Multi-sensor pedestrian position and attention tracking system

Master thesis

Stefan Ladstätter

Institute for Broadband Communication
Graz University of Technology

Institute for Information and Communication Technologies
Joanneum Research

Supervisor: Ao. Univ.-Prof. Dr. Erich Leitgeb
Co-Supervisor: DI (FH) Patrick Luley

Graz, October 2010

Abstract

This master thesis describes the theoretical fundamentals, necessary design steps and implementation of an outdoor pedestrian location and attention tracking system for consumer studies. The approach introduced combines sensor data from multiple sources in a hybrid Particle/Kalman filter model. This system makes it possible to specify the position of test participants with 5 meter accuracy independent of time and walking distance in the evaluation area. Beginning with an overview of already existing pedestrian positioning methods, the thesis describes the theoretical geodetic and mathematical fundamentals necessary for realizing the proposed localization system. After that, the used sensor devices and their measurement principles are introduced. The raw sensor readings are saved in a database specifically designed for geodetic measurements. The information stored is post-processed in iterative enhancement stages. The result of this data optimization is then visualized as a 3D model. These software components are described in the middle part of the thesis. In order to assess the performance of the system, two accuracy tests were conducted. The thesis concludes with the results of these evaluations and finally gives an outlook on further work.

Kurzfassung

In dieser Masterarbeit werden die theoretischen Grundlagen, die notwendigen Entwurfsschritte und die Realisierung eines Systems zur Aufzeichnung von Position und Aufmerksamkeit eines Fußgängers für Konsumentenstudien beschrieben. Der vorgestellte Ansatz kombiniert Sensordaten von mehreren Quellen in einem Partikel-/Kalman-Filter Modell. Dieses System ermöglicht es, die Position von Studienteilnehmern im Testgebiet unabhängig von Distanz und Zeit mit einer Genauigkeit von 5 Metern anzugeben. Beginnend mit einem Überblick über schon bestehende Fußgängerpositionierungsmethoden, werden die theoretischen geodätischen und mathematischen Grundlagen, die notwendig für die Realisierung des vorgeschlagenen Systems sind, beschrieben. Daran anschließend werden die verwendeten Aufzeichnungsgeräte und deren Messprinzipien vorgestellt. Die Sensorrohdaten werden in einer Datenbank gespeichert, die eigens für die Sicherung geodätischer Messungen entworfen worden ist. Die gespeicherte Information wird in mehreren aufeinander aufbauenden Verbesserungsschritten verarbeitet. Das Resultat dieser Datenoptimierung wird dann in einem 3D-Modell visualisiert. Diese Softwarekomponenten werden im Mittelteil der Arbeit beschrieben. Um die Leistungsfähigkeit des Systems bewerten zu können, wurden zwei Genauigkeitsanalysen durchgeführt. Abschließend werden die Resultate dieser Tests beleuchtet und schlussendlich die nächsten notwendigen Erweiterungsschritte dargestellt.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am ……. ……..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

……. ……..
date (signature)

Danksagung

Diese Masterarbeit wurde im Jahr 2010 am Institut für Breitbandkommunikation an der Technischen Universität Graz in Kooperation mit dem Institut für Informations- und Kommunikationstechnologien am Joanneum Research durchgeführt.

Zuerst möchte ich mich bei Dr. Erich Leitgeb für die ausgezeichnete Betreuung der Arbeit bedanken. Sein fachlicher Rat stand mir im Zuge der Durchführung dieses Projekts praktisch jederzeit zur Verfügung.

Gleichermaßen möchte ich meinen Kollegen Patrick Luley und Alexander Almer sowie Dr. Mathias Schardt besonderen Dank für die Unterstützung dieses Projekts aussprechen. Durch die flexible Zeitgestaltung und der Bereitstellung zahlreicher Ressourcen ermöglichte mir das Joanneum Research erst die Realisierung dieser Arbeit in Kooperation mit der TU Graz.

Ich hätte wohl nie mein Studium beginnen können, hätten mich meine Eltern sowie meine Geschwister: Elisabeth, Ingrid, Robert und Romana nicht immer tatkräftig in allen Belangen unterstützt. Vielen, vielen Dank, euch allen dafür!

Abschließend möchte ich jene Person besonders hervorheben, der ich diese Arbeit widme: Meine Lebensgefährtin und zukünftige Frau, Anna. Ohne Dich hätt' ich's nicht geschafft! Danke!

Graz, im Oktober 2010 Stefan Ladstätter

Contents

List of Figures

List of Tables

1 Introduction

Analysis of human attention is a crucial issue to understand the user's actual needs as it is investigated by commercial applications for market studies. In order to decrease processing time, an accurate position estimate of the test participants is necessary. Within this project, a tracking platform for pedestrians utilizing a number of wearable sensors is introduced.

Since the "selective availability" constraint for the United States' global positioning system named NAVSTAR-GPS has been removed in 2000 [51], its usage increased heavily [1]. At the time of writing, other global navigation satellite systems (GNSS) are currently under (re-)construction (Russia: GLONASS [43], Europe: GALILEO [15], China: COMPASS [57]) which will enable civilian usage too.

However, the accuracy of satellite based positioning methods is strongly dependent on clear viewing conditions to the sky. This is often not the case in urban canyons, where high buildings either prevent signal reception or induce multi-path wave propagation effects. Moving indoors naturally leads to a total loss of signal.

As a result, within this project, other sensor sources like received signal strength multilateration via available ground-based WiFi networks, digital maps and motion tracking are added to the location estimation provided by the GNSS through a hybrid particle/Kalman filter model. By fusing the data and incorporating its current reliability, the drawbacks of using a single sensor source can be compensated. The obtained information can be divided into data which is a direct result of user behavior and measurements of environmental indicators. While the first class of data only is generally recorded on-the-fly, the second requires a-priori information.

Since visual impulses are far prevalent when advertising products, behavior analysis heavily relies on tracking eye movements of test users. Due to the vast amount of aggregated image data already after short capturing intervals, identifying key features in video streams through an object recognition chain is not feasible.

To solve this problem, user parameters like current position and head orientation are correlated with pre-recorded points of interest (POIs), such as the coordinates of shopping windows, house facades or posters. As a consequence, the system can return a spatial estimate of the user's current field of vision at a certain time, which effectively reduces the number of possible objects in the current eye tracker frame. If the test person's field of view contains a POI and additional parameters (intersection angle, speed, distance to object) are met, the current time code is classified as a point of attention (POA).

In order to prevent the test participants from deviating from their normal behavior al

patterns, it is necessary that every piece of measurement equipment is placed in a non-intrusive way. The rise of microelectromechanical systems (MEMS) made it possible to miniaturize inertial measurement units. The devices used in this project are roughly comparable with a matchbox in size and weigh below 100 grams. Consequently, they can be carried easily in trouser or jacket pockets.

The high amount of recorded data requires a clear and intuitive representation. Therefore, a visualization showing the current user location, the field of vision and all POIs in three dimensional space has been realized. The visual representation is synchronized to the scene camera of the eye tracker. As a consequence, a direct comparison between estimated user gaze and actual field of view is available.

In the following section, an overview of existing pedestrian positioning approaches is given. In Chapter 2, the necessary geodetic and mathematical fundamentals for this project are outlined. Here, the theoretical background for the employed filter systems are illustrated too. Chapter 3 describes the sensor devices used for location and attention tracking. In Chapter 4, the implementation of the various software components is introduced. Following that, measurement results are shown in Chapter 5. Finally, the thesis concludes and gives information about the next necessary steps.

1.1 Related work

Various studies for pedestrian tracking incorporating multi-sensor measurements were already conducted. The approaches include both academic and commercial research. Examples for the latter are the Vectronix DRC [52] and the Honeywell DRM [24]. Prototypes, which were developed as partnership projects at Graz University of Technology are the GyroDRM [33], the mobileNAV [36] and SARHA [53].

The systems were gradually improved by both optimizing the post processing steps (map matching, neural networks, particle filters) and by incorporating more capable acquisition hardware. [17], [6] for example, mounted gyroscopes directly at the shoe of the test subjects and applied double integration on the measured acceleration values. Normally, the slightest drift in those values would cause such an approach to fail already after short data capture periods due to accumulation of error. However, assuming a normal walking pattern, the velocity of the test user's foot is zero at the stance phase. By resetting the potential falsely integrated error, a reliable estimate of the current speed is possible. This method is commonly known as zero velocity update (ZUPT). The main drawback of this approach is a high requirement to the angular resolution of the gyroscope (at least 500 degrees/sec).

Many approaches [33], [36], for pedestrian localization use a model proposed from [35] to infer stride length by a relation between the test subject's body characteristics and his or her current step frequency. Step detection on the other hand was done by means of autocorrelation and thresholding.
[5] no the other hand, employed a step length determination method proposed by [34]. Here, a feature set in a step detection pattern was isolated and used to train a feed forward neu-

ral network. The errors in path length estimated stayed below 5.2 percent of distance traveled.

[55] and [30] used a-priori known information of the target environment to employ map matching for an indoor application. This was done by using a particle filter system which corrected for noisy heading measurements in buildings. The tracking error reported by [55] in this study was below 0.73 meters after initialization of the particle cloud.

Recent research focuses on non-fix placement of measurement equipment on the test subjects. This provides for a non-intrusive setup in behavior al studies as well as potential future applicability in consumer markets. [7], for example uses heading measurements obtained from gyroscopes through principal component analysis (PCA). The actual orientation of the measurement device with respect to the body is irrelevant in this case. [31] applies a similar approach to accelerometer values. Here, PCA is applied on acceleration measurements resulting from the user's walking motion. [48] incorporated this method in a study with 8 test participants wearing different trousers and recorded 23 traces. The median error (resulting primarily from heading deviations) was 12.8 cm per step when applying a 5 Hz low passed filtered 2D-PCA. This particular study aims for a mobile phone style setup. In order to avoid inaccuracies caused by low cost sensors in mobile phones, the sensor actually employed was the XSens MTx [58].

1.2 Implications

As noted in the previous section, a wide range of different approaches in user tracking is available. These methods vary mainly in infrastructural preparation effort, degree of equipment intrusion, hardware cost and post processing time. The aim of this project is to find a suitable balance between these factors.

In addition to the tracking component of the system, an intuitive representation of data has been realized. In this three-dimensional representation, the current position and viewing angle of the pedestrian can be inspected. The position and alignment direction of the observing camera can be chosen arbitrarily.

2 Fundamentals

In this chapter, the mathematical foundations for navigation and tracking are outlined. The first section deals with reference frames, which are the basis for positioning and tracking in multi-dimensional space. Subsequently an outline of necessary coordinate transformations is given. Finally, the positioning methods used in this project are illustrated. The content of this chapter strongly follows [23].

2.1 Reference frames

In order to locate an object in space, a reference system is necessary. If a three-dimensional point is considered, an exact determination of it's location requires three coordinates. Movement of this point, as it is the case in navigation or tracking applications, add a fourth dimension (time) to that description.

2.1.1 Inertial frame

An inertial frame is a non-accelerated reference system, meaning it is either in standstill or in translation with a constant velocity. In inertial frames, the laws of Newtonian Motion can be applied:

- The first law of motion states: A body at rest or in uniform motion preserves its status unless an external force is applied
- The second law of motion states: A translational change of a body is proportional to the impacting force and occurs in the direction of the force's orientation vector.

The second law can be expressed as follows:

$$\boldsymbol{F}^i = m_i \ddot{\boldsymbol{x}}^i \tag{2.1}$$

Where m is the mass of the body, the force is applied to. The acceleration vector is described by $\ddot{x}$. The superscript i indicates the inertial system.
If the gravity component is added to (2.1), the following can be asserted:

$$m_i \ddot{\boldsymbol{x}}^i = \boldsymbol{F}^i + m_g g^i \tag{2.2}$$

Where $m_g g^i$ is the gravitational force impacting the body. Applying the equivalence principle of inertial and gravitional mass $m_i = m_g = m$ and dividing relation (2.2) by m leads to the fundamental relation:

$$\ddot{\boldsymbol{x}}^i = \boldsymbol{f}^i + \boldsymbol{g}^i \tag{2.3}$$

An inertial frame for the earth can be defined as such:

- The origin of the system is the geocenter
- The x-axis points to the vernal equinox
- The z-axis points towards the mean rotational axis
- The y-axis is orthogonal to the x- and y-axis

The vernal equinox can be defined either kinematically or dynamically. For the first case, radio signals from distant stars are utilized. For the second, the intersection point between the equatorial and elliptical plane of the earth serves as vernal equinox.

Since the moon and the sun induce (gravitational) forces onto the earth, the second law of Newtonian motion is compromised slightly. Therefore, a geocentric system with space-fixed orientation is called pseudo-inertial.

2.1.2 Terrestrial equatorial frame

A terrestrial equatorial frame differs from the inertial frame in the way the x-axis is defined. In this right-handed system, the x-axis points to Prime Meridian of the Earth, the Greenwich meridian. The z-axis is again defined to point along the mean rotational axis towards the conventional terrestrial pole (CTP) which is the North Pole. Accordingly, the y-axis complements the system.

In contrast to the inertial frame, the terrestrial equatorial system follows the rotation of the earth with the geocenter as pivot point. The angle between the vernal equinox and the Greenwich meridian is called Greenwich sidereal time Θ_0 and counted counter-clockwise.

The most prominent example for a terrestrial equatorial system is the World Geodetic System 1984 (WGS84) which is used by GPS.

2.1.3 Local level frame

A local level frame can be defined everywhere on or near the surface of earth and is used to describe geodetic observations. It is defined as follows:

- The x-axis points towards north
- The y-axis points towards east
- The z-axis points towards the center of gravity (geocenter)

Note that in a left-handed coordinate system, the z-axis points to the local zenith. However, in order to avoid coordinate frame transformations, right-handed local level coordinate systems are preferred.

2.1.4 Body frame

Body frames are used to determine the orientation of a moving object (pedestrian, aircraft, car, etc.) towards the local-level frame (earth, moon, etc.). The origin of the body frame is in most cases the center of mass of the moving object. For a three dimensional case, it is composed of three rotation axes which are called as follows:

- *Roll*: rotation about the x-axis (ϕ)
- *Pitch*: rotation about the y-axis (θ)
- *Yaw*: rotation about the z-axis (ψ)

If an aircraft is considered, *roll* would be the rotational angle about the fuselage, *pitch* would be the rotation about the wings and *yaw* the rotation about the axis, which is orthogonal to both *roll* and *pitch*. In the case of an object moving with respect to a geoid, *yaw* simply corresponds to the current azimuth.

2.2 Coordinate transformations

Assuming a geocentric elliptic coordinate system, there are two possibilities to express the location of a three-dimensional point:

- By using Cartesian coordinates x_1^e,x_2^e,x_3^e
- By using ellipsoidal coordinates ϕ (latitude), λ (longitude) and height

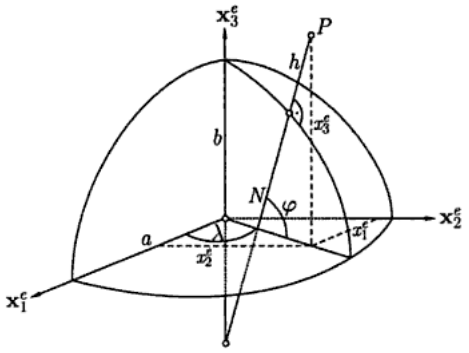

Figure 2.1: Cartesian and ellipsoidal point mapping (from [23])

The coordinate systems are related to each other as follows:

$$\boldsymbol{x}^e = \begin{bmatrix} x_1^e \\ x_2^e \\ x_3^e \end{bmatrix} = \begin{bmatrix} (N+h)\cos\phi\sin\lambda \\ (N+h)\cos\phi\cos\lambda \\ (\frac{b^2}{a^2}N+h)\sin\phi \end{bmatrix} \tag{2.4}$$

where a and b are the semimajor und semiminor axis of the ellipsoid. N is the radius of the curvature in the prime vertical obtained by [23]

$$N = \frac{a^2}{\sqrt{a^2 \cos^2 \phi + b^2 \sin^2 \phi}} \tag{2.5}$$

If an ellipsoidal coordinate system is considered, lines of **longitude** appear vertical and correspond to great circles of a sphere running from the north to the south pole. Lines of **latitude** appear horizontal and are circles of different radii which are parallel to the equator.

The reference line at which the longitude angle is zero passes through the British Royal Obversatory in Greenwhich. This reference line is called the prime meridian. As shown in Figure 2.1, the longitude of an arbitrary point on a sphere can be determined - assuming the center of the earth as vertex - by measuring the east/west angle between x_1^e (which passes through the equator and the prime meridian) and the line intersecting both the meridian of the point in question and the equator. The angle between the local vertical and the equator plane yields the latitude.

The longitude is measured from the prime median in east/west direction. Depending on the position, longitude coordinates have either an E/W suffix, or a $+/-$ prefix. So, travelling east from the prime meridian will increase the longitude from 0° to up to 180°, where the International Date Line (IDL) is reached. Likewise, traveling west from the prime meridian will decrease the longitude value to a minimum of -180 degrees.

As stated above, the latitude is the intersection angle of the local vertical of the point in question with the equatorial plane. Depending of whether the corresponding location is in the northern or southern hemisphere, coordinates are suffixed with either an N/S or a $+/-$. So, at the north pole, latitude is 90°, at the equator, latitude is 0° and at the south pole, latitude is -90°.

2.3 Positioning methods

The available methods for determining the position of an object can be divided into two main categories:

- **Position fixing**: The position can be determined directly from sensor measurements. No information about previous positions is necessary (absolute positioning).
- **Dead reckoning**: The estimation of the current location relies on knowledge about the previous positions (relative positioning).

In the subsequent sections, the methods used for the prototype are outlined in greater detail.

2.3.1 Position fixing

When performing position fixing, the coordinates of an unknown location are deduced by incorporating measurements to or from known control points. The available techniques are:

- Multiple plotting: The position of an unknown point is deduced from the different directions towards it from two control points. Considering two-dimensional space, the unknown point is defined through the intersection of two lines originating from the reference coordinates.
- Multiple ranging: The position of an unknown point is deduced from the distance towards known control points. Knowing the range to the control points, two circles can be defined. The unknown position is located at one of the intersection points. In most cases, one intersection can be eliminated through plausibility checks.
- pseudo range positioning: The position of an unknown point is determined according to multiple ranging. However, in this case, each obtained range measurement is distorted by an unknown common error. Deviations in ranging measurements are mostly the result of clock synchronization problems between transmitter and receiver. As shown in Figure 2.2, in the two-dimensional case, the uncertainty area is formed by a circle.

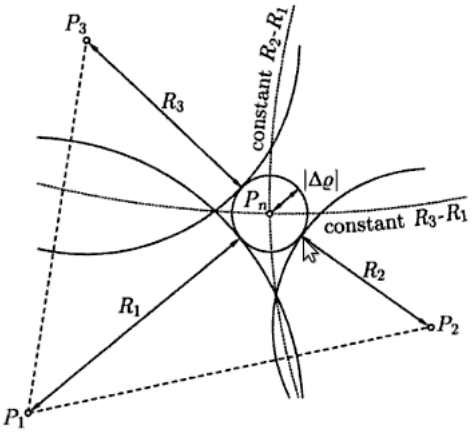

Figure 2.2: 2D-pseudoranging (from [23])

Pseudo range positioning also forms the fundamental principle for the satellite based localization system used within this project, GPS. In the following sections an overview of the system's architecture is given.

2.3.1.1 GPS system architecture

As already briefly noted in 2.3.1, as pseudorange measurement unit, a GPS receiver compares the travel time of signals sent by satellites orbiting the earth. The received messages include the precise transmission time, orbital position data of the sending satellite (ephemeris) *only* and the location of *all* other satellites (almanac). The difference between ephemeris and almanac data is that information about the former is very accurate and has to be refreshed in short intervals (30 seconds) whereas the latter provides the receiver merely with a rough estimate which satellites are trackable in principle.

Theoretically speaking, in three-dimensional space, signal lock on three satellites would be enough to solve the positioning problem. However, since the signals emitted by the satellites propagate with the speed of light, even the slightest deviation in the receiver's clock can lead to serious errors. Therefore, in practice, signal fix to a fourth satellite is necessary.

After having acquired at least four satellites as signal sources, the receiver performs the following steps:

1. Synchronize the clock
2. Calculate the transit time of two messages by subtracting the time of transmission from the current time of the receiver's clock: $transit_i = t_i - t_r$
3. Multiply the transit time with the speed of light to obtain the distance to the satellites: $d_i = c \cdot transit_i$. This yields two spheres with radius d originating at the former transmitted ephemeris position.
4. Intersect the spheres in a circle
5. Intersect the circle with the sphere obtained by the third satellite. The result is formed by two intersection points.
6. Select the intersection point, which is nearer to the earth's surface.

This algorithm assumes ideal signal propagation and clock synchronization. Due to various natural noise sources, such as interference in the same frequency band through solar bursts or scintillation in the earth's ionosphere, messages are either delayed or corrupted. As a consequence, signal lock to more than 4 navigation satellites is beneficial. This additional information can be incorporated in the positioning solution by applying least squares estimation [36].

2.3.1.2 WiFi based positioning

Another possibility to apply position fixing is to utilize signals from wireless local area network (WLAN) base stations. Since the implementation of the wireless lan standard 802.11b under the marketing name WiFi in 1999 [25], wireless LANs are available practically ubiquitously in many urban areas [47]. In contrast to the satellite based positioning method used within this project, multilateration can not be applied through measuring the time deltas of the propagating signal with off-the shelf hardware, though. With respect to WLANs, the only useable value is provided by determining the received signal strength (RSS). This value has to be converted to a distance identifier as shown in relation 2.6. The model used within the prototype is known as Log-normal-fading formulated by [40]. All power levels in the following equations are given in dB.

$$PL(d) = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma \tag{2.6}$$

Here, $PL(d_0)$ is an a-priori determined path-loss value at an reference distance from an WLAN transmitter, n is the path loss exponent and X_σ is a zero mean gaussian random

variable. Naturally, obstacles between the receiver and the base station attenuate the received signal strength more strongly. These effects can be modeled by introducing an additional parameter to relation 2.6 according to [12].

$$PL(d) = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma + \sum_{i=0}^{N_w} w_i \tag{2.7}$$

$N_w - 1$ denotes the number of walls the received signal traversed through, and w_i the attenuation in dB. The received power corresponds to

$$P_r(d) = P_t - PL(d) \tag{2.8}$$

Here, P_r is the received power and P_t is the power used at transmission of the base station. Combining 2.7 with 2.8, the received power now becomes a parameter for the distance to the base station:

$$d(P_r) = 10^{\frac{P_t - \left(P_r + PL(d_0) + X_\sigma + \sum_{i=0}^{N_w} w_i\right)}{10n}} \tag{2.9}$$

Without proper post postprocessing, the optimal achievable accuracy value for this approach corresponds to the Cramer-Rao-Bound [14], which gives the information theoretic minimum value of the error variance. This boundary is a function of angles from the mobile user towards the base stations. Consequently the accuracy will vary with the current position. E.g. [46] deployed three wireless base stations without obstacles at an area of 11m x 24m, with an varying accuracy between 6m and 15m. The resulting standard deviation from the true position was 7.4 meters and consistent with the CRB model.

