A Survey of Selected Indoor Positioning Methods for Smartphones

Pavel Davidson and Robert Piché

***Abstract*—This paper provides an overview of the most significant existing methods for indoor positioning on a contemporary smartphone. The approaches include Wi-Fi and Bluetooth based positioning, magnetic field fingerprinting, map aided navigation using building floor plans, and aiding from self-contained sensors. Wi-Fi and Bluetooth based positioning methods considered in this survey are fingerprint approaches that determine a user's position using a database of radio signal strength measurements that were collected earlier at known locations. Magnetic field fingerprinting can be used in an information fusion algorithm to improve positioning. The map-matching algorithms include application of wall constraints, topological indoor maps, and building geometry for heading correction. Finally, methods for step counting, step length and direction estimation, orientation tracking, motion classification, transit mode detection, and floor change detection in multi-storey buildings are discussed.**

***Index Terms*—Indoor positioning, WLAN, BLE, magnetometer, fingerprinting, map-matching, IMU.**

I. Introduction

Since every smartphone is equipped with GNSS or Assisted GNSS receiver, street navigation with smartphones is ubiquitous. However, there is no single technology that can provide reliable indoor positioning. Among technologies suitable for indoor positioning that potentially can be implemented on smartphones are:

- Use of radio frequency (RF) signals, either those already present such as wireless local area networks (WLAN), or cellular, or those generated by new and dedicated infrastructure (RFID/NFC, Bluetooth).
- Self-contained sensors: tri-axis accelerometer, tri-axis gyroscope, tri-axis magnetometer, barometer
- Indoor map (building floor plan)
- Magnetic field fingerprinting

None of these technologies is a complete solution to all indoor positioning needs on mass-market devices and their advantages and disadvantages depend on the nature of each technology. For example, UWB- and RFID- based solutions require deployment of new infrastructure and addition of new hardware components to smartphones. These additional costs currently limit use of tag-based solutions to Bluetooth Low Energy (BLE).

Typical tasks for indoor navigation with smartphone include locating people and places in public buildings like universities, malls and airports, at sport events, conventions, etc., keeping track of children and elderly people, and automatically tagging posts and pictures. To meet these requirements, positioning accuracy in a typical office building has to be at the room level and the floor has to be correctly identified. In the case of large open spaces such as airports and shopping malls an accuracy of several meters is enough to establish visual contact and find the required place, for example, an airport gate. For some applications like store navigation or a guided tour of museums better positioning accuracy is required. It can be achieved by BLE based positioning when the beacons are densely placed in the required areas.

Currently most of smartphones use WLAN based indoor positioning that usually satisfies the position accuracy requirements in areas rich with WLAN routers. However, the accuracy significantly depends on the density of WLAN routers and on the quality of the radio map which usually deteriorates over time and has to be updated on a regular basis. Improvement of WLAN/BLE based positioning can be achieved if additional sensors and sources of information are used, for example self-contained sensors that improve accuracy and reliability between WLAN based position updates and during gaps in WLAN/BLE coverage. Other commonly used methods to improve positioning are map-matching and magnetic field fingerprinting. They can be fused with WLAN/BLE and sensors to improve position accuracy. Magnetic field based positioning is conceptually similar to terrain navigation systems in military aircraft and submarines, in which altitude or depth are map-matched to estimate the position; in magnetic indoor positioning the features are the local distortions in magnetic field caused mainly by steel in the building structure.

The major components and data flow in a smartphone positioning system are shown in Fig. 1. These technologies meet the requirements for mass market deployment on smartphones, however not necessarily all of these components have to be present in the positioning system. Possible combinations of technologies for indoor navigation include:

- WLAN (map is used for display purpose only)
- WLAN and sensors
- WLAN and indoor map for map-matching
- WLAN, map-matching and sensors

Magnetic field fingerprinting and BLE based positioning can be added to any of these combinations. This paper describes

Manuscript received March 30, 2016; revised August 18, 2016 and November 2, 2016; accepted December 4, 2016. Date of publication December 13, 2016; date of current version May 31, 2017.

The authors are with the Department of Automation Science and Engineering, Tampere University of Technology, 37200 Tampere, Finland (e-mail: pavel.davidson@tut.fi).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Digital Object Identifier 10.1109/COMST.2016.2637663

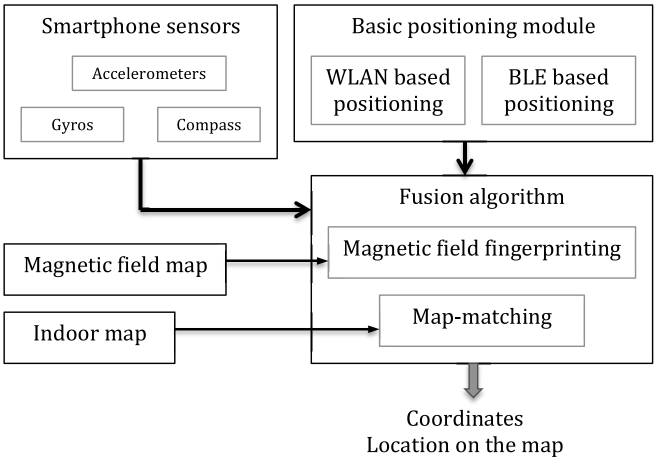

Fig. 1. Major components of a smartphone indoor positioning system.

the technologies and position-estimation methods of these technology components.

The paper is organized as follows. Section II considers WLAN and BLE based positioning systems. Magnetic field fingerprinting is explained in Section III. Map-matching approaches suitable for implementation in indoor environment are presented in Section IV. Section V surveys the use of smartphone sensors to improve positioning and to bridge gaps in WLAN and Bluetooth coverage. Finally, Sections VI and VII summarize this survey of approaches to indoor positioning using smartphones and give recommendations for future research in this area.

II. WLAN and BLE Based Positioning

WLAN access points (AP) broadcast beacon frames, which include the AP's media access control (MAC) address, typically every 100 ms to announce their presence in a certain area [1]. The mobile nodes receive these signals and can identify the AP according to its MAC address. Although WLAN was not designed for positioning, it can be used to estimate user location by exploiting the received signal strength (RSS) measurements. This technology is very attractive for positioning because WLAN APs are readily available in most buildings and every smartphone has WLAN connectivity. However, WLAN positioning requires the generation and maintenance of a radio map, which is a time-consuming and labor-intensive task. Using crowdsourcing for the radio map is difficult because ~~reference position data is usually not available indoors, and so collection of positioning data cannot be done in the background on user devices.~~ Smartphone implementation of WLAN based positioning is usually based on the RSS methods. Since the measured RSS data is very noisy only two-stage approaches that include the preliminary stage of the training data collection can provide required localization accuracy.

This section overviews some established fingerprinting algorithms such as nearest neighbour (NN), k-nearest neighbour (KNN), weighted k-nearest neighbour (WKNN), coverage area and path-loss (PL) which are the most common approaches in RSS based positioning on smartphones (Fig. 2) as well as recently emerging techniques such as state vector machines (SVM), multiple weighted decision trees, compressive sensing, channel state information (CSI) and deep learning.

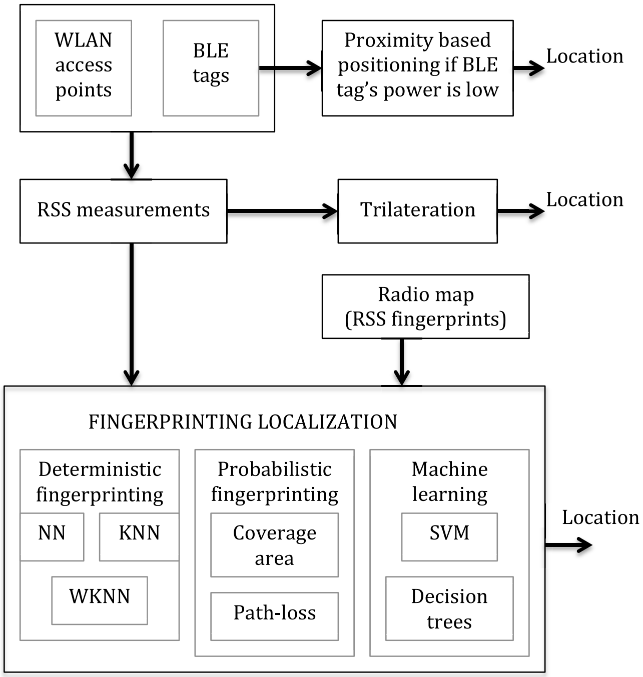
WLAN access points
BLE tags
Proximity based positioning if BLE tag's power is low
Location
RSS measurements
Trilateration
Location
Radio map (RSS fingerprints)
FINGERPRINTING LOCALIZATION
Deterministic fingerprinting
NN
KNN
WKNN
Probabilistic fingerprinting
Coverage area
Path-loss
Machine learning
SVM
Decision trees
Location

Fig. 2. Approaches for RSS based positioning.

A. RSS Based Positioning

A basic approach to positioning is trilateration (Fig. 3). This technique estimates the location of a mobile node by measuring its distances to several reference points. First, the RSS measurements are converted to distance from a mobile node to APs, then the location of a node is computed using the geometry of circles. However, in the case of WLAN based positioning the trilateration method does not give accurate results because of the errors arising when converting RSS measurements to distance. This problem is especially severe in indoor environments where WLAN signals can be reflected by walls and attenuated by people's bodies. In addition, RSS measurements are also affected by the smartphone orientation. The conversion of signal strength to distance is based on a path-loss model such as Eq. 6 which will be discussed in Section II-D2.

An alternative to explicit conversion of signal strength measurement into distance is the comparison of the RSS measurements with a radio map. This pattern recognition approach to localization is called location fingerprinting. There are two stages for location fingerprinting (Fig. 4): offline stage and online stage. In the offline stage, the site is surveyed and the RSS values ("fingerprints") are measured at different calibration points throughout the area of interest and recorded in a database. The online stage is the user positioning, when RSS measurements on the mobile device are compared with recorded fingerprints from the database. The set of "heard"

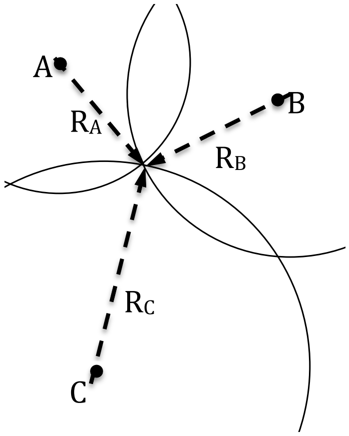

Fig. 3. Trilateration using signal strength (range) measured from three AP.

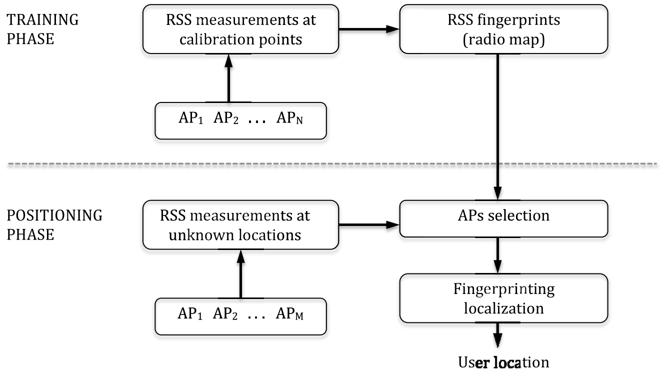
TRAINING PHASE
RSS measurements at calibration points
RSS fingerprints (radio map)
AP1 AP2 ... APN
POSITIONING PHASE
RSS measurements at unknown locations
APs selection
AP1 AP2 ... APM
Fingerprinting localization
User location

Fig. 4. Location fingerprinting approach based on RSS measurements of the signals from WLAN access points and BLE beacons.

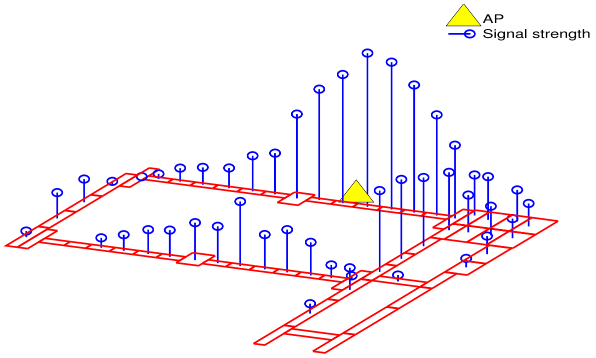

Fig. 5. Example of a radio map. (Reprinted from [4]).

APs at some particular location during the online stage can be different from a set of APs stored in the radio map because some APs can be removed and new APs added or transmitted power settings can change. Therefore, an AP selection process is included in online positioning. The location that best corresponds to the observed signal strengths is recognized as the user location. A challenge with fingerprinting is that the previously recorded RSS values may be quite different from the current values because of changes in the environment and in settings in APs. Removal or addition of new APs in the area of interest would require modifications in the database. The RSS measurements can also be affected by the type and model of mobile device. Despite these limitations, the fingerprinting technique can achieve reasonable accuracy and has become a mainstream approach to WLAN based positioning.

Fingerprinting algorithms can be deterministic or probabilistic. In deterministic matching, one or several survey locations are found that best match the observed RSS values with recorded fingerprints. In probabilistic matching, signal strength values are represented as a probability distribution, and the algorithm calculates the probability of a user's location based on the online measurements and RSS database.

