriminant function calculates a new variable representing a straight line, which then is able to predict a new unclassified activity by determining whether the value of the unclassified activity is smaller or larger than the straight line.

In Figure 3.4 again, the red and blue points represent values that have been used to train the classifier, i.e. these data additionally contain the information of the class affiliation. In contrast to KNN that classifies a new data set by assigning it the to class of its nearest neighbor, the DA classifier computes a variable representing a straight line that attempts to make the best possible limit between the blue and red points, which entails a strict distinction of these two classes. A new data set is now either classified as the class that is above the limit (inclination up) or below the limit (inclination down). In this case, the yellow point is classified as inclination up.

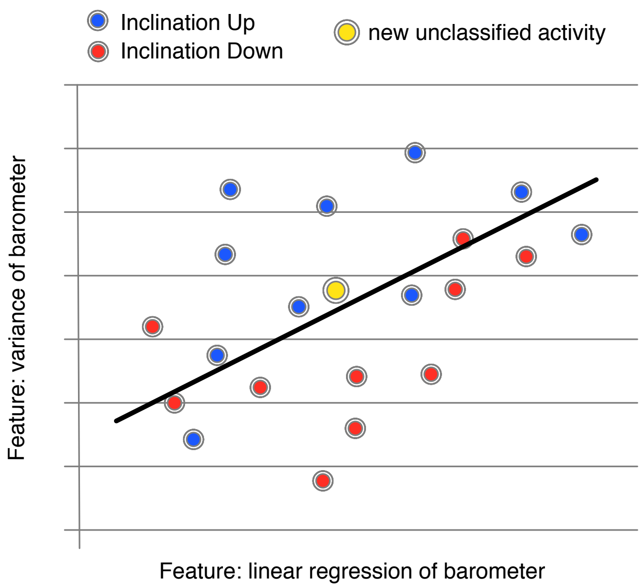


Figure 3.4: Process of DA classification algorithm.

Plurality Voting:
Each classification algorithm has advantages and disadvantages compared with other classifiers. Different algorithms follow different methods regarding the feature selection, i.e. weighting features as more or less significant, which leads to different classification results on the same data set. Hence, using more than one type of classification algorithms simultaneously, may perform better results.

Following pseudocode demonstrates the proceeding of plurality voting using several classifiers:

Algorithm Pseudocode for plurality voting

for each $classifier$ **do**
  $prediction \leftarrow classifier.predict(data)$
  for each $class$ **do**
    if $prediction = class$ **then**
      $class \leftarrow class + classificationAccuracy$
    end if
  end for
end for
$finalPrediction \leftarrow max(class_1, class_2, class_3, class_4, class_5, class_6, class_7)$

Plurality voting is especially used in political elections, but it can also be used for many other applications since its underlying understanding is very fundamental. Using plurality voting for classification purposes, makes the realization of getting a single classification result utilizing several classifiers possible. More precisely, each classifier attempts to assign the given data to one of the known activities. The activity, which has been assigned by the majority of the classifiers, is chosen as the final predicted activity [31].

In our case, an unclassified activity gets classified by each trained classifier (KNN, NB, DA, CT) that means that each classifier has a vote to predict the unclassified data as one of the investigated classes (standing, walking, running, stairs up, stairs down, inclination up, inclination down). To avoid an undecisive vote, e.g. when every classifier votes for another class or when two classes are voted by two classifiers each, we give different weights for each classifier. The weights are not chosen arbitrarily, they consist of the their classification accuracy. The final predicted activity is then chosen as that class that has the highest voting number.

3.2 Indoor Localization

In order to realize an indoor localization system that meets our requirements, we need to make some thoughts about the appropriate technology and technique.

As WLAN-based technology seems to be the most promising approach for our localization requirements due to its low complexity, low costs and acceptable accuracy, this thesis makes use of this technology.

3.2.1 Weighted Centroid

Since triangulation techniques have multiple drawbacks such as difficulties in exactly measure the angle of arriving signals in angle based methods, or challenges of time synchronization between sensor nodes in time based lateration methods, techniques like TOA, TDOA, RTOF or AOA are less common in WiFi positioning systems. In contrast, centroid localization provides a robust and efficient way for indoor localization.

The centroid approach counts as one of the simplest positioning algorithms since its calculations rely on basic geometric understandings. Using either RSS or TOA information, it determines the centroid of a defined geometric polygonal shape, which is the center of mass. However, it is different from angulation and lateration approaches and rather counts as an improved proximity approach [32].

Using the centroid algorithm in two-dimensional space requires the implementation of $n \geq 3$ beacons (B), also called access points (AP) or base stations (BS). Knowing exact coordinates of j^{th} beacon $B_j(x, y)$ makes it possible to determine the centroid between all beacon positions, which represents the position $P_i'(x, y)$ of the i^{th} mobile station (MS), also called mobile user, mobile customer or mobile device. Note that the real position $P_i(x, y)$ can strongly differ from the centroid.

$$P_i'\left(x, y\right) = \frac{1}{n} \sum_{j=1}^{n} B_j\left(x, y\right) \tag{3.6}$$

However, the centroid algorithm only performs the average value over all beacon coordinates that always leads to the same centroid for given beacon positions in the localization area. Certainly this delivers wrong location information since mobile stations move within the localization area. Therefore, using a weighted centroid approach, which implements distance information d_{ij} as weights w_{ij} between beacon B_j and MS_i, ensures an improved localization [32]. The formula to calculate the weighted centroid consists of the centroid algorithm with weighted beacon information, which provides a new position $P_i''(x, y)$.

$$P_i''\left(x, y\right) = \frac{\sum_{j=1}^{n}\left(w_{ij} \cdot B_j\left(x, y\right)\right)}{\sum_{j=1}^{n} w_{ij}} \tag{3.7}$$

The weight w_{ij} actually is defined as $1/d_{ij}$ since shorter distances are more weighted than longer distances. However, with RSS we are measuring power in dBm. Converting this unit to $watt$, we can directly use the converted power as weight. Thus, there is no need to derive distance information from RSS or TOA measurements.

$$w_{ij} = P_w = 10^{\frac{RSS_{dBm} - 30}{10}} \tag{3.8}$$

The localization error, used as accuracy information, is the most important requirement of positioning systems. Accuracy can be considered to be a potential bias of a positioning system. The higher the accuracy, the better the system. The accuracy is expressed in meters by computing the mean value of all localization error measurements. Typical WiFi positioning systems, which make use of RSS, achieve an accuracy of approximately 3-30m [23]. The localization error $f_i(x, y)$ is defined as distance between the real position $P_i(x, y)$ and approximated position $P_i''(x, y)$ [33].

$$f_i\left(x, y\right) = \sqrt{\left(x'' - x\right)^2 + \left(y'' - y\right)^2} \tag{3.9}$$

Figure 3.5 illustrates the way how the weighted centroid algorithm works. $P_i'(x, y)$ is the centroid position when the algorithm is unweighted. Weighting the algorithm shifts the weighted centroid position $P_i''(x, y)$ towards the real position $P_i(x, y)$, what reduces the localization error.

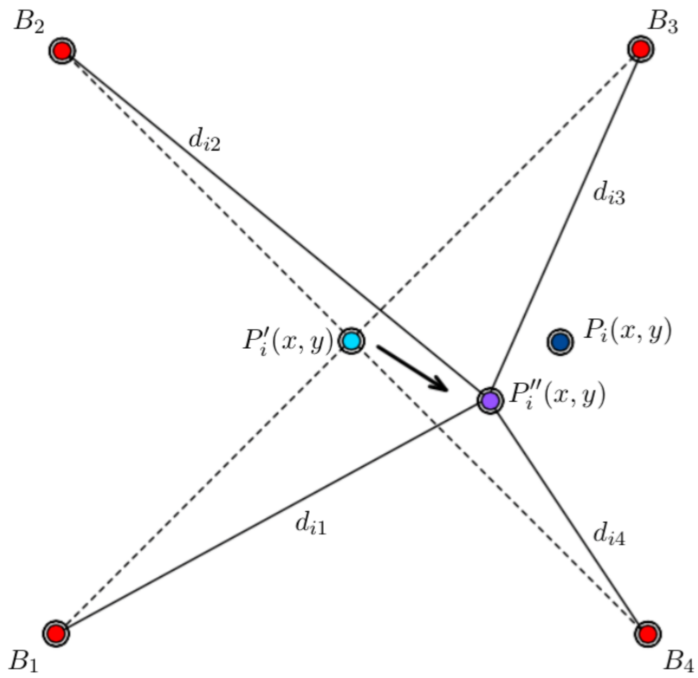


Figure 3.5: Exemplification of centroid and weighted centroid localization.

3.3 Combination of Mobility Recognition and Indoor Localization

Indoor localization in combination with mobility recognition has so far been implemented only in connection with dead reckoning positioning methods. However, dead reckoning methods do not distinguish between different activities, they merely make use of sensors to recognize steps,which allows to calculate position changes based on direction, velocity and number of steps. To our knowledge, this is the first thesis considering the implementation of human activity recognition to discern whether a mobile device should collect more or less RSS data.

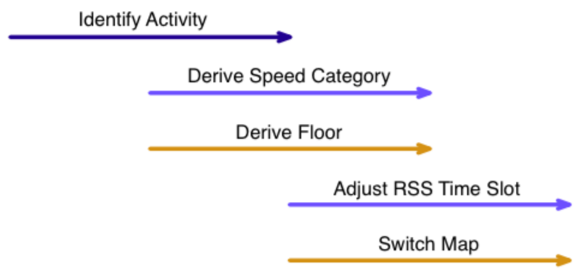


Figure 3.6: The use of mobility recognition for indoor localization.

As we can see in Figure 3.6, mobility recognition may be of advantage for indoor localization in two ways:

- A recognition system differentiating between activities such as climbing stairs up/down or walking an inclination up/down provides the useful information whether a mobile user has changed the floor and whether the switch was the floor above or below. This information can cause the localization system to continue on a different floor considering another set of access points, or to change the indoor map from one floor to another floor.

- Distinguishing between activities such as standing, walking or running gives the possibility to adjust the time slot, in which WiFi signals are gathered. Since localization with WiFi signals is often inaccurate due to measurement error and multipath propagation, the mean RSS value with an increased number of samples could reduce the deviation. In other words, increasing the number of samples may improve the RSS accuracy because an increased number of samples can improve the mean RSS value. However, increasing the number of samples needs to be adapted to the movement speed. Therefore, having knowledge of the current performing activity allows to regulate the RSS time slot accordingly.

In the first part of this thesis we investigate seven most practiced activities in indoor environments. How accurately they can be distinguished from each other and by which algorithms, sensors and features the best distinction can be achieved, are also parts of this thesis.

Further investigations refer to the second part of this thesis. In this part we examine accuracy changes of an indoor localization system when it is combined with a mobility recognition system. In this regard it is also of interest to verify the influence of the mobility recognition accuracy on the localization result. Due to time restrictions, we will not test a localization system performing results with all investigated activities. Only positioning while distinguishing between standing and walking in a two-dimensional space is object of this research. The implementation of the remaining activities may be considered as future work.

Chapter 4

Design and Implementation of the Mobility Recognition and Indoor Localization Application

4.1 Overall Structure

The overall structure illustrates the entire process of setting up the mobility recognition system, which then is implemented in the system for indoor localization. The structure is divided into a left and right part: the left part comprises steps using Android and the right part steps using Matlab. The upper half (blue) of the diagram shows the proceeding of mobility recognition and the lower half (red) the proceeding of indoor localization. A detailed description of each step is given in following sections.

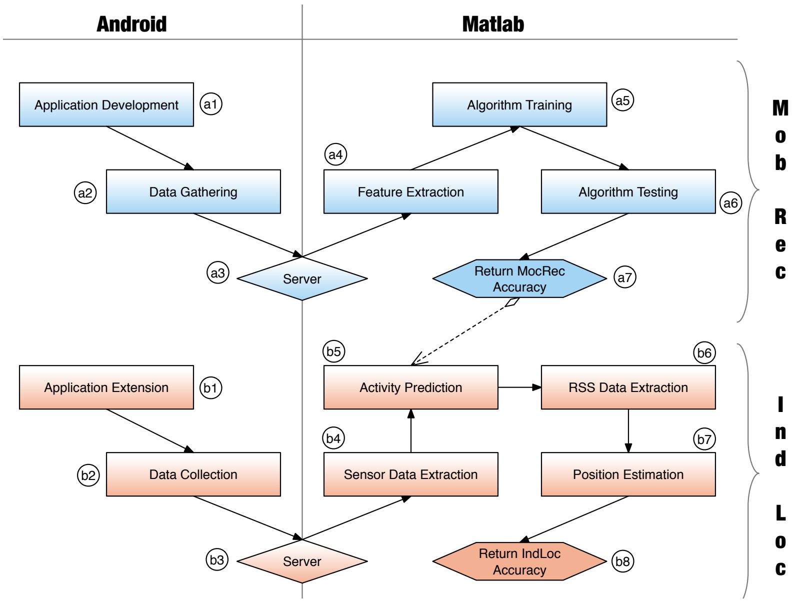


Figure 4.1: Overall structure (MobRec=mobility recognition / IndLoc=indoor localization).