The main drawback of the log normal method is its large inaccuracy if the view to an access point is obstructed. In many cases, deploying WiFi access points solely for positioning purposes is not feasible. In the context of this project, pedestrian tracking over an complete shopping street (100m x 20m) should be achieved. The following subsections explain the approach best suited for that scenario.

K-closest neighbors fingerprinting: Fingerprinting relies on geo-referenced RSS data which is pre-recorded for an specific area and then saved into a database. In order to achieve the best results, the RSS data has to be saved for as many possible user locations as possible. In cases where this is difficult to perform, the log-normal model as specified by 2.7 can be used as an interpolator. The database, which stores the RSSI information is outlined in 4.2.3.2.

Various location services ([47], [20], [49]) which offer coverage of all major cities are already available for application developers. The position reference for those databases is provided by GPS. By accessing them with a smart phone, for example, their users enhance the granularity implicitly.

Since the WiFi positioning system should serve as backup for GPS outtakes within this project, it is necessary to pre-map the RSS values. This process is described in section 4.1 in more detail.

The k-closest neighbour fingerprinting algorithm used in this work follows the implementation of [16]. This algorithm iterates through all RSS tuples in the database and calculates the Euclidian distance (in signal space) to the RSS values currently measured:

$$d(Z, Z_i) = \frac{1}{M}\sqrt{\sum_{j=1}^{M} \left(RSS_j(x, y) - RSS_j(x_i, y_i)\right)^2} \tag{2.10}$$

Here, Z is a vector containing $[RSS_1 \dots RSS_M]$ recently measured RSS values including MAC addresses at the unknown position $X = (x, y)$. Z_i denotes the different RSS tuples already stored in the database. After having iterated through all RSS footprints, the 4 lowest values [13] are extracted. The estimated position is given by the barycenter:

$$X = \frac{\sum_{j=1}^{4} \left(\frac{1}{d(Z,Z_i)} \cdot X_j\right)}{\sum_{j=1}^{4} \left(\frac{1}{d(Z,Z_i)}\right)} \tag{2.11}$$

Probabilistic estimation: As already noted above, a high granularity of RSSI footprints is necessary to provide accuracy values at the 10 metre level. If that cannot be granted, a probabilistic approach proposed by [42] can be used to weight an estimated location value by formulating the conditional probability of the location X given the signal vector Z by applying the Bayes rule:

$$P[X|Z] = \frac{P[Z|X] \cdot P[X]}{P[Z]} \tag{2.12}$$

In this equation, $P[X]$ denotes the probability of being at location X before knowing the observation value. Accordingly, $P[Z]$ specifies the probability of measuring Z and acts as normalizing constant. Its computation is described in 2.15 The likelihood function $P[Z|X]$ is the product of a Gaussian kernel method $p(z, X)$.

$$P[Z|X] = \prod_{z \epsilon Z} p(z, X); \tag{2.13}$$

and

$$p[z|X] = \frac{1}{n_X} \sum_{i \epsilon \Lambda_z} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z - z_i)^2}{2\sigma^2}\right) \tag{2.14}$$

In this method, a probability mass (the "kernel") is assigned around the observation z using the RSS training data z_i. Depending on how many training data is available for the specific location, the Kernel functions are added to each other and normalized to their number n_X. The parameter σ denotes the width of the kernel. It is possible that the RSS vector Z is lacking certain values for a specific location which were available in the calibration phase. In these cases, those missing values are simply replaced by small constants. The prior probability of measuring a specific signal vector Z is denoted by:

$$P[Z] = \sum_{X_i \epsilon \Lambda_x} P[Z|X_i] \cdot P[X_i] \tag{2.15}$$

Λ_x contains all possible location values.

Implementation issues: Besides the time consuming task of pre mapping RSS values, there are several practical problems in the tracking phase, when using WLAN positioning:

- When the user is in motion, there is only a limited amount of RSS observation values for a constant location available
- Even if the user stands still for a few seconds, averaging the acquired measurement data is difficult because of the inadequate sampling speed of off-the-shelf WLAN receivers. A typical scanning interval of integrated WiFi receivers (in notebooks or cell phones, for example) is in the range of 500 ms.
- The log-normal model used for multi lateration does not account for small scale fading, which can occur in crowded shopping streets (E.g.: other pedestrians, public transport lines, cars)

This results in a non-Gaussian probability distribution of the received signal strength for a specific access point at the same measurement location as depicted in figure 2.3.

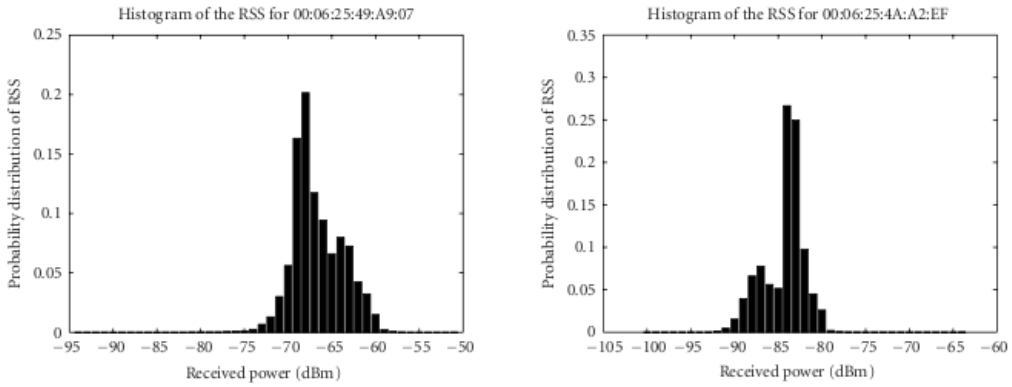

Figure 2.3: RSS distribution for 2 different access points (from [16])

To solve this problem, a filter system which can cope with non-Gaussian noise has to be used. A possible approach named particle or sequential monte carlo filter is described in section 2.4.2.

2.3.2 Dead Reckoning

By combining data of previous positioning results with transitional information, new locations can be estimated. The initial position has either to be known a-priori or be calculated by position fixing methods. By adding baseline vectors, new locations are derived. Dead reckoning approaches differ in the method, how these vectors are obtained.

The relative positioning method used within this project is named rho-theta technique. The transitional components are formed by an orientation angle θ and a range measurement ρ. The corresponding baseline vector $\boldsymbol{x_{12}}$ describing the position delta between points P_1 and P_2 is given by the following relation:

$$\boldsymbol{x_{12}} = \rho_{12} \cdot \begin{bmatrix} \cos(\theta_{12}) \\ \sin(\theta_{12}) \end{bmatrix} \tag{2.16}$$

In the context of this project, the directional angle θ is denoted by the cardinal orientation of the pedestrian. The range measurement ρ is obtained by multiplying the current step frequency with the step length.

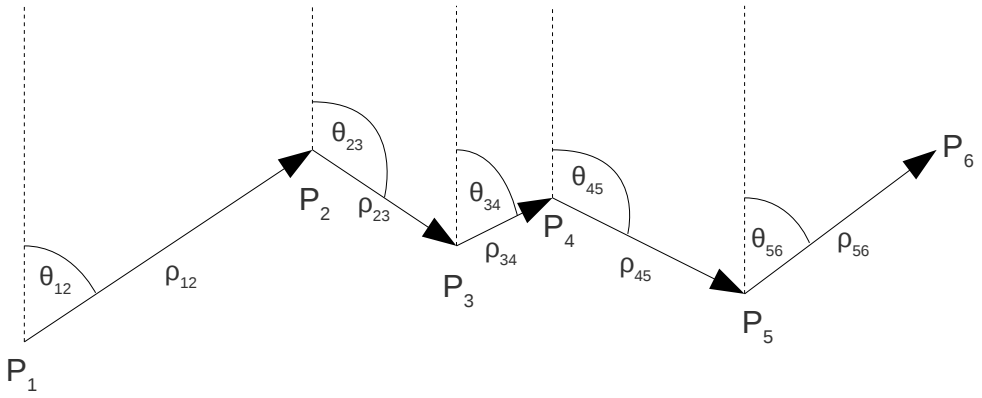

Figure 2.4: Position fixing with the rho-theta technique

2.4 Data fusion methods

The localization methods described in the previous sections differ in various aspects. Satellite based positioning methods generally offer the best coverage, but are dependent on clear viewing conditions to the sky. On the other hand, WiFi based positioning can also be performed indoors, but relies on infrastructural prerequisites. Finally, the introduced dead reckoning method can be employed ubiquitously, but has only acceptable accuracy in the short term.

Therefore, efficient methods for combining the data of the various sensors have to be used. In order to account for measurement and natural noise sources, a Kalman filter is applied to the dead reckoning subsystem. The user's speed and walking direction are then introduced in a particle filter. In the following sections, the theoretic fundamentals for both filter methods are outlined. The practical implementation with the specification of the system models and the corresponding parameter values are described in section 4.3 in more detail.

2.4.1 Kalman filter

Kalman filters are methods for predicting a system's state by weighting new observation values with their current confidence value. The underlying process model and dynamic ability to combine observations make it possible to use the system for data fusion tasks. The filter, which relies on a linear system model in its original form, was proposed first by Rudolf E. Kalman in 1960 [28].

2.4.1.1 Definitions

The general formulation for underlying dynamic system model is given as follows:

$$X_k = \Phi_{k-1} \cdot X_{k-1} + B_{k-1} \cdot u_{k-1} + w_{k-1} \tag{2.17}$$

The matrix Φ describes the transition between state $k-1$ to k. Furthermore, the system is influenced by deterministic control inputs u, which are incorporated into the system through the control input model B. The model description is completed by an nondeterministic additive white Gaussian noise component $w_{k-1} \sim N(0, Q_{k-1})$, with Q_{k-1} being the process covariance matrix. As a result, X_k is a random variable. Due to the fact that the state k is only dependent on its predecessor, the filter equation can be considered a Markov-chain.

The observation state space is modeled through the following relation:

$$Z_k = H_k \cdot X_k + v_k \tag{2.18}$$

The observable state space is limited by the capabilities of the measurement equipment and subject to additive white noise. These effects are modeled through the observation matrix H_k and the noise process $v_k \sim N(0, R_k)$, with the observation covariance matrix R_k.

2.4.1.2 Algorithm

At every sampling time k, the filter's state is defined by the following variables:

- $\hat{x}_{k|k}$: The estimate of the system state x at time k given all prior observations up to and including time k. This is normally a vector consisting of relevant measurable attributes.
- $P_{k|k}$: The error covariance matrix. These values provide a means to weight the confidence of the current state estimate.

Given a defined initial state estimate $\hat{x}_0$ and an error covariance P_0, the filter system calculates the next state by multiplying the current state with the transition model Φ. Note, that in this description the control input vector has been omitted.

$$\tilde{x}_{k+1} = \Phi_k \cdot x_0 \tag{2.19}$$

This is considered as **prediction phase**. Similarly, the error covariance matrix is updated:

$$\tilde{P}_{k+1} = \Phi_k \cdot P_k \Phi_k^T + Q_k \tag{2.20}$$

Where $Q_k = N_k \cdot R_k \cdot N_k^T$ is the covariance of the process noise. It is itself composed of N_k, denoting the general system noise and R_k describing the observation uncertainty.

In the next step, k is increased and the **Kalman gain** matrix is calculated:

$$K_k = \tilde{P}_k \cdot H_k^T \cdot S_k^{-1} \tag{2.21}$$

Here, H_k is the observation matrix. S_k is called the innovation covariance and defined as:

$$S_k = H_k \cdot \tilde{P}_k \cdot H_k^T + R_k \tag{2.22}$$

The Kalman gain matrix K_k denotes the weight of new observation values ultimately. The next state estimate is calculated as follows:

$$\hat{x_k} = \tilde{x_k} + K_k \cdot (z_k - H_k \cdot \tilde{x}_k) \tag{2.23}$$

This is called the **correction step**. Here, $\tilde{x_k}$ denotes the predicted value from the first step and z_k is the current observation data provided by a sensor. An overview of the recursion is depicted by figure 2.5. Note that in many applications, variables like the system transition

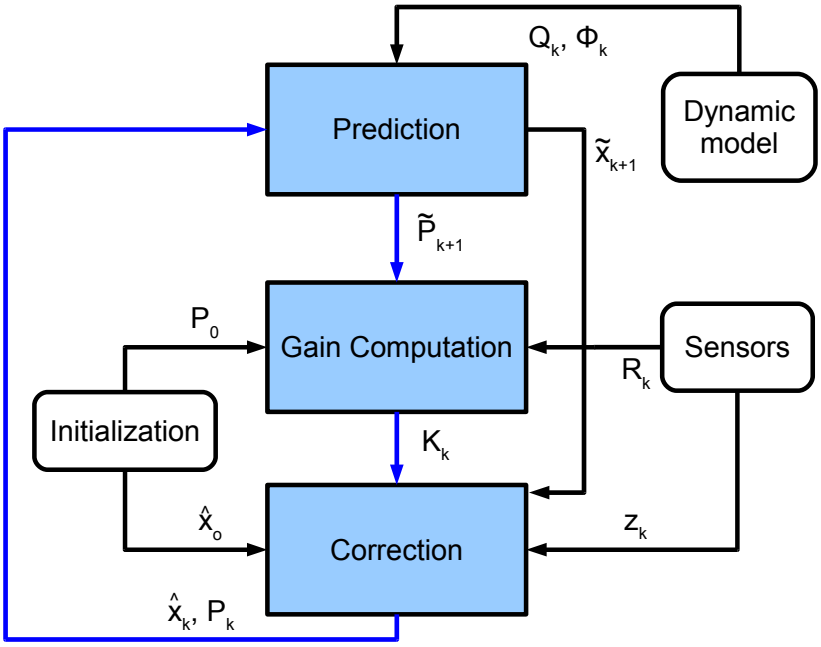

Figure 2.5: Kalman filter steps

model ϕ, the observation model H_k and the covariance matrices can be constant over time. A detailed description of the models and parameters used is given in 4.3.

Kalman filters perform generally well if the sensor variances are normally distributed and the process model is linear. As already noted in 2.3.1.2, by including WiFi measurements in the positioning approach, a non-Gaussian distribution of the RSS measuring error function has to be assumed. In addition to this, incorporation of digital map information as depicted in 4.3.5 introduces non-linearities into the motion model which the Kalman filter cannot cope with. Therefore, the sensor fusion model has to be enhanced.

2.4.2 Sequential monte carlo method

In contrast to Kalman filters, sequential monte carlo methods or particle filters are algorithms which can estimate the probability density function of the current state vector directly. This is achieved by creating a cloud of particles which consist of the elements of the state vector (2D coordinates in the tracking case) and a weight. The particles are moved forward according a specific motion model (see 4.3.2). After having advanced, every particle is assigned a solution function which depends on a metric of the new position and the current observation (RSS measurement vector). The better the return value of the solution function correlates with the current observation, the higher the particle weight.

2.4.2.1 Definitions

In more formal terms, the probability density function of the current state is calculated given all $(k-1)$ prior observations $Z_{0:k}$ directly through [41]:

$$P[x_k|Z_{0:k}] = \sum_{i=1}^{N} w(k,i)\delta(x_k - x_k^i) \tag{2.24}$$

Here, x_k^i describes a state estimate or a particle, which is weighted according to:

$$w(k,i) = w(k-1,i) \cdot P[x_k^i|x_{k-1}^i] \cdot P[z_k|x_k^i] \tag{2.25}$$

2.4.2.2 Algorithm

Following [41], a Sequential-Importance-Resampling (SIR) particle filter is realized:

1. The particle cloud is initialized in a normal distributed manner around the initial position

2. The particles are advanced according a specific motion model. This model is dependent on the data acquired by the dead reckoning component and described in 4.3.2.

3. Assign the particles a new weight according relation 2.25.

4. If the number of particles with an effective weight drops below a threshold, perform SI - resampling according to [41], to avoid degeneracy problems. The number of effective particles can be calculated following [45]:

$$N_{eff} = \frac{1}{\sum_{j=1}^{P}(\tilde{w}_t^j)^2} \tag{2.26}$$

   With $\tilde{w}_t^j$ being the newly assigned weight to particle j at time t. A good empirical criteria is $N_{eff} < \frac{P}{2}$.

5. If no external position update is available, repeat this process from step 2. Else, go to step 1 and reinitialize the cloud at the new position.

2.5 Measurement principles

The dead reckoning component of the system relies on converting measurable physical quantities like acceleration, changes in angular rate and the earth's magnetic field into the current inertial attributes of the observed body. These values include orientation and velocity. In the following subsection, the fundamental measurement principles utilized by microelectromechanical devices are outlined.

2.5.1 Accelerometer

In this subsection, the theoretical fundamentals of acceleration measurement are explained. Following that, a realization as a microelectromechanical system is illustrated.

2.5.1.1 Theoretical fundamentals

Generally speaking, accelerometers measure the force exerted on a proof mass which is connected to a casing through a flexible restraint. The delta from its initial position is then converted into an acceleration value. Depending on the type of device, several methods for sensing local displacement (capacitive, piezo electrical, mechanical, optical) are available. The rebalancing force applied by the restraining system can be modeled through the following relation:

$$F_{re} = m \cdot \frac{d^2x}{d^2t} + k_v \cdot \frac{dx}{dt} - k_e \cdot x \tag{2.27}$$

where m is the proof mass and x the displacement vector. The constants k_v and k_e describe the damping value and the stiffness coefficient of the used restraint. The homogeneous solution of this linear differential equation would be a damped harmonic oscillator. In the static case, Hooke's law can be formulated:

$$F_{re}^s = -k_e \cdot x \tag{2.28}$$

This relation can now be equated with the force exerted on the proof mass by applying Newton's second law of motion:

$$F = F_{re}^s = m \cdot a = -k_e \cdot x \rightarrow a(x) = \frac{-k_e \cdot x}{m} \tag{2.29}$$

2.5.1.2 MEMS accelerometers

MEMS accelerometers make use of the changes in the geometry of their integrated capacitors. A moving proof mass is extended by several cantilever beams and suspended by springs. These connections form, combined with a static component, a two plate capacitor. As depicted in figure 2.6, several capacitors are connected in parallel. This is necessary in order to obtain a measurable capacity value. The capacity of a two-plate setup is specified by:

$$C = \epsilon_0 \epsilon_r \frac{A}{d} \tag{2.30}$$

with ϵ_0 and ϵ_r being the vacuum and relative permittivity, A the area of the electrodes and d the distance between the electrodes. By increasing or reducing d with respect to its original

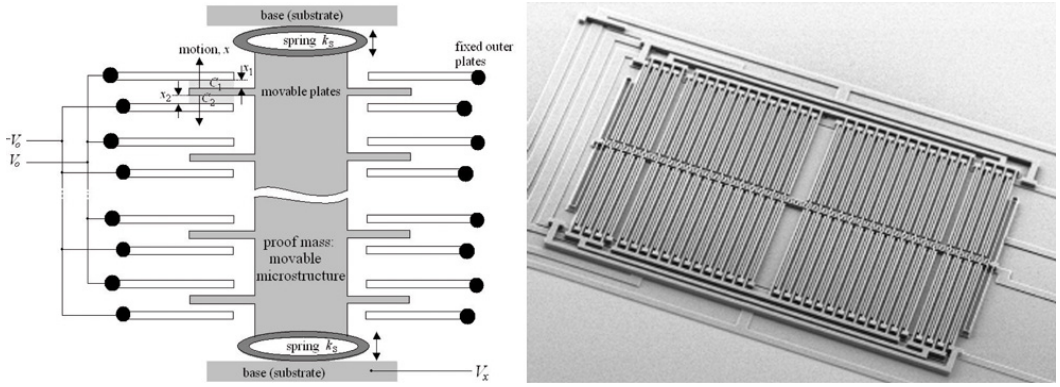

Figure 2.6: MEMS accelerometer schematic with REM image (from [3] and [44])

position by a displacement value x, two adjacent capacitors will differ in their capacitivity as follows [3]:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} \epsilon_A \frac{1}{d+x} = C - \Delta C \\ \epsilon_A \frac{1}{d-x} = C + \Delta C \end{bmatrix} \tag{2.31}$$

Here, the product of the electrical permittivity and the electrode area $\epsilon_0 \epsilon_r A$ have been replaced by $epsilon_a$ for reasons of simplicity. The difference in capacity values can be used to obtain x:

$$C_1 - C_2 = 2\Delta C = 2\epsilon_a \frac{x}{d^2 - x^2} \tag{2.32}$$

which leads to the quadratic equation:

$$\Delta C x^2 + \epsilon_A x - \Delta C d^2 = 0 \tag{2.33}$$

Since the term $\Delta C x^2$ is negligible for very small excursions (typically in the range of micrometers), the equation can be simplified to:

$$x = \frac{d^2}{\epsilon_A} \Delta C = d \frac{\Delta C}{C} \tag{2.34}$$

Two adjacent static capacitor plates are driven by rectangular reference voltages $V_0, -V_0$. By measuring V_x at the proof mass, a voltage divider can be formulated:

$$(V_x + V_0)C_1 + (V_x - V_0)C_2 = 0 \tag{2.35}$$

Combining this equation with 2.31 and 2.34 yields:

$$x = d \cdot \frac{V_x}{V_0} \tag{2.36}$$

As a consequence, the acceleration value can be expressed as a variable dependent on V_x by including relation 2.29:

$$a(V_x) = -V_x \cdot \frac{k_e d}{m V_0} \tag{2.37}$$

2.5.2 Gyroscope

Gyroscopes can be used to obtain the orientation of a body frame with respect to an inertial frame by sensing changes in the angular momentum of their rotational axis. In the following subsections, the basic measurement principle of directional changes with gyroscopes and an implementation as MEMS are outlined.

2.5.2.1 Theoretical fundamentals

The theoretical fundamentals for measuring changes in direction will be explained by a mechanical implementation of a gyroscope. If a rotational-symmetric body rotates frictionless about its center of mass, an angular momentum can be defined as:

$$\boldsymbol{L} = \boldsymbol{I} \times \boldsymbol{\omega} \tag{2.38}$$

Here, $\boldsymbol{I}$ denotes the moment of inertia of the rotating body and $\boldsymbol{\omega}$ its spinning axis. In a frictionless environment, this system will conserve its state unless a torque $\boldsymbol{\tau}$ is applied:

$$\boldsymbol{\tau} = \frac{d\boldsymbol{L}}{dt} \tag{2.39}$$

As the angular momentum $\boldsymbol{L}$ is the rotational equivalent of the linear momentum $\boldsymbol{p} = m\boldsymbol{v}$, the torque can interpreted as the rotational analog of force impacting at a distance $\boldsymbol{r}$ from the pivot point:

$$\boldsymbol{\tau} = \boldsymbol{r} \times \boldsymbol{F} \tag{2.40}$$

If the attack vector $\boldsymbol{r}$ has the same direction as $\boldsymbol{L}$, the absolute amount of the angular momentum remains the same. However, its orientation changes according to [23]:

$$d\boldsymbol{L} = \boldsymbol{L} \times d\boldsymbol{\psi} \tag{2.41}$$

Dividing by dt and substituting with 2.39 yields:

$$\boldsymbol{\tau} = \boldsymbol{L} \times \frac{d\boldsymbol{\psi}}{dt} = \boldsymbol{L} \times \boldsymbol{\omega_p} \tag{2.42}$$

Here, $\boldsymbol{\omega_p}$, is a vector both orthogonal to the torque $\boldsymbol{\tau}$ and the angular momentum $\boldsymbol{L}$. Subsequently, it can be used as indicator for orientation changes.