B. Radio Map

The radio map is created as a part of the offline learning phase and it contains the measured RSS values at known locations called calibration points. The advantage of this approach is that the characteristics of the signal propagation in indoor environment can be captured avoiding difficulties of modeling complex signal propagation. However, the data collection required for radio map generation is a time consuming and labor intensive process. The radio map generation starts by dividing the surveyed area into cells based on a floor plan. For each cell, the RSS transmitted by different APs is measured over a certain time period at a calibration point and stored in the radio map database. The i-th element in the radio map can be represented as [2], [3]:

$$M_i = (B_i, \{a_{ij} | j \in N_i\}, \theta_i), \qquad i = 1, \ldots, M, \tag{1}$$

where B_i is the i-th cell centered at i-th calibration point, array a_{ij} consists of the RSS measurements from AP$_j$, N_i is the set of all transmitting APs that can be heard at the ith calibration point, and the parameter vector θ_i contains any additional information that can be useful during positioning phase, for example, antenna orientation, time of day, etc.

The RSS values may be affected by the orientation and location of the smartphone with respect to the user's body during the measurements, device heterogeneity and also by the density of people within the building at this time, since the human body strongly attenuates signal at 2.4 GHz frequencies. The radio map accuracy also degrades over time because of the WLAN APs transmitting power variation and other environmental changes such as position of furniture and walls. Therefore, new fingerprints have to be collected periodically to keep the radio map up to date.

C. Deterministic Fingerprinting Algorithms

Most of the deterministic fingerprinting approaches compute position based on minimization of the Euclidean distance in the RSS measurement space considering all "heard" APs [5]. The user location estimate is chosen to be the calibration point x_j that is closest (Eq. 2), that is

$$\hat{\mathbf{x}} = \underset{x_j}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (r_i - \rho_i(\mathbf{x}_j))^2 \right) \tag{2}$$

where r_i is the measured RSS from the i-th AP during the online positioning phase, $\rho_i(\mathbf{x}_j)$ is the i-th AP's RSS fingerprint measured at the j-th calibration point during the site survey phase.

Examples of deterministic location fingerprinting algorithms include the nearest neighbour algorithm and its generalizations KNN and WKNN [3], [6], [7]. These methods determine the location of the mobile node by comparing the recorded RSS values from the database with the RSS values measured by a mobile node. k calibration points with the most similar RSS value sets are selected. In this approach, k is a design parameter that can be selected to achieve better accuracy. Usually, when the radio map density is high k should be small. For typical cases the optimal k value is about 3-4.

The estimated user location is computed giving equal importance to all k fingerprints. A generalization is to assign different importances to fingerprints, which is known as weighted KNN. The common choice for weighting is to give larger weights to the fingerprints that are closer in RSS space. The nearest neighbour algorithms are non-parametric methods. Their advantages are simplicity, robustness and reasonably good accuracy.

D. Probabilistic Fingerprinting Algorithms

Many approaches apply probabilistic fingerprinting algorithms [5], [8], [9] by estimating the probability distribution of the measured RSS values given the value of the user's location $p(\mathbf{x}|r_1, \ldots, r_n)$. The most likely user's location maximises the posterior probability. Therefore, the problem can be formulated as

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}}(p(\mathbf{x}|r_1, \ldots, r_n)) \tag{3}$$

This so-called posterior distribution of the location distribution is usually computed using Bayes rule:

$$p(\mathbf{x}|r_1, \ldots, r_n) = \frac{p(r_1, \ldots, r_n|\mathbf{x})p(\mathbf{x})}{p(r_1, \ldots, r_n)} \tag{4}$$

The prior distribution $p(\mathbf{x})$ encompasses additional information about personal user profile, for example, tendency towards some particular location [8]. In most publications it is assumed that there is no a priori information about the user position and all locations can be accessed with equal probability, i.e., the prior is uniform. The denominator also does not depend on the location and can be treated as a normalizing constant. In this case the problem is equivalent to the Maximum Likelihood estimation

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}}(p(r_1, \ldots, r_n|\mathbf{x})) \tag{5}$$

The solution of this problem requires knowledge of the likelihood function that describes the distribution of the fingerprint data at all possible locations. The likelihood function can be approximated by histograms or some parametric distributions, for example Gaussian distribution. This leads to non-parametric or parametric probabilistic methods. Parametric or model-based algorithms use models with small numbers of parameters to describe the fingerprint data. Compared to non-parametric methods they have one major advantage – reduced data transmission. Two widely used examples of probabilistic parametric location fingerprinting algorithms are coverage area models and path-loss models [3], [7], [10]. The former approach uses elliptical probability distributions for modeling the area in which an AP's signal can be received. The latter is based on the RSS power loss model calibrated from fingerprinting data.

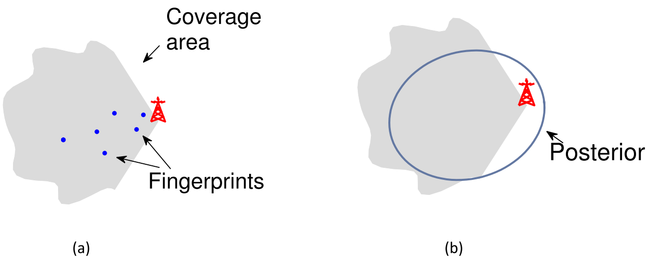

Fig. 6. Estimation of a coverage area from fingerprints. (Reprinted from [11]).

1) Coverage Area Models: A coverage area (CA) estimate is a probabilistic model of the set of locations at which signals from this specific AP were received in the learning phase. It is not a model of the actual geographical CA of WLAN AP (unless the survey points were uniformly distributed), and it does not model the AP location. If fingerprint location density matches user location density (which is reasonable assumption for fingerprints collected by crowdsourcing), the CA estimate models the location probabilities of the users within reception range of the WLAN AP, although the actual reception area can be much larger. The CA method considered in [7], [10], and [12] ignores the specific RSS values corresponding to IDs of observed APs. Therefore, it is more robust to changes in the radio environment than fingerprinting methods that use these values. However, CA methods have lower accuracy compared with nonparametric fingerprinting methods such as KNN and WKNN that record corresponding RSS values, and use them during positioning.

Fig. 6a conceptually shows the actual CA from WLAN access point, and the fingerprints collected during training phase. The CA is approximated by an ellipsoidal probability density shown in Fig. 6b. The distribution's location and shape parameters are computed from fingerprints using Bayesian estimation, which combines the prior information and information given by the data [12]. Given the coverage areas of all APs that can be "heard", the user position can be computed as a weighted average of the ellipsoid centers, with weights determined by the ellipsoid shape parameters, as shown in Fig. 7.

2) Path-Loss Models: PL models describe the RSS as a function of distance from the AP. In the simplest models the PL depends only on the transmission power and distance from AP. A standard assumption is that the signal attenuates logarithmically in terms of the distance between the AP and the user. A PL model typically contains several parameters that have to be determined before the positioning phase. A commonly used PL model [13] is

$$RSS = A - 10n\log_{10}\|\mathbf{x} - \mathbf{m}\| + w \tag{6}$$

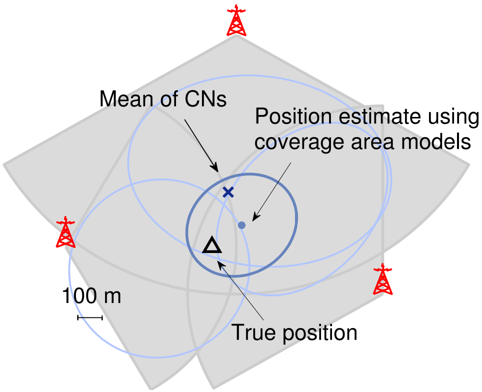

Fig. 7. User position estimation using coverage areas. (Reprinted from [11]).

where A is apparent transmission power, n is a parameter describing attenuation properties of the environment, $\mathbf{m}$ is AP location, $\mathbf{x}$ is a mobile device's location, and w is a zero-mean Gaussian random variable used for modeling the shadow fading. Positioning based on PL model has two phases: an offline learning phase and online positioning phase. In the learning phase RSS values are collected at known locations. Then unknown parameters A, n, $\mathbf{m}$ are inferred for each AP from the posterior probability distribution [14]

$$p(A, n, \mathbf{m}|RSS_{1:n}) \propto p(RSS_{1:n}|A, n, \mathbf{m})p(A, n, \mathbf{m}). \qquad (7)$$

Nurminen *et al.* [14] estimated AP position and PL model parameters simultaneously using the Iterative Reweighed Least Squares method and Bayesian approach, which permits updating AP position estimate and PL model parameters as new fingerprint data becomes available. During positioning phase the user position is inferred from the RSS that can be detected from several APs according to the following equations

$$\begin{aligned} &RSS_i|\mathbf{x}, A_j, n_j, \mathbf{m}_j \sim N\left(A_j - 10n_j \log_{10}\left\|\mathbf{x} - \mathbf{m}_j\right\|, \Sigma\right) \\ &p(\mathbf{x}, A_{1:k}, n_{1:k}, \mathbf{m}_{1:k}|RSS_{1:k}) \propto p(RSS_{1:k}|\mathbf{x}, A_{1:k}, n_{1:k}, \mathbf{m}_{1:k}) \\ &p(\mathbf{x}, A, n, \mathbf{m})p(\mathbf{x}|RSS_{1:k}) = \int p(\mathbf{x}, A_{1:k}, n_{1:k}, \mathbf{m}_{1:k}|RSS_{1:k}) \\ &\qquad\qquad \times dA_{1:k}dn_{1:k}d\mathbf{m}_{1:k} \end{aligned} \qquad (8)$$

where A_j is apparent transmission power from j-th AP, n_j is a parameter describing attenuation properties of the environment, $\mathbf{m_j}$ is j-th AP location, $\mathbf{x}$ is a mobile device's location, and Σ is a covariance of RSS measurement noise. From RSS measurements and PL models the distances between a set of reference nodes and the target node are estimated, which then enables estimation of the user's position. However, the position estimate is sensitive to signal noise and PL model parameter uncertainties, which are large because the distance-power gradient is relatively small. Consequently, these estimates are generally less accurate than radio signal based estimates that are derived using angle of arrival or time delay measurements.

E. Application of Machine Learning Methods and Compressive Sensing in Fingerprint Positioning

Support vector machines (SVM) is a technique for data classification, statistical analysis and machine learning that was recently applied for RSS fingerprinting positioning [15], [16]. The SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other. The best hyperplane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the parallel to the hyperplane that has no interior data points. Classification of the fingerprint database can obtain optimal nearest neighbor points. Wu *et al.* [15] mentioned that when SVM is applied to RSS fingerprinting localization the inference model has to be built in advance. Therefore, SVM does not solve the biggest disadvantage of traditional RSS fingerprinting related to adaptivity to changes in the environment. Feng *et al.* [16] proposed to use the SVM and bilinear median interpolation method to reduce the calibration effort on creating RSS fingerprint map while retaining the required positioning accuracy.

Sánchez-Rodríguez *et al.* [17] proposed an indoor localization system based on multiple weighted decision trees. In addition to the RSS from different APs, the training dataset includes the device orientation because the RSS is significantly influenced by the different device orientations. This approach is computationally efficient and can be implemented on a resource-constrained mobile devices. This localization algorithm is about 100 times faster than well known Horus [9] RSS fingerprinting positioning algorithms.

Feng *et al.* [18] applied the compressive sensing approach for the radio map reconstruction using only a subset of RSS fingerprints, thus significantly reducing the number of measurements during fingerprinting database update. Compressive sensing is a signal processing algorithm for sparse signals recovery from a small number of noisy measurements by solving an L_1-minimization problem. Their location estimator consists of a coarse localizer, where the RSS is compared to a number of clusters to detect in which cluster the node is located, followed by a fine localization step, using the theory of compressive sensing, to further refine the location estimation.

F. Difficulties With WLAN Based Positioning System Deployment

The accuracy of WLAN RSS-based positing systems can be seriously affected by attenuation of WLAN signals by people, heterogeneous mobile devices and their pose. The effectiveness of WLAN based fingerprinting system deployment is characterized mainly by the required time and efforts for site survey and maintenance of fingerprint database to keep it up-to-date.

1) Devices Heterogeneity and Orientation: RSS measurements made by different commodity smartphones can vary significantly even under the same wireless conditions since these devices are usually equipped with different WLAN chipsets and antennas. Viel and Asplund [19] observed the difference in RSS of up to 12 dBm from same signal measured by different smartphones. According to He and Chan [20] the major factors that affect the RSS measurement on smartphones are WLAN chipset sensitivity, antenna installation position and smartphone's operating system. Therefore, a

WLAN based positioning system that relies on RSS fingerprints generally does not perform well across heterogeneous devices.

Hossain *et al.* [21] presented new approach for RSS indicator calibration on heterogeneous devices that is robust to mobile device hardware heterogeneity. The approach is based on a new location fingerprint, namely, signal strength difference (SSD) that is free from hardware-specific parameters, such as antenna gain and transmitter/receiver power gain. The algorithm selects one AP measurement which will be deducted from all other RSS measurements. Therefore, the constant factor of antenna gain is deducted. However, this approach suffers from RSS noise fluctuations. The SSD fingerprinting is also applicable to BLE based positioning.

There are also approaches for offline calibration of heterogeneous smartphones. Laoudias *et al.* [22] noticed that RSS measurements on different devices are correlated and can be approximated by a linear model. They proposed to build a linear regression for RSS measurements on two different devices. The disadvantage of this method is that it requires offline training before an accurate mapping relationship can be learned.

Chapre *et al.* [23] measured RSS on a stationary laptop from 13 APs in a typical office environment. The data was collected for 4 directions such as east, west, north and south during 24 hours at a fixed location. The results show that the RSS variation for the different directions is on average 3 dBm. Rosa *et al.* [24] conducted similar experiments on smartphones and also reported that RSS measurements depend on the specific smartphone's orientation with respect to the AP. The RSS variation for different directions depends also on the type of smartphone. It was 3.6 dBm for Nokia N900 and 7.6 dBm for iPhone 4.