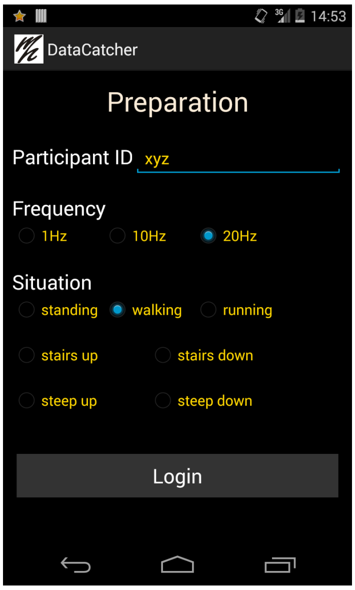
4.2 Mobility Recognition System

4.2.1 Application Development and Data Acquisition

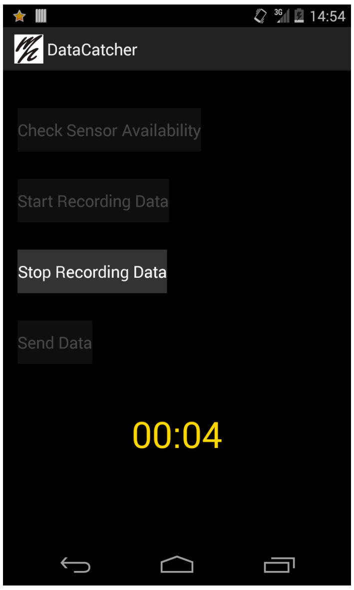
In order to realize an activity recognition system, sensor data needs to be gathered. Due to the fact that there is no smartphone App, which exactly meets our requirements for data acquisition, we develop an Android application *(step a1)* that is able to gather data of the sensors barometer (Bar), accelerometer (Acc), gyroscope (Gyr) and magnetometer (Mag) with a frequency of 20Hz. For our system, we use an LGE Nexus4 smartphone.

To gather data with this application *(step a2)* we found a couple of participants (family members and friends) performing different activities several times on different places. To do that, the participant first has to specify in the preparation page, which activity he is going to perform, as shown in Figure 4.2. This information is important since we train supervised classification algorithms, in which the data is labeled. The input field 'Participant ID' is optional and the frequency specification refers to the sampling frequency of the sensors. After the specification the participant passes through to the main page. In the main page the Android application first checks for the availability of the sensors before it starts recording. If that condition is satisfied, the participant pushes the button to start recording, puts the mobile phone in his trouser pocket and begins to perform the specified activity for about 10 seconds. In order to finish the record, the participant takes the mobile phone out of his trouser pocket and stops recording, and then pushes the button send data, which sends all sensed data as a csv file to an FTP server *(step a3)*.

All in all, each activity has been peformed 65 times that makes a total of 455 csv sensor data files stored in the private FTP server.



(a) Preparation Page



(b) Main Page

Figure 4.2: Screenshots of the mobility recognition application in Android.

The structure of the csv files saved in the private ftp server, which are sent by our Android application, is in the following order as we can see in Figure 4.3. This figure is an excerpt of a csv file containing the first second of sensor data of the activity 'walking'. Since we are sampling with 20Hz each second comprises 20 rows of data. It can also be seen that triaxial sensors do not only deliver one value, such as the barometer, but three; for each axis one value. For visualization purposes we cut the decimal numbers to one digit, which originally contained over 10 digits.

id,	bar,	x_acc,	y_acc,	z_acc,	x_gyr,	y_gyr,	z_gyr,	x_mag,	y_mag,	z_mag,
1,	948.1,	8.9,	-6.0,	6.7,	-0.9,	-1.3,	1.1,	9.4,	-20.2,	-38.0,
2,	948.1,	5.8,	-5.9,	3.0,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,
3,	948.1,	5.8,	-5.9,	3.0,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,
4,	948.1,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,
5,	948.1,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,
6,	948.1,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,
7,	948.2,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-6.5,	-10.3,	-34.4,
8,	948.2,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-6.5,	-10.3,	-34.4,
9,	948.2,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-6.5,	-10.3,	-34.4,
10,	948.2,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-6.5,	-10.3,	-34.4,
11,	947.8,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-14.5,	-4.1,	-31.7,
12,	947.8,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-14.5,	-4.1,	-31.7,
13,	947.8,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-14.5,	-4.1,	-31.7,
14,	947.8,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-14.5,	-4.1,	-31.7,
15,	948.0,	6.3,	-8.6,	0.1,	-0.5,	0.5,	1.6,	-21.7,	3.2,	-29.0,
16,	948.0,	6.3,	-8.6,	0.1,	-0.5,	0.5,	1.6,	-21.7,	3.2,	-29.0,
17,	948.0,	6.3,	-8.6,	0.1,	-0.5,	0.5,	1.6,	-21.7,	3.2,	-29.0,
18,	948.0,	6.3,	-8.6,	0.1,	-0.5,	0.5,	1.6,	-21.7,	3.2,	-29.0,
19,	948.0,	2.0,	-8.4,	0.1,	0.1,	0.3,	0.8,	-26.0,	10.9,	-25.6,
20,	948.0,	2.0,	-8.4,	0.1,	0.1,	0.3,	0.8,	-26.0,	10.9,	-25.6,

Figure 4.3: Excerpt of a stored csv file containing sensor data.

4.2.2 Feature Extraction and Algorithm Training

As mentioned, our recognition system is developed for offline processing. Therefore, the remaining steps continue in Matlab.

In order to train a classifier, it is needed to feed the algorithm with certain features of the data, which are characteristic for an activity, in order to distinguish it from other activities. As we can see in Figure 4.4, the measurement contains irregularities at the beginning and end. This is caused when the participant put and took out the smartphone from his trouser pocket. Therefore, we can not extract the mentioned features (mean value, standard deviation, axes correlation, linear regression) from raw sensor values of the entire measurement. Rather we extract the features *(step a4)* in a defined data window from somewhere in the middle. In addition, we will see that the length of that window affects the recognition accuracy. To investigate this parameter we compare the results of data window lengths 1, 2 and 3 seconds.

Before we can train classification algorithms with the extracted features, we divide the set of data into a train set and a test set. For our collection of samples, we decided to train the algorithms with 15 samples per activity and use the remaining 50 samples for testing. This makes a train set containing 105 csv files and a test set containing 350 csv files. Now we can train the classifiers KNN, NB, DA and CT with the extracted features *(step a5)*.

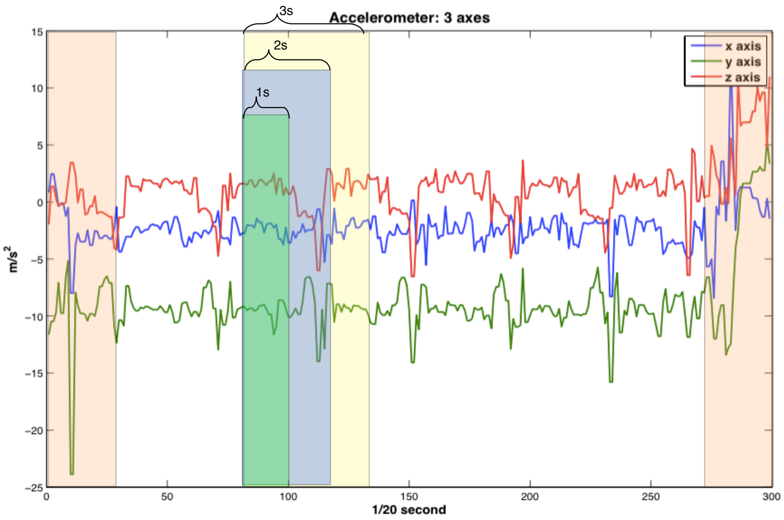


Figure 4.4: Define length of data window.

4.2.3 Recognition Algorithm Testing

To train an algorithms means to feed it with labeled data, such that it can learn and build a mathematical model, which then should be able to classify unlabeled data, what we call testing *(step a6)*. In our case for instance, every algorithm builds a mathematical model based on 7 activities with 4 features.

Such a mathematical model we can see in Figure 4.5. The CT classifier needs for its classification at the first node the standard deviation from the accelerometer at its x-axis to pass the condition. If that feature is smaller than a certain value, then this measurement is classified as standing, otherwise the CT will compute further calculations until it arrives a leaf.

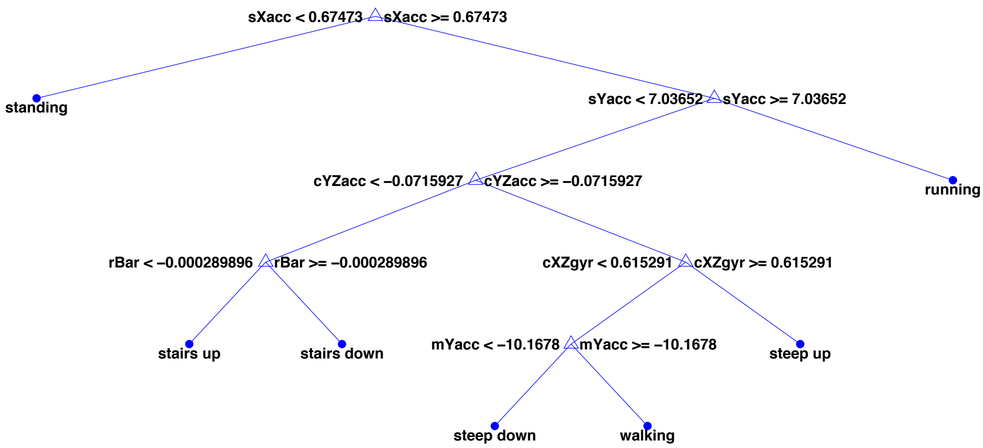


Figure 4.5: Classification tree classifier.

Performing the testing step on every of our 350 measurements, Matlab returns the results in a confusion matrix *(step a7)*. As we can see on Figure 4.6, a confusion matrix contains information about the actual and predicted classifications of a certain classifier. Suppose we test the classifier CT on the activities standing and walking, and both activities are tested with 10 samples each. When the result is presented in a confusion matrix like below, we can interpret the results as follows:

- standing is 7x predicted as standing and 3x as walking
- walking is always predicted as walking and has no misclassifications
- the accuracy of CT is $\frac{7+10}{7+3+0+10} = 85\%$

		PREDICTED	
		standing	walking
REAL	standing	7	3
	walking	0	10

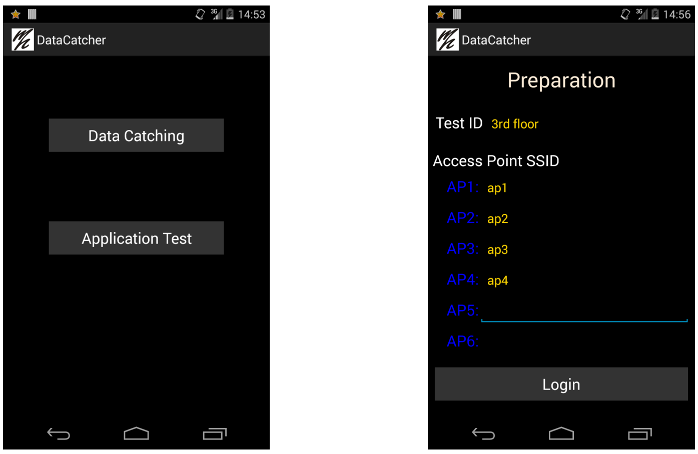
Figure 4.6: Example of a confusion matrix.

Since our goal is to maximize the recognition accuracy of every algorithm, we test the classifiers in different variations, i.e. verifying the influence of sensor combinations, feature combinations and data window length on the classification accuracy. The plurality voting algorithm then takes the highest result of each algorithm to provide a new confusion matrix, which is the prediction result of all four classifiers. This result is then used in our indoor localization system.

4.3 Indoor Localization System

4.3.1 Application Extension and Data Acquisition

As we aim to combine an indoor localization system with an activity recognition component, we develop our system for an offline execution since online executions suffer from loss of accuracy.



(a) Mode Page **(b)** Preparation Page

Figure 4.7: Screenshots of the indoor localization application in Android.

Again we need an application that is able to gather WiFi signal data from several access points in addition to sensor data. This should enable us to locate a mobile station, taking into account the prediction of the current performed activity. Therefore, we supplemented our Android application with an additional mode *(step b1)*. As we can see on Figure 4.7, the mode page shows the additional mode 'Application Test', which works in exactly the same way as the previous mode 'Data Catching' except the point that it additionally gathers RSS values of several predefined access points with a frequency of 20Hz.

To gather data *(step b2)* for our localization purposes, we deploy several APs on predefined locations in our environment and measure their coordinates. For comparative purposes we test our application in two environments, both are located on the 3rd floor in the IAM building of the University of Bern. The first environment covers an area of 13m x 16m with 4 APs, see Figure 4.8, and the second evironment an extended area of 24m x 16m with 6 APs, see 4.9. The red points on the map are the positions of the APs and the blue line marks the walking trace.

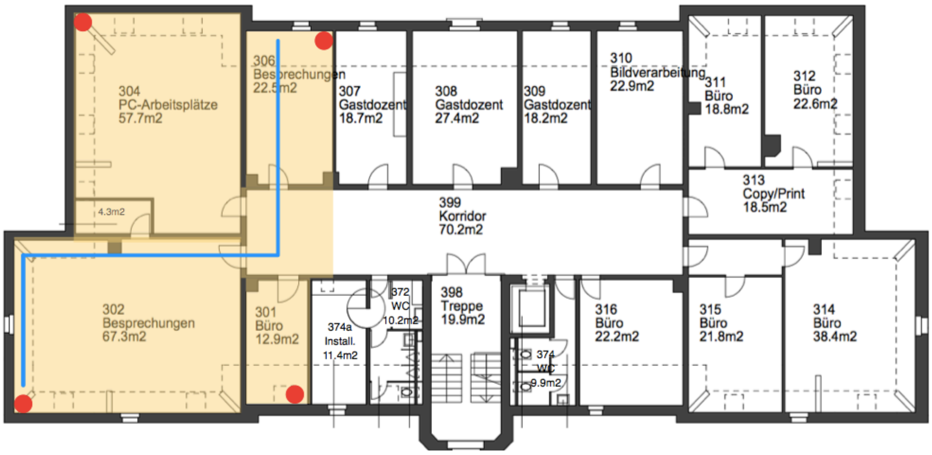


Figure 4.8: Map of first environment.