2.5.2.2 MEMS gyroscopes

For MEMS gyroscopes, the Coriolis effect is utilized. If an object moves in a rotating reference frame, a force defined as follows is exerted onto it:

$$\boldsymbol{F_c} = m \cdot (2\boldsymbol{v} \times \boldsymbol{\Omega}) \tag{2.43}$$

where $\boldsymbol{v}$ is the speed of the object and $\boldsymbol{\Omega}$ the rate of rotation. If the object is restrained by a spring, the displacement $\boldsymbol{x}$ can be specified with Hooke's law and 2.43:

$$\boldsymbol{x} = \frac{\boldsymbol{F_c}}{k_e} = \frac{m}{k_e} \cdot (2\boldsymbol{v} \times \boldsymbol{\Omega}) = \frac{1}{\omega_r^2} \cdot (2\boldsymbol{v} \times \boldsymbol{\Omega}) \tag{2.44}$$

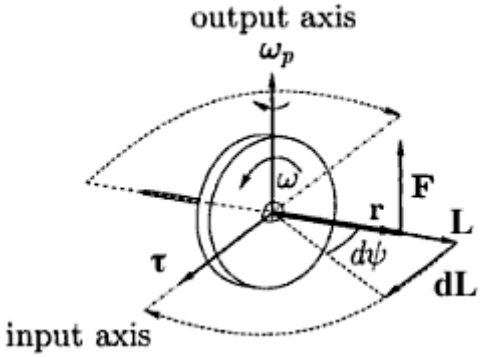

Figure 2.7: Mechanical gyroscope (adapted from [23])

with $\omega_r = \sqrt{\frac{k_e}{m}}$ being the resonant frequency. The physical quantity measured with a MEMS gyroscope is thus effectively an acceleration. In contrast to a an accelerometer, the proof mass has to be in motion constantly and its velocity vector $\boldsymbol{v}$ has to be known. This is realized through vibrating two planes of proof masses. An example for a two-dimensional MEMS gyroscope is given by figure 2.8. Here, a conceptual design of a two-dimensional vibrating

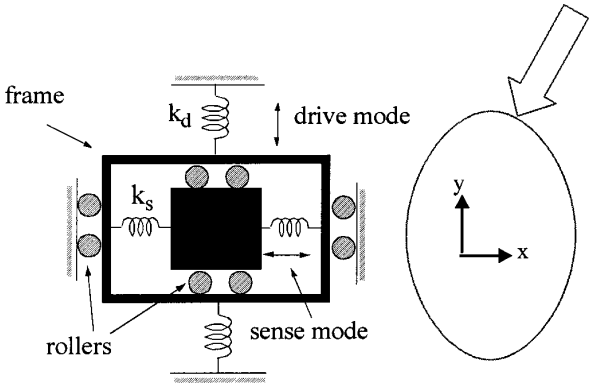

Figure 2.8: MEMS gyro schematic (from [56])

MEMS gyroscope is illustrated. The central proof mass is suspended by two springs on the left and right side. On the top and the bottom it is framed by two rollers. As a result, translation is only possible in one direction. This setup is surrounded by an outer frame. The outer frame corresponds to the second proof mass and is spring-restrained in y-direction. The inner frame is in sensing operation while the outer is driven by an external force. The displacement vertical displacement value can be specified as [56]:

$$y = Q_d \frac{F_{dm}}{k_d} sin(\omega_{r,d})t \tag{2.45}$$

Here, Q_d is the mechanical quality factor of the drive mode and k_d the spring constant. F_{dm} corresponds to the drive force and $\omega_{r,d}$ to the resonant frequency. The first order

time-derivative of y specifies the vibration velocity:

$$v_y = Q_d \frac{F_{dm}\omega_{r,d}}{k_d} cos(\omega_{r,d})t \tag{2.46}$$

Applying the second-order mechanical transfer function of the inner accelerometer, a phasor for the horizontal displacement can be defined [2]:

$$xe^{j\omega\phi_s} = K_s \frac{|a_s|}{w_{r,s}^2} e^{j\omega\phi_s} = K_s \frac{2|v_y||\Omega_z|}{w_{r,s}^2} e^{j\omega\phi_s} \tag{2.47}$$

Here, Ω_z is the rotation about the z-axis.

Several possible topologies for vibrating gyroscopes are available. These are depicted in figure 2.9. The subscripts "s" and "d" denote sense and drive modes respectively. The topologies

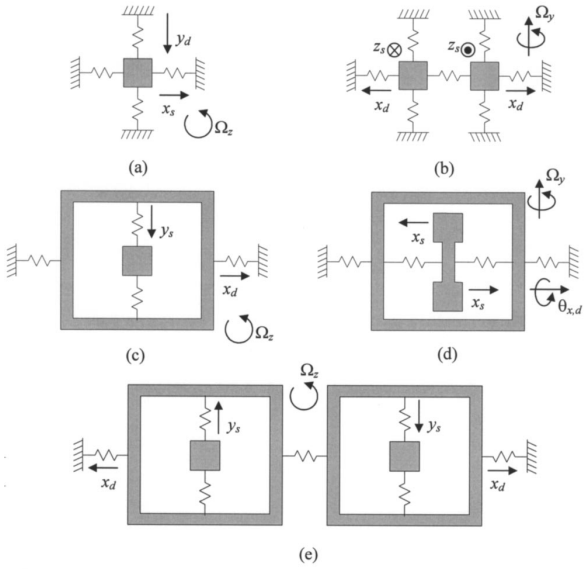

Figure 2.9: Vibrating gyroscope types (from [56])

are called as follows [2]:

- (a) Single spring mass with translational drive
- (b) Dual mass spring with translational drive (tuning fork)
- (c) Single gimbaled mass with translational drive
- (d) Single gimbaled mass with torsional drive
- (e) Dual gimbaled mass with translational drive

2.5.3 Magnetometer

When describing movements of a body frame in a three-dimensional local level frame, as defined in 2.1.3, a value for the cardinal direction is required. This can be done by measuring the Earth's magnetic field.

In the following sections, a basic explanation of how the magnetic field is sustained is given. In addition to this, the measurement principles are outlined. Finally, an example sensor implementation is described.

2.5.3.1 Theoretical fundamentals

The Earth's magnetic field can be approximated with a dipole model, with the magnetic south pole located near the geographic north pole. The field is sustained by large masses of flowing molten iron near the earth's core. Due to its conductivity and translation in a weak magnetic field, an electrical current is induced. This current ultimately creates the Earth's magnetic field.

When determining the current cardinal direction through a magnetometer, the difference between *magnetic* and *true* north has to be kept in mind. This variation is called **magnetic declination**. Throughout the globe, this value changes mainly because of irregularities in the flow of iron streams in the earth. Since records of magnetic declinations began, their values have changed considerably at some locations. The distance from true north in Iceland for example, changed from near 0° in 1600 to -40° in 1800. Now, the declination returned to -20° [26]. The components of the magnetic field of Earth are measured in a right-handed coordinate

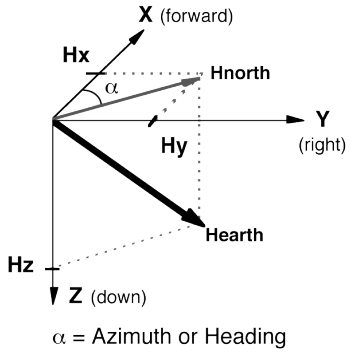

Figure 2.10: Body frame magnetic measurements (from [9])

system H_x, H_y, H_z as depicted in figure 2.10, where H_x, H_y are parallel to the surface and H_z points down. The angle between the magnetic field and the ground is dependent on location and called **magnetic inclination**. When determining direction, only the values parallel to the surface are used. It is important to note that this system is a body frame. When facing magnetic north, H_x should be at a maximum level, while H_y approaches zero. The azimuth can be determined by a variant of the arctangent:

$$\alpha = \text{atan2}(H_y, H_x) \tag{2.48}$$

where $\text{atan2}(y, x)$ is defined as:

$$\text{atan2}(y,x) = \begin{cases} \arctan(\frac{y}{x}) & x > 0 \\ \pi + \arctan(\frac{y}{x}) & y \geq 0, x < 0 \\ -\pi + \arctan(\frac{y}{x}) & y < 0, x < 0 \\ \frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases} \tag{2.49}$$

If the measurement device is tilted, the determined values for H_x and H_y do not longer correspond to the components of the magnetic field parallel to the surface. Therefore, the sensor has to be aware of changes in roll and pitch (as defined in section 2.1.4). This tilt compensation can be realized by measuring the magnetic field in all three axes and sensing orientation changes through gyroscopes or accelerometers. The received components H_x, H_y, H_z can be transformed to the the horizontal plane H_{xh}, H_{yh} as follows [9]:

$$\begin{bmatrix} H_{xh} \\ H_{yh} \end{bmatrix} = \begin{bmatrix} H_x \cos(\phi) + H_y \sin(\theta)\sin(\phi) - H_z \cos(\theta)\sin\phi \\ H_y \cos(\theta) + H_z \sin(\theta)) \end{bmatrix} \tag{2.50}$$

where ϕ is the rotation abound the x-axis (roll) and θ the rotation about the y-axis (pitch) of the reference body.

2.5.3.2 Magnetoresistive Sensors

In integrated circuits, magnetoresistive measurement methods are widely used. With this approach, changes in the electrical resistance of materials when exposed to a magnetic field are utilized. For electronic compasses, the anisotropic magnetorestistance effect (AMR) is well suited.

Suppose a very thin electrically conducting film which is connected to a voltage source. This voltage would drive an electrical current flowing through the film. If a magnetic field influences the current, an angle-dependent increase in electrical resistance can be observed. The physical reason for this is the difference in energy-level shift of electrons with opposing spin direction. The change in resistance can be formulated as follows [22]:

$$R(\theta) = \rho_{0,n}\frac{l}{bd} + \Delta\rho\frac{l}{bd}cos^2\theta \tag{2.51}$$

Here, $\rho_{0,n}$ and $\Delta\rho$ are material constants. The geometric attributes of the thin film overlay are denoted by l,b,and d which stands for length, width and thickness respectively. Normally, it can be assumed that: $l > b >> d$

3 Sensor Setup

In order to provide for a multi-sensor data acquisition environment, adequate measurement equipment was used. Since the recording hardware has to be carried by pedestrians, it was essential that the sensors were easily portable and placed in a non-intrusive way. A wide range of different sensors has been used. For visual tracking, a scene camera and an eye tracker mounted at an easy to wear cycling helmet were included in the test setup. In addition to this, two orientation and motion trackers were mounted on the helmet and the test user's belt respectively. Finally, the test participant was instructed to carry the smart phone responsible for GNSS and WiFi measurements in her trouser pocket. In figure 3.1, the placement of the different sensors onto the test person's body is shown.

Figure 3.1: Sensor setup

3.1 Position fixing component

As position fixing component, a sensor-enriched smart phone from manufacturer High Tech Corporation (HTC) named HTC Hero was used. The cellphone's operating system is Android 2.1, which is developed by the Open Handset Alliance. The OS supports interpretation of specifically generated bytecode through its virtual machine (Dalvik VM), or executes compiled code natively. For the platform, several compiler tool chains are available. Supported

programming languages are Java, C, C++ and Python.

Figure 3.2: HTC Hero

As far as the data acquisition is concerned, the handheld has to access the GPS receiver, listen to changes in WiFi signal levels and store the data. Since the performance critical part of these tasks is already implemented in the device's drivers and the corresponding APIs are well documented, Java was chosen as programming language for the measurement software.

3.1.1 GNS system

The position fixing component of the handheld is directlyintegrated into its Qualcomm MSM7200 hardware platform. Its brand name is *gpsOne*. In addition to the conventional GPS locking process, this chipset can use position information provided by the cellular network in order to reduce the time to first fix (TTFF). Unfortunately, the manufacturer does not specify comparable performance characteristics apart from a -160 dbM sensitivity [39]. However, in [21], a performance evaluation of various integrated smart phone GNSS receivers in an urban environment has been done. The results are illustrated in table 3.1.1. Due to the

Parameter	Value	Unit
Max Signal	41.13	db
Avg Signal	23.69	db
Valid Fix Time	94%	
HDOP	2.85	m

Table 3.1: MSM 7200 GNSS performance

fact that accuracy of the GNSS system is always dependent on parameters such as satellite constellation, signal attenuation and building environment, these numbers can vary, though.

3.1.2 WiFi system

The device uses a dedicated WiFi scanning platform named *WL1251* which is a system on chip (SoC) design. It supports the 802.11b/g standards, which operate in the 2.4 GHz spectrum.
In order to scan for WiFi levels, the measuring software has to register a receiver object at the operating system. After that, the wifi service can be requested to perform a scan. If new scanning results are available, a receive callback is invoked with a list of *ScanResult* objects.

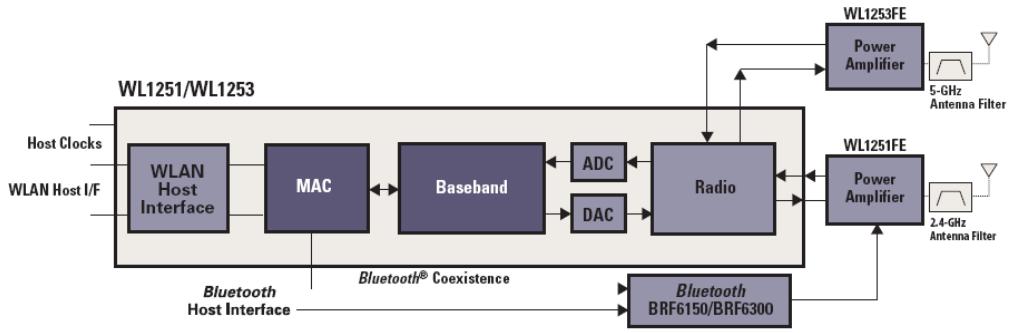

Figure 3.3: WiFi system block diagram from [50]

One object consists of the MAC address, operating SSID, frequency and the signal level of a detected access point in dBm.

3.2 Dead reckoning component

In order to both deduce position from the user's motion and to give an estimate of his or her current field of vision, inertial measurements are necessary. It is essential that the recording equipment does not interfere with the normal movement and behavior al patterns of the users. Therefore, light weight, easy to carry inertial measurement units were used for data recording.

The systems deployed were a Xsens MTi-G which was placed onto the eye tracker helmet and

Figure 3.4: Xsens MTx

a MTx which was placed at the test subjects hip. The IMU placed at the head is responsible for recording head movements, the device at the hip captures acceleration and magnetic field values for relative positioning.

3.2.1 Orientation Measurement

The orientation measurement utilizes both the sensing devices' internal gyroscope and magnetometer. The sensor implicitly returns an orientation estimate by fusing the data with an internal filter system. This filter can be optimized for several scenarios, such as aerospace,

maritime, automotive or pedestrian. The last profile is optimized for slow velocities and sudden accelerations, making it naturally the best suited setting.

The sensors utilized by the measurement device are an MEMS solid state gyroscope with beam structure and capacitive readout as well as a thin film magnetoresistive magnetometer. The measurement principle for both devices is described in sections 2.5.2 and 2.5.3. The performance of the internal sensors as specified by the datasheet is as follows:

Sensor	**Gyroscope**	**Magnetometer**
Full Scale (FS)	±1200 deg/s	±750 mGauss
Linearity	0.1% of FS	0.2% of FS
Bias stability	1 deg/s	0.1 mGauss
Scale Factor stability	–	0.5%
Noise	0.05 $\frac{m/s^2}{\sqrt{Hz}}$	0.5 mGauss
Bandwidth	40 Hz	10 Hz
Max update rate	512 Hz	512 Hz

Table 3.2: Xsens MTx orientation sensor performance

In addition to this, after combining the sensor readings with each other, the internal filter system produces the following output accuracy:

Angular Resolution	0.05° RMS
Static Accuracy (roll/pitch)	$< 0.5°$
Static Accuracy (heading)	$<1.0°$
Dynamic Accuracy	2° RMS

Table 3.3: Xsens MTx specified orientation estimation accuracy

It is important to note that the specified dynamic accuracy depends on the type of motion and does not comply with pedestrian movement. In section 5.1.1.1 more specific orientation measurement results are given.

3.2.2 Acceleration Measurement

The component for acceleration measurements in both motion trackers is a solid state MEMS accelerometer with capacitive readout. The basic functional principle of this module has already been described in section 2.5.1. The performance of the sensor according to the datasheet is specified as follows:

Full Scale	± 50 m/s^2
Linearity	0.2% of FS
Bias stability	0.02 m/s^2
Scale Factor stability	0.03
Noise	0.002 $\frac{\text{m/s}^2}{\sqrt{\text{Hz}}}$
Bandwidth	30 Hz
Max update rate	512 Hz

Table 3.4: Xsens MTx specified accelerometer accuracy

The acceleration data gained by the sensor then has to be converted into a velocity estimate. This process is explained in section 4.3.2.

4 Prototype Implementation

In this chapter, all necessary steps for implementing the prototype are illustrated. The first section describes the mapping process, in which geographic referencing and the recording of RSS data took place. Following that, the framework for managing all data sets is outlined in detail. In the succeeding section, the used algorithms for data post processing and the prerequisites for fusion are illustrated. Subsequently, the capabilities of the visualization component are introduced in the final part of this chapter.

4.1 Mapping

To avoid systematic measurement errors, highly accurate mapping was necessary. The acquisition area was a 100 x 20 meter sized part of Herrengasse, which is situated in the inner city of Graz. For initial bearing determination, a nearby cadastral map point was targeted with a laser ranging tachymeter from a known reference location. Subsequently, all remaining reference points were recorded from that position. With that approach, deviations could be limited to a maximum value of 10 cm. This data was then realigned graphically and inserted into the database. The targeted reference points incorporated house corners, facade boundaries, shopping windows and manhole covers.

The mapping of reference points was concentrated on probable user positions. Altogether, 140 points where recorded. If the acquisition area is reduced by the space claimed by tram rails (650 m^2), the average number of reference points per 10m x 10m area yields 1.04. The measuring process required one operator for the tachymeter and an assistant for holding the reflector. Due to the semi-automatic nature of the acquisition equipment, the geographic mapping was completed within 2 hours. After the geographic mapping process, RSS data for the fingerprinting database (see 4.2.3.2) was recorded. For this task, two capturing programs for both a Windows XP laptop and an Android smart phone were created. The acquisition software for the Windows notebook was based on the open source Inssider WLAN monitoring tool [38]. For the smart phone implementation, enabling the WiFi service and registering a listener to a WiFi update event called by operating system were the only required steps.

Both software components offer an user interface, which requires the operator to choose the id of the current reference point. Then, the RSSI values, the SSIDs and the MAC addresses of all WiFis in range are recorded. The signal strength indicators are then averaged for a period of 20 seconds. After having acquired all measurements, the data was checked for plausibility and subsequently uploaded to the database. This process required 3 hours of work.

Figure 4.1: Mapping process

4.2 Data management

In this section, the underlying data storage approach is outlined. In the first subsection, the software components which form the foundation of the database are described. Then, a general overview of is given. The system is required to combine pre-recorded information with measurements which are added after the deployment phase dynamically. Sections 4.2.3 and 4.2.4 describe all necessary tables as well as data fields for these tasks.

4.2.1 Software components

In order to cope with geo-referenced data efficiently, data types and methods which are optimized for spatial processing have to be incorporated into a classic object-relational database. Therefore, the popular PostgreSQL database with PostGIS was used. This software combination supports essential features, such as platform independency, user-defined functions and spatial extensions.

4.2.2 Overview

The database is composed of tables requiring preparatory work (like mapping or measuring RF fields) on the one hand and data sets which are recorded dynamically on the other. Tables which incorporate static data are denoted by an "s_" prefix. Information, which is acquired on the fly in a user context is stored in tables beginning with "u_".