2) Attenuation of WLAN Signals by People: Rosa *et al.* [25] studied the body-loss effect on the RSS measurements on smartphones. The experimental results show that the signal strength at a given location varies by up to 5 dBm depending on the direction that the user is facing because the user's body creates a systematic source of error in RSS measurements. Correcting this source of error during positioning phase can be difficult. Hand grip is another source of error in the RSS measurements on smartphones because of close proximity of hand to the antenna. Rosa *et al.* [25] made RSS measurements on a smartphone in the same environment for two different cases: (a) the hand is not covering the WLAN antenna and (b) the antenna is fully covered. The test results show that the RSS error because of hand grip can reach 12 dBm that corresponds to approximately 9 m of range error at the distance of 3 m from the AP.

3) AP Selection: Appropriate AP selection can reduce the computational burden and in some cases improve the WLAN based positioning accuracy. Youssef *et al.* [26] proposed to reduce computational cost by choosing only a subset of all "heard" APs with the strongest signal. In this case the position is computed using only information from the selected APs and other APs are discarded.

Fang and Lin [27] developed another approach for reduction of fingerprint dimension transforming RSS to a new set of variables that is comprised of principal components (PC). In this way the information for different APs is reorganized and squeezed into lower dimension by a linear transformation. The work also presents an algorithm for determining the number of PCs required to obtain good performance and make sure that the discarded information is in fact the redundant noise. He and Chan [20] assumed that a major concern with fingerprint dimension reduction proposed in [27] is that the selected APs and their weights derived from the offline training samples may be not optimal during online positioning because the environment dynamically changes. The importance of APs in online localization can be different if the set training samples is too small or outdated.

G. Positioning Accuracy of WLAN Based Approach

The positioning accuracy of WLAN based approaches depends mainly on the following factors: density of the WLAN APs, resolution of the radio map, degradation of radio map accuracy with time, and positioning algorithm. Therefore, the positioning performance of the same algorithms can be very different at different sites, or even in the same building if the radio map has not been updated for long time and collected fingerprint data does not represent well actual signals. Müller *et al.* [7] evaluated major algorithms for WLAN based positioning by analyzing their positioning accuracy during field tests in a university campus that represents a typical office building densely covered by WLAN APs. The goal of these experiments was to compare the performance of different fingerprinting algorithms: WKNN, coverage area (CA), path-loss (PL), and generalized Gaussian mixture (GGM) approximation of the PL model. The major effects of environmental changes such as density of APs and radio map accuracy degradation were also evaluated.

An example of estimated user position for three parametric fingerprinting algorithms and an up-to-date radio map is shown in Fig. 8. The actual user location is shown by a black line and the recorded fingerprints are depicted as grey dots. The position accuracy was slightly better for PL and GGM algorithms: 50% of time the error was below 8 m, compare to CA algorithm for which the error was about 10 m. However, CA algorithm requires 30-50 times less computational effort compare to PL and GGM. Non-parametric WKNN also showed good performance with position error below 6 m 50% of time. When 90% of access points were dropped the position error of CA algorithm slightly increased to about 12 m (50% of time). The GGM algorithm's position error was also about 10 m, and PL algorithm's position error was about 15 m. However, the performance of WKNN algorithm deteriorated significantly exhibiting the largest position errors among all of these algorithms. Finally, the performance of these algorithms was also compared when the fingerprints data was not up-to-date: data for generating the radio maps and data for positioning were collected with a time gap of several months to evaluate the performance degradation over time. All parametric methods preserved their positioning performances, while the WKNN algorithm's performance degraded significantly [7].

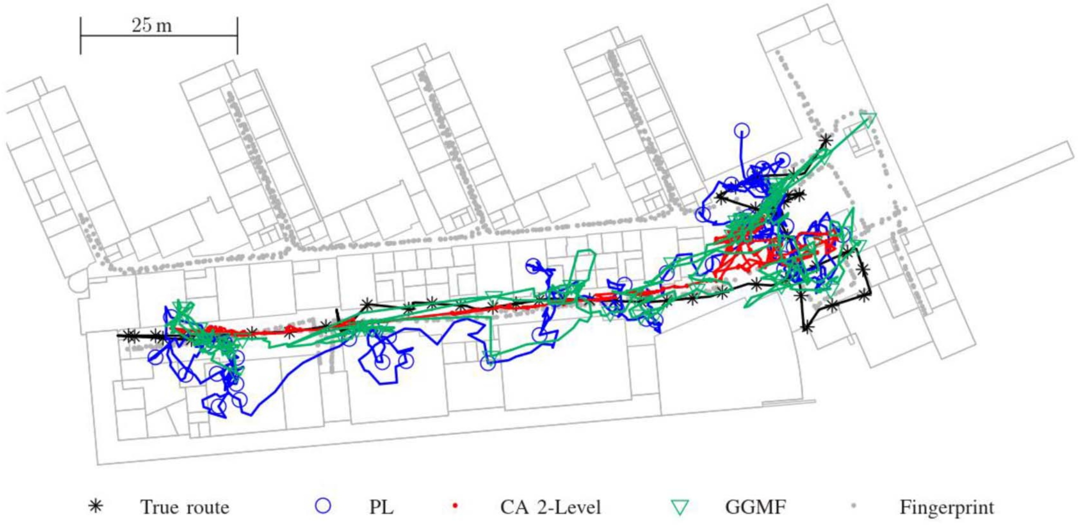

Fig. 8. Field test results for three commonly used WLAN based positioning algorithms. (Reprinted from [7]).

H. *Future Trends in WLAN Based Positioning*

Unlike the RSS values that essentially represent the total received power, the channel state information (CSI) contains information about the channel between sender and receiver at the level of individual data subcarriers for each pair of transmit and receive antennas and reports the amplitude and the phase of each of the subcarriers [28]–[30]. The CSI describes the combined effect of scattering, fading, and power decay with distance. It achieves higher robustness than traditional RSS fingerprinting and it can potentially replace RSS in WLAN based positioning systems. However, the WLAN cards on modern commodity smartphones do not support the CSI data collection. Current prototype systems for CSI data collection are laptop based and use the Intel WiFi Link 5300 network interface card, which is the only commercial tool for CSI data recording based on the 802.11 standard. The example of WLAN fingerprinting positioning systems are given in [31] and [32]. Wang *et al.* [31] proposed a deep learning based fingerprinting scheme that uses CSI information for all the subcarriers from three antennas. Another version of fingerprinting positioning approach is based on calibrated phase information of CSI [32].

I. *BLE Based Positioning*

The emergence of Bluetooth Low Energy (BLE) beacons opens up a new generation of indoor positioning systems that can be more accurate, reliable and efficient. BLE can be used in new applications for a multitude of new markets such as IoT, connected home, health, omnichannel retail, ambient intelligence, augmented reality, and mobile advertising. ABI Research's report, "BLE Tags: The Location of Things" states that total BLE beacon shipments could exceed 400 million units in 2020. The Bluetooth Special Interest Group predicts that by 2018 more than 90 percent of Bluetooth-enabled smartphones will support the BLE standard, so this technology will become ubiquitous.

BLE devices operate in the same 2.4 GHz band as WLAN and Classic Bluetooth 2.1. However, BLE has some special characteristics such as high scan rate, short handshake procedure and very low power consumption that make it different from these technologies. Since BLE beacons are powered by their own batteries they can be placed everywhere, unlike WLAN AP, which are typically placed near power outlets in such a way as to optimize signal coverage with minimal infrastructure deployment. So, another advantage of BLE is in the freedom to place beacons to provide good signal geometry [33].

1) BLE Beacons: At the moment, there are several types available, each with their own standards and advantages. The most popular BLE beacon ecosystems are Apple's iBeacon, Google's URIBeacon and Eddystone, and Radius Networks' AltBeacon. iBeacon reliably works with Apple's iOS and compatible with Google's Android. Other beacons have open source protocol and are compatible with different mobile operating platforms. A BLE beacon is self-contained and very small, roughly the size of a coin and it is in sleep mode most of the time and only wakes up when a connection is initiated [34]. Power consumption is kept low because the actual connection times are only a few milliseconds. The maximum power consumption is 15 mA, and the average power consumption is only about 1μA. A BLE beacon is usually a 1-way transmitter that transmits a 16-24 digit string of characters, which can identify the individual beacon.

2) Positioning With BLE Beacons: Indoor positioning based on BLE can use the same methods as WLAN based positioning, namely, fingerprinting. One difference is that BLE is available in class 1 and class 2 versions, where class 1 has a data transmission range of up to 100 meters in open spaces while class 2 range is about 10 m. Radio maps for BLE can include one additional parameter describing the class of Bluetooth. Zhao *et al.* [35] compared BLE and WLAN based positioning and came to conclusion that BLE is superior for indoor localization for the following reasons: (a) channel hopping mechanism, (b) lower transmission power, and (c) much higher scan rate than WIFi. Frequent channel hopping can average out interference in a given channel or completely remove interference when BLE radios hop to the next channel. This makes the RSS less noisy. The lower transmission power of BLE can reduce the multipath effect in some scenarios. The high scan rate makes it possible to average out the occasional outliers caused by interference or multipath effect and improve the tracking accuracy.

Another difference is that BLE can provide very accurate (submeter-level) positioning in proximity mode when the transmitting power is set to low levels. The proximity is usually detected applying thresholds on RSS measured on a smartphone. To reduce fluctuations in the measured RSS the data can be smoothed, for example, by applying exponentially weighted moving average [36]. Apple and Qualcomm have released frameworks that exploit the proximity mode of BLE to provide "microlocation" and detect the presence/absence of a device within a certain radio range. Based on this, notifications and special offers can be sent to the user's smartphone.

In close proximity (less than one meter) to a beacon the distance estimation based on RSS is very accurate because the signal power decreases as the lognormal attenuation model, which relates the RSS to distance d with the following function [35], [36]:

$$RSS(d) = RSS(d_0) + 10n\log(d/d_0) + X_\sigma \tag{9}$$

where $RSS(d_0)$ is the value of RSS (dBm) at the reference distance d_0, n is the path loss exponent, and X_σ is the zero-mean Gaussian noise with variance σ^2. The path loss exponent has a theoretical value of 2 for unobstructed free space propagation, corresponding to a loss of -20 dB/decade. However, in real environments path loss typically is much higher due to shadowing and multipath interference. Therefore the ranging performance deteriorates with range, and at 10 m the ranging error would be around 5 m [33].

3) Accuracy of BLE Based Positioning: Zhao *et al.* [35] conducted extensive experiments in indoor environments and compared the positioning accuracy of BLE and WLAN in almost identical environment and conditions. The results showed that BLE based positioning is more accurate than WLAN by around 27 percent. The authors also demonstrated that BLE propagation model can describe RSS as function of range better than WLAN. This can be explained by the lower transmitter power of BLE and, as a consequence, smaller range and less multipath. This is the major reason why BLE is better suited for positioning than WLAN and potentially can be more accurate when used in indoor navigation.

Lohan *et al.* [37] also compared positioning accuracy of WLAN and BLE based positioning. In their setup the number of WLAN routers was almost three times larger than the number of BLE beacons. So, the environment and conditions were not identical. The results showed that the positioning accuracy was almost the same for WLAN and BLE. The authors made a general conclusion that the WLAN and BLE signal propagation in multi-floor buildings is very similar, and that a PL model with a floor-loss factor can be successfully employed in order to achieve good floor detection probabilities and low position estimation errors.

J. Summary of WLAN and BLE Based Positioning

Currently, WLAN and/or BLE based positioning is a backbone for indoor navigation with a smartphone. The advantage of this technology is in the wide dissemination of WLAN infrastructure and the growing number of BLE beacons and also the presence of integrated WLAN and BLE radio in almost every smartphone. The main drawback is radio map creation and maintenance. One solution to this problem is to use crowdsourcing. However, how to implement this is still an open problem. Other disadvantages of WLAN based positioning are the low scan rate of WLAN RSS, usually 0.2-0.3 Hz and large errors in RSS measurements due to device heterogeneity, smartphone's orientation, and attenuation of signals by people.

In typical office buildings or shopping malls WLAN based positioning provides an accuracy of about 10 meters, which is not always enough to meet the requirements for LBS services of a room level place recognition and correct floor identification. The BLE positioning can be potentially very accurate with up to one meter position accuracy in proximity mode. However, the BLE beacons are not so ubiquitous as WLAN APs. The WLAN based positioning can be improved if it is aided by other technologies such as map-matching and self-contained sensors.

III. Magnetic Field Fingerprinting

Magnetic field fingerprinting uses a map of magnetic field distribution inside buildings for indoor positioning. Disturbances of the Earth's magnetic field indoors are caused mainly by the metal structure of buildings. The positioning approach consists of two phases [38], [39]: offline mapping of the magnetic field measurements at known locations and online positioning by matching the measured magnetic field with the fingerprints from the database. Unlike WLAN or BLE based positioning, magnetic fingerprinting provides only local positioning because of its spatial ambiguity.

This approach has one important advantage that makes it attractive for indoor positioning: the magnetic field is everywhere, it is relatively stable [40]–[42], and consequently no pre-installed infrastructure is required. Since the output rate for a typical smartphone's magnetometer is about 10 Hz the generation of magnetic field map and positioning can be continuous [43]. However, there are also serious disadvantages: (a) a single fingerprint consists of only a small number of parameters, at most three, but usually two or even one; (b) the magnetic field gradient sometimes can be very steep [40]; (c) intermittent magnetic interferences can be significant.