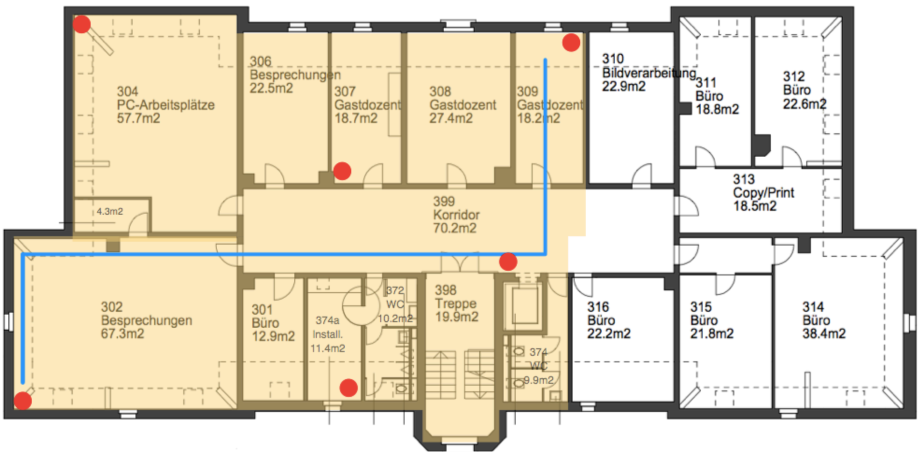


Figure 4.9: Map of second environment.

Like in the mobility recognition part, the sensor and RSS data are stored into a csv file, which is then sent to a private ftp server *(step b3)*. Figure 4.10 shows an excerpt of a csv file containing RSS values (red) in addition to sensor data (blue).

id,	bar,	x_acc,	y_acc,	z_acc,	x_gyr,	y_gyr,	z_gyr,	x_mag,	y_mag,	z_mag,	ap1,	ap2,	ap3,	ap4
1,	948.1,	8.9,	-6.0,	6.7,	-0.9,	-1.3,	1.1,	9.4,	-20.2,	-38.0,	-87,	-87,	-68,	-79
2,	948.1,	5.8,	-5.9,	3.0,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,	-87,	-87,	-68,	-79
3,	948.1,	5.8,	-5.9,	3.0,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,	-87,	-87,	-68,	-79
4,	948.1,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,	-87,	-87,	-68,	-79
5,	948.1,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,	-87,	-87,	-68,	-79
6,	948.1,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	1.6,	-15.7,	-37.0,	-87,	-87,	-68,	-79
7,	948.2,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-6.5,	-10.3,	-34.4,	-87,	-87,	-68,	-79
8,	948.2,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-6.5,	-10.3,	-34.4,	-87,	-87,	-68,	-79
9,	948.2,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-6.5,	-10.3,	-34.4,	-87,	-87,	-68,	-79
10,	948.2,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-6.5,	-10.3,	-34.4,	-87,	-87,	-68,	-79
11,	947.8,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-14.5,	-4.1,	-31.7,	-87,	-87,	-68,	-79
12,	947.8,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-14.5,	-4.1,	-31.7,	-87,	-87,	-68,	-79
13,	947.8,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-14.5,	-4.1,	-31.7,	-87,	-87,	-68,	-79
14,	947.8,	7.4,	-4.7,	1.8,	1.5,	-2.5,	2.2,	-14.5,	-4.1,	-31.7,	-87,	-87,	-68,	-79
15,	948.0,	6.3,	-8.6,	0.1,	-0.5,	0.5,	1.6,	-21.7,	3.2,	-29.0,	-50,	-69,	-71,	-78
16,	948.0,	6.3,	-8.6,	0.1,	-0.5,	0.5,	1.6,	-21.7,	3.2,	-29.0,	-50,	-69,	-71,	-78
17,	948.0,	6.3,	-8.6,	0.1,	-0.5,	0.5,	1.6,	-21.7,	3.2,	-29.0,	-50,	-69,	-71,	-78
18,	948.0,	6.3,	-8.6,	0.1,	-0.5,	0.5,	1.6,	-21.7,	3.2,	-29.0,	-50,	-69,	-71,	-78
19,	948.0,	2.0,	-8.4,	0.1,	0.1,	0.3,	0.8,	-26.0,	10.9,	-25.6,	-50,	-69,	-71,	-78
20,	948.0,	2.0,	-8.4,	0.1,	0.1,	0.3,	0.8,	-26.0,	10.9,	-25.6,	-50,	-69,	-71,	-78

Figure 4.10: Excerpt of a stored csv file containing additionally RSS values.

4.3.2 Data Extraction and Position Estimation

Here as well, we implement our system in an offline mode since we require high computation power due to accuracy purposes. Therefore, we will continue the further work in Matlab.

Since we now have csv files, which not only contain sensor values, but also RSS values of either 4 or 6 APs, we need first to extract the features of the sensor data *(step b4)* in order to predict the activity with plurality voting *(step b5)*. Note that the classifiers predict the activity based on those features, which have yielded the best accuracy. Based on the prediction we can then extract a number of RSS data *(step b6)* to use them for the weighted centroid algorithm that yields approximated location information *(step b7)*. The difference between the real and approximated position provides us the localization error *(step b8)*.

How the localization depending on the prediction exactly works, we see on the next two Figures. For this we imagine two time windows; a left window (blue) that contains sensor data and a right window (red) that contains RSS data. The left window is responsible to predict the activity every second, which means that this window always reads 20 rows (1 second always comprises 20 rows of data because we sampled with a frequency of 20 Hz) of sensor data, extracts the features, lets every trained algorithm make a prediction, and chooses that activity as the final predicted activity that has been predicted by most algorithms (implementation of plurality voting). The information about the predicted activity can now be used by the right window that is responsible for the localization. However, the right window is not fixed on reading only 20 rows like the left window, it rather adjusts the number of rows to the prediction because it is configured in such a way that it keeps its length also to 20 rows if the prediction of the left window is the activity 'walking'. However, if the left window predicts the activity 'standing' then the right window enlarges its length as long as 'standing' is predicted. For instance, it computes the position of a walking user by reading 20 rows, calculating the mean of these 20 values for each AP, and using the weighted centroid algorithm to compute the coordinates. On the other hand, when the user is standing, for instance 3 seconds, then it does not compute 3 times the mean value of 20 rows, instead it computes the position based on the mean value of 60 rows.

Step by step explanation:
In Figure 4.11 we see a csv file containing sensor data on the left side, and RSS data on the right side. Since the left window (blue) is responsible to predict the activity every second, it reads the first 20 rows (row numbers 1-20) and predicts the activity. Assume, the prediction is 'walking'. This information is now sent to the right window. The right window (red) knows that it must read also 20 rows to compute the position if the activity is 'walking'. Therefore, it computes the mean value of row numbers 1-20 to return the position coordinates.

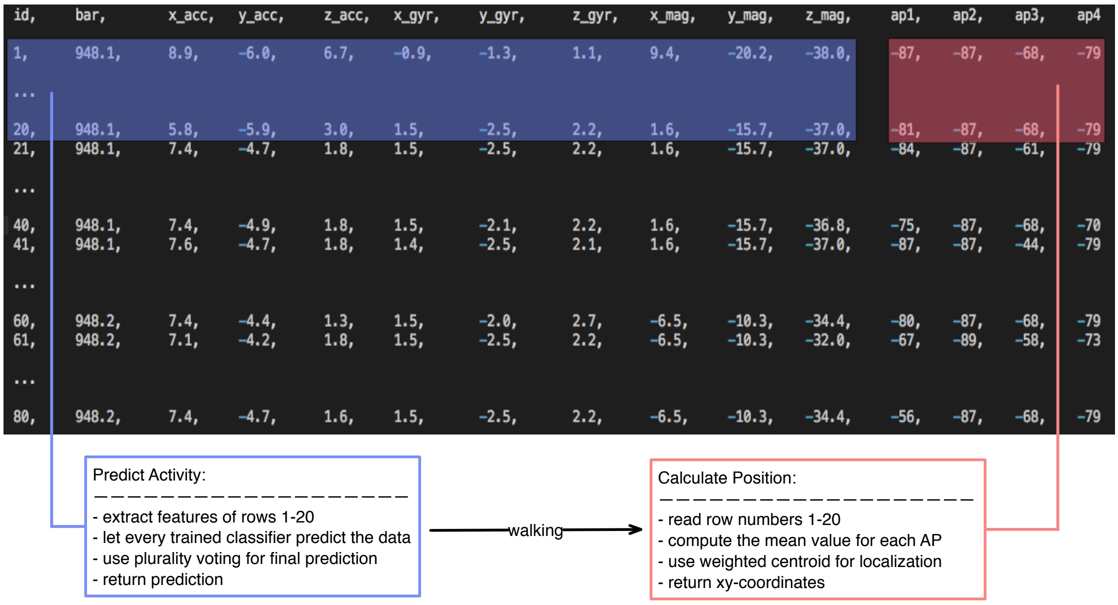


Figure 4.11: Adjustment of the time window while walking.

In Figure 4.12 we see the next step. Now the left window takes the next second, i.e. the next 20 rows (row numbers 21-40). Again the left window makes a prediction. Assume, the prediction is 'standing'. When this information is sent to the right window, the right window does not calculate the position. It only remembers the row number, which was the starting point of the activity 'standing' (in this case row number 21). Assume, the left window again predicts the activity standing (of row numbers 41-60). This time again, the right window does not calculate the position, it only remembers the starting point of the activity standing (still row number 21). When now the left window predicts the activity 'walking' on row numbers 61-80, the right window recognizes that the 'standing' activity is over. Therefore, it takes row number 60 as the end point of the activity 'standing' and calculates the position with the mean value of the rows 21-60. In addition to that, this prediction (rows 61-80) does not only signifies the end of the activity 'standing', it does also give the information of the next prediction, i.e. compute the position of rows 61-80 for the activity 'walking'. So, a walking mobile user is located every second, whereas a standing mobile user is located once (until the user again begins walking).

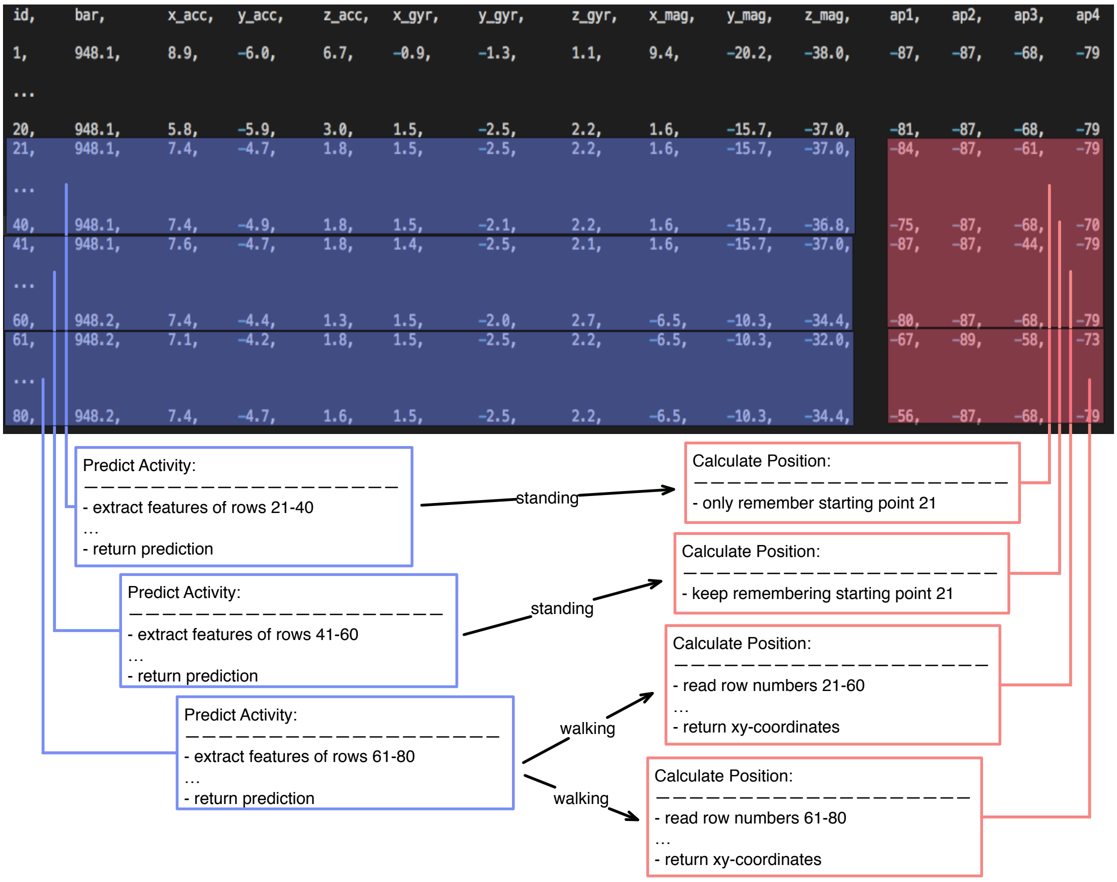


Figure 4.12: Adjustment of the time window while standing.

Chapter 5

Evaluation

5.1 Results and Analysis for Mobility Recognition

In mobility recognition systems several parameters can affect the recognition accuracy. In order to maximize the accuracy of our activity recognition system we consider the influence of several features and sensors on the accuracy of 4 classification algorithms, which are used to distinguish between 7 activities (standing, walking, running, stairs up/down, inclination up/down). Another parameter that also can affect the accuracy is the length of the time slot, that is used to define the duration of training for a sample.

Since classifiers, which are trained only for 1 second, yield another result than the same classifier trained for 3 seconds, we examine this parameter as well. But what happens, if we train an algorithm for 3 seconds and test it on samples that are trained for 1 second? Also interesting is the question, whether the activity 'walking an inclination' is accurately distinguishable from other activities, since its investigation is not common. Is the accuracy of plurality voting higher than the accuracy of individual classifiers? We attempt to answer these and other questions in the following sections.

As we already mentioned, we investigate in our indoor localization system only the influence of mobility recognition distinguishing between 2 activities. The results for the recognition of 7 activities can be used in a future work, where the RSS time slot adjustment can also be used for the running activity or for indoor positioning in different floors.

5.1.1 Distinction between 7 Activities

To receive an answer to the question 'which features and sensors provide the best accuracy in which time slot', we represent the results in 3 parts. Each part handles a certain data time slot (1, 2 or 3 seconds), in which each classifier consists of tables. These tables contain achieved recognition accuracy results (in %) for every feature and sensor combination. In order to infer knowledge about the most significant sensor and feature, we marked for each sensor and sensor combination the best result with the color blue. Additionally, the best achieved accuracy in a table is marked with the color red, which represents that combination of features and sensors that achieved the best result for a classifier. This combination is then used for the plurality voting algorithm.