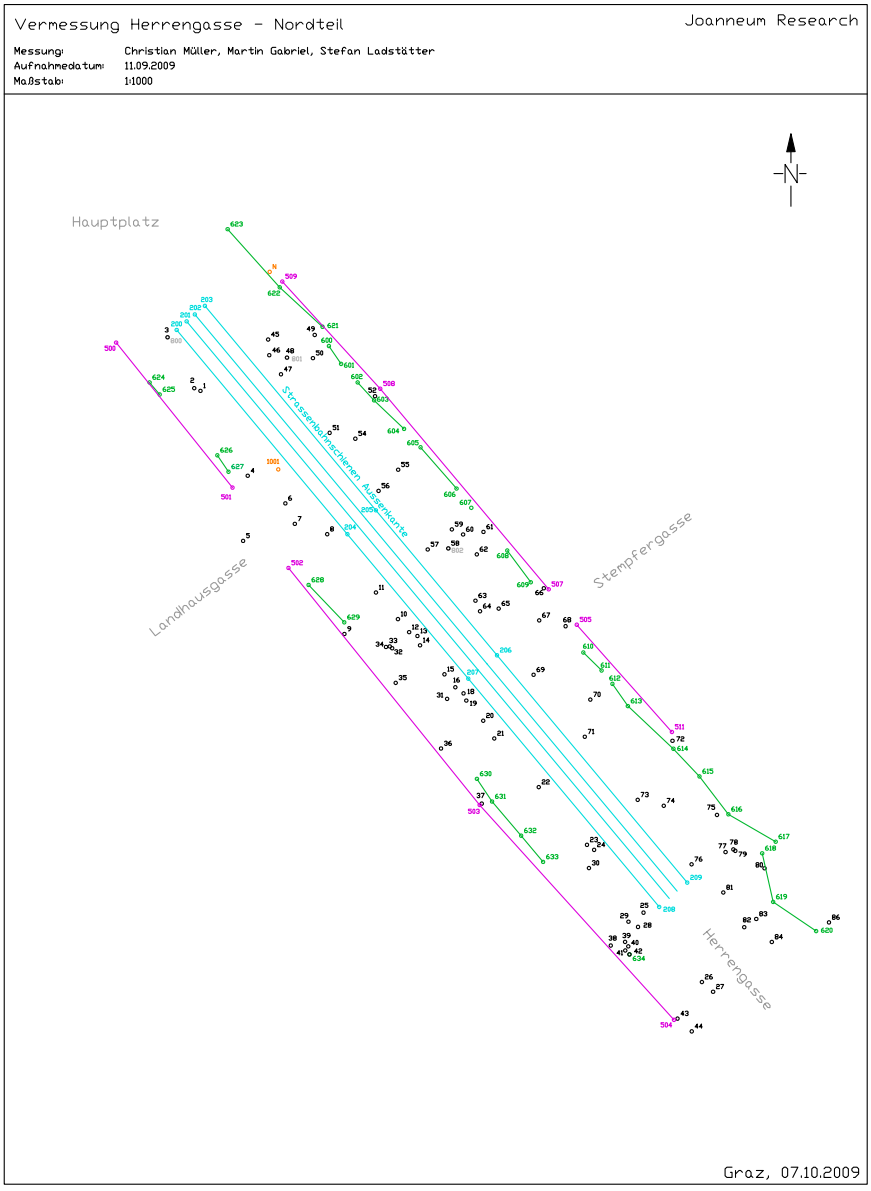

Figure 4.2: Acquisition of reference points

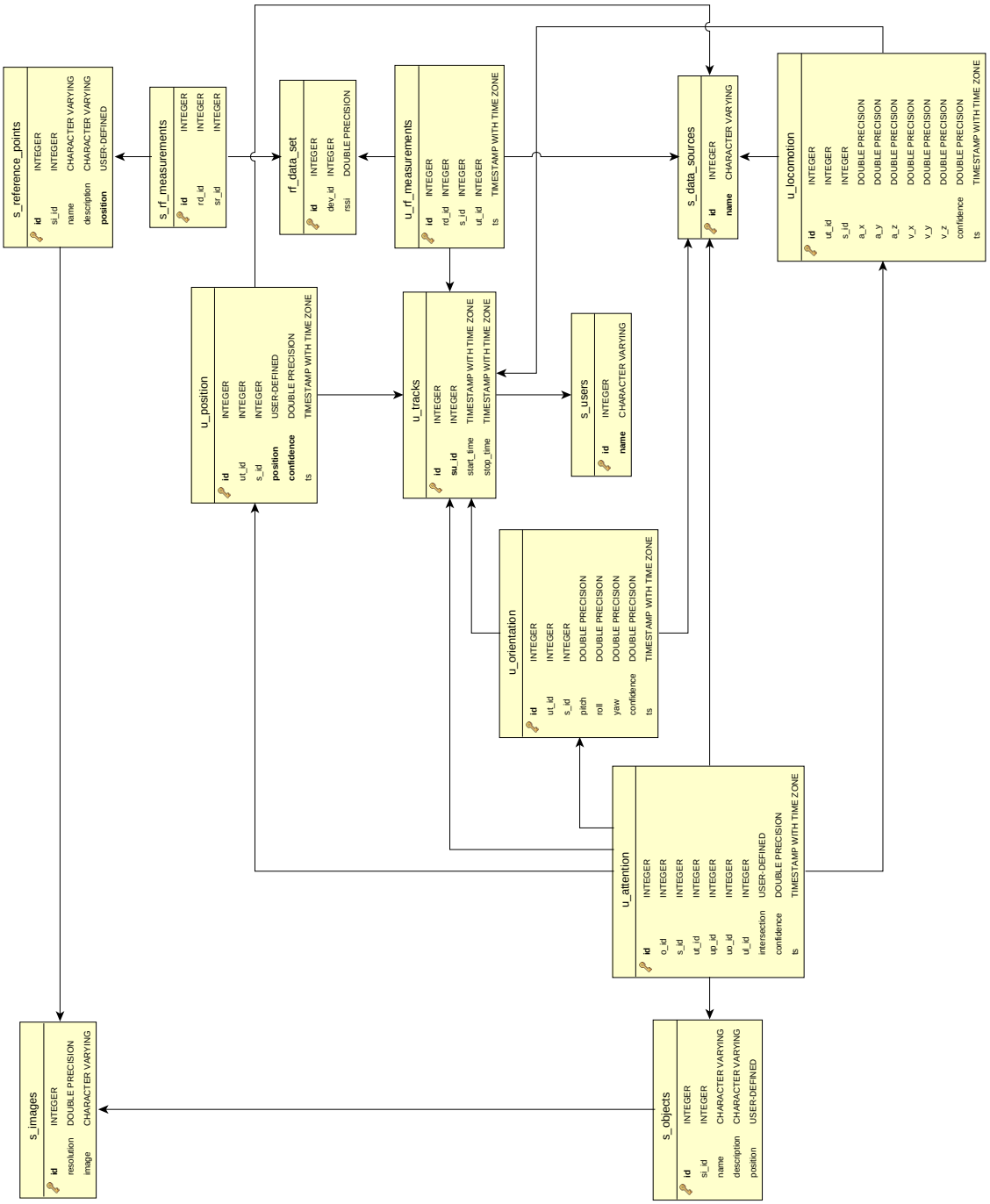

Figure 4.3: Database overview

4.2.3 Static Data

Every piece of data which is a prerequisite for system operation is considered static. Data capturing is done with specialized equipment before conducting the user study. Hence, information about elements belonging to that category can be acquired usually in a very precise manner.

4.2.3.1 Objects

Generally speaking, every data set in the system to which user attention is measurable, is classified as object. Examples for objects could be house facades, shopping windows or company logos. Every object is classified by an unique identifier, a name and an optional description. For reasons of simplicity, geographical position and height are saved as cuboids with infinitely small depth. That approach enables intersecting the test user's line of sight with the object's surface.

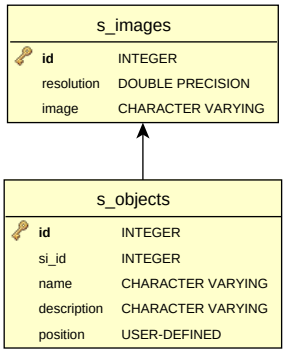

Figure 4.4: Objects

In order to make image processing possible, a corresponding texture for the object has to be specified. The texture of the object has to be cropped in such a way that its corner points match the extent of the cuboid. Therefore, specification of only one texture object (o_images) suffices for unambiguous pixel to coordinate mapping.

4.2.3.2 Reference points

Reference points are pre-mapped markers in the test area which are easy to identify visually (e.g.: sewer / duct covers, facade boundaries, hydrants). Similarly to objects, reference points are denoted by an unique id, name and description. However, the referenced image is only used as a means to ease identification. Furthermore, for unambiguous localization, the center of the reference point is saved. In addition to this, information from nearby RF fields is acquired initially. A RF data set is composed of an unique id, an device identifier (E.g.: MAC address of an WiFi access point) and the radio signal strength indicator (RSSI) in decibels. This information is used for positioning estimates in the post processing phase.

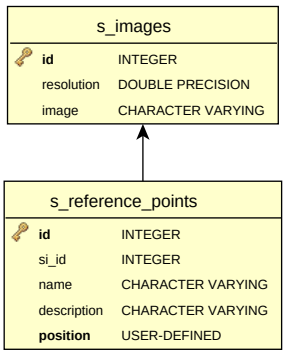

Figure 4.5: Reference points

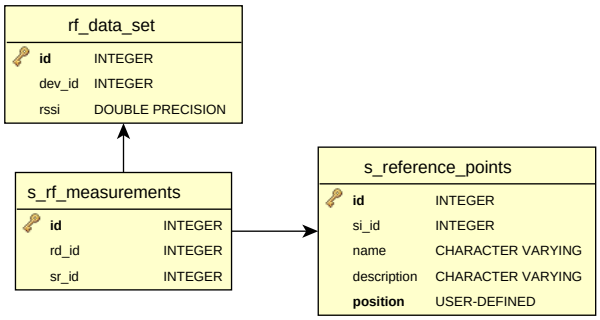

Figure 4.6: Relation between Reference points and RF measurements

4.2.3.3 Data sources

In order to differentiate the various pieces of acquired sensor information and estimations resulting from post processing, the corresponding data sources have to be saved too. A data

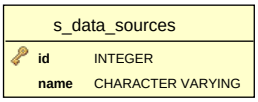

Figure 4.7: Data sources

source is denoted by an unique id and a name. Sources can be divided into sensors, which perform live capturing and post processing methods. The interdependencies with dynamic data assets are described in section 4.2.4 in more detail.

4.2.3.4 Users

Every dynamically recorded data set is assigned an unique user id and name. In this way, identification and later assignment of the recorded data is possible.

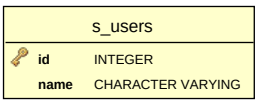

Figure 4.8: Users

4.2.4 Dynamic data

Dynamic data is all information which is recorded during the field study on the fly or is result of post processing. In the following sections, the database tables dealing with that kind of information class are illustrated.

4.2.4.1 Tracks

In order to identify beginning and termination of a test user session unambiguously, all data dynamically captured is assigned to a single track.

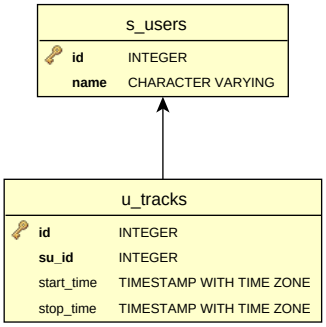

Figure 4.9: Track relations

The mapping to a specific user is achieved by incorporating her or his id as foreign key. In addition to this, to guarantee global integrity, the current time zone is stored as well.

4.2.4.2 Locomotion

To achieve an accurate behavior al mapping of the test users, recording their locomotion patterns is essential. Since various types of sensors and different places for mounting are used, acquired data has to be handled accordingly. This is done via a reference to a corresponding data source. As every locomotion data set is part of a test track, a link is saved as foreign key. As primary locomotion parameters, current velocity and acceleration in three axes (x - North, y - East, z - center of earth) is saved. The recording of the current speed could be theoretically avoided by integrating the acceleration measurements over time. The different data output characteristics of the various sensor devices and varying temporal resolutions make live capturing of either velocity or acceleration feasible, though. Furthermore, the reliability of the current measurement is denoted by the normalized standard deviation of the sensor and saved as confidence value. To maintain the time context, the capturing date is written in the ts field.

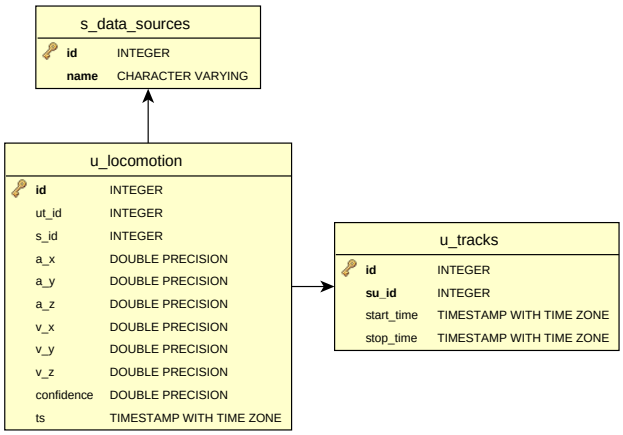

Figure 4.10: Locomotion relations

4.2.4.3 Orientation

In order to be able to estimate the user's gaze and the direction of her or his heading, an accurate recording of the orientation values is necessary. Data sources for that type of information include gyroscopes, electronic compasses, differential positioning methods and image processing. For differentiating between walking direction and line of sight, sensors are placed on both head and foot. Again, a foreign key to "s_data_sources" stored in every "u_orientation" data sets provides that unique mapping. In addition to this, the recording is assigned a reference to the corresponding track. Orientation information is saved as Euler

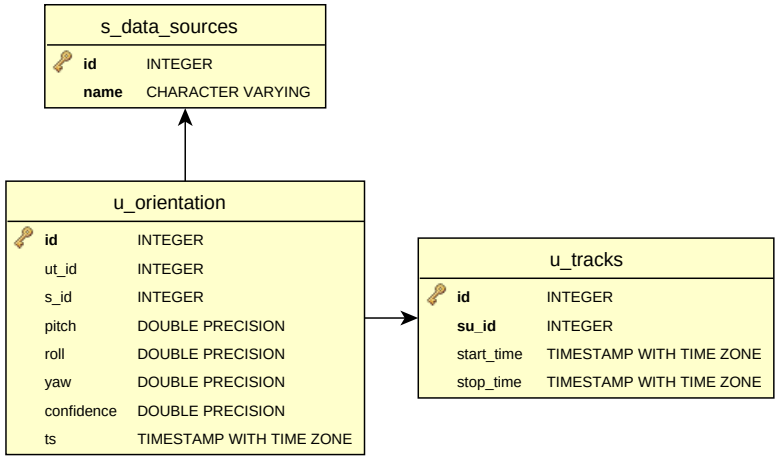

Figure 4.11: Attention relations

angles, which describe rotation around three reference axis according to section 2.1.4. The measurement uncertainties are expressed through a normalized confidence value. Again, the point of capture time is saved.

4.2.4.4 Position

User position can be either deduced from satellite multi lateration directly or estimated in post processing with help from inertial measurement or map matching data. The sensor source and the associated user track are referenced as foreign keys.

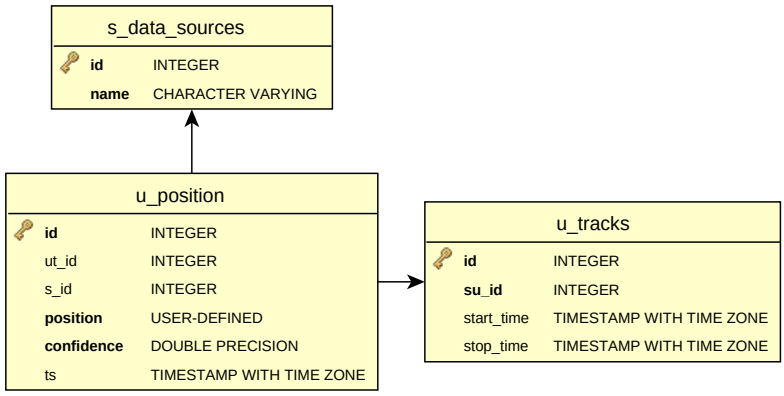

Figure 4.12: Orientation relations

4.2.4.5 RF measurements

While moving in the test area, measurement equipment on the test user currently keeps track of the signal levels of surrounding high frequency radio fields. This information is combined with the preliminary RF measurements surrounding the reference points to obtain a position estimate. Similarly to the prior user data sets, this information is assigned to a track and tagged with respect to time.

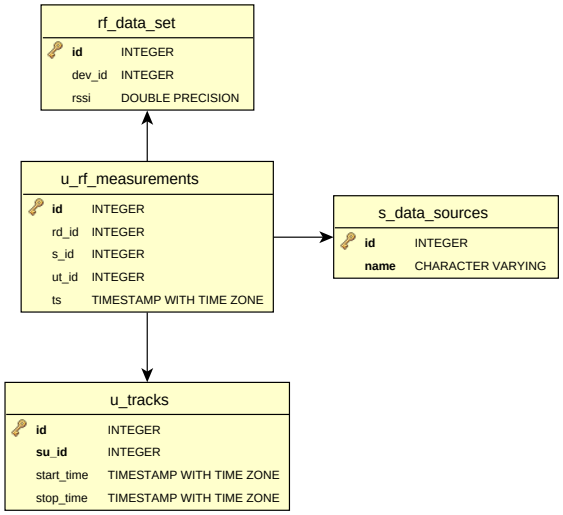

Figure 4.13: RF measurement relations

4.2.4.6 Attention

In the context of this project, attention is deduced by combining all dynamic data classes mentioned in the preceding sections with each other. It is self-evident that performing such a task is only possible in post processing. All incorporated data sets are referenced through foreign keys. In the course of data post processing, the intersection point of the user's line

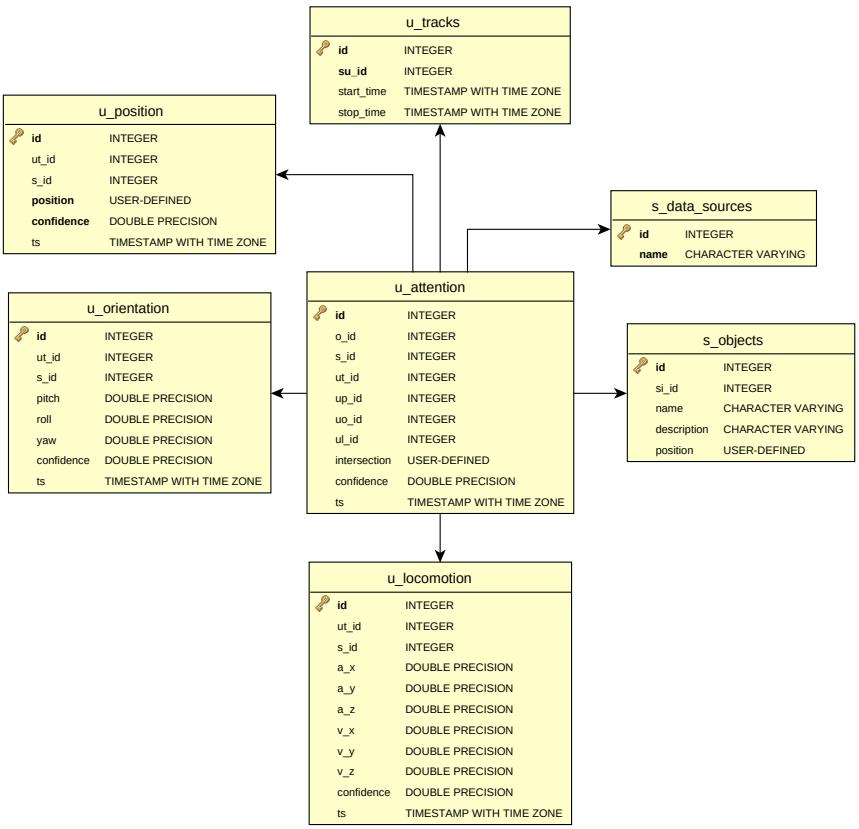

Figure 4.14: Attention relations

of sight with an object in the database is stored. In addition to this, the accuracy of this intersection is saved as normalized confidence value. The point of time, when a user has paid attention to a specific object, is an estimation.

4.3 Data fusion

This section describes all necessary steps for converting the returned sensor data like orientation, acceleration and GNSS values into usable information. The desired output values are optimal 2D localization values and the current movement speed of the user. In addition to this, tilt, cardinal orientation and roll of the test person's head is acquired.

An overview of the employed data fusion strategy is given in the first subsection. Subsequently, the required algorithms for data conversion are introduced.

4.3.1 Sensor fusion approach

Two methods for incorporating new data into the system are supported by the data fusion approach:

1. Integrate sensor data into the motion model and thus change the direction of particle propagation
2. Reassign particle weights depending on new input data

In this project, a combination of Kalman and particle filtering is implemented. The Kalman filter method is aimed at smoothing the input of the dead reckoning subsystem, which is responsible for step and stride length detection as well as for determining the user's body and head orientation. In addition to this, position fixing data obtained from the GNS subsystem is weighted against its current reliability. The particle filter on the other hand uses both the smoothed absolute and relative localization information to reinitialize and advance the estimation cloud respectively. Additionally, the particle weights are weighted according to the WiFi fingerprinting database. In the same way, map matching is achieved. An overview of the approach is given by figure 4.15.

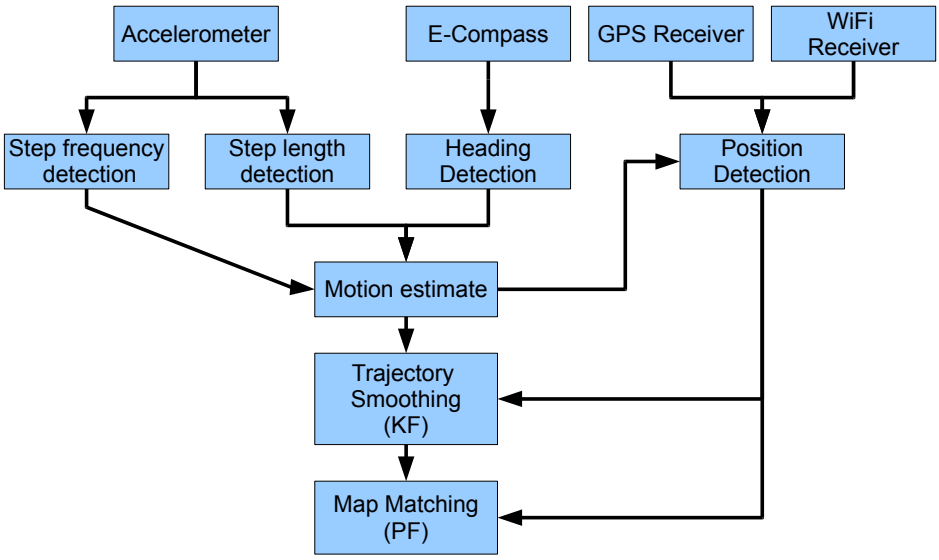

Figure 4.15: Data fusion overview

For the first implementation, a dedictated MATLAB toolbox was used for the particle filter component [11].

4.3.2 Motion model

The translational model is designed to mimic human movement patterns by adhering to Newton's laws of motion. It is the basis for advancing the particles in the primary filter system and defined following [16]:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_s \cos\theta_k \\ 0 & 1 & T_s \sin\theta_k \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ v_k \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \end{bmatrix} \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} \tag{4.1}$$

Given a known starting location x_0, y_0, this model considers the current heading θ and the most recently determined velocity value v_k. Furthermore, a Gaussian noise error process is simulated by the last additive term. Because the underlying movement model is designed according to Newton's laws, the error component has the form of an acceleration value.

4.3.2.1 Step frequency determination

Depending on the placement of the sensor, various step counting methods are available. Similar to heading detection, measurements of periodic changes in the angular momentum or variations in the magnetic field could be used to detect steps. Additionally, sensor data from a downward oriented accelerometer will roughly oscillate around a mean value of the Earth's gravity.

Due to possible x/y rotation of the sensor in the test person's trouser pocket, an algorithm which utilizes the magnitude of the whole three-axis acceleration data is used. A method proposed by [27] was implemented with slight modifications:

1. Compute the magnitude of the acceleration data:

$$a_i = \sqrt{a_x^2 + a_y^2 \cdot a_z^2} \tag{4.2}$$

2. Compute the local acceleration variance to highlight foot activity and to remove gravity

$$\sigma_{a_i}^2 = \frac{1}{2w+1} \sum_{j=i-w}^{i+w} (a_j - \overline{a_j})^2 \tag{4.3}$$

$$\overline{a_j} = \frac{1}{2w+1} \sum_{q=j-w}^{j+w} a_q \tag{4.4}$$

   $\overline{a_j}$ denotes the local acceleration value and w the sampling window.

3. Apply a threshold to detect the motion phase. A threshold value of $B > 2\frac{m}{s^2}$ has been gained empirically.

4. If the acceleration variance drops below the formerly applied threshold, a stance phase is detected

When using this algorithm, it is essential to provide the sampling window with valid data for all times. In a real-time processing environment, the filter delay would be at least four times the sampling window.
In figure 4.16, the detection process is illustrated. Note that the acceleration pattern consists of 18 peaks, with 9 peaks having a higher amplitude. That is the result of placing the sensor in vicinity to a specific leg making its vertical acceleration variance prevalent. When using this method, the number of detected steps has to be multiplied by two.