Measurements of the magnetic field vector come from the smartphone's 3D magnetometer in the device coordinate system. Since the smartphone pose can change freely the knowledge of the transformation between the device and global frames is required. Accurate computation of smartphone's 3D pose is a difficult task because the heading measurements are unreliable when magnetic field disturbances are large. In many real life scenarios the direction of gravity (measured by accelerometers) is the only pose component that can be computed, thus reducing the number of fingerprint parameters to two [44]. If even the direction of the gravity cannot be computed, then only the magnitude of the magnetic field intensity $\|m\|$ can be used as a single fingerprint parameter since magnitude $\|m\|$ is independent of magnetic sensor pose [38]. The small dimension of fingerprints makes the magnetic field fingerprinting unreliable and spatially ambiguous. In addition to this it also sets stringent requirements for fingerprint database creation. The large gradients of magnetic field are advantageous for positioning making possible sub-meter position accuracy. However, it also requires a very high resolution magnetic field map, thus making efficient magnetic field mapping one of the biggest challenges for this approach.

Other challenges of using the magnetic field for positioning with smartphones include diversity of devices and usage scenarios, and significant variability of magnetic field with altitude. Shu *et al.* [42] demonstrated that for the same magnetic field, different devices show different readings of the magnitude of the collected magnetic signals along exactly the same path. Also for the same device, the data collected at different device's altitudes is different. Therefore, to achieve good accuracy the magnetic field map should be three-dimensional. Also during positioning phase the altitude has to be accurately estimated which is impossible in the case of pedestrian navigation with smartphones and usually only correct floor can be identified.

A commonly used approach for incorporating measured magnetic field intensity into a positioning system is to use a statistical model. If magnetic field measurement errors are Gaussian the likelihood function is given by a multivariate normal distribution with mean $h(\mathbf{x})$ and covariance $\mathbf{R}$ [38], [45]

$$p(\mathbf{z}|\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N|\mathbf{R}|}}\exp\left(-\frac{1}{2}(\mathbf{z}-h(\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{z}-h(\mathbf{x}))\right) \tag{10}$$

where $\mathbf{z}$ is the magnetic field intensity measurement, $\mathbf{x}$ is the location, and the function $h(\mathbf{x})$ represents magnetic field map or fingerprints generated during training phase. If the magnitude of the magnetic field intensity is used as an observation, then the multivariate normal distribution in Eq. 10 is replaced by a single variable normal distribution.

The framework for position estimation is similar to terrain navigation with matching of altitude (for aircraft) or depth (for ships) [46]. This approach requires data fusion with a positioning system that produces estimates of the user's location. In the case of indoor pedestrian navigation this can be PDR, WLAN and/or BLE based positioning or any combination of these systems. The dynamic system model including propagation and measurement equations can be defined as [45]

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{u}_t + \mathbf{v}_t; \quad t = 0, 1, 2, \dots \tag{11a}$$

$$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{e}_t; \quad t = 1, 2, 3, \dots \tag{11b}$$

where $\mathbf{u}_t$ is the estimated displacement from the previous position, $\mathbf{x}_t$, at time t, $\mathbf{v}_t$ is the random error associated with the navigation system that provided $\mathbf{x}_t$, $h(\mathbf{x}_t)$ is the function that maps a specific value of $\mathbf{x}_t$ to the measurement domain, and $\mathbf{z}_t$ is the actual measurement collected by the sensor, which has a measurement error, $\mathbf{e}_t$. For the case when the measurement noise $\mathbf{e}_t$ is Gaussian, the estimation algorithm can be the maximum likelihood estimator (MLE) with likelihood function given by Eq. 10. The MLE can be combined with the position estimate from the propagation equation using Bayes rule and the law of total probability. According to the Bayesian view, the posterior probability density function (pdf) $p(\mathbf{x}_{0\dots t}|\mathbf{z}_{1\dots t})$ contains all the statistical information available about the state vector sequence $\mathbf{x}_{0\dots t}$ given the information in the $\mathbf{z}_{1\dots t}$ measurements. If the probabilistic model of the transitional density is described by a Markov process of the first order, then the posterior pdf can be calculated recursively as [47]

$$p(\mathbf{x}_{0\dots t}|\mathbf{z}_{1\dots t}) \propto p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0\dots t-1}|\mathbf{z}_{1\dots t-1}) \tag{12}$$

where $p(\mathbf{z}_t|\mathbf{x}_t)$ is the likelihood of $\mathbf{x}_t$ given the measurement $\mathbf{z}_t$ (Eq. 10), and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the mobility model based on the propagation equation described in Eq. 11a. However, very often the analytical solution for Eq. 12 is not tractable. Therefore, the problem is usually solved numerically by means of a particle filter, where the posterior pdf is approximated by a particle set [48], or by a point mass filter, where the densities are approximated using a grid of point masses [46].

IV. Map-Aided Navigation Indoors

The idea of using building geometry for reduction of position and heading errors in autonomous positioning systems has been extensively exploited in the last several years. In the case of pedestrian indoor navigation, building floor plans represent constraints that restrict movements. For example, people cannot walk through walls and floor changes can occur only via staircases or elevators. The goal of map-aided navigation is to exploit prior information contained in maps or building plans to improve positioning accuracy. The problem is to do in an automated way what our eyes/brains can do so well. When an indoor map is available in addition to a WLAN or BLE based solution it can improve the positioning accuracy. There are currently three approaches to map aided navigation indoors, all of which can be implemented on smartphones: probabilistic map matching based on particle filtering using wall constraints, topological map matching based on link-node representation of a building plan, and reduction of heading error by comparison with building cardinal heading. The purpose of these algorithms is to improve positioning and heading by adjusting the estimated path to the building plan. The major components of an indoor map-matching approach are shown in Fig. 9. In some cases some sensors may not be present in a smartphone. In this case estimations of velocity and heading that are necessary for particle propagation in map-matching algorithms can be obtained from other sources, for example, WLAN based positioning.

A. Map-Matching Indoors Using Wall Constraints

Indoor versions of probabilistic map matching are based on numerical solution of the Bayesian filtering equation using

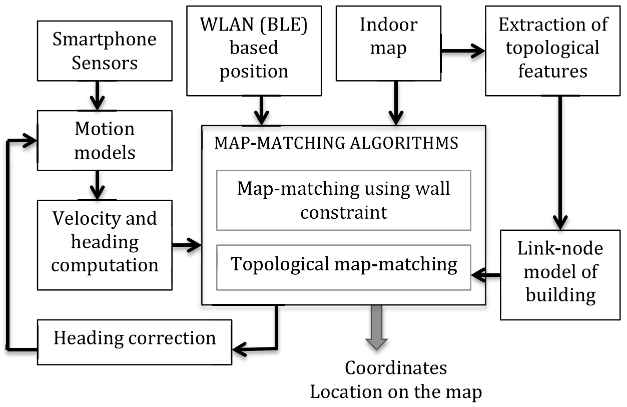
Smartphone Sensors
WLAN (BLE) based position
Indoor map
Extraction of topological features
Motion models
MAP-MATCHING ALGORITHMS
Map-matching using wall constraint
Velocity and heading computation
Topological map-matching
Link-node model of building
Heading correction
Coordinates
Location on the map

Fig. 9. Approaches to map-matching indoors.

algorithms known as particle filters [49]–[55]. These algorithms use a set of weighted random samples (particles) to represent the posterior density of the unknown position in a dynamic state estimation framework. The particles are distributed over the digital building plan where walls represent impassable obstacles. A particle that collides with a wall is excluded from the Monte Carlo simulation.

The goal of the Bayesian filter is to recursively solve for the *posterior* distribution $p(\mathbf{x}_{0\ldots k}|y_{1\ldots k})$, i.e., the conditional distribution of the states $\mathbf{x}_{0,\ldots,k}$ at time instants $0, 1, \ldots, k$ given the sequence of observations $y_1, \ldots, y_k$. In a case of pedestrian navigation it is common to assume that the state is described by a discrete-time Markov process. In other words the state at time step t_k depends on the previous state at time t_{k-1} according to the probabilistic model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The state of the Markov chain usually consists of the pedestrian's horizontal position, heading and, optionally, the floor number: $\mathbf{x}_k = [x_k \; y_k \; \psi_k \; f_k]^T$. Here, the k subscript corresponds to the t_k time instant. The evolution of the horizontal position in time can be described with the aid of a constant velocity model or dead-reckoning equations if velocity and heading are computed using phone's sensors. In many cases, an additive zero-mean Gaussian noise in the speed and heading estimations provides a good approximation. Then the particles can be simply sampled from the transitional prior described by

$$p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}\right) = \mathcal{N}(\mathbf{x}_k; a_k, \Sigma_k). \quad (13)$$

In this formula, the i-th particle is a candidate state vector $\mathbf{x}^{(i)}$ with associated weight $w^{(i)} \in [0, 1]$, the mean a_k for the normal distribution of position and heading which is calculated based on the motion model. The standard deviation Σ_k approximately corresponds to the uncertainty in velocity and heading. However, Σ_k can be larger to accommodate not only the motion model uncertainty, but also the uncertainty induced by the approximate nature of the particle filter algorithm [56].

In partially observable Markov chains the entire state cannot be measured. Usually only stochastic projection of the true state is available. In the case of indoor pedestrian navigation with WLAN or BLE based positioning the application of the map constraint expresses the fact that people cannot move through the building walls. In the proposed map-matching scheme, the map information is incorporated into the particle's weight during update step where the measurement likelihood is defined as

$$p\left(y_k|\mathbf{x}_k^{(i)}\right) = \begin{cases} 0 & \text{if there is a wall between } \mathbf{x}_{k-1}^{(i)} \text{ and } \mathbf{x}_k^{(i)} \\ p\left(y_k|\mathbf{x}_k^{(i)}\right) & \text{otherwise} \end{cases} \quad (14)$$

here $p(y_k|\mathbf{x}_k^{(i)})$ is the measurement model corresponding to the user's location measurement y_k computed by the WLAN or BLE based positioning system.

This problem can be stated as the estimation of the sequence of states $x_{0\ldots k} = \{x_0, \ldots, x_k\}$ in partially observed Markov sequence given a map of the environment, and subject to the motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and measurement model $p(y_k|\mathbf{x}_k)$. The prior probability at t_0, $p(\mathbf{x}_0)$ is assumed to be known. The goal is to find the best trajectory in terms of the minimum mean square error (MMSE) criterion. The required posterior probability distribution can be expressed according to Bayes' law as

$$p(\mathbf{x}_{0\ldots k}|y_{1\ldots k}) = \frac{p(y_k|\mathbf{x}_{0\ldots k}, y_{1\ldots k-1})p(\mathbf{x}_{0\ldots k}|y_{1\ldots k-1})}{p(y_k|y_{1\ldots k-1})} \propto p(y_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{0\ldots k-1}|y_{1\ldots k-1}). \quad (15)$$

In this particular application, a particle filter provides a natural and intuitive way of incorporating building plan information into position estimation, by applying the direct constraint to each particle as was implemented in [49]–[51] and [55]. All these approaches use the same method of building constraints implementation. The differences lie in the particle propagation and measurement update computation. These algorithms also use the smartphone's sensors to compute the particle movement. Each particle is propagated in a procedure analogous to Kalman-type filters and can be represented by the following steps (Fig. 10):

1) Prediction step: The particles are projected to the next time step by sampling from a proposal distribution which can be chosen to be the transitional model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$.
2) Application of wall constraint by reducing a particle weight if it crosses a wall.
3) Update step: Applying measurement update based on proximity of particle's location to the WLAN or BLE computed position. The weights are updated according to

$$w_k^{(i)} \propto w_{k-1}^{(i)} p\left(y_k|\mathbf{x}_k^{(i)}\right). \quad (16)$$

4) Resampling.

Since the particles represent a Monte Carlo approximation of the posterior probability density function expectation, the user location can be estimated as their weighted mean. The user's path estimated in this way is optimal in terms of the minimum mean square error. Note that the path computed this way can cross walls because the particles can be in different parts of the building.

Khider *et al.* [53] used different particle propagation models to increase the robustness of map-matching algorithm,

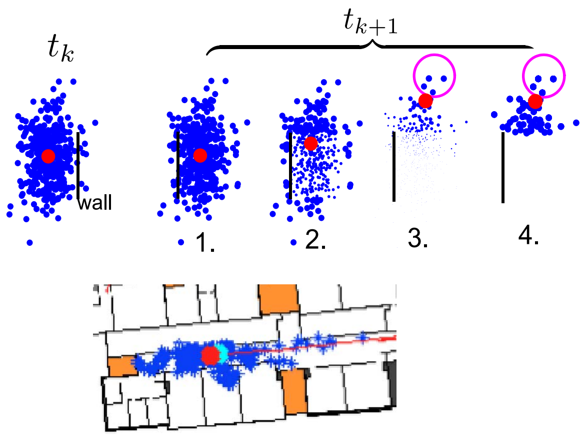

Fig. 10. Application of wall constraints using particle filter. (Reprinted from [57]).

for example, the stochastic behavioral model, the diffusion movement model and map-enhanced combined model. Later the authors also addressed floor changes in [58] adapting diffusion movement models for 3D case. The vertical movement computation module is turned on if the person is detected to be in any of the stairs areas, because in this case the floor change is possible. The authors also proposed an intermediate virtual floor between staircases, which they called x1/2 level, to approximate motion between floors inside the staircase. User motion inside the staircase is represented by an extended Markov model. The new floor level is set depending on the model based information. The vertical speed in the staircase is adjusted depending on the user climbing up or down. In this approach the altitude measurements are not used although they could increase the robustness of the floor transition estimation. Also elevators or ladders for industrial facilities are not addressed.