Individual classifiers trained and tested with 1 second time slot length

It is noticeable that the highest accuracy for KNN in Table 5.1 is achieved by 2 sensor combinations (AG and BAG), so the additionally used Barometer does not bring any gain in accuracy. The same effect we see for the CT in Table 5.4 where the combination AGM and BAGM provide same results.

Concerning features, we see that using the mean value and correlation solely does not provide best results for any sensor combination. Some features provide worse results when they are combined with other features, and some of them still provide the same result even if an additional feature is used, such as red marked MS and MSC in Tables 5.3 and 5.4.

Table 5.1: KNN: 7 activities / time slot 1 second

	M	S	C	MS	MC	SC	MSC
B	13.1	13.1	13.1	13.1	13.1	13.1	13.1
A	31.7	59.1	25.1	62.9	35.7	61.7	66
G	37.7	47.4	26.3	57.1	31.7	52.9	53.7
M	30.6	42	19.7	31.7	31.4	34.3	32
BA	31.7	59.1	24.9	62.9	35.7	61.7	66
BG	37.4	47.7	26.3	57.1	31.7	52.9	53.7
BM	30.6	42.6	19.4	31.7	31.4	34.3	32
AG	34	68	31.4	66.6	44	72	69.4
AM	31.4	62.6	24.3	44.3	33.4	62.6	45.1
GM	30.6	58.6	27.7	31.7	31.7	48.6	32.3
BAG	34	68	31.7	66.6	44	72	69.4
BAM	31.4	62.6	24.3	44.3	33.4	62.6	45.1
BGM	30.6	58.6	27.7	31.7	31.7	48.6	32.3
AGM	31.4	68.9	35.1	44.6	33.4	68.3	45.7
BAGM	31.4	68.9	35.1	44.6	33.4	68.3	45.7

Table 5.2: NB: 7 activities / time slot 1 second

	M	S	C	MS	MC	SC	MSC
B	16	16	16	16	16	16	16
A	36.9	62.9	32	68	50.3	65.7	68.9
G	39.4	58.3	28.6	63.7	49.1	60.3	62.9
M	24.6	36.3	16.3	36	20.9	34	34.9
BA	36.3	62.6	31.4	66.6	46.9	65.4	68
BG	37.1	56.3	26.9	63.1	47.1	59.4	62.9
BM	21.1	36	16	37.1	20.9	34.3	36.9
AG	47.4	77.4	43.4	80.3	60	77.1	80.9
AM	36.9	62.3	33.1	68.6	53.4	63.1	67.4
GM	43.4	59.7	31.1	66.3	50.9	65.4	68.6
BAG	47.4	76.9	40.3	80	59.1	76.3	79.4
BAM	38.6	63.4	34.6	70	52.9	62.3	67.7
BGM	41.4	60.6	32	66	54	64	68.3
AGM	48.9	73.4	42.3	78.3	60	74.6	79.7
BAGM	48.9	74.3	43.1	78	60.3	74	78.3

Table 5.3: DA: 7 activities / time slot 1 second

	M	S	C	MS	MC	SC	MSC
B	13.1	13.1	13.1	13.1	13.1	13.1	13.1
A	36	61.4	32.6	66.6	42	63.1	67.1
G	33.4	53.1	28.3	61.1	34.3	55.7	58.9
M	13.1	32.9	17.4	30.9	20.6	30.9	33.4
BA	34.6	60.3	33.4	67.1	42	63.1	67.4
BG	30	51.7	28.9	60.9	35.4	55.7	58.6
BM	13.1	31.1	16.3	31.1	19.4	32.3	33.7
AG	40	75.1	40.9	78.3	55.7	72.3	76.9
AM	32.3	59.1	34.9	59.7	42	62.3	64.9
GM	28.3	59.1	29.1	60.6	34.9	62.9	64.3
BAG	40.6	75.7	40	78.9	54.9	72.9	78.9
BAM	32.6	58	33.7	60.3	42	63.4	64.6
BGM	28	58.3	29.1	60.3	36	63.1	62
AGM	40.3	70.9	40.3	74.3	55.4	69.1	72.9
BAGM	40.3	70.6	41.1	75.4	54.6	68.9	72.9

Table 5.4: CT: 7 activities / time slot 1 second

	M	S	C	MS	MC	SC	MSC
B	18.9	18.9	18.9	18.9	18.9	18.9	18.9
A	27.1	63.1	27.1	66.9	37.1	62.3	58
G	31.4	45.4	30	47.4	44.6	54	56.9
M	28.3	39.4	17.7	37.7	27.7	42	36.3
BA	27.1	63.1	26	67.7	37.1	61.4	60
BG	33.4	46.9	29.1	48.3	34.6	53.7	56.3
BM	26.6	34.6	20.3	34.9	31.4	34	34.6
AG	38	64	35.7	66	43.4	66.6	67.4
AM	30.3	58.9	26.6	59.4	39.4	57.4	60.6
GM	32.3	54.9	27.7	57.7	32.6	59.7	58.6
BAG	38	64	35.4	65.7	43.1	66.6	67.4
BAM	30.6	58.9	24.6	64.9	39.4	56.9	58.6
BGM	32.3	54.9	27.7	57.7	34.3	59.1	58.6
AGM	45.1	65.7	35.7	68	45.4	65.1	68
BAGM	44.3	65.7	35.7	67.7	45.4	65.1	68

LEGEND
Sensors: B = barometer / A = accelerometer / G = gyroscope / M = magnetometer
Features: M = mean value / S = standard deviation / C = correlation
Algorithms: KNN = k-nearest neighbors / NB = naive Bayes / DA = discriminant analysis / CT = classification tree

Individual classifiers trained and tested with 2 seconds time slot length

Here again, we see that sensor combinations AG and BAG provide the same results in Tables 5.5 and 5.6. And again the reliability of the Barometer can be questioned. It is striking that the best results always include the sensor combination AG. The use of more sensors, than only AG, does not seem to increase the accuracy. Merely the DA in Table 5.7 and the CT in Table 5.8 provide a hardly noticeable improvement when additionally using the magnetometer or the barometer.

Comparing these results with those of time slot length 1, we notice that the red marked accuracy results are higher. However, we cannot recognize any patterns of distribution in the blue marked results.

Table 5.5: KNN: 7 activities / time slot 2 seconds

	M	S	C	MS	MC	SC	MSC
B	18.6	18.6	18.6	18.6	18.6	18.6	18.6
A	37.4	66	31.7	70.6	42	72	73.7
G	39.1	48.9	34.6	50.9	36	62.9	62
M	34	37.4	21.4	34.9	33.7	37.7	34.9
BA	37.4	66	31.7	70.6	42	72	73.7
BG	39.1	48.9	34.6	50.9	36	62.9	62
BM	34	37.4	21.4	34.9	33.7	37.7	34.9
AG	38	74.6	45.1	74	53.1	77.1	79.4
AM	36	66	26.9	48	36.6	68.3	48.6
GM	34	52.6	34	34.9	34.6	55.1	35.4
BAG	38	74.6	45.1	74	53.1	77.1	79.4
BAM	36	66	26.9	48	36.6	68.3	48.6
BGM	34	52.6	34	34.9	34.6	55.1	35.4
AGM	35.7	70	40.9	48.3	36.9	73.4	49.4
BAGM	35.7	70	40.9	48.3	36.9	73.4	49.4

Table 5.6: NB: 7 activities / time slot 2 seconds

	M	S	C	MS	MC	SC	MSC
B	25.7	25.7	25.7	25.7	25.7	25.7	25.7
A	40.9	70.6	35.7	77.4	51.4	73.1	78.9
G	38	58.6	34.3	65.1	46.6	62.3	65.4
M	22.6	39.4	13.1	40	22.3	34.6	37.4
BA	47.4	71.7	42	81.1	55.1	75.1	80
BG	41.7	60.3	37.1	64.6	49.1	62.3	66.6
BM	28	39.7	22	44	26.9	40	41.1
AG	50.9	76	55.1	84.6	65.7	84	88.3
AM	41.1	67.1	34.3	72.9	51.1	72	75.4
GM	42.6	63.7	32	68.9	49.4	69.4	70.6
BAG	57.1	76.6	57.7	86.3	67.7	83.4	88.3
BAM	44.6	70.3	40	76.6	54.6	75.4	78.3
BGM	46.6	64	36.3	69.7	50.6	68.9	70.9
AGM	52.6	76.9	51.4	82	61.4	79.7	84.9
BAGM	54.9	77.4	54	82	62.3	80.9	84.6

Table 5.7: DA: 7 activities / time slot 2 seconds

	M	S	C	MS	MC	SC	MSC
B	24.6	24.6	24.6	24.6	24.6	24.6	24.6
A	39.4	67.1	40.6	75.1	48.6	74.9	76.6
G	31.4	56.9	36.6	58.6	31.1	59.7	60
M	13.1	32	14	32	22	37.4	35.4
BA	46.3	69.4	42	80.9	50.9	75.7	78.9
BG	31.7	56.9	33.7	58.3	34	57.4	58
BM	18.6	35.7	22.6	33.4	25.7	39.1	37.7
AG	47.4	78.6	53.4	86	58	82.9	86.9
AM	38.9	68.6	37.1	73.1	49.7	70.9	75.4
GM	25.4	64	30.3	61.7	31.1	63.4	61.4
BAG	53.1	79.4	54.9	83.7	59.1	81.4	84.3
BAM	45.4	72.9	37.1	78	52.3	71.7	75.4
BGM	25.7	63.4	32.6	64	33.4	63.4	61.1
AGM	46	80.6	49.4	87.7	58.6	82	83.4
BAGM	49.7	80	50.9	83.7	60.6	82.6	82

Table 5.8: CT: 7 activities / time slot 2 seconds

	M	S	C	MS	MC	SC	MSC
B	18.9	18.9	18.9	18.9	18.9	18.9	18.9
A	44.9	62.9	30.6	65.7	45.1	70	67.7
G	37.7	48.6	36.9	43.7	41.4	56.9	49.7
M	28.3	35.4	18.3	37.1	21.1	29.7	27.7
BA	43.7	70.9	39.1	74.3	47.7	67.4	74.3
BG	40.3	50	28.6	47.4	36.6	56.3	50.9
BM	26.6	34.9	26	38.9	28	40	38.9
AG	47.1	67.4	42	72.3	50	70.9	75.4
AM	44.9	62.9	25.7	67.7	47.1	65.4	70.6
GM	36.6	54.6	35.4	52.6	38.3	52.3	49.7
BAG	47.4	66.3	44.9	71.1	49.1	71.1	75.7
BAM	43.7	68	28.6	74.3	50.9	67.4	74.3
BGM	39.7	53.7	35.1	53.4	41.4	55.4	52.9
AGM	48	67.4	43.1	72.3	45.1	70.9	75.7
BAGM	47.4	66.3	47.7	71.1	48.6	71.1	76

LEGEND

Sensors: B = barometer / A = accelerometer / G = gyroscope / M = magnetometer
Features: M = mean value / S = standard deviation / C = correlation
Algorithms: KNN = k-nearest neighbors / NB = naive Bayes / DA = discriminant analysis / CT = classification tree

Individual classifiers trained and tested with 3 seconds time slot length

Again we have the same effects like in time slot length 1 and 2, namely the sensor combination AG is included in every red marked result, compared to time slot length 2 the achieved accuracy results are increased again but only slightly and finally, the feature standard deviation is included in every red marked result (here in particular the correlation feature as well).

Even if not clearly visible, here we can detect a subtle change concerning the distribution of the blue marked results. As we see on Tables 5.9 - 5.12, the blue marked results in this slot length are primarily located in the right half of the tables, what suggests the use of several combined features.

Table 5.9: KNN: 7 activities / time slot 3 seconds

	M	**S**	**C**	**MS**	**MC**	**SC**	**MSC**
B	17.7	17.7	17.7	17.7	17.7	17.7	17.7
A	39.4	66.9	32.6	72.3	42	71.7	72.3
G	37.1	54.6	42.9	58.6	36.6	74.6	70.9
M	32	37.7	21.7	33.7	32.3	38	34.3
BA	39.4	66.9	32.6	72.3	42	71.7	72.3
BG	36.9	54.6	42.9	58.6	36.6	74.6	70.9
BM	32	37.7	21.7	33.7	32.3	38	34.3
AG	41.4	76.9	55.1	76.9	56.3	81.7	80.6
AM	34.6	60.9	37.4	46.9	34.6	67.4	47.4
GM	32	51.1	32.6	33.7	32.6	56	34.9
BAG	41.4	76.9	55.1	76.9	56.3	81.7	80.6
BAM	34.6	60.9	37.4	46.9	34.6	67.4	47.4
BGM	32	51.1	32.6	33.7	32.6	56	34.9
AGM	34.6	64.9	45.7	47.1	35.1	74.9	47.7
BAGM	34.6	64.9	45.7	47.1	35.1	74.9	47.7

Table 5.10: NB: 7 activities / time slot 3 seconds

	M	**S**	**C**	**MS**	**MC**	**SC**	**MSC**
B	27.7	27.7	27.7	27.7	27.7	27.7	27.7
A	45.1	69.7	38.9	78.3	56.6	76	78.3
G	40.3	64	35.7	64.3	42.6	67.1	64
M	23.4	37.7	18.9	38.9	24.6	37.4	38
BA	48.9	77.4	43.7	84.6	60.6	79.1	84.6
BG	44.9	69.7	39.7	67.1	47.4	68.3	67.1
BM	27.1	39.7	28	42.9	31.7	42	43.1
AG	54	80	55.7	88.6	67.4	86	88.9
AM	44	64.6	41.1	73.1	55.7	72.9	74.9
GM	43.4	62	36.6	65.4	44	66	65.4
BAG	59.1	81.7	57.7	87.7	69.7	85.1	88
BAM	50.3	72.9	44.3	79.4	61.1	76.3	80
BGM	44.6	65.1	42	66.9	48.3	68.6	66.9
AGM	54	78	51.7	84.3	67.1	83.7	86.6
BAGM	58.9	80.3	55.4	84	68	85.1	85.1