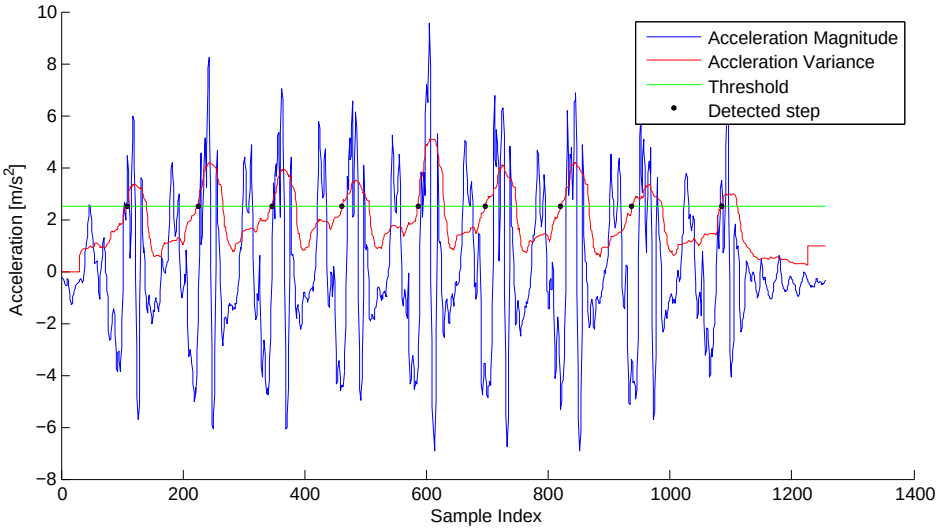

Figure 4.16: Detection of 18 steps

4.3.2.2 Step length model

The stride length of a person is dependent on body height and the current step frequency. For the test scenario, a simple expression proposed by [54], which assumes that step length is proportional to vertical acceleration changes, is well suited:

1. Calculate the magnitude of the acceleration data
2. Low pass filter that signal
3. Apply the Weinberg-Expression:

$$l_{Weinberg} = K \cdot \sqrt[4]{max_{j=i\pm w}^{\tilde{a_j}} - min_{j=i\pm w}^{\tilde{a_j}}} \tag{4.5}$$

Here, K denotes a scaling constant depending on the body height of the test person, $\tilde{a_j}$ is the low-pass filtered data set and w the averaging window.
Figure 4.17 shows the step length estimation of the 18-step acceleration pattern. The acceleration values themselves have been normalized to one. As depicted in the figure, the step length estimation depends on the difference between the current minimal and maximal acceleration value. Its mean value amounts to approximately 0.7 meters.

4.3.2.3 Filter integration

Integrating the inertial measurement data into the motion model described in 4.3.2 is done by means of a Kalman filter which can reduce the effects of observation and processing noise.

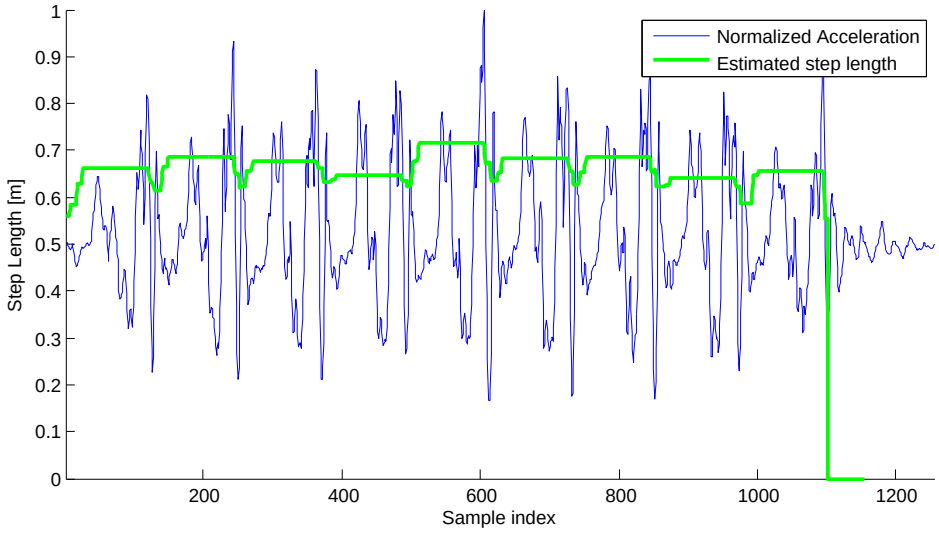

Figure 4.17: Step length estimated by Weinberg expression

The basic equations for the system are defined as follows (adapted from [16]):

$$
\begin{aligned}
\begin{bmatrix} \hat{\theta} \\ \hat{v} \end{bmatrix}_k &= \begin{bmatrix} \theta \\ v \end{bmatrix}_{k-1} - \begin{bmatrix} \dot{\theta} \\ \dot{v} \end{bmatrix}_k \Delta T \\
\hat{\boldsymbol{P}}_k &= \boldsymbol{Q} + \boldsymbol{P}_{k-1} \\
\boldsymbol{K}_k &= \hat{\boldsymbol{P}}_k \left[\hat{\boldsymbol{P}}_k + \boldsymbol{R} \right]^{-1} \\
\begin{bmatrix} \theta \\ v \end{bmatrix}_k &= \begin{bmatrix} \hat{\theta} \\ \hat{v} \end{bmatrix}_k + \boldsymbol{K}_k \begin{bmatrix} \theta_{fix} - \hat{\theta} \\ v_{fix} - \hat{v} \end{bmatrix} \\
\boldsymbol{P}_k &= (I - \boldsymbol{K}_k) \hat{\boldsymbol{P}}_k
\end{aligned}
$$

After having estimated the kth iteration from the preceding values, the current inertial state is integrated in the motion model. The corresponding covariance matrices for the process ($\boldsymbol{Q}$) and observation noise ($\boldsymbol{R}$) limit the potential negative effect on the accuracy. Both matrices have diagonal shape and include the specific situation dependent standard deviations. A more detailed list of accuracy values can be found in section 5.1.1.1.

In order to be able to estimate the next filter states, it is necessary to access the first order time derivative of the output values. These are:

- The angular rate: $\dot{\theta} = \frac{d\theta}{dt}$.
- The movement acceleration: $\dot{v} = \frac{d(L_{step} \cdot f_{step})}{dt}$

The angular rate $\dot{\theta}$ is returned by the used motion tracker itself with the same sampling rate as the accelerometer readings. The current movement acceleration however, has to be sensed by measuring differences in the step length and step frequency considering the previous values.

In addition to being able to predict the next filter state, it is necessary to include a second estimation. In this case, the position fixing system provides the filter with both a velocity and a heading value.

4.3.3 GNSS data

Considering the quality of satellite based positioning, the number of satellites, the current dilution of position (DOP) and the signal strength are relevant parameters [23].

The GPS Subsystem of the mobile device already offers a quality indicator which incorporates these three components. It simply returns the accuracy of the current positioning fix as floating point value in meters. As this value corresponds to a two-dimensional location, it has to be adjusted by a factor of $\sqrt{2}$ when applied in the filter system. This indicator is referenced to as ACC.

4.3.3.1 Filter integration

New measurements from the GNSS receiver can be weighted with the following covariance matrix:

$$R_{GPS} = \begin{bmatrix} \left(\frac{ACC}{\sqrt{2}}\right)^2 & 0 \\ 0 & \left(\frac{ACC}{\sqrt{2}}\right)^2 \end{bmatrix} \tag{4.6}$$

This model assumes uncorrelated observations which is actually not the case for the internal measurement process (both latitude and longitude are determined simultaneously using the same measurement data). By incorporating only valid GPS fixes, this simple model has proven to be satisfactory, though [36].

It is important to note that the returned accuracy value from the GNSS receiver does not account for multipath effects. Therefore, GPS signals were only used in two cases in the the test area. First, the signals had to fit the motion model described in 4.3.2. Second, the position estimate had to correlate with the result from the WiFi localization.

4.3.4 WiFi data

The inclusion of WiFi measurement data works by assigning new weights to the propagated particles according to relation 2.25. The conditional probability of z_k given x_k^i can be formulated through the nearest neighbour approach described by 2.11:

$$P[z_k|x_k^i] = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{||X_{zk} - X_{x_k^i}||}{2 \cdot \sigma^2}\right] \tag{4.7}$$

The parameter σ should be chosen indirect proportional to the precision of RSS measurements. As already noted in 2.3.1.2, the RSS values obtained by the users will not be properly averaged, so relatively large variations can be expected. Consequently, the σ value has to be adapted to a broad range of errors.

4.3.5 Map matching

Map matching is also achieved by adjusting the particle weights. In contrast to the WiFi positioning method however, this method does not take an additional measurement into account but checks the location of the propagated particles for plausibility. In more formal terms, the conditional probability $P[x_k^i|x_{k^i-1}]$ in relation 2.25 is set depending on the general possibility on whether particle i can have the state x_k.

This state corresponds to a two-dimensional location in the context of this project. The weighting of a particle when considering only its location is straightforward:

$$P[x_k^i|x_{k-1}^i] = \begin{cases} 1 & \text{if } x_k^i \text{ is in map} \\ 0 & \text{if } x_k^i \text{ is not in map} \end{cases} \tag{4.8}$$

So, an automatic plausibility check for outliers coming from the absolute positioning methods is established implicitly. Particularly in indoor localization tasks, were the freedom in movement is strongly limited, this method is very efficient.

4.3.6 Attention detection

In order to get a clear overview of the points of attention in the user's track, the following filter procedure is performed:

1. Load all track points, velocity measurements and line of sight recordings into a buffer
2. Check for each track point if the user's line of sight intersects with a potential pre-saved annotation region (shopping window, company logs, house facades)
3. Take the points of LOS intersection and calculate the probability of attention via:

$$p_a(d, \phi, v) = \frac{1}{2}\left(w_d \frac{d_{min}}{d} + w_v\left(1 - \frac{v}{v_{max}}\right)\right) \cdot w_i |\tan\phi| \tag{4.9}$$

$$w_d + w_i + w_v = 1 \tag{4.10}$$

   Here, d denotes the current distance to the intersection point of the user's line of sight with an object of interest and d_{min} is the minimum distance feasible. v is the pedestrian's current velocity and with respect to that, v_{max} the maximum velocity walked in the recording phase. Both parameters are multiplied by the tangent of ϕ, which describes the angle between the user's LOS and the perpendicular towards the shopping window. The output of this function is illustrated in figure 4.18. Finally, w_d,w_i and w_v are weights for the various parameters.

4. If the probability of attention for the specific point reaches a higher value than 0.5, mark the track point as point of attention.
5. Send this point of attention to an image recognition component
6. Save the results in the database

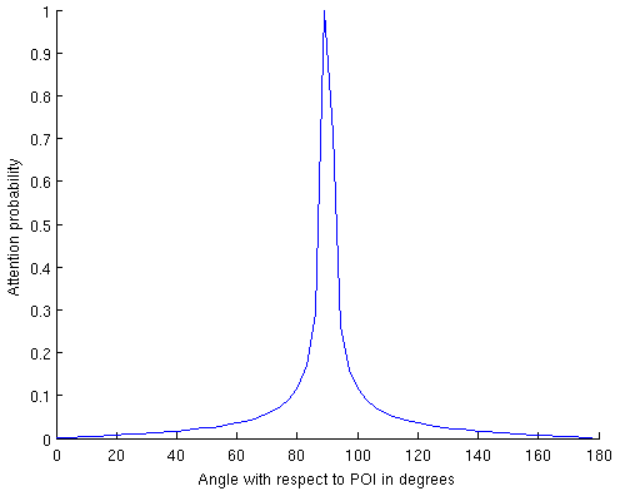

Figure 4.18: Angle dependent attention probability

4.4 Visualization component

In order to offer a general overview of the recorded tracks, an efficient as well as clear visualization is necessary. To satisfy both requirements, the viewing software has to account for the currently necessary level of detail automatically. In addition to this, the spectator has to be equipped with a range of customizable options.

The amount and type of recorded data exceeds traditional tracking systems: Raw data capturing is done at a rate of 25 Hz for inertial- and orientation measurement. In order to be able to perform in-depth analysis, this capturing rate is constant in raw mode. Assuming a moderate walking speed at around 1.2 m/s, a track with a length of about 100 meters already produces roughly 2090 samples. Naturally, one has to keep in mind that motionless phases produce the same amount of data in the same time because of the invariant capture interval. This high sample rate would introduce an unnecessary challenge to the rendering engine. Consequently, when a post-processed track is accessed from the database, after initial downsampling, the amount of points being loaded was made dependent on the user's current velocity.

In addition to the recording of the walking trace of the pedestrian, orientation measurements of the head were performed. This data is also downsampled according to the maximum rotation and tilt speed which could be expected by the test subjects.

The target platform for the visualization component is a Microsoft Windows desktop computer. Within the software, features like fast database access, 3D rendering, multimedia playback and an intuitive graphical user interface are required. The high number of different versatile class libraries and the interfacing capabilities of the .NET 3.5 Framework in combination with the Windows Presentation Foundation suits these requirements well.

4.4.1 Rendering Engine

The combination of both track and orientation measurement with a virtual representation of the test scenario requires a simulated three-dimensional environment. This environment has to support superimposed images, animation and 3D modeling. To meet these requirements, a visualization framework which uses the Google Earth API as foundation was developed.

The rendering engine makes it possible to view geospatial imaginary as well as customly created models from arbitrary angles and distances. If available, it utilizes the computer's integrated graphics accelerator adapter for optimal perfomance. Access to the rendering

Figure 4.19: View at the inner city of Vienna with Google Earth

engine is achieved via a COM type library. This component is available via a web browser plugin obtainable at [18]. In order to extend the interaction capabilities of the graphics renderer, integration within a desktop application is necessary. Fortunately, the library is embeddable within a Winforms managed code assembly through a common language runtime wrapper provided by [10].

Using this component, it is possible to improve the versatility of the stand-alone Google Earth viewer to a high degree. The main extensions to the standard viewer are synchronizing the eye tracker's video stream with the estimated position of the user, the possibility to customly control the animation, a head tilt and rotation visualization and a method for highlighting points of attention (POAs). These extensions to the standard geo viewer are now outlined in greater detail.

4.4.2 Video stream synchronization

In order to have a direct comparison between the deduced field of view of the pedestrian and the data recorded by the eye tracker, corresponding synchronization between the different data sets is essential. In addition to this, the component requires tolerance with respect to

dropped frames in the recorded videos.

The eye tracker capturing software uses XVID/MPEG4 [59] compression with an average bitrate of 1.2 Mbit/sec in its acquisition process. This algorithm uses (apart from other optimization methods) different types of frames for storing image data:

- Intra-coded frames (I-frames), on which JPEG compression is applied to
- Predicted frames (P-frames), which save the deltas with respect to their preceding frames
- Bi-predictive frames (B-frames), which are constructed by differences between both an preceding P-frame and a succeeding I-frame.

As a consequence, when selecting a specific image from the recorded video at a certain time code, the frame has either to be an I-frame or be reconstructed. Considering the high data rate of the video stream and the subsequent necessary fast decompression requirements, a C-sharp wrapper [29] library accessing the MediaDet Object provided by the DirectShow framework [37] was used. This component can be easily integrated within a .NET managed code assembly. Using these components, fully user track synchronized real time video playback

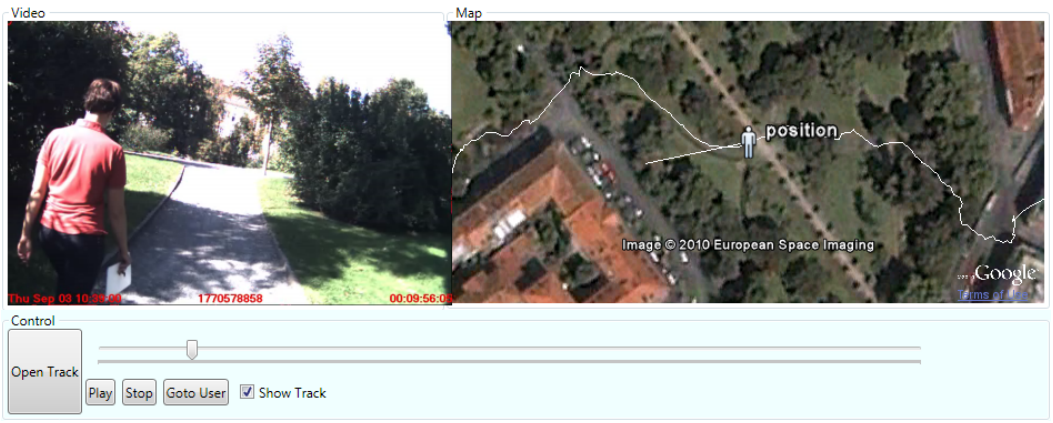

Figure 4.20: Video synchronization example

can be established. By panning the slider, the time code can be changed arbitrarily.

4.4.3 Track visualization

Within the track visualization component, it is possible to switch between the different levels of post processing as well as display the raw data. The post-processing chain for accuracy improvement yields the following track types:

1. Dead reckoning trajectory estimate (green)
2. Wifi and GPS based position fixing components (orange)

3. Fusion result from DR and absolute position fixing data (red)
4. Track realignment result from map matching (blue)

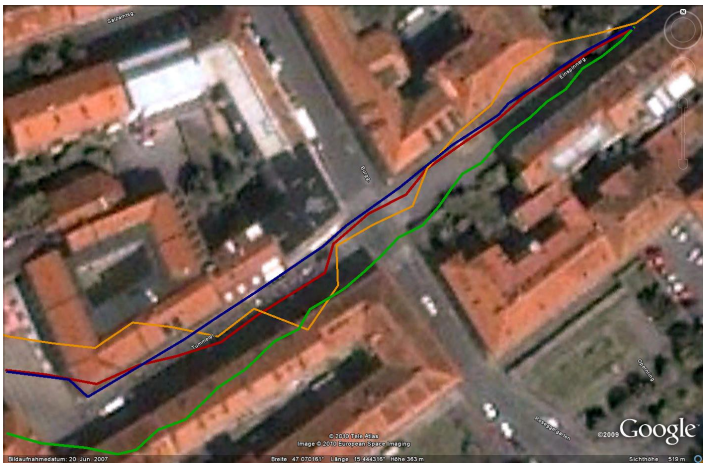

Figure 4.21: Different levels of positioning accuracy.

4.4.4 Orientation visualization

As far as orientation measurements are concerned, three parameters are the most prevalent. These are head direction and tilt as well as the current body orientation.

Figure 4.22 shows an example plot of orientation measurements. Both head (green) and body (blue) orientation are denoted using a 360° division of the cardinal directions, where 0 means north, 90° means east, 180° means south and 270° means west. As shown by Figure 4.22, at sample number 2840 up until 2930, the test person is moving straight ahead heading east. At sample 2860, the head is rotated towards south for a short period of time. The pedestrian continues to look slightly to the right. When inspecting the head tilt measurements, peaks at 2915 (looking up) and 2940 (looking down) become visible.

Despite the fact that the 2D-plot helped to identify and evaluate specific behavioral patterns of the test persons (described in section 4.3.6), it surely is not very intuitive. Therefore a straightforward representation of the user's current cone of vision and his body orientation was created within the 3D environment. This representation is illustrated in Figure 4.23.

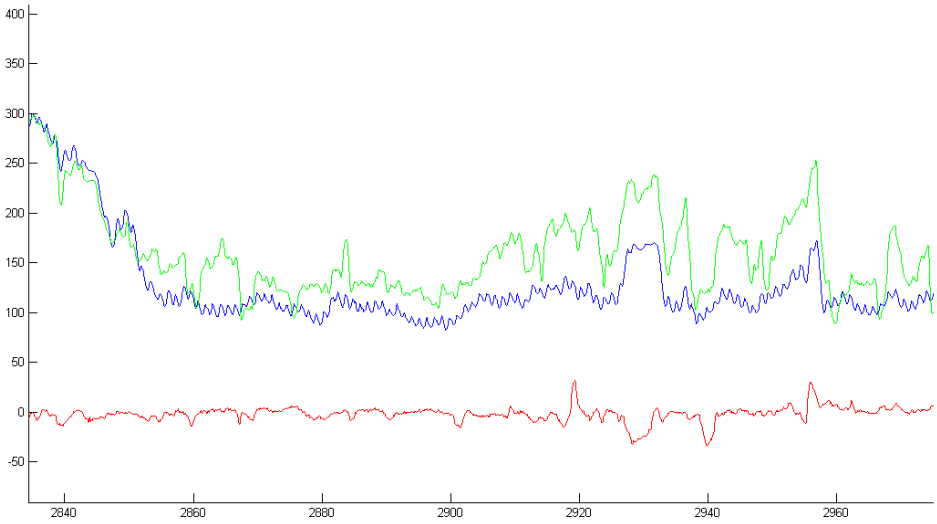

Figure 4.22: Example of orientation measurements. Body orientation is denoted blue, head direction is green, head tilt is red

Figure 4.23: Example of orientation visualization. This image corresponds to sample number 2880 in Figure 4.22

5 Measurements

In this chapter, the performance of the positioning system is outlined. It is divided into two parts: An initial small scale test and the description of the actual deployment of the system in the pre-mapped evaluation area.

In the small scale test, the stand-alone accuracy of the varying sensor components is illustrated first. Then, the improvement of the overall performance after applying various enhancement stages is shown.

The second part of this chapter contains an accuracy evaluation of several tracks in the inner city of Graz. Here, the actual practical benefits of the filter system are outlined.

5.1 Small scale test

In order to obtain necessary calibration data and to provide a clear overview of the various performance enhancement stages, a small scale test was conducted. The test area's building characteristics are similar to those of the deployment zone, so comparable precision checks for the GNS and WiFi positioning subsystem can be provided.

Figure 5.1: Calibration test area

As depicted in figure 5.1, the area is bordered by an inner and outer trapezoid. This shape defines a centered path which is composed of four straight lines with a length of 53 meters from the lower left to the upper left corner, 48.4 meters from the upper left to the upper right corner, 40.6 meters from the upper right to the lower right corner and finally a 10 meter segment which closes the track. So, the total length of the path amounts to 152 meters.

The track shape has been been mapped with Google Earth Pro Edition by overlaying a high resolution (10cm/pixel) aerial image onto the test area and visual identifying the corner points. In this way, fast mapping results were available while maintaining overall accuracy at the meter level.