In [51], [55], and [58] the map-matching algorithm was also extended with 3D movement in multi-storey buildings. The vertical movement can be detected by the smartphone's sensors. To monitor a floor change Ascher *et al.* [55] proposed a new representation of the map with three possible objects:

- Rooms: bounding walls, transitions to other objects
- Stairs: inclined planes with bounding lines and transitions to other objects
- Ladders/elevators: vertical rectangle with bounding lines

Transition between floors is only possible through stairs, ladders and elevators. Vertical movement is detected using barometer and inertial sensors.

B. Topological Map-Matching Indoors

Similar to the street map-matching the topological algorithms for indoor applications use a link-node representation of a building plan [59]. A node is a point defined by its coordinates including altitude. The altitude can be given in terms of a floor number. The nodes correspond to the junctions and to the points of interest in the building. A link is a straight line connecting two consecutive nodes. The links may correspond to the corridors, staircases, entrances to elevators, passages between buildings, etc. The consecutive links are connected and represent a network similar to a road network (Fig. 11), where link-node representation of building is depicted by blue lines and dots. The link-node model can be generated automatically from a building floor plan using, for example, Dijkstra's algorithm [60] or Voronoi-Diagram [61].

However, developing a set of map matching functions for pedestrian navigation is a challenge because the trajectory of people is not always similar to the geometry of the mapping data. Development of algorithms indoors is based on the comparison of topological elements from the trajectory and the database. The goal is to identify the correct link and then location of a user on this link. A topological map matching for indoor applications relies on the similarity of the trajectory geometry and the topological features of the link-node graph. The approach described in [59] is based on Bayesian inference where the estimation is computed considering the traveled distance and azimuth. Spassov [62] used the Fréchet distance as a degree of similarity between two polylines.

Another approach is to use a particle filter in which the particles always stay on links [60], [63]. Propagating the particles on the links imposes an additional constraint on particle movement and allows the use of a smaller number of particles. However, this approach provides an adequate model of pedestrian movement only in places like corridors where the most likely route will be along the corridor. Using a link-node graph it is difficult to model pedestrian movements in large open spaces where people can stay and walk anywhere. In practice, better results can be achieved when a link-node model of building is combined with wall constraints [60]. In this implementation corridors and small rooms are represented by graphs whereas wall constraints are applied for large open spaces.

Hilsenbeck *et al.* [61] proposed to use a grid-based Bayesin filter in which the continuous state space is replaced by a sparse graph structure that represents a link-node model of a building and the posterior probability distribution is computed only in the selected nodes. Their results demonstrate at least tenfold reduction of computational burden compared to state-of-the-art particle filter solutions while ensuring robust localization. Additional reduction of computational complexity comes from a coarse, adaptive quantization of the indoor space, which takes into account typical motion patterns of pedestrians in buildings. The proposed approach is able to reliably track a pedestrian through an indoor environment using only 50 particles to process pedometer measurements in combination with WiFi signal strength measurements. In addition, the algorithm is able to adapt to characteristic gait patterns for individual users. The approach also addressed navigation in open spaces by adaptively constructing the graph with respect to the available space.

The performance of a positioning system based on WLAN and building floor plan is shown in a supplementary downloadable video Clip 1, see supplementary material. On this video the true user's position is shown by a cyan diamond, the WLAN based positioning by black asterisk, and the position estimated as a result of fusion of WLAN with map is shown by a red circle. The particles that are propagated using link-node

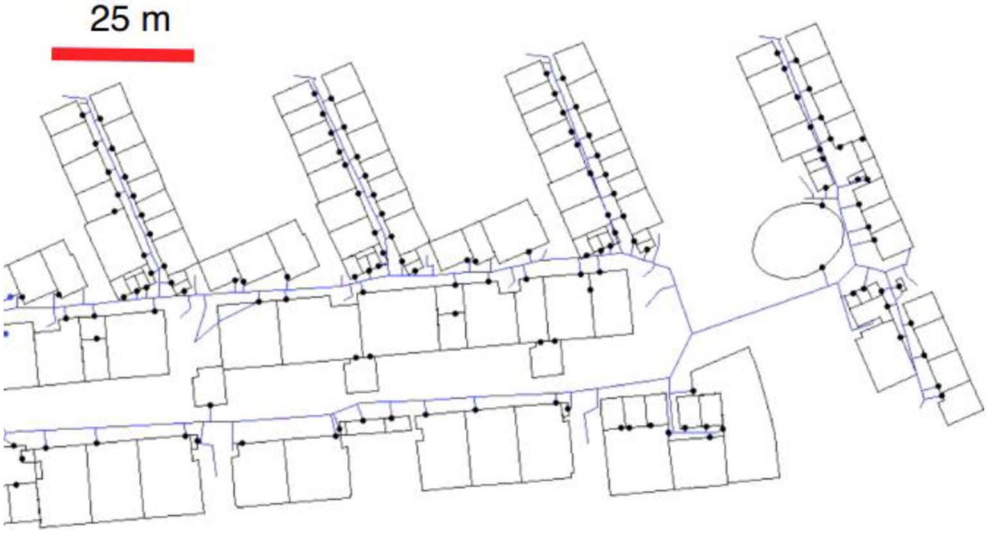

Fig. 11. Link-node representation of a building. Links are shown by blue lines.

model are depicted as blue dots. The particles that are propagated using wall constraint are depicted as red dots. The field tests results indicate that most of time the map-aided positioning is more accurate than WLAN only positioning. Note also that link-node map-matching better preserves the topology: the estimated position follows the true path without crossing the walls and going to the wrong places. Movement between floors is also implemented using the vertical link representing elevators and staircases.

C. Heading Correction Through the Knowledge of the Building Orientation

The algorithms for heading correction with the aid of a map fall into two groups: Algorithms that explicitly generate heading measurements from the knowledge of the building orientation; Algorithms that exploit the information contained in building floor plans through map-matching.

Abdulrahim *et al.* [64], [65] proposed the approach to generate heading measurements from the basic knowledge of the orientation of the building in which the navigation system is operating. The solution is based on the assumption that most buildings are constructed with a rectangular layout where most rooms and corridors are also rectangular, thus constraining the direction in which the user can move throughout the building into one of four principal headings. The disadvantage of these methods is that a user can actually move in a different direction than the "cardinal heading" of the building. Therefore the method may fail in the following cases [64]: (a) a person is walking in circles or curvilinear lines for long period of time, (b) the building does not conform to the simple geometry, and (c) internal rooms and corridors are not parallel to external walls.

Borenstein and Ojeda introduced a method called "Heuristic Drift Reduction" (HDR) [66], [67]. HDR makes use of the fact that many streets or corridors are at least partially rectilinear [66]. At any moment, the HDR method estimates the likelihood that the user is walking along a straight line; if that likelihood is high, HDR applies a correction to the gyro output that would result in a reduction of drift if indeed the user was walking along a straight line. If the algorithm assesses that the user is not walking along a straight line, then HDR does nothing. The limitation of the HDR method is that when not moving straight, at best we can expect HDR to notice that and suspend its operation [67]. During that time, gyroscope drift accumulates and the integration of the rate of turn results in heading errors. Then, when moving straight again, new heading errors are prevented, but those heading errors that were accumulated while HDR was suspended remain in the system and cannot be eliminated. The effect of the HDR method is thus that it reduces heading errors due to drift substantially, but cannot totally prevent the unbounded growth of heading errors.

Jiménez *et al.* [68] analyzed the shortcomings of HDR algorithm, which can even degrade the navigation solution when used in complex buildings where corridors are curvilinear and not aligned to a rectangular layout or where there are large open areas. They proposed a method, called improved Heuristic Drift Elimination (iHDE), that includes a motion analysis block to detect straight-line paths and an adaptive on-line confidence estimator for the heading corrections.

Davidson *et al.* [69] proposed an approach for preventing the unbounded error growth by correcting position and heading errors of autonomous navigation systems operating indoors. Although this method was applied to vehicle navigation it can be extended for pedestrian navigation with a smartphone. The algorithm comprises three steps: (a) the autonomous sensor data is processed to obtain position, velocity, and attitude, (b) map-matching corrections are applied to the trajectory calculated by the dead reckoning system, and (c) the most accurate estimation of vehicle's path is computed as optimal fusion of map-matching and dead reckoning solutions.

D. Summary of Map-Aided Positioning

Two different approaches to the fusion of map data with WLAN based positioning were studied: particle filter with wall constraints, and particle filter with combined representation of building plan (link-node and wall constraint). In both cases the performance is better than performance of only WLAN based positioning. A brief summary of pros and cons for each map-matching approach is given below. Map-matching based on particle filter with wall constraints has the following properties:

- Floor plans are readily available
- Implementation is relatively simple
- The computed user's path can cross walls
- In the case of only WLAN based position estimates, the propagation of particles may be not accurate

Map-matching based on particle filter using link-node representation of buildings has the following properties:

- Node-link model of building has to be obtained
- Implementation is complex, but more stable numerically and requires small amount of particles
- The topology is better followed, however it is not well suited for large open spaces
- Work well even when only WLAN based position estimates are available.

V. Using Smartphone Sensors for Positioning

Inertial and other self-contained sensors can improve smartphone based positioning. However, the accuracy of these sensors in a typical smartphone is low. Therefore, a conventional strap-down inertial navigation approach is not suitable for implementation because of rapid accumulation of position, velocity and attitude errors. An approach to using smartphone sensors for positioning can be useful only if it can cope with the changing orientation of a phone without any constraint on usage scenarios which typically include carrying phone in a pocket, bag, hand or arm-band, making a call, texting, or looking at the display. The major challenge when the phone is not assumed to be constrained is the determination of the heading offset between the direction a smartphone is facing and the direction the user is moving. There are approaches that attempt to solve this problem [70]–[72], however it is still an open research problem. Different transit modes such as pedestrian, elevators and escalators make the sensor based positioning even more complicated. The approaches also have to work on different types of surface and across wide group of people regardless of their height, weight, age, gender, physical condition, etc.

Smartphone sensors can assist in pedestrian positioning by performing the following tasks (Fig. 12): walking parameters estimation including step counting and step length and direction estimation, motion classification, transit mode detection, and floor change detection in multi-storey buildings. These tasks are carried out using a so called pedestrian dead reckoning (PDR) algorithm that computes position by integrating the displacement vectors, which represent steps. If no steps are detected, the system is assumed to be stationary. The PDR's performance is quite robust to the sensor quality, which makes it suitable for implementation on smartphones.

Any PDR system performs the following tasks [73]:

- Step count or step segmentation

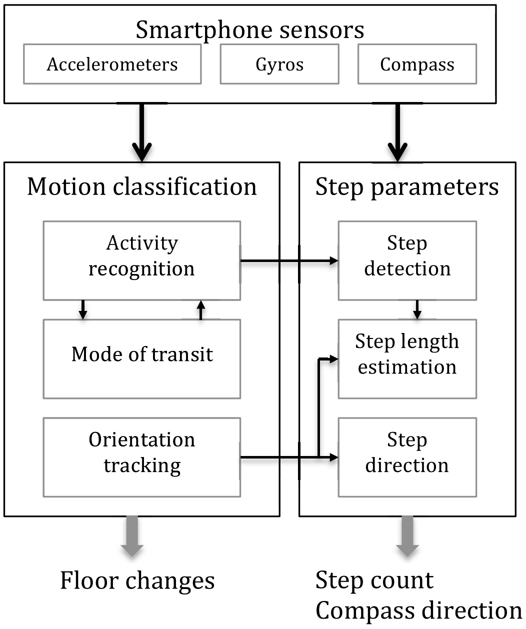

Fig. 12. Major tasks that can be performed using smartphone sensors.

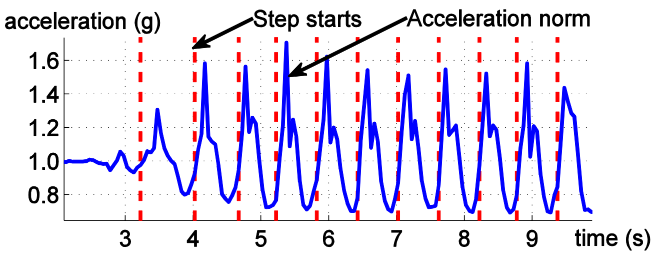

Fig. 13. The norm of the acceleration signal during normal walking as it measured by a torso mounted accelerometer.

- Step length estimation
- Step direction estimation

In the remainder of this section different approaches for implementation of the algorithms shown in Fig. 12 will be described: step detection, step length and direction estimation, orientation tracking, multiple usage scenarios and activity recognition. Special attention will be paid to the methods that do not restrict a smartphone placement and usage: it may be firmly held in place, or free to move from its current position without warning.

A. Walk Detection and Step Counting

Most walk detection and step counting methods are based on the assumption that pedestrian motion has a cyclic nature. The algorithms search for the repeating patterns in accelerometer or gyroscope data during walking (Fig. 13). They can be generally categorized into the following groups: thresholding and peak detection, auto-correlation, and spectral analysis. The pros and cons of different walk detection and step counting approaches are summarized in Table I.