Table 5.11: DA: 7 activities / time slot 3 seconds

	M	**S**	**C**	**MS**	**MC**	**SC**	**MSC**
B	30.9	30.9	30.9	30.9	30.9	30.9	30.9
A	45.1	66.9	40.3	78.6	56	76	78.9
G	40.3	62.9	35.1	61.7	34.6	66.3	63.7
M	12.6	35.7	19.1	32.9	22.6	36.6	36.6
BA	52.9	73.1	41.4	84.3	60.9	77.4	79.4
BG	40.6	71.1	44.3	64.6	37.1	73.4	67.7
BM	22.9	40	28.9	36.6	30	42.6	42
AG	58.3	83.4	55.1	87.7	61.4	88	89.1
AM	44	64.3	43.1	73.4	56.6	76	76.9
GM	33.1	64	33.7	64.9	33.4	65.7	68
BAG	62.3	84.3	56.3	87.4	62	86.6	88
BAM	56	70	44.9	76.9	58.6	77.4	79.1
BGM	39.1	68.3	43.7	66.9	39.4	67.1	69.7
AGM	52	81.4	53.1	85.7	61.1	88.3	86.6
BAGM	57.7	82.6	56.9	85.4	64.3	87.7	86.9

Table 5.12: CT: 7 activities / time slot 3 seconds

	M	**S**	**C**	**MS**	**MC**	**SC**	**MSC**
B	17.4	17.4	17.4	17.4	17.4	17.4	17.4
A	45.1	61.1	32.9	69.1	50.9	72.3	69.7
G	36.9	52	36	51.7	47.7	60.3	60
M	26.6	38	20.3	36	21.7	34.3	35.1
BA	44.9	71.4	40	72.9	46.3	74.3	73.4
BG	36.9	50.9	49.1	52.9	46.9	61.1	60.9
BM	23.4	36.6	22.3	41.1	26.6	35.1	37.4
AG	40	73.4	47.1	75.1	54.9	80.6	78.9
AM	46.3	63.1	38.3	68.6	49.7	69.7	68
GM	38.6	52.9	37.7	57.1	43.4	60.6	61.1
BAG	46	77.4	51.1	76.6	57.1	79.4	78.9
BAM	43.4	71.4	42	74.6	50.9	74.6	74.6
BGM	35.1	53.1	46.9	56.9	42.6	63.4	63.1
AGM	40.3	73.4	50	75.1	53.1	80.6	79.7
BAGM	46	77.4	53.1	76.6	54.9	79.4	79.4

LEGEND
Sensors: B = barometer / A = accelerometer / G = gyroscope / M = magnetometer
Features: M = mean value / S = standard deviation / C = correlation
Algorithms: KNN = k-nearest neighbors / NB = naive Bayes / DA = discriminant analysis / CT = classification tree

Considering all 3 time slot lengths, it cannot be made the statement, the more features and the more sensors, the better the result. The optimal combination of sensors and features is highly dependent on the algorithm and time slot length. Nevertheless, it can be said that the feature standard deviation is the most significant feature since it appears in every feature combination that yielded highest results (red marked results).

Concerning the sensors, it is obviously visible that in every red marked result the combination of accelerometer and gyroscope is included. Looking at their individual results, it can be stated that the accelerometer is the most significant sensor. Using the accelerometer in combination with the gyroscope brings a significant gain in accuracy, whereas on the other hand the other two sensors seem to be quite useless. Neither the magnetometer nor the barometer can significantly improve the accuracy.

It is understandable that the magnetometer does not really increase the accuracy but it is unexpected for the barometer, because it is the only sensor that provides height changes. Actually, the barometer should significantly improve the accuracy since we test activities such as climbing stairs up/down and walking an inclination up/down, which could easily be distinguished from walking by comparing their barometric pressure values. Probably the movement itself, which is catched by the accelerometer and gyroscope, is more characteristic than the information about a change of the height. Or it might be possible that the feature linear regression is not significant enough for a classification algorithm. Another explanation is that the technical level of the barometer sensor is not yet at the required level.

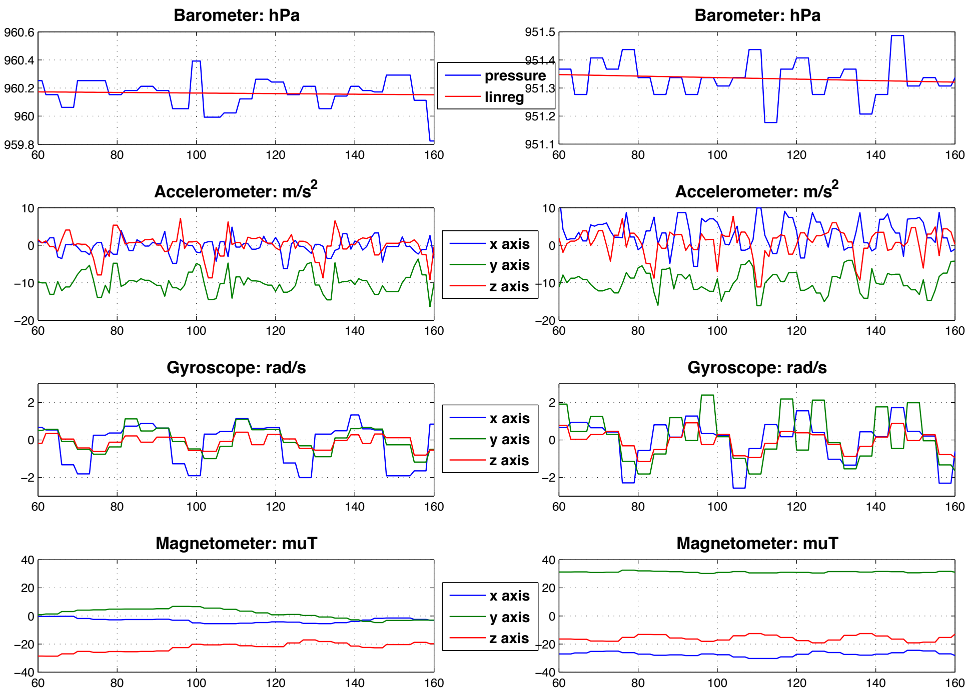


Figure 5.1: Graph of sensor values of activity 'walking' (left column) and activity 'walking an inclination up' (right column) in a 5 second window.

As we can see on Figure 5.1, the slope of the linear regression line in the barometer value is weakly decreasing for the activity 'walking an inclination up' (air pressure decreases in altitude). This value could be used to differentiate between activities performing a height change and activities such as standing, walking and running. However, the problem is that even during standing, walking and running the line of the linear regression can slightly increase or decrease (such as in our left barometer plot) since the air pressure value of the barometer is not reliable, such as visible on the deviation of the barometric sensor value (blue line). The barometer may be pretty useful for activities performing a large difference in altitude in a short time.

Also the magnetometer provides any usable values for the distinction between activities. Although its graphs seem to provide reliable results for recognition, our result tables have shown that the additional use of a magnetometer does not provide any improvement to the accuracy. The reason may be that its values change too slowly.

On the other hand, the values of the accelerometer and gyroscope show characteristic repeating patterns, which indicate a strong sensitivity of the sensors. These values are characteristic enough to discriminate between very similar activities, such as walking and walking an inclination, even without the values of the barometric pressure.

Accuracy of plurality voting

We have seen that the accuracy of single classifiers in our work lies between 68 and 89.1%. Likewise, we have discussed the results of different sensor and feature combinations in individual algorithms. Now, the question comes up, how much the accuracy can be increased, when we use all these classifiers in our plurality voting (PV) algorithm.

In Table 5.13 we see the best achieved accuracy (red marked results in previous tables) for each classifier in a specific time slot length. In tables, which have multiple red marked results, we just take the top left-most result since these results require less computation.

	1 sec	**2 sec**	**3 sec**
KNN	72	79.4	81.7
NB	80.9	88.3	88.9
DA	78.9	87.7	89.1
CT	68	76	80.6
PV	82.3	90.9	90.6
train 2/test 1	-	80.6	-
train 3/test 1	-	-	78.6

Table 5.13: PV results in different time slot lengths

As we can see, the accuracy of each classifier increases with growing time slot length. But this observation applies only for single classifiers. In the PV algorithm the accuracy slightly drops between time slot length 2 and 3. Thus, the statement, the longer the time slot, the better the accuracy, is not valid for the PV. Besides, activity recognition systems need to keep the time slot length as small as possible in order to correctly recognize a movement change, especially in short activities. Since the largest accuracy gain lies between time slot length 1 and 2 seconds, recognition systems should operate in this range.

It remains the question, whether it could be useful to train the algorithms with larger time slots but test the algorithms on short time slots. For this purpose we tested the classifiers that have been trained with a time slot length of 2sec and 3sec, on a test set with 1sec time slot. The results, i.e. 80.6% when trained with 2sec and tested with 1sec and 78.6% when trained with 3sec and tested with 1sec, show that this way is no solution to realize a system with accurate recognition since single classifiers provide better results. Thus, a recognition system needs to be tested with the same time slot length as it was trained with.

Regarding the algorithms, it is clearly visible that NB and DA provide the best accuracy in every time slot length. Between the most accurate and the least accurate algorithm, which is the CT, is a big difference (up to 12%). Therefore, the choice of the right algorithm has great impacts on the mobility recognition accuracy.

In every time slot the PV algorithm provides the most accurate result. However, there is no large growth in accuracy between the PV result and the most accurate individual algorithm (only about 2%). Therefore, in online recognition systems it may be quite useful to use only 1 classifier due to low computation power.

Analysis of confusion matrices

All our calculated accuracy results follow from confusion matrices. A confusion matrix does not only provide information about the accuracy of a classifier, it rather shows us which activities can correctly be classified and which activities are confused with others. Considering the best achieved accuracy for a single classifier in a time slot length compared with the corresponding PV result, we can gain knowledge in which activities exactly the PV makes better predictions.

In time slot length 1 the NB classifier achieved the best accuracy. In Table 5.15 we can see that the PV algorithm only makes better classifications in 2 activities (see green marked cells). The gain in accuracy of PV (4 less misclassifications in walking inclination down and 1 misclassification less in stairs up) is only marginal.

Also marginal is the gain in accuracy of PV in time slot 2 and 3 (Tables 5.16 - 5.19). The only significant improvement of the PV algorithm is in Table 5.17, where walking is misclassified 6 times less. Thus, the use of PV is not profitable in our case since it provides only marginal improvements for high computation resources.

Regarding the misclassifications themselves, it can be seen that walking and inclination up have been classified worst. While walking is mostly confused with inclination up, the activity inclination up itself is mostly confused with inclination down or walking. The fact that not even inclination up is correctly distinguishable from inclination down enforces the ineffectiveness of the barometer. Regarding the activity stairs up, which has a change in altitude too, we see that its recognition is much better in longer time slot lengths. The reason for its high recognition (in contrast to inclination up/down) might be the characteristic body movement of climbing stairs.

While the magnetometer is useless for activity recognition, the barometer would be very helpful to distinguish between activities with altitude change. Muralidharan et al. show in [29] that the barometer can successfully be used for floor changes when pressure information is saved as a fingerprint. However, in our case we need a barometer that reliably and quickly provides pressure information without deviation.

LEGEND
a = standing / b = walking / c = running
d = stairs up / e = stairs down / f = inclination up / g = inclination down

Table 5.14: NB confusion matrix (time slot = 1s)

	a	b	c	d	e	f	g
a	50	0	0	0	0	0	0
b	0	31	0	2	7	9	1
c	0	0	50	0	0	0	0
d	0	11	0	36	2	1	0
e	0	4	0	0	46	0	0
f	0	4	0	1	0	32	13
g	0	6	1	0	0	5	38

accuracy = 80.9%

Table 5.15: PV confusion matrix (time slot = 1s)

	a	b	c	d	e	f	g
a	50	0	0	0	0	0	0
b	0	31	0	3	8	6	2
c	0	0	50	0	0	0	0
d	0	8	0	37	4	1	0
e	0	2	0	1	46	0	1
f	0	3	1	3	0	32	11
g	0	3	0	0	1	4	42

accuracy = 82.3%

Table 5.16: NB confusion matrix (time slot = 2s)

	a	b	c	d	e	f	g
a	50	0	0	0	0	0	0
b	0	39	0	0	6	3	2
c	0	0	50	0	0	0	0
d	0	0	0	48	2	0	0
e	0	3	0	0	47	0	0
f	0	2	0	1	0	34	13
g	0	6	0	0	0	3	41

accuracy = 88.3%

Table 5.17: PV confusion matrix (time slot = 2s)

	a	b	c	d	e	f	g
a	50	0	0	0	0	0	0
b	0	45	0	0	1	3	1
c	0	0	50	0	0	0	0
d	0	0	0	48	2	0	0
e	0	1	0	0	49	0	0
f	0	5	1	3	0	34	7
g	0	3	0	0	2	3	42

accuracy = 90.9%

Table 5.18: DA confusion matrix (time slot = 3s)

	a	b	c	d	e	f	g
a	50	0	0	0	0	0	0
b	0	45	0	0	1	2	2
c	0	0	49	0	1	0	0
d	0	0	0	49	1	0	0
e	0	5	0	0	45	0	0
f	0	5	0	6	0	32	7
g	0	7	0	0	0	1	42

accuracy = 89.1%

Table 5.19: PV confusion matrix (time slot = 3s)

	a	b	c	d	e	f	g
a	50	0	0	0	0	0	0
b	0	45	0	0	1	2	2
c	0	0	50	0	0	0	0
d	0	0	0	49	1	0	0
e	0	4	0	0	46	0	0
f	0	6	0	2	0	34	8
g	0	6	0	0	0	1	43

accuracy = 90.6%

5.1.2 Distinction between 2 Activities

As mentioned, we only implement the activities ’standing’ and ’walking’ into our indoor localization system. Therefore, we have to set up a mobility recognition system that can differ between only standing and walking. Since this activity pattern has only 2 activities, the recognition accuracy should be much higher than in the pattern with 7 activities.