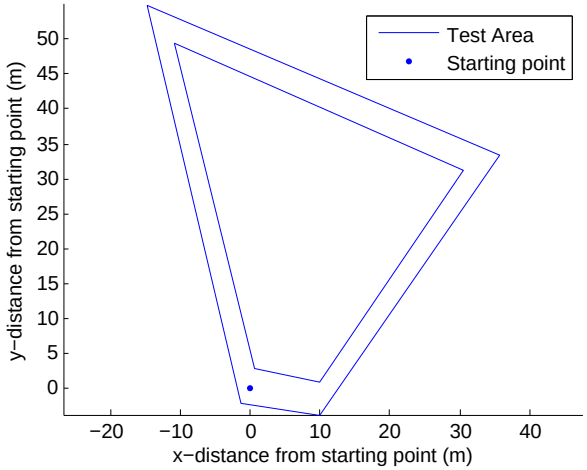

Figure 5.2: Basic 2-dimensional track template

The basis of the coordinate system is the World Geodetic System 1984 (WGS84). For two-dimensional cartesian operations, the mapped points where converted into the Universal Transverse Mercator projection in zone 33 north. These coordinates were then reduced by the x/y value of the starting point of all test tracks. In figure 5.2, the fundamental two-dimensional representation of the test area is depicted.

5.1.1 Accuracy tests

In this section, the results of the stand-alone performance of the used positioning sensors are shown. Since the data fusion system relies on accurate sensor integrity information the tests conducted in this chapter are essential for overall performance.

Beginning with an evaluation of the relative localization component, this section concludes with the measurement results of both the WiFi and satellite based positioning methods.

5.1.1.1 Dead Reckoning

The overall performance of the relative positioning system depends on the quality of the components defined in table 5.1.1.1:
The sensor used for dead reckoning already includes an internal data fusion algorithm which combines readings from the gyroscope and the magnetometers in order to give an optimal orientation estimate [58]. However, this system cannot cope with the inherent drift of all relative positioning methods and needs periodic updates. With the following measurements,

Detection Method	**Sensors**	**Unit**
Heading	E-Compass, Gyroscope	deg
Angular rate	Gyroscope	deg/s
Step frequency	Accelerometer	1/s
Step length	Accelerometer	m

Table 5.1: Specific methods for relative positioning

the interval demanding new location estimates is determined.

Figure 5.3 shows the results of the heading accuracy test.

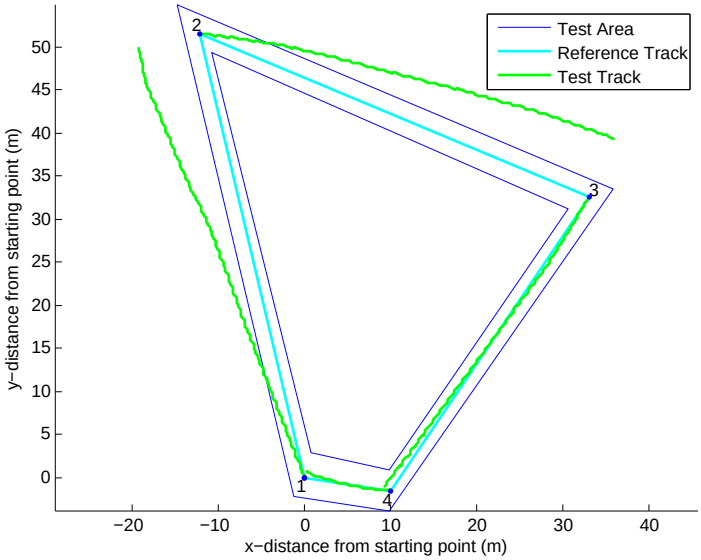

Figure 5.3: Heading test

The test track was traversed beginning from the starting point at coordinates (0,0) following the four quadrants of the trapezoid in clockwise manner. In order to avoid influences of inaccurate velocity determination, the test person was instructed to maintain a constant walking speed.

In addition to this, static heading deviations had to be considered. The magnetic declination at the test area had a value of 2°58' East at the time of writing. Furthermore, the sensor displacement on the test subject's body with respect to the cardinal direction amounted for 7° East. Consequently, all bearing measurements were reduced by 10°.

Figure 5.3 shows a varying degree of heading deviation depending on the direction currently travelled. This is mainly a result of distortions of the Earth's local magnetic field induced by ferromagnetic materials such as cars, duct covers or traffic sign bars. If the

sensor's orientation had remained still throughout the whole movement process, the internal data fusion algorithm could have utilized the gyroscope data more efficiently. However, the placement of the sensor at an acceleration sensitive spot automatically results in periodic orientation changes. Detailed heading determination results are given by table 5.1.1.1.

Path	**Path travelled**	**Distance from target**	**Avg. Deviation**	
-	[m]	[m]	[m/m]	[°]
1-2	53.3	7.2644	0.1363	7.7612
2-3	48.4	7.4335	0.1516	8.6204
3-4	40.6	0.9394	0.0231	1.3255
4-1	10.2	0.6958	0.0689	3.9415

Table 5.2: Heading accuracies for different walking directions

This yields an average value of 5.4121° for the heading data. In addition to the current bearing, speed measurements are necessary for relative positioning. The step frequency is obtained according to the method in section 4.3.2.1 on page 40. To determine the accuracy of this approach, three tracks with different velocities where recorded between points 1 and 2 in figure 5.3. The exact distance traveled amounts to 53.3 meters. Table 5.1.1.1 shows the results of the detection algorithm.

Average walking speed	**Actual Steps**	**Steps counted**	**Error**
[m/s]	-	-	[%]
1.2	83	81	3
1.4	75	74	1.4
1.6	68	67	1.5

Table 5.3: Step detection accuracy

Due to the prevalent nature of peaks in the acceleration patterns, the step detection algorithm proves to be very reliable. The remaining detection error can be explained by steps made at the beginning of walking phases. However, those steps are typically shorter in length than regular strides, so the negative effect is limited.

For step length detection, the approach already described in 4.3.2.2 is used. The stride length estimated by this method is a result of differences between vertical acceleration values scaled by an user-dependent constant. This constant should be optimized for the most frequently used walking pace. The accuracy for three different velocities is shown by table 5.1.1.1.
Due to the calibration of the scaling constant according to the normal walking velocity of the test user, the step length estimation was optimal for an average speed of 1.4 meters per second. It is important to note that the overall detection accuracy of the total step length depends both on the frequency of velocity changes and on the quality of the step counting method. So, in this test, the detection error rates of 3%, 1.5% and 1.6% for slow, normal and

Avg. Speed	**Total length detected**	**Mean**	**Error per step**
[m/s]	[m]	[m]	[cm]
1.2	54.0430	0.6464	2
1.4	53.45	0.6999	0.4
1.6	50.7826	0.7445	7.5

Table 5.4: Step length detection accuracy for a path of 53.3 meters.

fast walking speeds respectively apply. In addition to this, the length estimate only considers the walking pace of one foot. As a consequence, the stride distance has to be multiplied by two.

The resulting track when combining step frequency and length detection as well as heading measurements is depicted in figure 5.4. In this test, the user was instructed to move

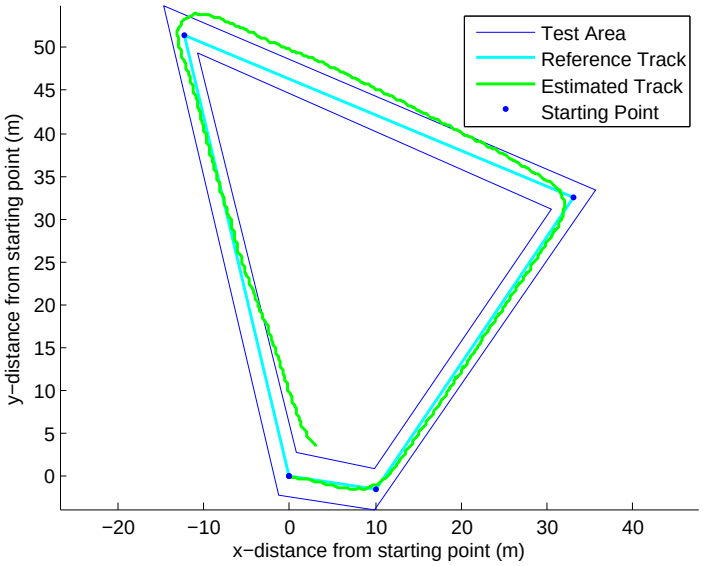

Figure 5.4: Dead reckoning test

counterclockwise along the trapezoid track. The heading deviations were in accordance with the average value already obtained from the first test measurements. After the 152 meters travel distance, the motion model estimated a position with an offset of 4.80 meters, which results in an average error of 3 cm per meter travelled. This accuracy would be good enough for behavior analysis, for longer term applications, periodic position updates are necessary, though. Examples for these methods are described in sections 5.1.2.2 and 5.1.2.3.

5.1.1.2 WiFi positioning

The first absolute positioning method used in this first small scale test relies on nearby WiFi signals. Before the tracking test, the signal vectors of 15 reference points were recorded with

a distance of 10 meters and then inserted into the fingerprinting database. Following that, the k-closest neighbour approach described in 2.3.1.2 was used to find the most likely position from the training data.

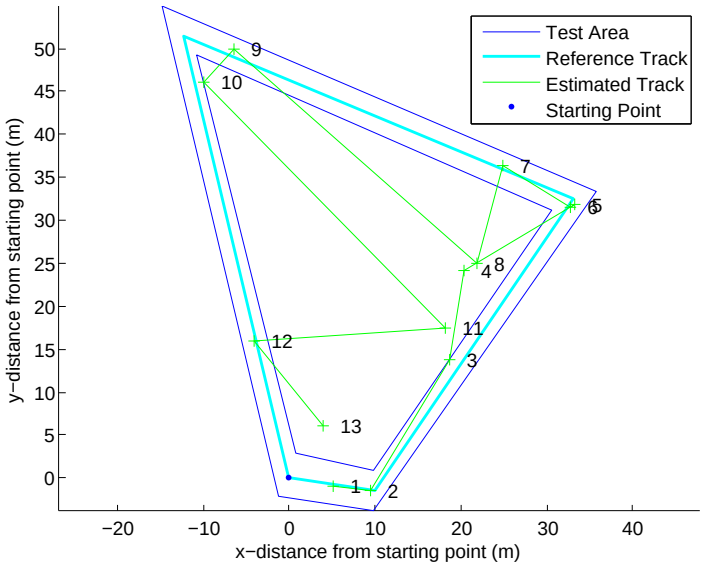

Figure 5.5: WiFi based positioning test

Figure 5.5 shows the result of a test track which started at coordinates (0,0) and was traversed again in counterclockwise manner at a velocity of 1.4 meters per second. The surrounding WiFi signal level data was recorded every 2 seconds and averaged after 4 iteration steps.

While the general shape of the track is recognizable, the results show severe outliers on positions 8 and 11. By complementing the position estimation with the probabilistic approach introduced in 2.3.1.2, these false estimations can be filtered. This method relies primarily on plausibility checks considering a pedestrian's motion model and the general probability of a location. Since the employed data fusion model already includes the latter algorithm, only the motion model integrity check is performed.

When using the plausibility verification, outliers can not be corrected but rather have to be discarded in most cases. Thus, the granularity of the WiFi based measurement data can diminish heavily. However, in the test scenario, users can be expected to stop rather frequently, making it possible to gain more RSS information. In addition to this, as already shown in the preceding section, inertial sensors allow for longer waiting periods before position updates are necessary.

5.1.1.3 GNSS measurements

Since the proposed WiFi fingerprinting method requires a certain amount of pre-mapping work, the location system should also be able to process GNSS based positioning data. The

user was instructed to traverse the same path as in the preceding tests. As shown in figure 5.6,

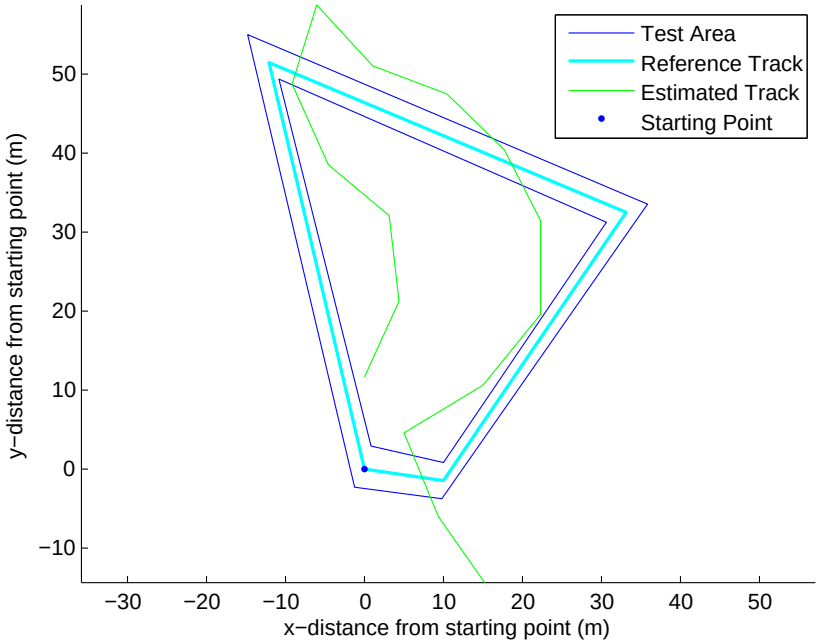

Figure 5.6: GNSS based positioning

the GNSS based positioning system suffers from moderate multipath and signal obstruction effects. These problems are particularly severe at the starting position. In order to improve the positioning results from this sensor, a similar probabilistic estimation method as used for the WiFi localization system can be utilized.

At the starting point, the GNSS position estimate deviates 15 meters from the actual location. In the right hand part of the track, the estimation shortly improves to 5 meter accuracy. However, signal obstructions caused by surrounding buildings make a position estimate impossible in corner 2, though. Before the third turn, location estimates with up to 10 meter deviation are returned. While returning to the starting position, which is illustrated at the left hand side, the accuracy of the GNSS system stayed between 10 and 15 meters again.

5.1.2 Sensor fusion

If the several positioning approaches are compared with each other, it is interesting to note that the unfiltered dead reckoning system performs best if the starting location is known. In this section, the accuracy improvement of the various data fusion methods is documented.

5.1.2.1 Trajectory smoothing

When inspecting the result trajectory of the dead reckoning component closely, an oscillation pattern perpendicular to the travelling direction becomes clear. This is a result of the moving

hip of the test person while walking. These oscillations amplify the negative effects of a short term heading deviation. By sensing the current angular rate and including trajectory estimates from the position fixing system, the current bearing of the test user can be stabilized.

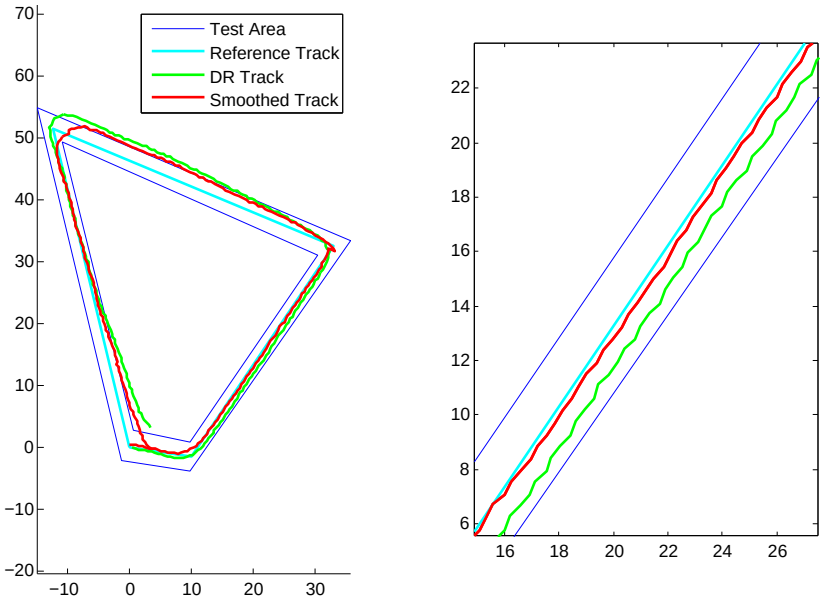

Figure 5.7: Smoothed trajectory

In figure 5.7 the trajectory estimate is depicted on the left. The filter is able to slightly reduce the overall heading deviation. On the right-hand side, a section of the resulting trace is magnified. Here, the smoothing effect of the filter is visible more clearly.

It is important to note that this approach only yields acceptable results if the expected trajectory is mainly composed of straight walking patterns and when there is a reasonable amount of position fixing data available.

5.1.2.2 Map matching

One of the most powerful methods for correcting noisy measurements is checking for plausibility with regard to a digital map. Combined with a motion model, this approach makes outlier detection very efficient.
In figure 5.8, the track estimation of the relative positioning method has been matched to the underlying digital map. In this way, the impact of heading deviations can be reduced notably. While the total positioning error when only applying dead reckoning has been 4.8 m (3 cm per meter travelled), the map matching approach further reduces this deviation to 1.58 m after 153 m distance, which results in an average error of 1 cm per meter covered. However, it is important to note that this average error is not distributed equally.

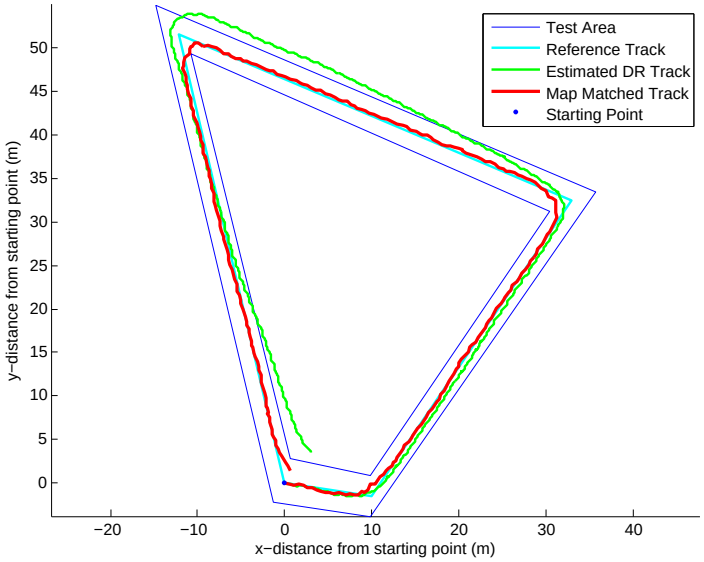

Figure 5.8: Map Matched DR track

5.1.2.3 WiFi integration

When inspecting figure 5.8 of the preceeding section more closely, it becomes obvious that the average accuracy strongly depends on the number of turns in the test trajectory. Since a heading deviation either results in an early or late turn at a corner point, the filtered result tends to adapt to the boundary. The impact of this problem can be reduced by integrating data from the position fixing components.
In figure 5.9 the map matched trajectory degeneration because of high initial heading deviation at the lower and upper right corner points are visible. In contrast to this, due to particle weight adaption to a reference WiFi signal, the map matching estimation follows the reference track at the meter level in the upper left corner. Figure 5.10 shows a magnification of the upper left corner and the distribution of the particles.
In this plot, the particles are denoted by crosses of different color. The WiFi location estimation algorithm assigns the red particles a higher weight. Green particles are the remaining valid state estimates. The black particles lie beyond the zone of possible state estimates and are therefore discarded.

When using periodic absolute position updates such as WiFi based localization, it is important to note that the overall system performance cannot exceed the accuracy of the position fixing systems. So, in order to prevent outliers to corrupt the whole state estimation, the WiFi positioning system relied on plausibility checks and was operated in transponder mode. That means that RSS measurements were only considered if they corresponded to the reference data. In addition to this, no signal space based relocation was performed.

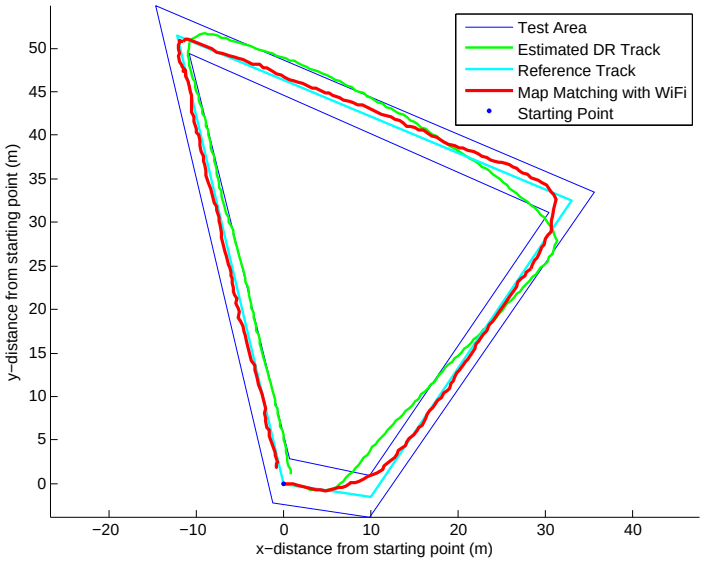

Figure 5.9: Map Matched DR track with WiFi estimate

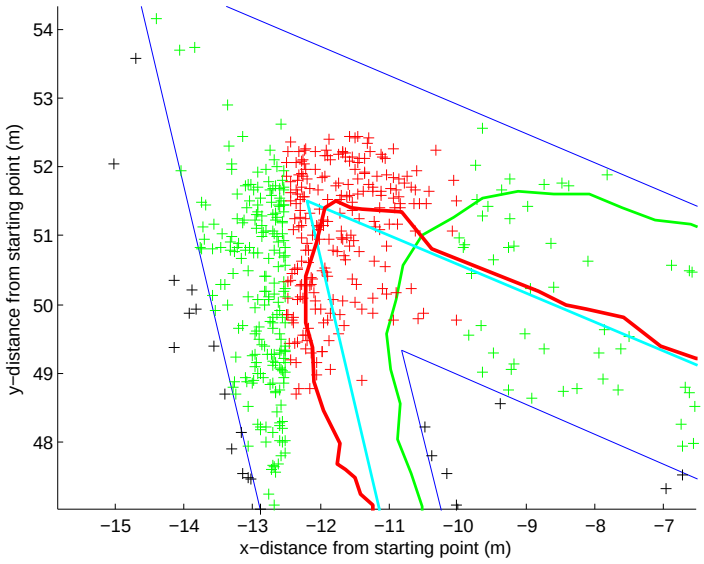

Figure 5.10: Particle distribution in upper left corner

5.2 Test area deployment

In order to assess the practical applicability of the positioning system, a second test in the designated deployment area was conducted. The overall structure of this section is equal to the preceding test. As already described in section 4.1 on page 29, the deployment test area was a 100x200 meter sized part of a busy shopping street in the inner city of Graz. An aerial

image of the test area is depicted by figure 5.11.