1) Thresholding and Peak Detection: Thresholding and peak detection algorithms are often used together: the walking or running motion is detected applying thresholds and

TABLE I
SUMMARY OF SMARTPHONE-BASED WALK DETECTION AND STEP COUNTING

Method	Advantages	Disadvantages
Threshold on magnitude of the acceleration	Simplicity, low computational burden Robust to the phone's orientation	Walking motion detection may be triggered by any motion with a similar periodicity to walking May not work on unconstrained smartphones
Threshold on vertical acceleration	Simplicity, low computational burden	Requires accurate smartphone's orientation May not work on unconstrained smartphones
Threshold on features computed from windows of accelerometer and gyroscope measurements	In addition to walking other activities can be identified	Optimal threshold value can change significantly across different users, surfaces and movements
Peak detection for step counting	Simplicity, low computational burden	A foot impact may generate multiple local peaks May not work on unconstrained smartphones
Normalized auto-correlation based step counting	Can be used for unconstrained smartphones	Requires learning phase for each user Time lag 0.8-2 sec
Dynamic time warping	Good performance in walk detection and step counting	High computational burden A stride template is formed offline
Short-term Fourier transform	The same algorithm works for walk detection and step counting Can be used for unconstrained smartphones Can distinguish irregular motion from walking	A window length of 5-7 seconds is required to capture cycles in activities such as walking and running
Wavelet decomposition	The same algorithm works for walk detection and step counting Can be used for unconstrained smartphones	The data has to include at least two cycles

then steps are counted by counting peaks [74]. In most cases steps are detected and counted using accelerometers [75]–[83] and, in some cases, gyroscopes [84]. Simple algorithms apply a threshold on some parameters derived from the accelerometer signal. Walking and other activities are identified by comparison of different features with the pre-defined thresholds [74], [85]. The features are computed from windows of accelerometers and gyroscopes measurements, and usually include mean, standard deviation, energy, correlation, frequency-domain entropy, and FFT coefficients [74], [85], [86].

According to [86], a window length of 5-7 seconds is sufficient to capture cycles in activities such as walking, running, window cleaning, or vacuuming. The mean and energy of acceleration sufficed for accurate recognition of these activities. For computation of features in the frequency domain a 512 sample window enables fast computation of FFTs. The energy feature can be calculated in both time and frequency domains. In the time-domain it is computed by squaring the norm of the accelerometer and gyroscope data and summing and normalizing them over a moving window. In the frequency domain it can be computed as the sum of the squared FFT component magnitudes of the signal. The sum is divided by the window length for normalization. Additionally, the DC component of the FFT should be excluded from this sum since the presence of a non-zero DC component can hide important information and reduce the effectiveness of frequency domain estimation techniques [85], [86].

Some algorithms use thresholds on vertical acceleration assuming that sensors are firmly attached and their orientation with respect to the geographical frame is known. The performance of these algorithms can significantly deteriorate if a smartphone is not firmly attached to the body. To resolve the orientation problem the signal magnitude can be used instead. Another difficulty with the threshold approach is that the optimal threshold value can change significantly across different users, surfaces and movements. This method may not work correctly for unconstrained smartphones when the measured acceleration does not match the regular acceleration patterns during walking motion and the thresholds can be triggered mistakenly.

Another useful feature for walking motion detection is zero-crossing [75], [76], [78]. It is related to threshold methods and can be computed based on accelerometer data. The decision on whether a step is being made is done by analyzing the magnitude of the acceleration vector, which can be computed by subtracting the local gravity from the measured magnitude of the specific force. When the sign changes from negative to positive, a new step is counted. The zero-crossing method is based on the cyclic movement of the human body and requires a triaxial accelerometer. Since the norm of the acceleration vector is used the orientation of the sensor unit has no effect on the measurement. This is a popular choice for pedometers or activity monitors due to its simplicity.

The steps can be counted by identifying and counting the peaks (maximum or minimum detection) in the measured acceleration. These peaks are emphasized in vertical acceleration during walking because of heel impacts especially when sensors are attached to the body [79], [80]. The peaks can be also detected in the norm of acceleration vector for other smartphone locations including texting or carrying in a bag. However, their magnitude is usually smaller. The difficulty with this approach is that each foot impact may generate multiple local peaks which can significantly increase the algorithm complexity. The algorithm performance can be improved if accelerometer measurements are pre-processed by applying band-pass filtering. For example, Ladetto applied the wavelet transformation to remove high-frequency harmonics from the signal [79]. A disadvantage of this approach is that walking motion detection may be triggered by any motion with a similar periodicity to walking.

2) Correlation Based Algorithms: An alternative to peak detection in acceleration signal is computing cross-correlation of accelerometer measurements with a pre-defined stride template or auto-correlation function. The template can be previously recorded walking data from the same person [73]. Correlation based algorithms detect peaks in the mean-adjusted computed correlation of a temporal sequence of acceleration, usually the norm of the measured acceleration vector. This approach can be used for different sensor locations.

Rai *et al.* [83] proposed Normalized Auto-correlation based Step Counting (NASC). They observed that if the user is walking, then the auto-correlation will spike at the correct periodicity of the walker. Given a sequence of acceleration samples $a(k)$, the normalized auto-correlation for lag τ at the m-th sample is

$$\chi(m,\tau) = \frac{\sum_{k=0}^{k=\tau-1}(a(m+k)-\mu(m,\tau))(a(m+k+\tau)-\mu(m+\tau,\tau))}{\tau\sigma(m,\tau)\sigma(m+\tau,\tau)} \tag{17}$$

where the value of a time lag τ is chosen between 0.8 and 2 sec. According to [83] when the person is walking and the step frequency is equal to τ, the normalized auto-correlation is close to one. Since, the value of τ is not known a priori, the algorithm computes the auto-correlation for different values of τ within a pre-defined range and finds the value of τ for which the $\chi(m,\tau)$ becomes maximum. NASC can also detect other repetitive activity such as running.

Dynamic time warping (DTW) algorithm attempts to recognize each step within the data. A stride template is formed offline and cross-correlation with the accelerometer signal is computed online [87]. DTW gives better results since it allows a non-linear mapping between the template and measured acceleration. However, good performance comes at the price of a considerable computational overhead. Compare to DTW, computation of auto-correlation is faster and its computational burden can be further reduced by evaluating the auto-correlation only for a subset of time lags corresponding to the frequencies of interest [88].

3) Spectral Analysis Based Algorithms: Another version of peak detection algorithm uses wavelet decomposition to extract dominant frequency component from subsets of measured acceleration [79] within the time-interval that includes at least two cycles. In this approach the accelerometer signal is split into its low- and high-frequency components and the dominant frequency taken as the walking frequency. By iterating this process N times on the resulted low-frequency approximation, the dominant low-frequency signal with much smoother shape is generated. The frequency of this signal corresponds to walking cadence. This is called the wavelet decomposition tree, and N is the number of computed levels. Ladetto [79] performed decomposition at level 4 using the Meyer wavelet function. The decomposed at this level signal has only one maximum within each cycle and therefore steps can be easily identified and counted.

Susi *et al.* [85], [89] applied the short-term Fourier transform (STFT) to evaluate the frequency content of successive windows of accelerometer signal. The basic assumption of this technique is that any non-stationary signal can be regarded as stationary for a short time duration. The walking cadence is visible in the frequency peaks of the spectrogram obtained as the square absolute value of the STFT. They also analyzed the dominant frequency by evaluating the largest maximum in the spectrogram. When the user was walking and carrying the inertial measurement unit (IMU) in the texting mode and with the hand swinging, the walking cadence was visible in the spectrogram's frequency peaks and the strongest frequency for both IMU placements was coupled with the step event. Another observation was that when the user was performing an irregular motion, e.g., looking for a mobile phone in a bag, the two main frequencies showed a very irregular trend. The spectrogram was completely different when the user was static: the dominant frequencies did not exhibit strong peaks. So, the frequency analysis can be used to distinguish irregular motion modes from normal walking [85].

Brajdic and Harle [74] used STFT for both walk detection and step counting. In walk detection they used STFT to split the signal into successive time windows and labelled each as walking if it contained significant spectral energy at typical walking frequencies. For step counting they used STFT to compute a fractional number of strides for each window by dividing the window width by the dominant walking period it detected. These fractional values were then summed to estimate the number of strides taken.

4) Walk Detection and Step Counting on Unconstrained Smartphone: It should be noted that most implementations claim to use only the vertical acceleration, but do not compensate for changes in the global pose of the sensors during walking. Instead they assume that one of the accelerometer axes remains vertical throughout. This assumption in general is not valid for unconstrained smartphones. However, there are some efficient and robust walk detection and step counting algorithms designed for unconstrained smartphones. These approaches can be classified into the following groups: (a) allowing arbitrary pose and carrying location, (b) tracking the phone orientation, and (c) performing motion classification and adapting different algorithms for different motions.

Pan and Lin [90] proposed a step counting algorithm for unrestricted smartphone allowing arbitrary device pose during walking. The algorithm computes the phone pose transformation based on gravity vectors. Then it looks for correlated segments from the collected raw data. The authors adopted the concept of correlation coefficients to identify whether the collected sensing measurements exhibit similar tendencies and then perform step segmentation. It was claimed that the step detection error is quite small when the phone is carried without changing the carrying mode. This approach can be applied only for walking.

Susi *et al.* [85] analyzed biomechanics of walking and found that the swinging of the arm is synchronized with the foot motion, thus making walk detection and step counting possible when sensors are carried in hand. The authors proposed to use the gyroscope data for step counting because a swinging arm produces a notable cyclic pattern in angular velocity. So, the steps can be counted by counting the peaks of the gyroscope signal.

The same phenomena was also exploited in [91] where Renaudin *et al.* showed that the strongest frequency of the signal extracted from the handheld IMU is coupled with the step frequency of the walking person and can be used to estimate the cadence. The relationship between step and hand frequencies was analyzed for different hand motions and phone carrying modes. The approach for reliable step frequency estimation on handheld device is based on STFT. The field tests showed that the error in distance estimation with this algorithm was between 2.5–5% of the travelled distance [91], which is comparable with the accuracy achieved by other algorithms that are designed only for body fixed sensors.

Steinhoff and Schiele [92] experimented with a medium accuracy IMU (Xsens MTx) carried in trousers' front pockets during walking. The main contribution of their work is experimental evaluation of novel and existing PDR algorithms in scenarios when an unconstrained IMU is carried in pocket. To cope with unknown and changing IMU pose the approach computes its global orientation. Then it computes the vertical acceleration, which was found to be a feature that is robust to changing orientation. To make the algorithm more robust the signal was pre-processed with a median filter for noise removal. Also a threshold on the vertical acceleration variance was applied to avoid spurious step detections from small peaks. Based on a field test with eight people it was found that the step detection accuracy was not affected by the IMU position inside the pocket or the shape of the pocket.

Summarizing the walk detection and step counting algorithms for unconstrained smartphones it can be concluded that all successful algorithms are able to recognize (or know a priori) how a smartphone is being carried and then adapt the algorithms for better performance. Some approaches even demonstrated the ability to cope with ill-defined movements such as shuffling or side steps [74].

5) Step Counting During Walking on Stairs or Ramps: Another common assumption in step detection is that most algorithms are developed and tested only for pedestrian motion on flat surfaces. Ladetto [79] tested step counting algorithms on ramps and found that most algorithms are robust to small slopes in surface and start to break down only on inclines of 10% or more. More recently, Wang *et al.* [93] have demonstrated that different gait patterns corresponding to different inclines can be distinguished autonomously with accuracy exceeding 90%. From this we can conclude that algorithms for walk detection and step counting can be adapted to cope with long ramps such as those for wheelchair access, which can be found in many buildings.

Diaz and Gonzalez [84] proposed step detection algorithms for a pocket-mounted sensor unit that can work not only during walking on flat surfaces, but also during climbing and descending stairs. This algorithm exploits the use of the pitch angle computed by the IMU. In addition to step segmentation it was demonstrated that using the pitch angle, it is possible to identify five basic physical activities: standing, sitting, walking, walking up and down stairs. However, this algorithm heavily relies on the assumption that the sensor unit is in a pocket. When the unit is taken out of the pocket the algorithm may perform incorrectly.

6) Performance of Step Counting Algorithms Implemented on Smartphones: The most comprehensive presentation of field test results for walk detection and step counting on unconstrained smartphones is given in [74]. The authors proposed metrics and scenarios that allow uniform comparison of algorithms on commodity smartphones. The test scenarios included different activities such as picking up and putting down the phone, texting, revolving on an office chair, transitioning between standing and sitting, and between walking and standing. Step counting errors are computed for eight different algorithms based on a set of 130 sensor traces from 27 distinct users. All algorithms require only acceleration magnitude data gathered from a smartphone.

The results for walk detection showed that the best performance was achieved by the two threshold based algorithms: one with the threshold on standard deviation and another on signal energy. They exhibited total median error in walk detection of about 1% including both false positive and false negative errors. The overall best step counting results were obtained using the windowed peak detection algorithm with a median error of 1.3%. Except for the placement in a back trouser pocket, the majority of placements have little effect on the step counting performance.

The threshold based algorithms can have false-positive detection errors, reporting steps even when users are not walking if they make sudden movements such as hand gestures, turning or shifting in the chair etc. When walking is disrupted by sudden movements Brajdic and Harle [74] used normalized auto-correlation algorithm combined with the threshold on standard deviation. This approach is more robust since it exploits the repetitive nature of walking.

If the phone is placed in a trouser pocket the step counting can be improved by adding gyroscopes data to the algorithm. Diaz and Gonzalez [84] presented a step detector and step length estimator based on the opening angle of the leg. It was shown that the movement of the user's leg is strongly correlated with walking cadence and the step length. This approach can also reliably work during climbing and descending stairs. The error in step detection is less then 1%, however, the field tests were carried out with Xsens IMU, which has much more accurate gyroscopes than any smartphone.

B. Step Length Estimation

Simple algorithms only count steps assuming that the step length is just the average for that user. Advanced algorithms also perform accurate step segmentation and analyze the accelerometer signals to estimate the magnitude of each step individually. However, most of these systems require calibration to an individual user because everyone's gait has different acceleration profiles.