As we need a recognition system that can predict the activity every second (due localization purposes), we have trained the algorithms accordingly (only with time slot length 1sec).

We can see on Tables 5.20 - 5.23 that many feature and sensor combinations achieve the same best result (red marked results). Due to computation resources we take the top left-most result (orange marked results) for the PV algorithm, that of course achieves an accuracy of 100% as well. Even if the PV cannot bring any additional accuracy gain, we will use it though. The reason for that is because these algorithms achieved an accuracy of 100% (except the DA) on our test set. Testing these algorithms in a completely other environment may provide less accurate results.

Table 5.20: KNN: 2 activities / time slot 1 second

	M	S	C	MS	MC	SC	MSC
B	55	55	55	55	55	55	55
A	63	100	59	96	79	100	100
G	91	99	70	98	81	93	94
M	51	95	59	51	50	83	52
BA	63	100	59	96	79	100	100
BG	91	99	70	98	81	93	94
BM	51	95	59	51	50	83	52
AG	68	100	72	97	79	100	99
AM	48	98	61	60	50	97	61
GM	51	97	77	52	51	97	53
BAG	68	100	72	97	79	100	99
BAM	48	98	61	60	50	97	61
BGM	51	97	77	52	51	97	53
AGM	48	98	80	61	50	100	61
BAGM	48	98	80	61	50	100	61

Table 5.21: NB: 2 activities / time slot 1 second

	M	S	C	MS	MC	SC	MSC
B	53	53	53	53	53	53	53
A	70	100	61	98	77	100	96
G	99	100	82	100	100	100	100
M	57	94	58	93	60	92	93
BA	68	100	64	98	74	100	96
BG	99	100	82	100	100	100	100
BM	55	93	63	93	67	92	92
AG	95	100	81	100	93	100	100
AM	70	100	62	99	77	100	99
GM	99	100	79	100	99	100	100
BAG	95	100	78	100	93	100	100
BAM	66	100	64	99	76	100	99
BGM	98	100	79	100	99	100	100
AGM	95	100	81	100	93	100	100
BAGM	95	100	79	100	93	100	100

Table 5.22: DA: 2 activities / time slot 1 second

	M	S	C	MS	MC	SC	MSC
B	52	52	52	52	52	52	52
A	80	92	57	95	72	99	90
G	80	98	81	94	82	99	96
M	54	89	55	95	63	92	94
BA	78	93	57	97	73	98	90
BG	78	97	77	93	83	98	94
BM	58	90	57	92	59	91	91
AG	83	93	86	96	80	98	92
AM	77	97	54	91	71	99	95
GM	78	98	79	95	83	95	94
BAG	79	88	84	91	81	95	94
BAM	77	97	58	94	72	99	95
BGM	79	99	79	93	81	94	92
AGM	82	97	85	78	68	95	90
BAGM	80	99	82	76	72	95	88

Table 5.23: CT: 2 activities / time slot 1 second

	M	S	C	MS	MC	SC	MSC
B	55	55	55	55	55	55	55
A	72	100	53	100	77	99	99
G	89	99	64	99	83	99	99
M	54	96	68	96	73	91	91
BA	72	100	63	100	77	100	100
BG	89	99	65	99	65	99	99
BM	56	96	51	96	58	96	96
AG	90	99	69	99	83	99	99
AM	72	100	54	100	77	99	99
GM	88	99	77	99	83	99	99
BAG	90	99	65	99	65	99	99
BAM	72	100	63	100	77	100	100
BGM	88	99	65	99	65	99	99
AGM	90	99	69	99	83	99	99
BAGM	90	99	65	99	65	99	99

5.2 Results and Analysis for Indoor Localization

Our main goal in this thesis is to show whether an indoor localization system can be improved when it is combined with a mobility recognition system. For this purpose we implement our recognition system that can distinguish between the activities standing and walking with an accuracy of 100%.

Like in our activity recognition system, the accuracy of indoor positioning is dependent on several parameters. In order to investigate the influence of parameters, which can affect the accuracy, we take following points into consideration:

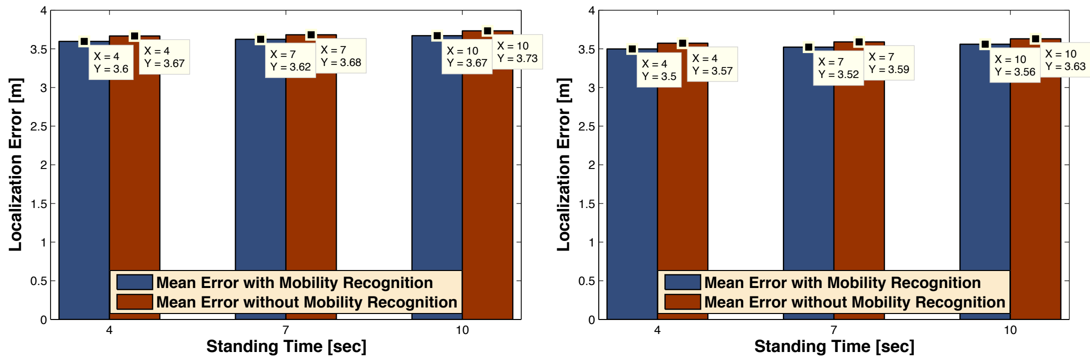
- We performed our localization tests in 2 environments. The first environment includes 4 APs and the second environment 6 APs. The second environment (24x16m) covers a larger localization area than the first environment (13x16m).
- In the first environment we differentiate the results between 3 and 4 APs and in the second environment between 3, 4, 5 and 6 APs.
- Each test execution was performed in single as well as in double speed. The walking velocity of single speed is 0.5m/sec and that of double speed 1m/sec.
- As we attempt to locate a mobile user while walking and standing, the duration of standing may affect our results. Therefore, we consider the standing times 4, 7 and 10 seconds.
- Finally, in order to compare localization results including mobility recognition with localization results without implementing mobility recognition, we calculate the localization accuracy once with and once without mobility recognition.

We present our results in graphs showing the mean localization error of a specific execution. For a better overview in order to compare the results of the first with results of the second environment, we calculate the cumulative distribution function of the mean localization errors. Finally, we visualize these results on maps showing traces of different execution modes.

5.2.1 First Environment

Figure 5.2: Mean Error in Single Speed.

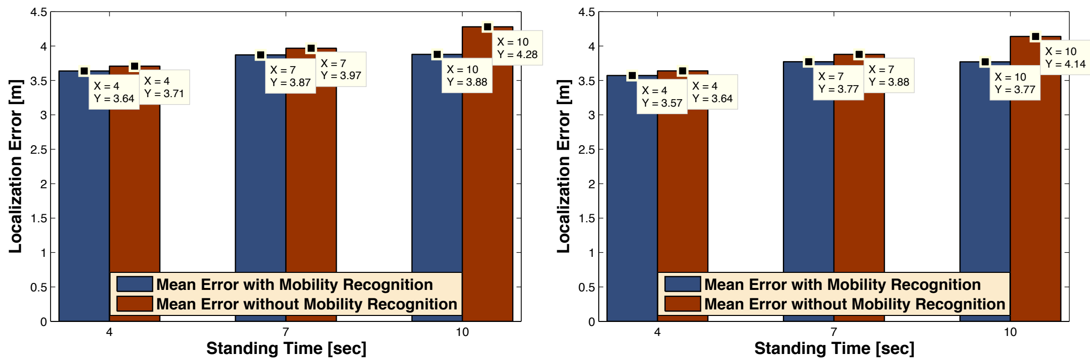
(a) 3 access points **(b)** 4 access points



It is clearly visible in Figure 5.2 that in every standing time the localization error is smaller when using mobility recognition. The distance between the results of with and without mobility recognition in each standing time varies in a very small range (0.06-0.07m). Comparing the results between the use of 3 and 4 APs, we see that using 4 APs constantly delivers slightly lower errors (about 10cm).

Figure 5.3: Mean Error in Double Speed.

(a) 3 access points **(b)** 4 access points

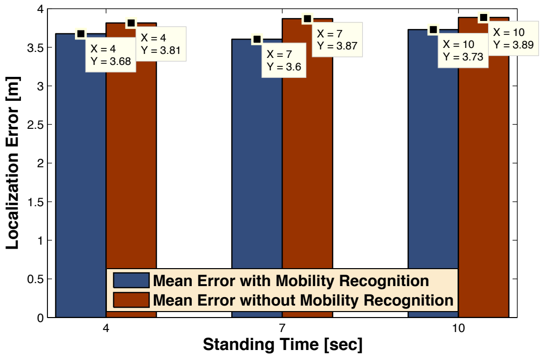


For the double speed we can see on Figure 5.3 quite similar effects. Here again, the localization error is smaller when combined with mobility recognition, and using 4 APs delivers better results than only 3 APs. Comparing single and double speed, it is apparent that in single speed the localization error increases very slightly from short standing times to longer ones, whereas in double speed this increase is much bigger. In other words, in shorter standing times the difference in localization error between single and double speed is smaller than in longer standing times.

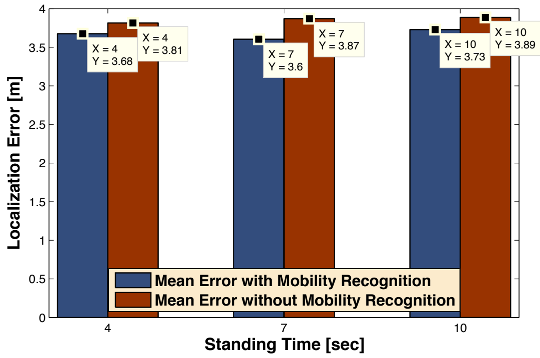
5.2.2 Second Environment

Figure 5.4: Mean Error in Single Speed.

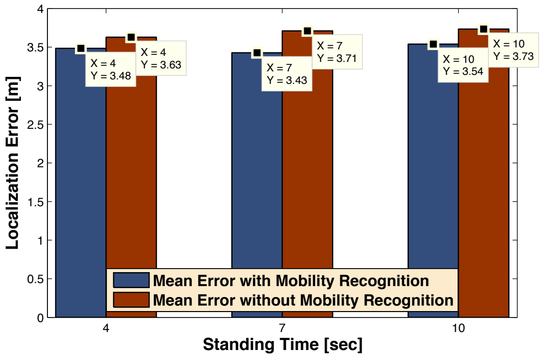
(a) 3 access points



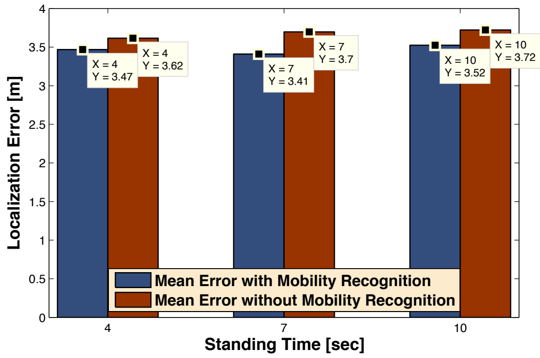
(b) 4 access points



(c) 5 access points



(d) 6 access points



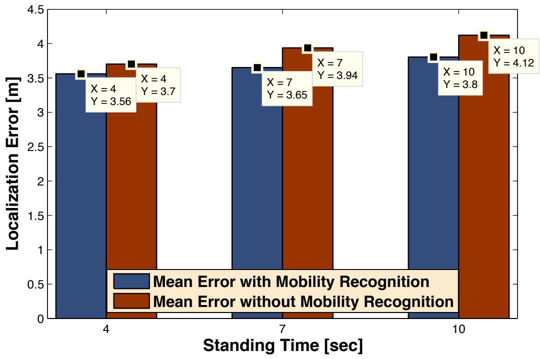
The results of the second environment in single speed show again that mobility recognition improves the indoor localization accuracy, as we can see on Figure 5.4. Concerning the number of used APs, there is a strong improvement when using 5 APs as compared to using only 4. Between 5 and 6 APs there is only a marginal improvement visible and between 3 and 4 APs even no difference, which is probably caused by the inaccessibility of some AP signals due to large distances in that localization area. Nevertheless, we can make the statement that the more access points are used, the better is the localization.

Unlike the first environment, the mean error by using localization with mobility recognition does not increase with longer standing times. The mean error drops by standing time 7 and then increases by standing time 10.

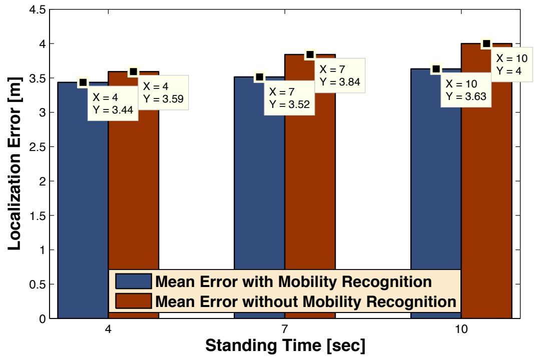
Actually we expected better localization results in longer standing times since longer standing times can gather more RSS data and thus provide a more accurate position. However, the results show the contrary.

Figure 5.5: Mean Error in Double Speed.

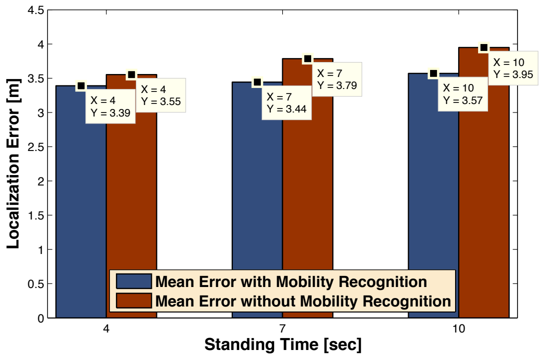
(a) 3 access points



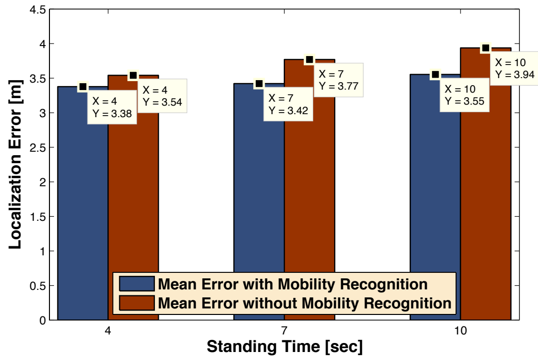
(b) 4 access points



(c) 5 access points



(d) 6 access points



Also in double speed we have the unexplainable phenomena of increasing mean errors with longer standing times, as we can see in Figure 5.5. And again, the use of all APs provides the best localization in both modes (with and without mobility recognition).