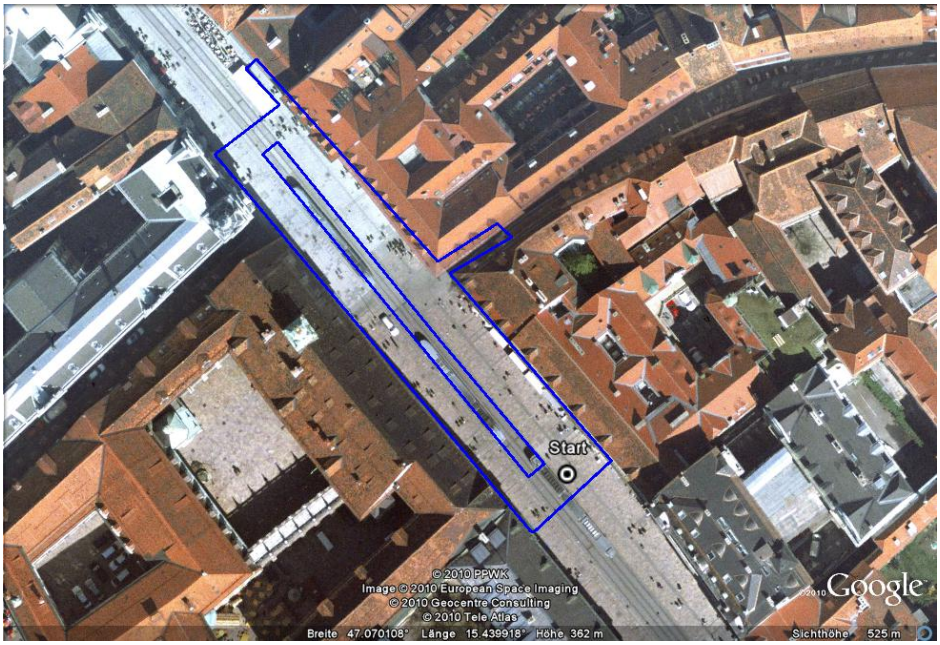

Figure 5.11: System deployment test area

Due to the high number of other pedestrians, frequent changes in velocity and direction are necessary in a real world scenario. Additionally, it cannot be assumed that a person which is interested in the contents of specific shopping windows, sticks to an overall average speed.

Therefore, a test trajectory was designed which accounts for both evasive actions and longer periods of slow walking next to shopping windows. Figure 5.12 shows the reference trace. In order to detect deviations from the reference path instantly, the test person was instructed to move over the pre-mapped reference points periodically. For reasons of clarity, only every odd source point is displayed.

Between points 1 and 5 avoidance of a large group of people is simulated. After that, a 35 meter long walk with minor directional changes is performed up until point 7. Up until this location, the walking speed was constant at approximately 1.2 meters per second. Following that, the trajectory is complemented by a curved short term visit of an adjacent street at slow speed (0.9 meters per second). Between points 11 and 15, a group of people is avoided again at normal walking speed. Then, a stage of very slow movement along the shopping windows is started. The average velocity at this position amounts to 0.5 meters per second. In addition to this, the test user was instructed to stop arbitrarily up until point 19.

At the upper end of the evaluation area, the test person turned and resumed normal walking speed again. At point 21, the user stopped again and traversed the tramway rail lines with high walking speed (1.5 meters per second). This speed was maintained up until point

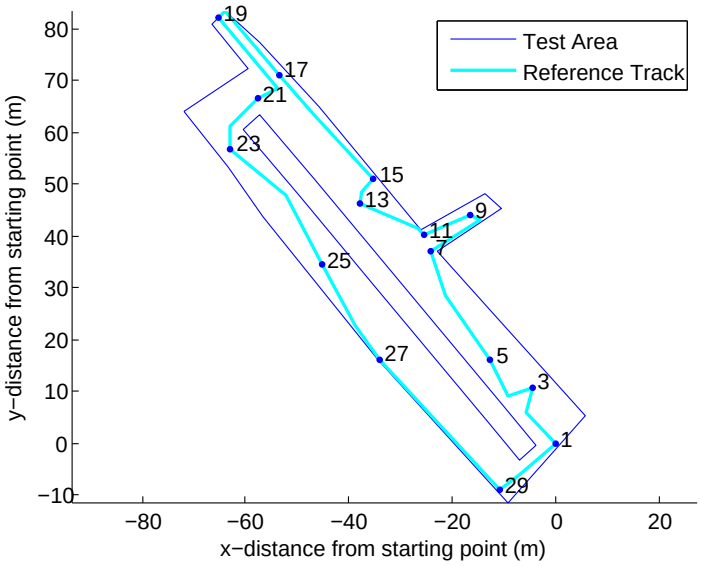

Figure 5.12: Deployment test - Area reference trajectory

27, where again a stage of slow walking with arbitrary stops was performed. After having traversed the distance between points 27 and 29, the track was closed at normal walking speed of 1.2 meters per second. The total length of the traversed path amounts to 266 meters.

5.2.1 Accuracy tests

In the following sections, the performance of the various sensor components in single operation mode is shown. The section begins with the results in accuracy of the dead reckoning subsystem and concludes with a performance evaluation of both the WiFi and GNSS based positiong approach.

5.2.1.1 Dead Reckoning

In the first test, the heading and stride detection capability by using inertial sensors only is evaluated. The results of are shown in figure 5.13.
In spite of the sharp changes in direction and the velocity variations, the resulting trajectory shows a good estimation result between segments 1 to 11. The maximum deviation in this part of the track amounts to 1.4 meters. However, starting at point 13, the resulting track suffers from a clearly visible heading deviation, which results in a 7 meter difference from the reference. The decrease of accuracy in that part of the track could be explained by the close vicinity to shopping windows, which are apparently responsible for local distortions of the earth's magnetic field.

A similarly strong heading deviation can be identified between points 23 to 27: While the slight bearing change to the left after point 23 is still recognized, the estimated heading vector at 25 does not point as strongly to the right again. In that case, magnetic distortions

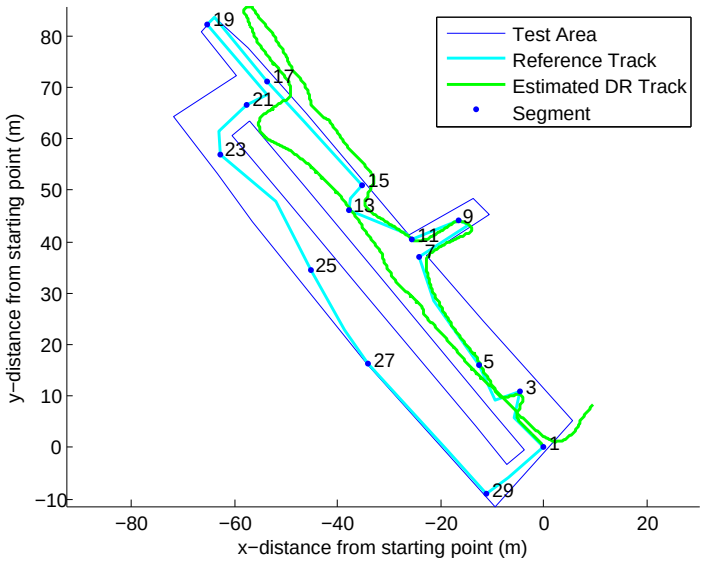

Figure 5.13: Deployment test - Dead reckoning evaluation

induced by a traversing tram probably corrupted the orientation estimate.

Despite of the good short term performance of the inertial measurement component, this result clearly shows the necessity of a periodic position update. The used methods will be shown in the next two sections.

5.2.1.2 WiFi positioning

As already described in section 4.1 on page 29, the signal strength of nearby WiFi access points were recorded at the location of the pre-mapped reference points. In that way, a backup position fixing system for GPS outtakes has been established. Despite of the higher number of reference measurements, an averaging interval of at least 8 seconds for useable accuracy was necessary. Consequently, the WiFi measurements' reliability depends on the current speed of the test user. When keeping in mind, that the dead reckoning component has the highest drifts at already known locations, the number of WiFi measurement integrations can be limited, though.
In figure 5.14 the result of the WiFi positioning estimate is shown. As already proposed in section 5.1.1.2, this trajectory has already been filtered for outliers. When inspecting the trace more closely, it becomes clear, that this approach is unable to sense small directional changes. This is visible between points 5 and 9, for example. In addition to this, due to the outlier filtering, some vital track information has been discarded between points 19 and 25.

However, the long term stability of the WiFi positioning approach make it possible to compensate for the drawbacks of the dead reckoning component. A detailed description of the data fusion result is given in section 5.2.2.

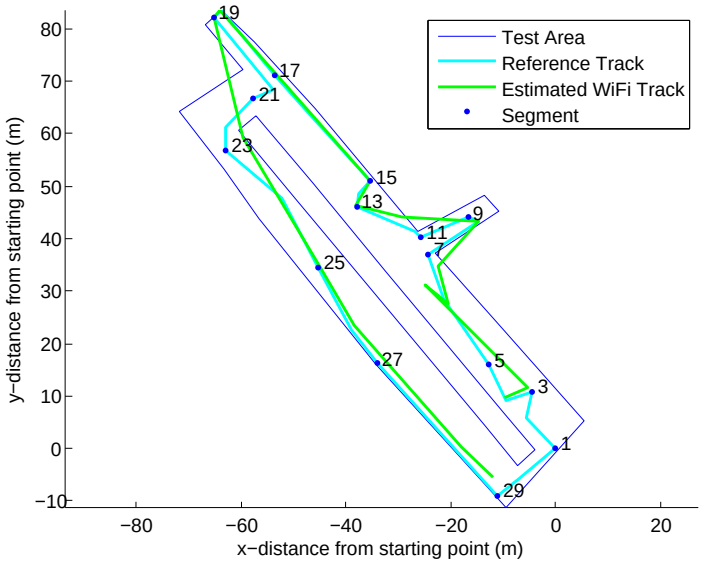

Figure 5.14: Deployment test - WiFi positioning evaluation

5.2.1.3 GNSS measurements

In addition to WiFi positioning system, the performance of GPS in the deployment area was evaluated. The results of this test are shown by figure 5.15. By inspecting the estimated trace, it becomes clear that the GNSS based position method has particularly problems at sections 7 to 11, 15 to 19 and 27 to 29. The first segment mentioned concerns the short visit in the adjacent street. Here, a total loss of signal occurred and therefore no positioning information whatsoever could be inferred. The two other segments mentioned have in common that the user moved directly along shopping windows and a walls respectively. While the view to the sky was blocked by the house facade similarly in both situations, the satellite signal was attenuated even more severely in the first case by awnings mounted on nearby shops.
The accuracy results of this performance evaluation clearly imply the necessity of a position fixing backup to GPS. However, as already outlined in section 4.3.3 on page 43, by incorporating a motion model, plausibility checks for the GNSS position estimate can be given. Additionally, by comparing the localization results of both WiFi and GPS positioning, specific positions can be weighted higher, if they match within a radius of 5 meters.

5.2.2 Sensor fusion

In this section, the track results after fusing the sensor data for the deployment test scenario are described. When considering the amount of necessary directional changes in the test area and the lack of a short term absolute positioning update, it becomes clear that trajectory smoothing, as described in sections 4.3.2.3 on page 41 and 5.1.2.1 on page 56 would be counter productive. Therefore, only map matching with the integration of WiFi and GPS measurements is applied in this section.

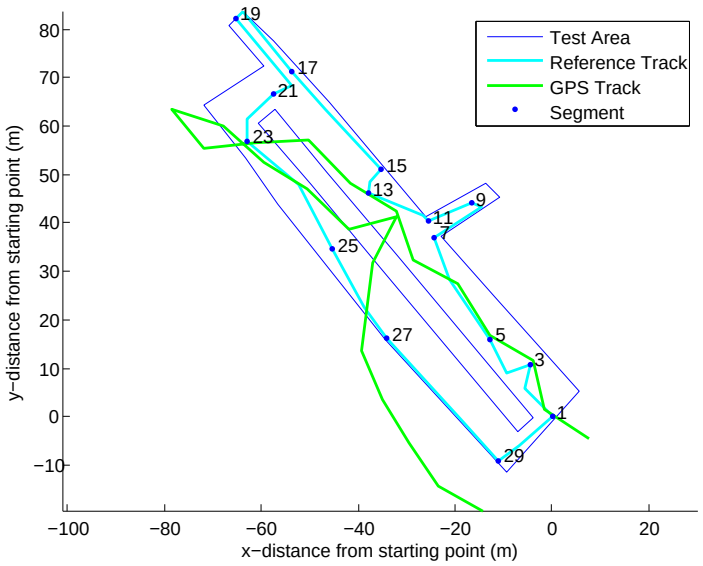

Figure 5.15: Deployment test - GPS positioning evaluation

5.2.2.1 Map Matching

In contrast to the small scale test, the degree of freedom in movement is considerably higher in the deployment area. Therefore, a lower improvement in accuracy when applying map matching only has to be expected.

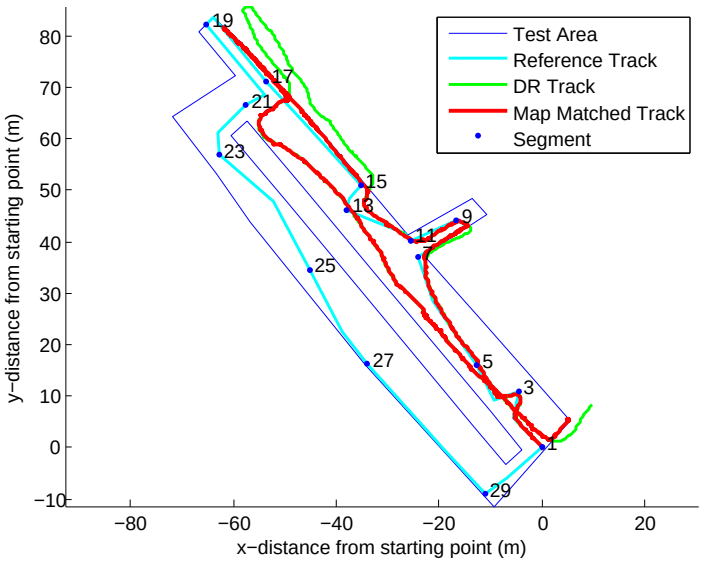

Figure 5.16: Deployment test - Map matching evaluation

In figure 5.16, map matching has been applied to the dead reckoning estimation. The track

shows that as long as the heading deviation stays limited, the estimation mimics the reference track well. In addition to this, the filtering adapts the output trace to the area of possible movement in segment 15 to 19. This adaption comes with a drawback, though: The used particle repositioning approach treats estimations in forbidden movement zones equally. So, the slope shaped trace at positions 17 and 19 degenerated into a straight line. This also replicates the problem from the dead reckoning only positioning: Since the turning point to the left at position 19 is not integrated into the filtered trajectory, the output lacks vertical extent to the left at point 21. Because of the fact that the filter estimates the track at the opposite position of the tramway rails, it actually increases the deviation from the true path between points 21 and 29.

This example clearly shows the limitations of map matching in an environment were movement is not restricted as heavily as, for instance, in the small scale test. In order to compensate for the loss of heading information, data from the position fixing system has to be used. Additionally, a metric, which weights outlying particles according to their distance from the allowed movement zone could also be implemented. However, this is not a trivial task, because it would require the heading to deviate uniformly for a specific section, which cannot necessarily be assumed.

5.2.2.2 Position fixing integration

For the deployment test, a configuration had to be found in which the number of required absolute positioning updates could be limited. In the case of the evaluated shopping street, it soon became obvious that the most problematic sections were those with the largest distortions of the magnetic field. By correcting the position at these sections of the test track, a near optimal replication of the ground truth could be achieved.

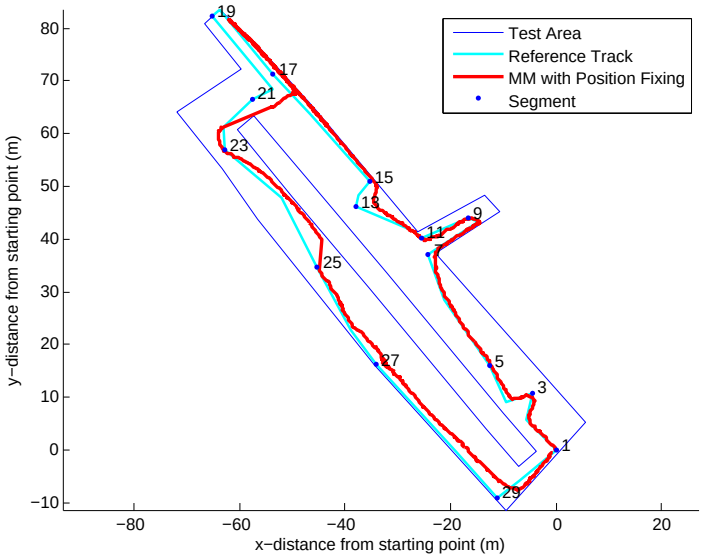

Figure 5.17: Deployment test - Map matching with two position updates

Figure 5.17 shows the resulting map matched trajectory after inclusion of two updates from the position fixing system. These location corrections were made after the trajectory information loss induced by walking along shopping windows in section 15 to 19. Furthermore, the magnetic interference caused by traversing trams between points 23 and 25 has been compensated too. As a result, it was possible to achieve a positioning estimate which stayed below the 5 meter error boundary.

6 Conclusions and Outlook

In this final chapter, the conclusions drawn from this project are outlined. Subsequently, the next necessary steps to enhance the system are described.

6.1 Summary

The experiences made within this project have shown that the proposed approach yields acceptable long term position and accuracy stability for position and attention tracking in urban environments. By utilizing already existing WiFi signals in combination with poor GPS positioning data, a viable alternative for otherwise necessary infrastructural modifications has been demonstrated.

By using an motion model, which was constantly updated through inertial measurements, the essential task of assessing the current quality of data provided by the GNSS and WiFi based absolute positioning methods was handled effectively. The additional matching of the position estimation with an pre-recorded reference map reduced the impact of the inherent drift of the inertial measurement system.

The visualization system developed in the course of this project provides a quick overview of the pedestrian's current position as well as the current field of vision. Due to the geographic referencing of potential points of attention, the positioning system could greatly reduce the number of objects to process for the eye tracker chain. However, in order to minimize the dependency on image processing, a transponder based approach with a range of max. 1 meter per node would be necessary.

In the accuracy tests, the average error for the dead reckoning velocity determination component stayed below 2% of distance travelled. The dominating source of position deviation, however, is posed by the heading detection system. Depending on magnetic field variations, the bearing estimation error was in the range of 1° to 9°.

When comparing the map matching results from both small scale and deployment test, it becomes obvious that the smaller the area of allowed movement is, the greater the benefit of adapting the track estimation to reference boundaries becomes. Particularly if the standard deviation of the used absolute positioning method exceeds the current zone of possible motion, map matching should be applied.

The availability of a digital map and a motion model was also essential for outlier detection of the absolute positioning system. Before checking the returned WiFi and GPS data for plausiblity, false position estimates of up to 40 meters where returned, thus corrupting

the current data fusion filter state. By comparing the estimations of WiFi and GNSS based positioning with each other and additionally checking the returned data against both the motion model and the underlying digital map, an average accuracy of 5 meters could be achieved.

6.2 Further Work

The test results for the velocity determination method have shown that the system still suffers imperfections in step detection. Errors in step counting mainly occur in acceleration phases. The reason for this problem is posed by a constant threshold value used for vertical acceleration value classification. In the next development stage of this project, this threshold will be replaced by a dynamic value.

While conducting the deployment test, it soon became clear that position updates are only needed in sections with strong magnetic disturbances. By pre-mapping local distortions of the Earth's magnetic field in advance, the position fixing system could be optimized to those regions, thus making an uniformly scattered grid of reference points obsolete.

Since the device responsible for heading estimation was also used for step counting, it was necessary to place the sensor at a position which is sensitive for vertical acceleration variations. In this project, the sensor was mounted on the test user's hip. As a consequence, the measured heading was constantly oscillating according to the horizontal movement of the test user's pelvis. This made it difficult for the sensor's internal filter system to stabilize the estimated bearing. Due to the availability of low cost acceleration sensors, it would be reasonable to place a dedicated step detection device directly at the test user's leg. In that way, the sensor responsible for heading detection could be mounted at a position which is not so much affected by orientation changes.

Finally, the proposed WiFi positioning and map matching approach should also be evaluated in an indoor scenario. Disturbances in the magnetic field can be expected to be more severe indoors, however, due to the confined movement space, map matching should generally perform better.

List of Abbreviations

ARM Advanced Risc Machine (former meaning, now a pseudo-acronym)

AMR Anisotropic Magneto Resistance

GCC Gnu Compiler Collection

GNU Gnu's Not Unix (recursive acronym)

CPU Central Processing Unit

CTP Conventional Terrestrial Pole

DMIPS Dhrystone Million Instructions Per Second

DVM Dalvik Virtual Machine

GLONASS GLObal'naya NAvigatsionnaya Sputnikovaya Sistema

GNSS Global Navigation Satellite System

GPS Global Positioning System

LBS Location Based Service

HDOP Horizontal Dilution of Position

JRE Java Runtime Environment

JVM Java Virtual Machine

MEMS Micro MEchanical System

WLAN Wireless Local Area Network

PCA Principle Component Analysis

POI Point Of Interest

POA Point Of Attention

RISC Reduced Instruction Set Computer

RSS Received Signal Strength

SIR Sequence Importance Resampling

SOC System On Chip

TTTF Time To First Fix

UTM Universal Transverse Mercator

WGS84 World Geodetic System 1984

ZUPT Zero Velocity Update

Bibliography

[1] ABIresearch, *Ics, chipsets, modules, technology developments and business factors*, GPS Semiconductors (2009).

[2] Acar C. and Shkel A., *Mems vibratory gyroscopes: Structural approaches to improve robustness*, Springer, October 2008.

[3] Andrejasic M., *Mems accelerometers*, Seminar Project at the University of Ljubljana, Department of physics, 2008.

[4] Asahi Kasei Corporation, *Ak8973 3-axis electronic compass*, 2008.

[5] Beauregard S., *A helmet-mounted pedestrian dead reckoning system*, Proceedings of the Third International Forum on Applied Wearable Computing, 2006.

[6] ———, *Omnidirectional pedestrian navigation for first responders*, Proceedings of the 4th Workshop on Positioning, Navigation and Communication, 2007, pp. 33–36.

[7] Blanke U. and Schiele B., *Sensing location in the pocket*, Adjunct Poster Proceedings of the 2008 Conference on Ubiquitous Computing (UbiComp08), 2008.

[8] BOSCH GmbH, *Bma-150: Digital, triaxial acceleration sensor, data sheet*, May 2008.

[9] Caruso M., *Application of magnetoresistive sensors in navigation systems*, Sensors and Actuators (1997), 15–21.

[10] Chapman F., *The winforms geplugin control library*, Available from `http://code.google.com/p/winforms-geplugin-control-library/`, June 2010.

[11] Chen L., Lee C., Budhiraja A. and Mehra R., *Pflib: an object oriented matlab toolbox for particle filtering*, Proceedings to SPIE Signal Processing, Sensor Fusion, and Target Recognition XVI, vol. 6567, April 2007.