The following two formulas can be used to estimate step length when the smartphone is in a jacket or trouser pocket. Since the formulas use vertical component of the acceleration to compute a step length, the mobile device's pose has to be known. An empirical relation between maximum and minimum of the vertical acceleration and the step length is given by [77]

$$SL_1 = K\sqrt[4]{a_{\max} - a_{\min}} \tag{18}$$

where $a_{\max}$, $a_{\min}$ are the maximum and minimum values of the vertical acceleration during one step, and K is a calibration constant that can adjust step length estimation to a specific person. This formula is based on the bounce movement of the hip while walking. The second step length estimator that was developed for implementation on smartphones when they are carried in pockets [94], [95] is

$$SL_2 = 0.1\sqrt[2.7]{\frac{\sum_{i=1}^{k=N}\|a_{\mathrm{vert,i}}\|}{N}\sqrt{\frac{K}{\sqrt{T\cdot a_{\mathrm{peak}}}}}} \tag{19}$$

where T is a step duration, $a_{\mathrm{vert,i}}$ is a sequence of measured vertical accelerations during one step, a_{peak} is the difference between maximum and minimum vertical acceleration during the same step, and K is a calibration constant.

Ladetto [79] proposed an empirical formula for step length that uses the variance of acceleration instead of the vertical component of acceleration. His formula is the following relation between the step length, step frequency and variance of acceleration signal:

$$SL_3 = K_1 + K_2 f + K_3\,\mathrm{Var} + w \tag{20}$$

where K_1, K_2, K_3 are user-specific calibration parameters, f is an estimated step frequency, Var is the variance of the measured acceleration signal, and w is a Gaussian noise. The accuracy of this formula was analyzed during field tests with 20 persons who walked with a constant cadence. The results indicate that the step length computed by Eq. 20 is quite accurate for fast walking with the variance of 4% at 130 steps/min. However, the variance increased to 15% at a cadence of 60 steps/min. Fortunately, since the mean error is much smaller the error in traveled distance is less than 2%.

Gusenbauer *et al.* [96] also used a linear equation to model the relation between the step length and cadence, but without the acceleration variance term:

$$SL_4 = K_1 + K_2 f + w \tag{21}$$

They regarded the step frequency as a crucial parameter for the step length estimation. The natural asymmetry in left and right steps is accounted for by a Gaussian noise term w. Based on field tests with different subjects, the standard deviation for the Gaussian noise distribution was defined as

$$\sigma = 0.24 - 0.09f \tag{22}$$

A model for reliable step length estimation on handheld devices was proposed in [91]. It includes the step frequency, height, and three calibration parameters:

$$SL_5 = h\cdot(K_1\cdot f + K_2) + K_3 \tag{23}$$

where h is the user's height. The authors proposed to use two sets of parameters: universal and calibrated. The universal parameters are not calibrated to a specific person and usually give the first approximation for the calibration procedure. The calibrated parameters are computed using recursive least-squares by minimizing the sum of squared residuals between the true step length and the predicted step length for n particular people. The relationship between walking cadence and hand swinging frequency was analyzed for different hand motions and sensor carrying modes based on the frequency content of synchronized signals collected with two sensors placed in the hand and on the foot.

The approach proposed in [84] works only when the mobile device is carried in trouser pockets during walking and it computes the step length using the opening angle of the leg. Based on extensive field tests the authors found that this relationship can be applied in general for all users regardless of other parameters such as the user's height, leg length, weight, gender and the pocket's size. The step length is computed using the formula

$$SL_6 = K_1 + K_2\,\Delta\theta \tag{24}$$

where $\Delta\theta$ is the pitch angle measured by an IMU in trouser pocket, and K_1, K_2 are user-specific calibration parameters. The results showed that the estimation of the step length is insensitive to speed changes.

1) Performance of Step Length Estimation Algorithms With Unconstrained IMU: Jahn *et al.* [95] evaluated the performance of the step length estimation algorithms based on Eq. 18 and Eq. 19. In the first case the Xsens IMU was attached to the back approximately at the center of mass and in the second case the IMU was carried in a trouser pocket. In both cases the error in step length estimation was about 2–3%. The standard deviation in the second case was higher. The measurement data was collected from five participants during indoor field tests on a flat concrete area. The subjects were asked to complete the test course at three different speeds: "normal", "slow" and "fast". The calibration constant was computed based on test data for different walking speeds.

Fang *et al.* [80] tested the step length estimator based on Eq. 18 on a 1.2 km course in indoor and outdoor environment on different terrains including swamp terrain. The overall distance estimation error with personal calibration was about ±3% and ±8% without calibration across the variety of subjects. The authors noticed that the estimation accuracy degrades rapidly when a person is walking, for example, on non-flat terrain or climbing on stairs.

Ladetto [79] tested his approach with a body mounted IMU during experiments with 20 people. He could achieve an error in distance estimation of about 1% with parameters calibrated to each person. The tests also revealed that slopes smaller than 10% have little effect on distance estimation.

Renaudin *et al.* [91] evaluated performance of the step length estimation algorithm with hand-held IMU during field tests involving 10 test subjects. They also compared the accuracy of travelled distance estimation of the same algorithm between the case when universal parameters were used and when a set of parameters was calibrated for each subject. The algorithm with user-specific parameters showed an error between 2.5 and 5% of the travelled distance, which is comparable with that achieved by models proposed in the literature for body fixed sensors only. The algorithm with universal parameters showed an error between 3.2 and 9% of the travelled distance.

C. Walking Direction Estimation

Walking direction estimation on an unconstrained smartphone is the most difficult problem that has to be solved to achieve capabilities for autonomous navigation. This is still an open problem since most of the proposed solutions can work only with IMUs that have much better accuracy than modern smartphone sensors. The algorithms for walking direction estimation usually include two important tasks: (a) computation of the smartphone's pose with respect to some global frame; (b) estimation of walking direction in the global frame using the smartphone's sensors.

1) Computing Smartphone's Orientation in Global Frame: The smartphone's orientation is usually computed with respect to a global frame N as shown in Fig. 14. This frame is fixed to the earth and its x_n and y_n axes are aligned with the local horizontal plane pointing north and east respectively, and the z_n axis is aligned with the local vertical. This way an orthogonal right-hand coordinate frame is formed, sometimes referred to as east-north-up (ENU)? frame. The sensor or body frame B is centered in the smartphone and moves with it. Ideally, its origin would be located in the accelerometer to avoid measuring accelerations due to pure rotation. However, since the accelerometer's exact location is unknown, the origin is assumed to be in the phone's geometrical center. If the screen is facing up, the x-axis points right, the y-axis points forward, and the z-axis points up. Both the Android and the iOS APIs use the same definition of the coordinate frames. Mathematically, orientation of the sensor frame with respect to the global frame can be described by a transformation matrix $\mathbf{R}_\mathbf{n}^\mathbf{b}$ which is composed of the direction cosine values between the two coordinate frames.

Tracking orientation using smartphone's gyroscopes is a difficult task that requires frequent updates from other aiding sources, for example, magnetometers and accelerometers. However, magnetometers are not reliable indoors because of magnetic field distortions and they cannot track fast movements like those when a phone is swinging with the hand. One solution to this problem is based on the assumption that during walking and some other activities the average acceleration due to the movement is almost zero. Then, the direction of gravity can be measured by the accelerometers and used to correct the horizontal angles. This assumption is valid for most practical cases encountered in pedestrian navigation, since the acceleration of a person's body during walking is cyclic, and therefore, if it is averaged over one or several cycles it will be almost zero. The accuracy of the gyroscopes must be good enough to track the orientation during the time between attitude updates to a high degree of accuracy. Since the gyros in smartphones are not accurate this time must be very short, only 1-2 seconds.

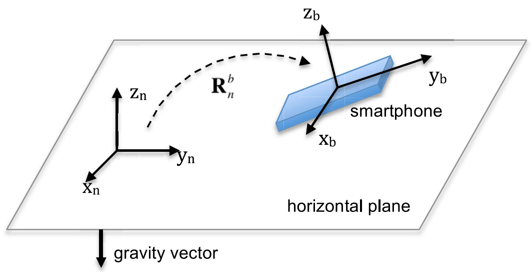

Fig. 14. Relation between global and smartphone's body (sensor) frames. $\mathbf{R}_\mathbf{n}^\mathbf{b}$ is the rotation matrix from body frame to global frame.

The algorithm for pitch and roll angles computation on a smartphone based on the complementary filtering technique and quaternion mechanization was proposed in [70] and [97]. In the filter prediction stage the state mean and covariance are propagated using the gyroscope measurements, while the measurements from the accelerometers and magnetometers are used during the update stage. The approach relies on the capability to calculate and resolve the absolute instantaneous roll and pitch angles of the device by using measurements from accelerometers. The data is first low-pass filtered or averaged to reduce noise and then the roll and pitch are computed using the equations [70]:

$$\begin{aligned} \text{roll} &= \arctan(-f_x/f_z) \\ \text{pitch} &= \arctan\left(f_y \Big/ \sqrt{f_x^2 + f_z^2}\right) \end{aligned} \tag{25}$$

where f_x, f_y, f_z are the instantaneous or averaged acceleration measurements along x, y and z axes. Averaging accelerometer signals reduces not only noise, but also the sensor's bandwidth. As a consequence, the averaged accelerometer signals mainly measure the gravity components, which leads to more accurate computation of pitch and roll. In real-time implementations the noise can be also reduced by a moving average method either applied to the calculated roll and pitch or applied to the accelerometer readings themselves before the calculation of roll and pitch [70].

In [98] another approach for horizontal angles estimation was proposed based on low-cost IMU and external velocity aiding obtained from a model of human gait kinematics, which provides an alternative way to calculate traveled distance and velocity averaged during the step. The algorithm was applied to pedestrian navigation systems that included a torso mounted IMU. The kinetic model of gait is considered as a virtual sensor that can be used to estimate the step events and step size for a person while walking. The term "virtual" emphasizes the fact that there is no separate instrument for direct speed measurement. The speed is estimated using accelerometer measurements and additional information about the pedestrian's movement which is derived from the model of human gait kinematics. The different characteristics of errors in INS output and in this virtual measurement make it possible to apply complementary filter methodology and significantly improve INS performance by keeping the horizontal velocity and tilt errors small.

After the horizontal angles are computed the azimuth can be determined. If a smartphone is being used, combining a gyro and a magnetic compass will give better results than using these sensors separately. The magnetic compass is able to provide the external azimuth to update the parameters of the gyro and the gyro can be used to detect the disturbances that can be as large as tens of degrees. Ladetto [99] proposed using a Kalman filter for coupling a magnetic compass with a low-cost gyroscope. In this case, the strengths of one technology can compensate the weaknesses of the other. If we compare the rate of change of both signals while measuring the strength of the magnetic field, it is possible to detect and compensate magnetic disturbances. In the absence of such disturbances, the continuous measurement of the azimuth allows to estimate and compensate the bias and the scale factor of the gyroscope. The reliability of indoor and outdoor navigation improves significantly thanks to the redundancy in the information.

2) Walking Direction Estimation: The ability to accurately estimate walking direction, regardless of smartphone orientation, is extremely important for long term positioning. The difficulty in walking direction computation with unconstrained smartphones comes from the fact that the direction that a smartphone is facing is often different from the direction the user is moving. So, even if the smartphone's pose with respect to a global frame is accurately determined this is not enough to estimate the step direction.

Traditional IMU can determine the direction of travel by computing vectors of acceleration or velocity increments in the global frame, if the horizontal angles are accurately computed. This approach is usually not possible to implement with smartphone sensors. Other methods for walking direction estimation are focused on estimation of the misalignment between a smartphone and the pedestrian, or in other words the misalignment between the frame of the sensor assembly in the device and the frame of the pedestrian [100], or walking direction [72]. Algorithms for angular misalignment estimation between the smartphones pointing direction and pedestrian walking direction include [72]:

1) The principal component analysis method (PCA)
2) The forward and lateral accelerations modeling (FLAM)
3) The frequency analysis of inertial signals (FIS)

The first two algorithms require only the acceleration measured by a smartphone. The frequency analysis method also requires the angular velocity in addition to the acceleration. These algorithms can be potentially implemented on smartphones in different carrying modes including handheld.

The prerequisite for the above mentioned methods is computation of the smartphone's pose in the global frame, since the measured acceleration in the sensor frame has to be transformed to the global frame. In the PCA method the first step is to compute horizontal acceleration components and remove the gravity value from the acceleration. Then the two horizontal acceleration components are fed to the PCA algorithm, which generates the principal components based on two features of human walk [72], [92], [100]–[102]: (a) over a gait cycle the variance of the horizontal acceleration is maximum along the forward direction; (b) the horizontal acceleration variance is minimum along the lateral direction. The directions are computed using eigenvalues: the eigenvector corresponding to the largest eigenvalue gives the forward direction, and the second eigenvalue's eigenvector gives the lateral direction.

The FLAM method for estimating the angular misalignment is based on modelling of the forward and lateral accelerations by a sum of sinusoids and computing the correlation function for the modelled and measured acceleration. When the angle parameter of this function is equal to the walking direction the correlation function reaches its maximum [103].

The FIS method also exploits the fact that the horizontal acceleration is maximum along the forward direction. It uses a cost function containing the power functions of the acceleration and angular velocity in the forward and lateral directions. These power functions are computed using a short-time Fourier transform. The walking direction is estimated by maximizing this cost function over the angle parameter [72].

The uDirect approach [104] determines a user's facing direction with a mobile phone, regardless of the mobile phone's orientation. A user's forward direction is computed based on accelerometers, exploiting the physiological walking behavior of a human. By analyzing the acceleration pattern of the thigh movement during walking locomotion, the algorithm is able to identify the optimum moment where the captured acceleration signal mainly consists of the forward direction, minimizing the noise of unwanted side components. The underlying mathematical model using the quaternion presentation further simplifies computation of the rotation compared to a vector based approach.