In single speed we assumed that the zero difference between using 3 and 4 APs happened coincidentally. This assumption is confirmed when we see that there is a significant improvement between using 3 and 4 APs in double speed.

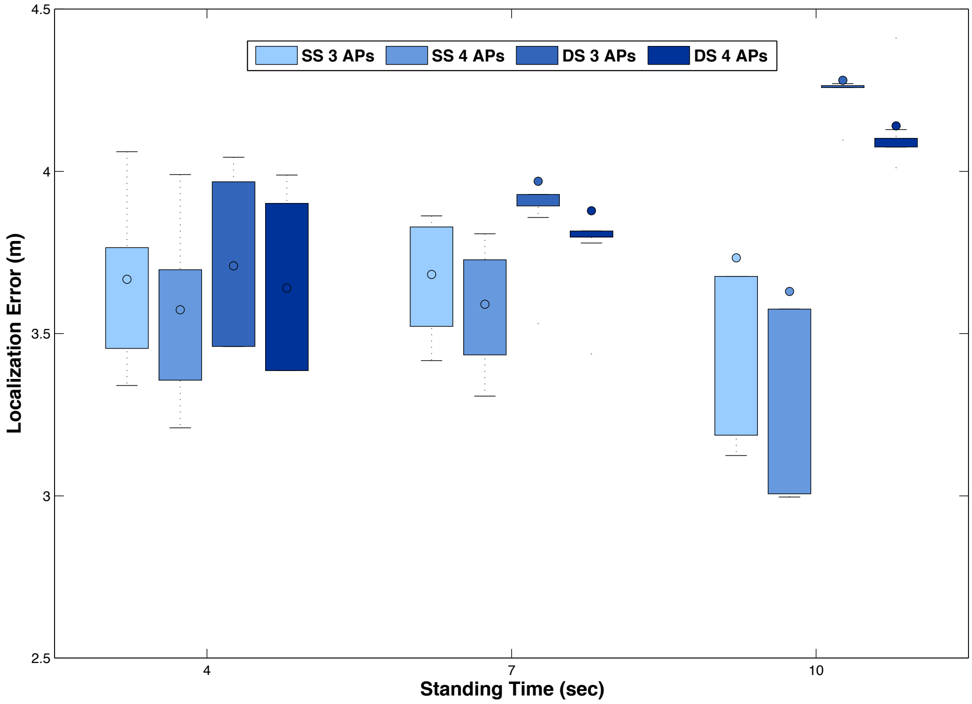
Unlike the first environment, the localization error in double speed is not generally higher than in single speed, rather it is lower in shorter standing times. However, these differences are too small to infer general statements of it.

Comparing with the first environment, the results of the second environment generally have a larger difference between using and not using mobility recognition. Therefore, the use of mobility recognition in indoor localization may be even more important in large areas.

In Figures 5.6 and 5.7 we can see the confidence intervals (CI) of all indoor localization results. Note that the little circle denotes the mean localization error.

Figure 5.6: CI of first environment in single speed (SS) and double speed (DS).

(a) Indoor Localization without Mobility Recognition



(b) Indoor Localization with Mobility Recognition

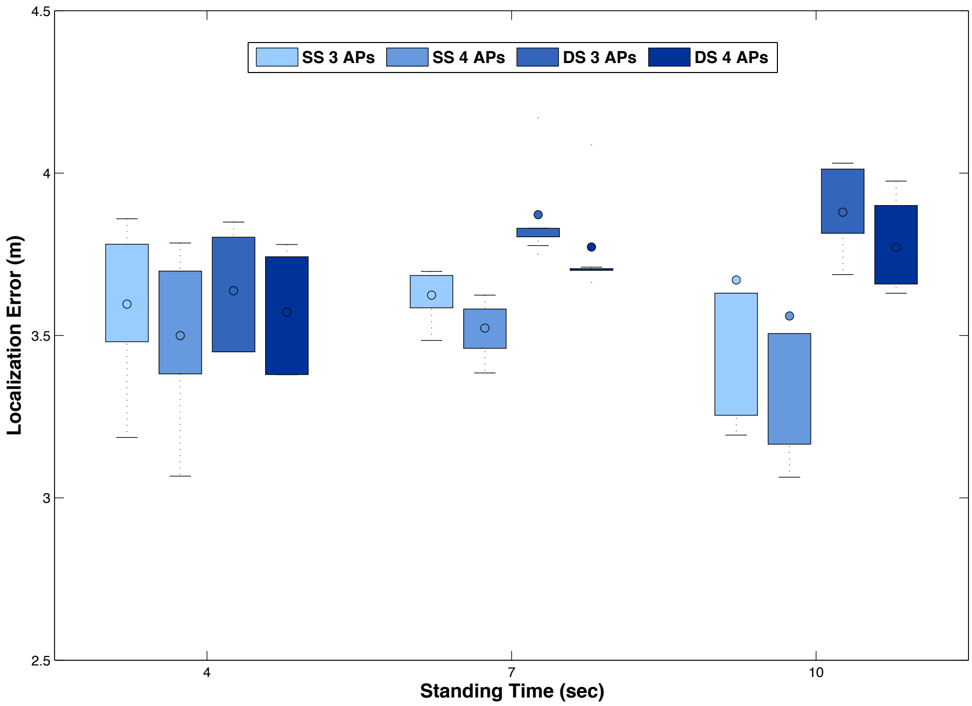
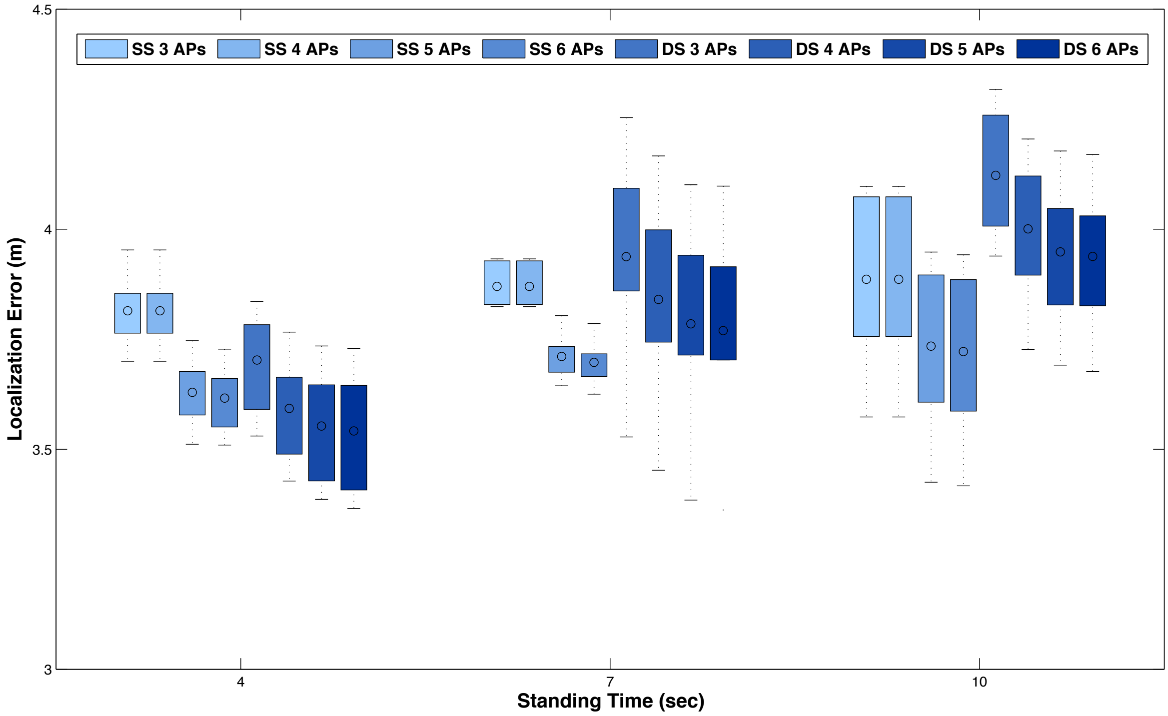
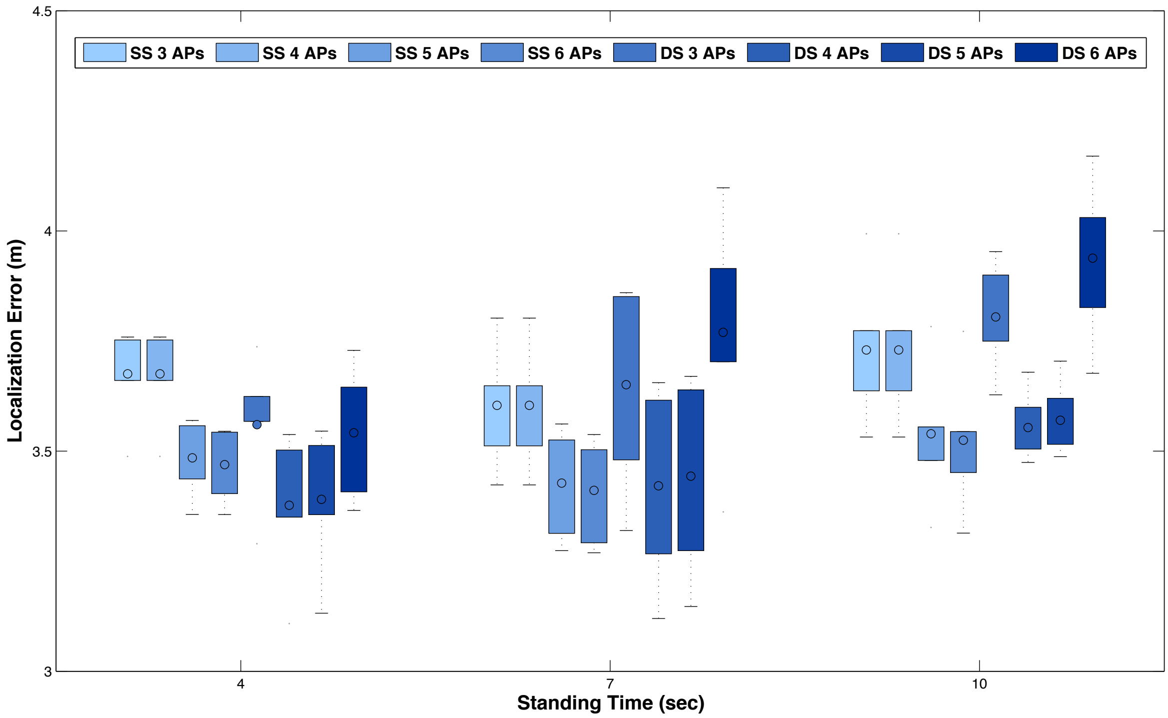


Figure 5.7: CI of second environment in single speed (SS) and double speed (DS).

(a) Indoor Localization without Mobility Recognition



(b) Indoor Localization with Mobility Recognition



For comparisons between the first and second environment we can compute the empirical cumulative distribution function of the mean localization errors. As we see in Figure 5.8, in both environments the localization error is smaller when using mobility recognition. It is also visible that the distance between using and not using mobility recognition (MobRec) is larger in the second environment (distance between green and light blue line) than the distance in the first environment (distance between dark blue and red line). This significant distance for the second environment proves the benefit of implementing activity recognition systems for indoor localization, particularly in larger environments.

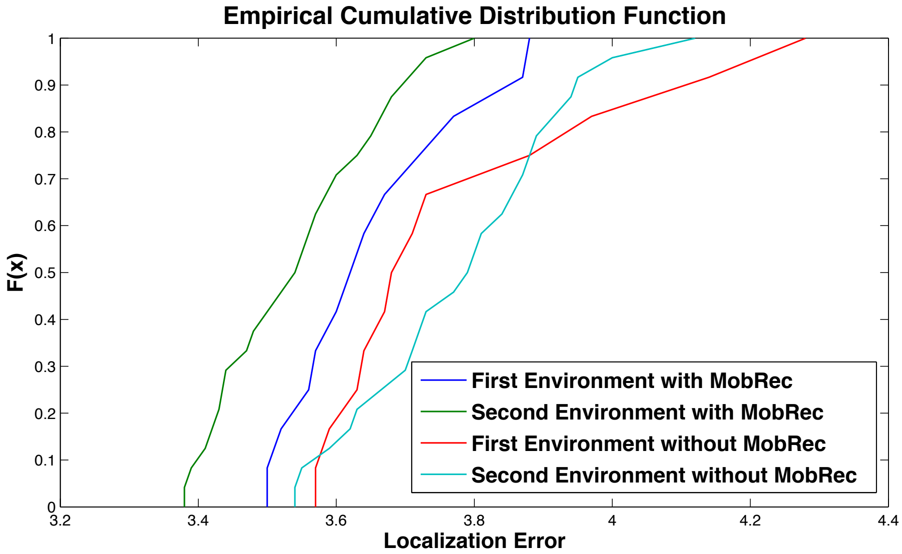


Figure 5.8: Empirical cdf of localization error.