[12] Chen Y. and Kobayashi H., *Signal strength based indoor geolocation*, Proceedings of the 19th annual joint conference of IEEE Computer and Communication societies (Infocom2000), 2002, pp. 436–439.

[13] Cheng Y., Chawathe Y., LaMarca A. and Krumm J., *Accuracy characterization for metropolitan-scale wi-fi localization*, Proceedings of the 3rd international conference on Mobile systems, applications, and services, 2005, pp. 233 – 245.

[14] Cramér H., *Mathematical methods of statistics*, Princeton University Press, 1946.

[15] European Space Agency, *Galileo full deployment and operations*, Available from `http://www.esa.int/esaNA/SEMF5OZCU8E_galileo_0.html`, April 2010.

[16] Evennou F. and Marx F., *Advanced integration of wifi and inertial navigation systems for indoor mobile positioning*, EURASIP Journal on Applied Signal Processing (2006), 164–175.

[17] Foxlin E., *Pedestrian tracking with shoe-mounted inertial sensors*, Computer Graphics and Applications, IEEE **25** (2005), no. 6, 38–46.

[18] Google Inc, *Google earth api*, Available from `http://code.google.com/apis/earth/`, June 2010.

[19] Google Inc., *Nexus one phone - feature overview and technical specifications*, Available from `http://www.google.com/phone/static/en_US-nexusone_tech_specs.html`, January 2010.

[20] Google Mobile Blog, *My location now with wi-fi*, Available from `http://googlemobile.blogspot.com/2008/10/my-location-now-with-wi-fi.html`, October 2008.

[21] GPSPassion, *Gps performance compared : Siratlasiv impresses*, Available from `http://www.gpspassion.com/fr/articles.asp?id=175&page=8`, February 2008.

[22] Hauser H., Hochreiter J., Stangl G. and Chabicovsky R., *Anisotropic magnetoresistance effect field sensors*, no. 216, 2000.

[23] Hofmann-Wellenhof B., Legat K. and Wieser M., *Navigation. principles of positioning and guidance*, SpringerWienNewYork, 2003.

[24] Honeywell Semicondicutors, *Honeywell dead reckoning module datasheet*, Available from `http://www.ssec.honeywell.com/magnetic/datasheets/drm5.pdf`, May 2006.

[25] IEEE 802.11 Working Group, *Official ieee 802.11 working group project timelines*, Available from `http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm`, 2010.

[26] Jackson A., Jonkers Art R.T. and Walker M, *Four centuries of geomagnetic secular variation from historical records*, Papers of a Discussion Meeting held at The Royal Society **358** (2000), no. 1768, 957.

[27] Jiménez A., Seco F., Prieto C. and Guevara J., *A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu*, Proceedings of the 6th IEEE International Symposium on Intelligent Signal Processing, 2009.

[28] Kalman R., *A new approach to linear filtering and prediction problems*, Journal of basic engineering (1960).

[29] Klint J., *Framegrabber: A class for extracting individual frames from a video as bitmaps*, Available from `http://www.codeproject.com/KB/graphics/FrameGrabber.aspx`.

[30] Krach B. and Robertson P., *Integration of foot-mounted inertial sensors into a bayesian location estimation framework*, 2008.

[31] Kunze K., Lukowicz P., Partridge K. and Begole B., *Which way am i facing: Inferring horizontal device orientation from an accelerometer signal*, Proceedings of the 2009 International Symposium on Wearable Computers (ISWC09), 2009, pp. 149–150.

[32] Ladstätter S., *Enhanced positioning system for sensor enriched smartphones*, Master Seminar Project at Graz University of Technology, Institute for Broadband Communication, 2010.

[33] Legat K., *Pedestrian navigation*, Ph.D. thesis, Graz University of Technology, 2002.

[34] Leppakoski H., Kappi J., Syriainne J. and Takala J., *Error analysis of step length estimation in pedestrian dead reckoning*, Proceedings of the 2002 Institute of Nagiation GPS Meeting, 2002, pp. 1136–1142.

[35] Levi R. and Judd T., *Dead reckoning navigational system using accelerometer to measure foot impacts*, United States Patent 5583776, 1996.

[36] Macheiner K., *Performance analysis of a commercial multi-sensor pedestrian navigation system*, Master's thesis, Graz University of Technology, 2004.

[37] Microsoft Corporation, *Directshow documentation*, Available from `http://msdn.microsoft.com/en-us/library/ms783323.aspx`, 2010.

[38] Putney C. and Rasmussen N., *Inssider*, Available from `http://www.metageek.net/products/inssider`, March 2010.

[39] Qualcomm Inc, *gpsone technical specifications*, Available from `http://www.qualcomm.com/products_services/mobile_content_services/location/gpsone.html`, 2010.

[40] Rappaport T., *Wireless communications: Principles and practice*, Prentice Hall, 2002.

[41] Ristic B., Arulampalam S. and Gordon N., *Beyond the kalman filter: Particle filters for tracking applications*, Artech House Publishers, Februrary 2004.

[42] Roos T., Myllymaki P. and Tirri H., *A statistical modeling approach to location estimation*, vol. 1, IEEE, 2002.

[43] Russian Space Agency, *Glonass constellation status*, Available from `http://www.glonass-ianc.rsa.ru/pls/htmldb/f?p=202:20:389395745467348::NO`, April 2010.

[44] SensorDynamics AG, *Sensordynamics photo service: Mems structure of an accelerometer*, Available from `http://www.sensordynamics.cc/cms/cms.php?pageId=51`, 2010.

[45] Shah Z. et. al, *Particle filters and position tracking in wi-fi networks*, Proceedings of the 63rd IEEE Vehicular Technology Conference, September 2006, pp. 613 – 617.

[46] ———, *Experimental deployment of particle filters in wifi networks*, Proceedings of the 7th International Conference on ICC Communications, August 2007, pp. 4692 – 4697.

[47] Skyhook Wireless, *Skyhook performance*, Available from `http://www.skyhookwireless.com/howitworks/performance.php`, 2010.

[48] Steinhoff U. and Schiele B., *Dead reckoning from the pocket - an experimental study*, Proceedings of the 8th IEEE conference on pervasive computing (Percom2010), 2010.

[49] TechCrunch, *In april, apple ditched google and skyhook in favor of its own location databases*, Available from `http://techcrunch.com/2010/07/29/apple-location/`, July 2010.

[50] Texas Instruments Inc., *Wireless handset solutions guide*, Available from `http://focus.ti.com/pdfs/wtbu/wl1251_1253_prod_bulletin.pdf`, 2008.

[51] The White House, *Statement by the president regarding the united states' decision to stop degrading global positioning system accuracy*, Available from `http://www.ngs.noaa.gov/FGCS/info/sans_SA/docs/statement.html`, May 2000.

[52] Vectronix AG, *Vectronix dead reckoning compass*, Available from `http://quentin.ladetto.ch/DRC_flyer.pdf`, 2005.

[53] Weimann F., *Development of a pedestrian navigation system*, Ph.D. thesis, Graz University of Technology, 2008.

[54] Weinberg H., *Using the adxl202 in pedometer and personal navigation applications*, Available from `http://www.analog.com/static/imported-files/application_notes/513772624AN602.pdf`, 2002.

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

[56] Xie H. and Fedder G., *Integrated microelectromechanical gyroscopes*, Journal of Aerospace Engineering **16** (2003), no. 2.

[57] Xinhua News Agency, *China to have global satellite navigation system by 2015*, Available from `http://news.xinhuanet.com/english/2009-01/19/content_10684404.htm`, January 2009.

[58] Xsens Technologies B.V, *3dof orientation tracker*, Available from `http://www.xsens.com/en/general/mtx`, April 2010.

[59] XVID Development team, *The xvid codec project*, Available from `http://www.xvid.org/Project-Info.46.0.html`, June 2010.

Appendix

Since smart phones equipped with microelectromechanical systems have become increasingly available, it stands to reason to also utilize their internal sensors as backup for GNSS based positioning. So, in addition to this thesis, an evaluation of the positioning performance on a modern smart phone has been conducted [32]. In the following, an excerpt of this project including test platform, fundamentals and measurement results is appended.

Sensor Overview

The test plattform, the Google Nexus One [19], is a last-generation smart phone, with an ARM style central processing unit (CPU) operating on the QSD 8250 Snapdragon chipset of Qualcomm and manufactured by HTC. The hardware environment integrates a number of microelectromechanical systems (MEMS) which are capable of measuring several physical quantities. In addition to this, the platform allows for direct processing of these values on the device itself.

Figure C.1: Test platform: Google Nexus One

In the following sections, the processing environment is described in further detail. Next, the operating system is described briefly. Following that, the capabilities of the built-in MEMS are outlined. Finally, some short notes about communication hardware with respect to positioning methods are given.

Processing environment

The core of the Snapdragon platform named “Scorpion” is a 1-GHz processor operating within a reduced instruction set (RISC) architecture. The set’s version corresponds to ARMv7,

which is also implemented by a variety of other manufacturers. The main features of the architecture are:

- Superscalar, dual issue 1-GHz CPU capable of 2100 Dryhstone million instructions per second (DMIPS)
- 600 MHz clocked Digital Signal Processor (DSP)
- Support for NEON (optimized DSP instruction set)
- VFPv3 floating point instruction set
- Support for MEMS
- 3G and WiFi wireless connectivity

Operating System

The operating system is part of a software stack named Android. Having been started by a company with the same name, it was later acquired by Google. Now, it is being developed further by the Open Handset Alliance . The basis of the system is a modified version of the Linux kernel and provides a Java runtime environment (JRE) capable of running bytecode. The virtual machine responsible for executing the code was particularly optimized for mobile devices. However, compared to traditional Java virtual machines (Java VMs) , the architecture is different. While Java VMs are stack machines, the VM used in Android, Dalvik, is a register-based architecture. In addition to this, it is possible to execute ARM native code which can be compiled with a native development kit (NDK).

Since the operating environment provides a number of Java libraries, which can be handled easily, most software is developed with the Java programming language. The current compliance level is 1.6. Android comes with a compiler toolchain which translates source code into bytecode first and then converts the resulting .class files into the .dex format. As far as mobile devices are concerned, this has several advantages:

- More instances of the DVM possible (beneficial for security reasons)
- Reduction of the instruction set (compared to standard bytecode)
- Swapping of the byte-order possible for certain data sets
- Inline linking of simple datastructures and function libraries
- Short-circuit evaluation

It is important to note that the underlying Class library does not conform with any standard platform profile issued by Sun (Java SE, J2ME, AWT, SWING). Instead, a subset of the Apache Harmony Project is being used. Therefore, porting existing Java projects to Android is not necessarily a trivial task.

For performance critical applications, it is often beneficial to have direct access to the system's core cpu without having a virtual machine as intermediate layer. By using the provided NDK, C-libraries can be precompiled into ARM native code.

In this project, the performance of the DVM was sufficient. As a consequence, the programming language of choice was Java.

MEMS

In this section, a short description of the micromechanical systems utilized by the smart phone is given.

Accelerometer

The accelerometer integrated in the device is a triaxial digital sensor named BMA-150 and manufactured by BOSCH GmbH. The device has the following relevant characteristics:

Parameter	**Min**	**Typ**	**Max**	**Unit**
Acceleration range	−2		2	G
Sensitivity	246	256	266	LSB/g
Output resolution				10 Bit
Refresh rate	2700	3000	3300	Hz
Output noise		0.5		$\frac{\text{mg}}{\sqrt{\text{Hz}}}$

Table C.1: BMA-150 specifications (from [8])

Magnetometer

The magnetic sensor of the device is an electronic compass, which is tilt compensated. This is achieved via measuring the magnetic flux density in three axes according to the Hall-effect. The sensor's relevant characteristics are specified as follows:

Parameter	**Min**	**Typ**	**Max**	**Unit**
Magnetic flux range	−300		+300	mT
Sensitivity	420	600	780	LSB/g
Output resolution				8 Bit
Refresh rate	66	79	100	Hz

Table C.2: AK8973 specifications (from [4])

Heading determination

Heading determination can be done in different ways. If reliable position fixing methods are available, the moving direction can simply be deduced by comparing the differences in received

location updates. Another method is to utilize gyroscopes. If the initial heading is known, the change in angular velocity denotes a direction change. This approach is particularly beneficial in indoor environments, were the earth's magnetic field is heavily warped. However, it is only reliable for a few seconds.

If on the other hand the magnetic field in proximity of the user is homogeneous, *magnetic* north can be determined by measuring the horizontal flux density. If the sensor is not aligned to the horizontal plane, the sensor results have to be adjusted by the tilt angle:

$$\begin{bmatrix} b_{x_{corr}} \\ b_{y_{corr}} \end{bmatrix} = \begin{bmatrix} b_x * \cos(\theta) + b_y * \sin(\phi) * \sin(\theta) - b_z * \cos(\phi) * \sin(\theta) \\ b_y * \cos(\theta) + b_z * \sin(\theta) \end{bmatrix} \tag{C.1}$$

If no dedicated sensor readings for tilt values are available, accelerometer data can be used to simplify relation C.3. As shown in Figure C.2, acceleration measurements in x-direction yield an angle α when incorporating the gravity component g. By reducing α by 90 degrees, the pitch angle θ can be obtained:

$$\theta = \sin^{-1}(\frac{acc_x}{g}) \tag{C.2}$$

Likewise, the roll angle ϕ can be obtained by using accelerometer values from an both to the x- and z-axis orthogonally aligned sensor. When applying C.2 to C.3 and considering trigonometric identities, the tilt compensated magnetic field measurements can be expressed as:

$$\begin{bmatrix} b_{x_{corr}} \\ b_{y_{corr}} \end{bmatrix} = \begin{bmatrix} b_x * \sqrt{1 - acc_{xg}^2} + b_y * acc_{yg} * acc_{xg} + b_z * \sqrt{1 - acc_{yg}^2} * acc_{xg} \\ b_y * \sqrt{1 - acc_{yg}^2} + b_z * acc_{yg} \end{bmatrix} \tag{C.3}$$

The suffix g to the subscript denotes a normalization to gravity: $acc_{xg/yg} = \frac{acc_{x/y}}{g}$

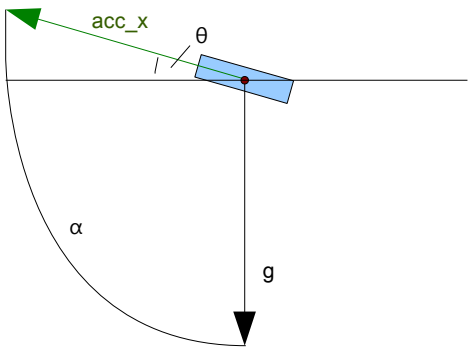

Figure C.2: Deducing pitch angle with accelerometer values

Data Fusion

In this section, the data fusion strategies for the tracking system are outlined.

Overview

The accuracy of the currently acquired measurement data depends on various parameters. E.g.: Positioning data quality from GNSSs will increase with the number of visible satellites. Additionally, deviations in step detection and heading determination have to be taken into account when applying the dead reckoning tracking method. As a result, the data fusion algorithm has to include an amplification / attenuation model.
A Kalman filter proposed by Legat [33], proved to be the best choice for the application platform. If a pedestrian walking pattern is considered, velocity and direction changes happen rather frequently. However, these high order effects are very difficult to model. Therefore, a simple dynamic model with relatively large system noise coefficients is being used.

Implementation details

The state vector x_k consists of the following components:

$$x_k = \left[\begin{array}{cccc} x & y & \dot{x} & \dot{y} \end{array} \right]_k^T \tag{C.4}$$

The time update equation yields:

$$\left[\begin{array}{c} \tilde{x} \\ \tilde{y} \\ \tilde{\dot{x}} \\ \tilde{\dot{y}} \end{array} \right]_{k+1} = \left[\begin{array}{cccc} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} \hat{x} \\ \hat{y} \\ \hat{\dot{x}} \\ \hat{\dot{y}} \end{array} \right] \tag{C.5}$$

The state vector is influenced by the next order time derivates of the independent parameters according to:

$$\Delta x_{k+1} = N_k n_k \tag{C.6}$$

N_k is the noise transition matrix and n_k includes hypothetical distortions of the state vector elements.

$$\left[\begin{array}{c} \Delta x \\ \Delta y \\ \Delta \dot{x} \\ \Delta \dot{y} \end{array} \right] = \left[\begin{array}{cc} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{array} \right] \left[\begin{array}{c} \ddot{x} \\ \ddot{y} \end{array} \right] \tag{C.7}$$

Dead Reckoning

In this section, the observation and error model for the dead reckoning component are described.
The state vector's velocity components are influenced as shown by the following observation equation:

$$\left[\begin{array}{c} \dot{x}_{DR} \\ \dot{y}_{DR} \end{array} \right]_k = L_{step}(k) f_{step}(k) \left[\begin{array}{c} \cos(\alpha_{DR}(k)) \\ \sin(\alpha_{DR}(k)) \end{array} \right] \tag{C.8}$$

where L_{step} is the step length, f_{step} the step frequency and α_{DR} the pedestrian's heading. All values are dependent on the current epoch k.
The error covariance matrix is derived according [33] from (C.8) by applying the laws of error propagation.

$$R_{DR} = \begin{bmatrix} \cos^2(\alpha\sigma_v^2) + v^2 \sin^2(\alpha\sigma_\alpha^2) & \cos(\alpha)\sin(\alpha(\sigma_v^2 - v^2\sigma_\alpha^2)) \\ \cos(\alpha)\sin(\alpha(\sigma_v^2 - v^2\sigma_\alpha^2)) & \sin^2(\alpha\sigma_v^2) + v^2\cos^2(\alpha\sigma_\alpha^2) \end{bmatrix} \tag{C.9}$$

Heading accuracy

To test the static and the dynamic heading accuracy of the devices, measurements were done in standstill as well as in motion. While the accuracy of the e-compass in the outdoor scenario was always within 1°, the deviations are considerably larger in the kinematic case. In addition to this, abrupt directional changes lead to a distorted heading reading as well. This is probably the consequence of acceleration shocks affecting the tilt compensation unit of the magnetometer.

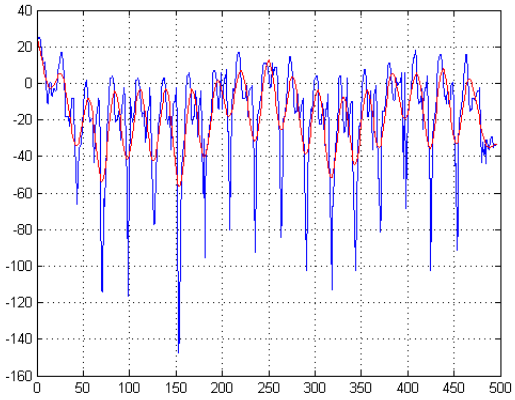

Figure C.3: Oscillation of azimuth angle while walking at a speed of 1 m/sec

Figure C.3 shows the large deviations at normal walking speeds. The test person was instructed to move along a straight path which had a length of 30 meters. When inspecting the illustration, it becomes clear that the heading measurement is far from stable with outliers ranging to up to 120 degrees. However, by low pass filtering the signal (red in Figure C.3) a more accurate representation of the user's direction can be given. The resulting empirical determined values are as follows:

Tracking test

In the tracking test, three different walking patterns were evaluated.

1. normal walking speed with medium distance to facade boundaries (1 – 1.2 m/sec, 3-5 m distance)

σ^2 [°]	v [m/s]
0.6	0
27.0056	0.6
25.7392	1.1
21.8209	1.3

Table C.3: Standard deviations for heading for different walking speeds

2. slow to zero walking speed with low distance to facade boundaries (0.0 – 0.6 m/sec, 0.5 – 3 m distance)

3. fast walking speed with large distance to facade boundaries (1.25 – 1.5 m/sec, >5 m distance)

To assess the dynamic adaptability of the filter, test users were instructed to stop walking at certain points. In addition to this, in Phase 2, the test subjects were told to move slowly with arbitrary stops.

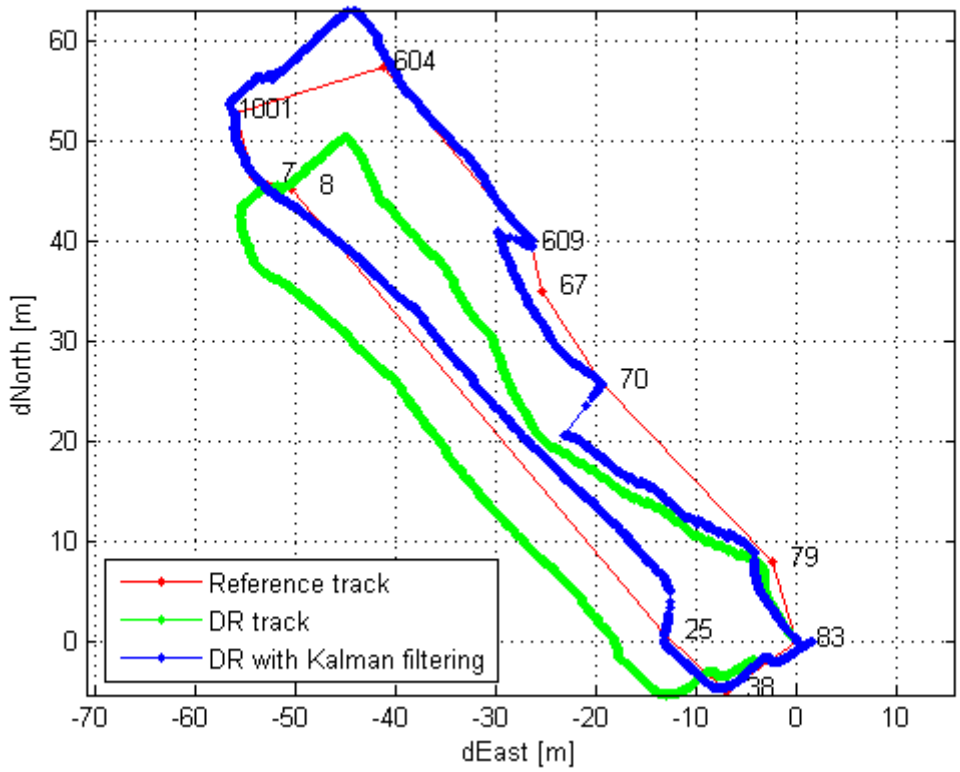

Figure C.4: Results of the tracking test

Figure C.4 shows the performance of the tracking system by comparing the ground truth (red) with both a dead reckoning only estimate and a Kalman filtered solution with position fixing updates. The Kalman filtered track receives position updates at 70, 690, 1001 and 25. Between reference points 79 and 70 (distance 25.5 meters), the large heading deviation can be observed. By specifying the corresponding direction variance, the Kalman filter limits the direction error slightly. For walking speeds higher than 1 m/s step detection and length estimation worked rather reliably with an average error of +/- 3m at a distance of 100 meters.

In contrast to this, velocity determination fails at slow speeds (609 – 604 - 1001). This is explained by the lack of large vertical acceleration variations. The Weinberg expression returns a very low threshold value in these cases. Therefore, the number of counted steps is too high.