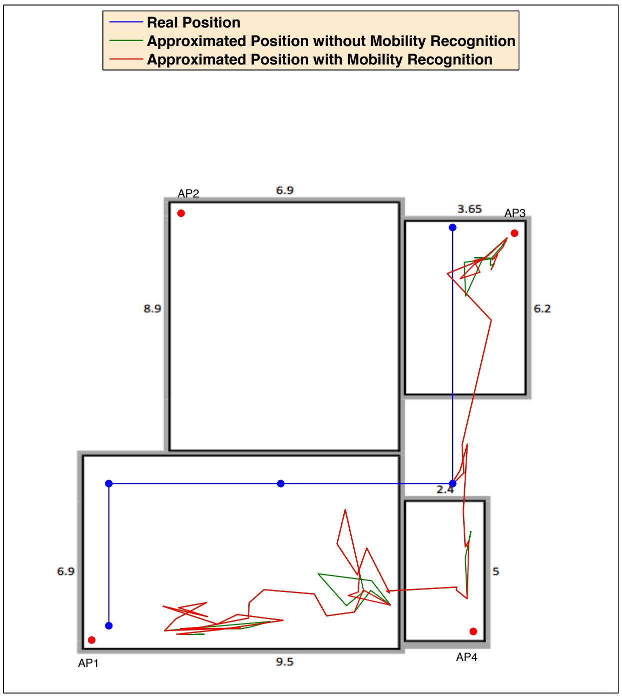
Figure 5.9 visualizes a map containing the traces of the first environment using 4 APs. It is apparent that longer standing times provide approximated positions closer to AP2, which, however, do not necessarily lead to a lower localization error. The localization error only indicates the distance between real and approximated position; hence, even the approximated trace of standing time 10 seems to be closer to the real position than the trace of standing time 4, it is not the case here. Note that the blue points represent standing positions.

The maps of the second environment, see Figure 5.10, show that in longer standing times the approximated positions are more centralized than in shorter standing times. Here again, the centralized positions do not necessarily provide smaller errors since these positions can be centralized on a large distance from the real position. A modified algorithm that is adapted to this environment regarding the signal loss in different positions, might provide localization errors not increasing with longer standing times, but decreasing with longer standing times.

Anyway, the improvement of using mobility recognition in indoor localization is not visible on these maps since the average improvement of the entire trace is less than 0.5m that is hard to perceive, especially as the approximated trace is not a straight line, but a scattered line.

Figure 5.9: Map of first environment in single speed with 4 access points.

(a) Standing Time 4



(b) Standing Time 10

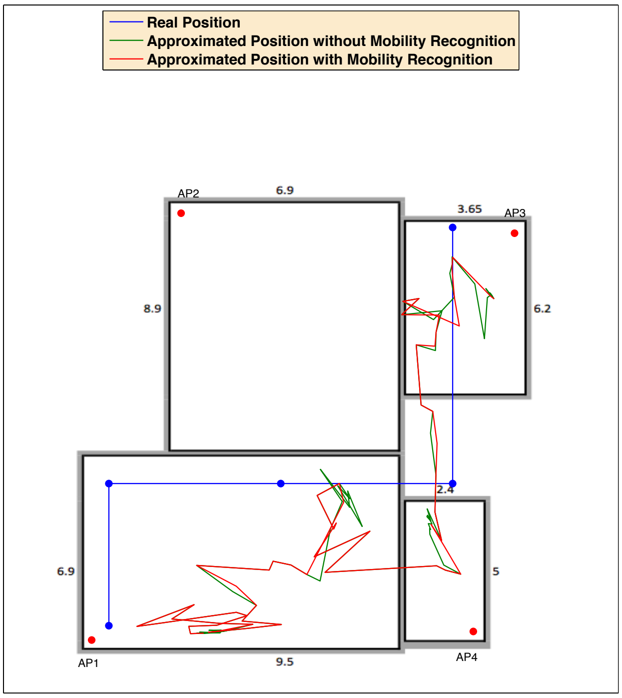
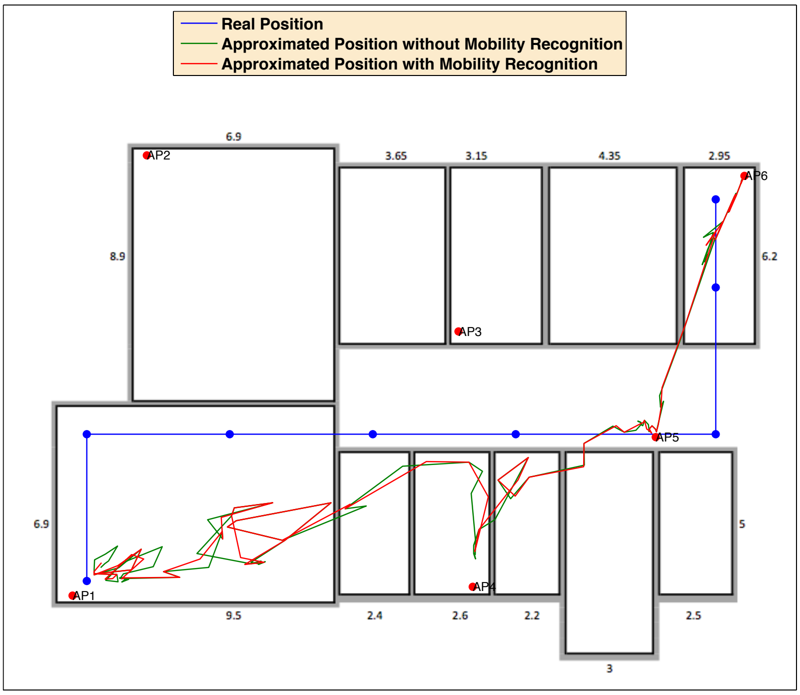
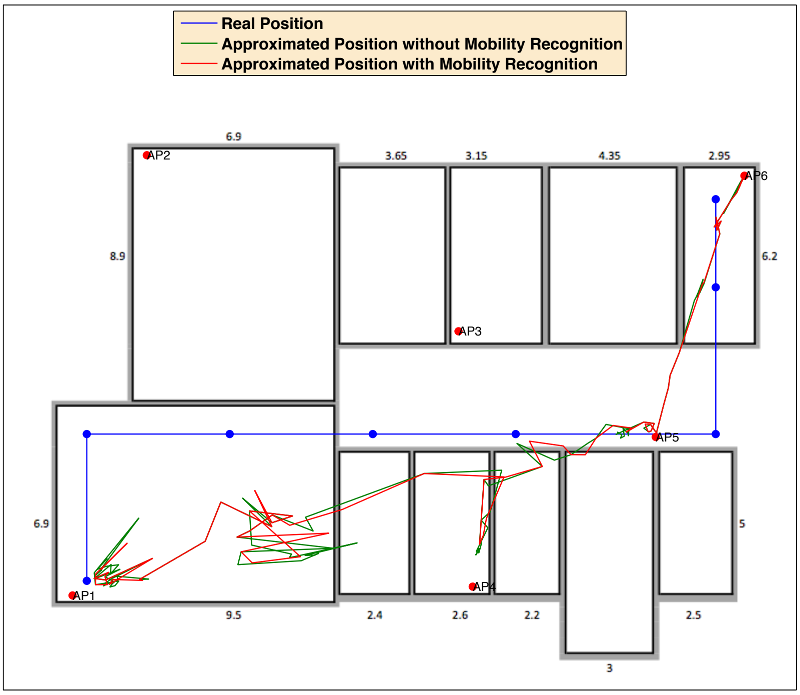


Figure 5.10: Map of second environment in single speed with 6 access points.

(a) Standing Time 7



(b) Standing Time 10



Chapter 6

Conclusions and Future Work

6.1 Conclusions

It is clearly visible that an indoor localization system can be improved when it is combined with mobility recognition. In the first as well as in the second environment, in single speed as well as in double speed, in every standing time and every number of used access points, there is an explicit difference between localization using mobility recognition and localization without using mobility recognition. The differences vary between 0.06-0.4m, which may be crucial in indoor environments.

Comparing the number of used access points, the first environment provides best results with all 4 access points instead of using only the 3 strongest RSS signals. Likewise, the second environment provides best results with all 6 access points. Hence, there is no need to implement mechanisms filtering out weak RSS signals since more access point signals provide more accurate localization results.

The observation, which does not meet our expectations is the rising mean error with longer standing times. Actually, it should be the reverse way, because increasing the number of samples, which is achieved by longer standing times, delivers more data of the same position, and taking the mean value of more RSS data should minimize the distraction of gathered signals, and therefore, longer standing times should lead to smaller errors. However, this is not the fact and this effect is unexplainable for us.

Concerning single and double speed, no general statement can be made. In the first environment the double speed localization provides more error than walking in single speed, but in the second environment it is vice versa. However, most people walk with a velocity between 0.5m/s (single speed) and 1m/s (double speed).

Between the small (first) and large (second) environment there are obvious differences. Although in our small environment the localization with mobility recognition provides better results than without mobility recognition, the difference between these two modes is small. This difference grows in our second environment since large areas have more signal attenuation and therefore, gathering for more RSS samples provide better results. Thus, using mobility recognition systems for indoor localization can be even more important for large environments.

6.2 Future Work

Since we developed an activity recognition system that can distinguish between several most observed activities in indoor environments (more or less accurately), future works can make use of this system in order to verify the improvement of an indoor localization system, which is not only adapting the RSS time slot, but also can provide position information on different floors.

It should be noted that the recognition of walking an inclination is probably not accurate enough for a successful implementation into indoor localization systems. Therefore, an accurate recognition for this activity remains a future work.

In this thesis we defined two options for the adjustment of the RSS time slot: 1. enlarge window as long as standing and 2. set the window to 1 second if walking. Thus, walking in single and double speed, both were handled with a 1 second time slot. In a future work we could define more options inside the category of walking in order to enable a denser adjustment between velocity and window length. However, this system will need a recognition application that not only identifies the current performed activity, but also delivers information about the velocity. This is possible with step detection mechanism, which are mainly used in dead reckoning systems.

Bibliography

[1] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *J. Syst. Softw.*, vol. 84, no. 11, pp. 1928–1946, Nov. 2011.

[2] R. Mautz, "Indoor positioning technologies," Ph.D. dissertation, ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, Institute of Geodesy and Photogrammetry, 2012.

[3] O. Lara and M. Labrador, "A survey on human activity recognition using wearable sensors," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 3, pp. 1192–1209, Third 2013.

[4] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 48–53, Jan 2010.

[5] A. Wilde, "An Overview of Human Activity Detection Technologies for Pervasive Systems," Seminar paper, Department of Informatics, University of Fribourg, Switzerland, 2010.

[6] A. Rasekh, C. Chen, and Y. Lu, "Human activity recognition using smartphone," *CoRR*, vol. abs/1401.8212, 2014.

[7] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Pattern Recognition and Image Analysis*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6669, pp. 289–296.

[8] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, ser. IAAI'05. AAAI Press, 2005, pp. 1541–1546.

[9] D. Maguire and R. Frisby, "Comparison of feature classification algorithm for activity recognition based on accelerometer and heart rate data," 9th. IT and T Conference, Dublin Institute of Technology, Dublin, Ireland, 2009.

[10] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6406, pp. 548–562.

[11] A. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis," in *Future Information Technology (FutureTech), 2010 5th International Conference on*, May 2010, pp. 1–6.

[12] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, ser. IJCAI'11. AAAI Press, 2011, pp. 2545–2550.

[13] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Pervasive Computing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 3968, pp. 1–16.

[14] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3001, pp. 1–17.

[15] M. Abdullah, A. Negara, M. Sayeed, D. Choi, and K. Muthu, "Classification algorithms in human activity recognition using smartphones," vol. 6, no. 8, pp. 372 – 379, 2012.

[16] S. Milker, "Bewegungserkennung mit Smartphones mittels deren Sensoren," Bachelor thesis, Institute for Web Science and Technologies, Universitaet Koblenz-Landau, Germany, 2012.

[17] A. M. Khan, "Recognizing physical activities using the axivity device," in *The Fifth International Conference on eHealth, Telemedicine, and Social Medicine*, 2013.

[18] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, April 2006, pp. 4 pp.–116.

[19] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[20] "Android Developers sensors overview," http://developer.android.com/guide/topics/sensors/sensors_overview.html, accessed: 2014-10-15.

[21] J. Torres-Solis, T. H. Falk, and T. Chau, "A review of indoor localization technologies: towards navigational assistance for topographical disorientation," in *Ambient Intelligence*. Bloorview Research Institute and University of Toronto, 2010, pp. 51–84.

[22] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and systems," *Journal of Computer Networks and Communications*, vol. 2013, 2013.

[23] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Trans. Sys. Man Cyber Part C*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[24] Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *Communications Surveys Tutorials, IEEE*, vol. 11, no. 1, pp. 13–32, First 2009.

[25] D. Goehl and D. Sachs, “Motion sensors gaining inertia with popular consumer electronics.” USA: InvenSense Inc., Coronado Drive, Santa Clara.

[26] F. Ichikawa, J. Chipchase, and R. Grignani, “Where’s the phone? a study of mobile phone location in public spaces,” in *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*, Nov 2005, pp. 1–8.

[27] M. Schirmer and H. Hoepfner, “Smartphone hardware sensors,” https://www.uni-weimar.de/medien/wiki/images/Zeitmaschinen-smartphonesensors.pdf, accessed: 2014-27-04.

[28] U. Shala and A. Rodriguez, “Indoor Positioning using Sensor-fusion in Android Devices,” Master graduation work, School of Health and Society, Department Computer Science, Kristianstad University, Sweden, 2011.

[29] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal, “Barometric phone sensors: More hype than hope!” in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile ’14. New York, NY, USA: ACM, 2014, pp. 12:1–12:6.

[30] L. Chen and C. Nugent, “Ontology based activity recognition in intelligent pervasive environments,” *International Journal of Web Information Systems*, vol. 5, no. 4, pp. 410–430, 2009.

[31] X. Lin, S. Yacoub, J. Burns, and S. Simske, “Performance analysis of pattern classifier combination by plurality voting,” *Pattern Recognition Letters*, vol. 24, pp. 1959–1969, 2002.

[32] S. Hirsbrunner, “Indoor Localisation In Wi-Fi-Networks Using An Improved Centroid Approach,” Bachelor thesis, Research group Communication and Distributed Systems, Universitaet Bern, Switzerland, 2014.

[33] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, “Weighted centroid localization in zigbee-based sensor networks,” in *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, Oct 2007, pp. 1–6.