3) Performance of the Walking Direction Estimation Algorithms Implemented on Smartphones: The Humaine [101] positioning system was implemented on Android devices. The field tests included 170 experiments on different testbeds and phone positions and showed that the system works accurately indoors and outdoors, without user involvement or prior configurations. Humaine achieved 15 degrees of median accuracy in walking direction estimation. The uDirect [104] autonomous positioning system was tested only with a smartphone carried in trouser pocket and it could estimate walking direction with 22 degrees of error.

Other algorithms for walking direction estimation have been tested only with tactical grade IMUs that included a tri-axial accelerometer, a tri-axial gyroscope and a tri-axial magnetometer: ADIS-16488 in [72], ADIS-16407 in [71], Xsens MTx in [92] and [102], etc. The reported walking direction estimation accuracy for these algorithms with IMU carried in hand or in trouser pockets is about $3 - 5$ deg. However, it is not possible to predict what these algorithms' accuracy in walking direction estimation would be if the high quality IMU were replaced by a smartphone.

D. Multiple Usage Scenarios and Transit Modes

Identification of transit mode, type of activity and phone placement can significantly improve the accuracy and reliability of PDR and map-matching. For typical smartphone deployment scenarios it will be enough to identify the following modes of transit and activities [74], [83], [105]: idle, picking up and putting down the phone, texting, transitioning between standing and sitting, level walking, up- and downstairs climbing, riding on elevators and escalators. In addition to these activities some common phone placement scenarios have to be considered: hand-held, front trouser pocket, back trouser pocket and in a backpack or handbag. All movements can be performed at different speeds and intensity levels.

1) Motion Classes: For the purpose of pedestrian navigation the motion classification can be simplified by limiting the number of classes to only five, similar to what was proposed in [85]:

1) All situations when both the device and the user are static, i.e., the location does not change significantly during some time with the allowed amount of movement not exceeding the predefined threshold.
2) All cases when a user is walking except when a phone is held in swinging hand. This class includes the following cases: (a) A phone is in pocket, bag, arm-band or any other type of attachment; (b) Texting or watching screen; (c) Making or receiving a call.
3) All cases when the user is walking while holding the mobile device in his/her swinging hand.
4) Moving on elevators or escalators.
5) Irregular motions that the user performs while standing or sitting. During this time the user's location does not change.

2) Feature Sets: Time domain features include basic signal characteristics and statistics. The following features can be used to identify different types of activities for pedestrian navigation: magnitude, variance and energy of accelerometer and gyroscope measurements [74], [85]. Frequency-domain features can be used to capture periodicity in data and distinguish cyclic activities from irregular movements. They can be computed using coefficients of Fourier transform or fast Fourier transform (FFT), which also provide the dominant frequencies in the measured acceleration and angular velocity.

Since the bandwidth of human movements is usually below 10 Hz, the sensor data can be low-pass filtered, for example, using a 10th order Butterworth filter with a 15 Hz cut-off frequency [85]. Filtering can improve motion classification because the smartphone sensors are noisy. During fast movements, for example, when a smartphone is swinging with the hand, the norm of the acceleration and angular velocity can be considered. This removes dependence on the phone pose and makes motion classification more robust.

3) Motion Classifier: Pedestrian motion can be classified by a decision tree, which uses a flowchart-like graph of decisions and their possible consequences covering all of the possible classes. Each internal node represents comparison of features with the thresholds, each branch represents the outcome of this comparison and each end node represents a class label. The paths from root to leaf represents classification rules. A decision tree consists of 3 types of nodes: decision nodes, chance nodes, end nodes.

Susi *et al.* [85], [106] used a multivariate decision tree. At the first node the algorithm evaluates the energies and variances of gyroscope and accelerometer signals to determine static and dynamic activities. Then the features describing periodicity are checked to separate irregular motion from walking. Irregular motion is usually characterized by very high values of the gyroscope and accelerometer variances in short temporal periods. Finally, the gyroscope measurements are evaluated to identify walking with a swinging hand, which is characterized by high values of the gyroscope and accelerometer variances and energies, and frequency peaks in the STFT spectrogram of the gyroscope signal.

E. Summary of Sensor Based Positioning

Although in contemporary smartphones the sensor based positioning cannot operate as a stand-alone technology it can be very useful when combined with WLAN/BLE based positioning or map-matching [107]. The major advantage of this technology is availability of accelerometers, gyroscopes and magnetometers in almost every smartphone. The major difficulty in implementation of these tasks on smartphones is freedom of movement and arbitrary placement. The major tasks that can be performed using smartphone sensors are step count, distance and walking direction computation, activity recognition, vertical movements and floor change detection. The most difficult task is a smartphone orientation tracking and walking direction estimation.

Examples of autonomous positioning implemented on a typical Android smartphone are shown in [108] and [109]. During the test, the user was carrying the phone in hand and pocket, texting and making a call with the phone on the ear. During two minutes of autonomous navigation the position error did not exceed 7 meters. In another test the user was walking indoors for 4 minutes and the maximum error stayed within 13 meters, even with the phone changing orientation with respect to the user. This fully autonomous solution can be further improved when combined with WLAN measurements and the building floor plan.

VI. Open Research Issues

Smartphone based indoor positioning systems are being deployed on Apple and Google smartphones and provide reasonably good positioning accuracy in areas rich with WLAN APs and BLE beacons. The companies also provide a public API for Wi-Fi, BLE, inertial and magnetic sensors data that can be used by third party developers. However, the positioning performance still can be improved and open research issues include autonomous indoor positioning, integrity monitoring, fingerprint database generation and maintenance, seamless positioning between indoors, outdoors and different modes of transit, evaluation and comparison of different positioning algorithms

1) Autonomous Indoor Positioning: Currently, there is no effective approach for long-term autonomous indoor positioning when absolute position fixes from WLAN or BLE are unavailable. A possible solution can be based on pedestrian dead-reckoning (PDR) combined with maps. However, PDR implementation for unconstrained phones and different modes of transit is complicated.

2) *Integrity Monitoring:* Receiver autonomous integrity monitoring is commonly used in GNSS. There are also some approaches for integrity monitoring of INS/GNSS integrated navigation systems. In indoor positioning integrity monitoring makes it possible to improve operational robustness and eliminate outliers, in particular, to isolate WLAN APs and BLE beacons that disagree with the fingerprint database. There is a need for such methods on smartphone based indoor positioning systems. Possible solutions can use hardware and analytical redundancy: multiple WLAN APs and BLE beacons, PDR, map-matching and magnetic field fingerprinting.

3) *Effective Generation and Maintenance of Fingerprint Database:* Since the data collection required for WLAN and BLE fingerprint database generation is a time consuming and labor intensive process, the crowdsourcing approach will play a critical role in reducing time and costs in the future. The major difficulty is in providing reference position indoors.

4) *Positioning Performance Evaluation:* There is no standard procedure for positioning accuracy evaluation of different algorithms. Most publications present only walking tests for a single user in a small area of an office building filled with WLAN APs and BLE beacons. The tests are performed in a very controlled environment in which the same device and the way of carrying is used during the training and positioning phases. In some cases the reported position accuracy can be 1-2 m. However, in real life situations the position accuracy is usually significantly worse, especially when the fingerprint database is not up-to-date or the WLAN and BLE coverage is not complete or the fingerprints are not collected in all areas. The accuracy can degrade also because people use different types of phones and carrying modes.

A possible solution to this problem can be creation of a public database that can be used as a benchmark during evaluation of different indoor positioning algorithms. The general requirements to this database can be summarized as

- The data has to be recorded on several different commodity smartphones and include RSS measurements from all available WLAN APs and BLE beacons, acceleration, angular velocity, magnetic field, GNSS or assisted GNSS.
- The WLAN and BLE fingerprints as well as the reference smartphone position has to be provided.
- The test path has to be long enough to include different types of buildings such as office buildings, shopping malls, convention centers, airports, etc, floor change in multi-storey buildings, using elevators and escalators, and different modes of transit between the buildings (car, bus, subway, and train).
- The data is recorded during field tests with many different users who follow the same pre-defined scenario that includes carrying the phone in hand and pocket, texting and making a call with the phone on the ear.

The work presented in [43] is a step towards such a public benchmark.

VII. Conclusion

In this paper we have surveyed the most significant algorithms that can be implemented in smartphone positioning systems: WLAN/BLE based positioning, map-matching, magnetic field fingerprinting, and motion tracking using sensors. The default approach for indoor navigation on a smartphone is based on WLAN. The positioning accuracy depends on density of WLAN access points and calibration points, and can be sufficient for many applications in typical public buildings like shopping malls, universities, airports etc. The disadvantage of this approach is the labor-intensive process of building and maintaining the WLAN database.

Improvement of WLAN based positioning can be achieved by using indoor map information. The assumption of indoor map availability is justified since the map is necessary to communicate results to the user and also for creating the radio map. So, we can conclude that map aiding does not require any additional investments in hardware or infrastructure. This capability is enabled by software that implements a map-matching algorithm.

Many modern smartphones also contain accelerometers, gyroscopes and magnetometers. The use of sensors can significantly enhance the positioning accuracy of WLAN/BLE based solution because the advantages and disadvantages of the two systems broadly complement each other: sensor based position errors growth can be curbed using WLAN positions and gaps in WLAN/BLE coverage can be filled by sensor based positioning. However, because smartphone sensors are not accurate, long term autonomous positioning without WLAN/BLE position fixes currently is not possible. Walking direction computation is the biggest challenge in implementation of PDR on smartphones and it is still an open research problem.

Magnetic field fingerprinting can be added to other indoor positioning methods when high position accuracy is required. Although this method can potentially provide sub-meter position accuracy its implementation on smartphones is difficult, mainly because of stringent requirements to magnetic map database. Magnetic field gradients can be very steep indoors and, therefore, a dense grid of calibration points is necessary. Since the magnetic field may also vary significantly with height, the calibration points also have to be at different heights. Methods have to be developed to create the database in an efficient manner. Finally, transient magnetic interferences should be considered, especially in areas close to elevators.

The most accurate navigation solution is based on a combination of WLAN, BLE, map, magnetic field and sensor based positioning. The trend for augmentation of WLAN/BLE based positioning by all possible sensors will continue in the future smartphone indoor positioning systems. In addition to WLAN and BLE absolute position fixes can be also obtained from LTE, 5G or other future technologies. These measurements will be fed into data fusion filter that can combine them with map, magnetic field fingerprints and sensor measurements and produce accurate and reliable position estimates.

my thought

Acknowledgment

The authors thank Laura Wirola, Dr Helena Leppäkoski, Dr Philipp Müller, Henri Nurminen, Ville Honkavirta, and Dr Simo Ali-Löytty for permission to use their material. The authors would also like to thank the reviewers of this paper for their detailed and thoughtful comments that considerably improved this work.

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Comput. Soc. LAN/MAN Standards Committee, Washington, DC, USA, 1997.

[2] V. Honkavirta, "Location fingerprinting methods in wireless local area networks," M.S. thesis, Dept. Math., Tampere Univ. Technol., Tampere, Finland, 2008.

[3] V. Honkavirta, T. Perälä, S. Ali-Löytty, and R. Piché, "A comparative survey of WLAN location fingerprinting methods," in *Proc. 6th Workshop Position. Navig. Commun. (WPNC)*, Hanover, Germany, Mar. 2009, pp. 243–251.

[4] S. Ali-Löytty, T. Perälä, V. Honkavirta, and R. Piché, "Fingerprint Kalman filter in indoor positioning applications," in *Proc. IEEE Control Appl. (CCA) Intell. Control (ISIC)*, Saint Petersburg, Russia, Jul. 2009, pp. 1678–1683.

[5] F. Seco, A. R. Jimenez, C. Prieto, J. Roa, and K. Koutsou, "A survey of mathematical methods for indoor localization," in *Proc. Intell. Signal Process.*, Budapest, Hungary, 2009, pp. 9–14.

[6] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[7] P. Müller, M. Raitoharju, and R. Piché, "A field test of parametric WLAN-fingerprint-positioning methods," in *Proc. 17th Int Conf. Inf. Fusion*, Salamanca, Spain, Jul. 2014, pp. 1–8.

[8] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to WLAN user location estimation," *Int. J. Wireless Inf. Netw.*, vol. 9, no. 3, pp. 155–164, 2002.

[9] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst. Appl. Services*, Seattle, WA, USA, 2005, pp. 205–218.

[10] L. Koski, R. Piché, V. Kaseva, S. Ali-Löytty, and M. Hännikäinen, "Positioning with coverage area estimates generated from location fingerprints," in *Proc. 7th Workshop Position. Navig. Commun. (WPNC)*, Dresden, Germany, Mar. 2010, pp. 99–106.

[11] L. Koski, R. Piché, and S. Ali-Löytty, "Positioning with Bayesian coverage area estimates and location fingerprints," in *Proc. Eur. Conf. Math. Ind.*, Wuppertal, Germany, 2010, pp. 99–106.

[12] L. Koski, T. Perälä, and R. Piché, "Indoor positioning using WLAN coverage area estimates," in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, Zürich, Switzerland, Sep. 2010, pp. 1–7.

[13] H. Nurminen, "Position estimation using RSS measurements with unknown measurement model parameters," M.S. thesis, Dept. Math., Tampere Univ. Technol., Tampere, Finland, 2012.

[14] H. Nurminen *et al.*, "Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks," in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, Sydney, NSW, Australia, Nov. 2012, pp. 1–9.

[15] C.-L. Wu, L.-C. Fu, and F.-L. Lian, "WLAN location determination in e-home via support vector classification," in *Proc. IEEE Int. Conf. Netw. Sensing Control*, vol. 2. Taipei, Taiwan, 2004, pp. 1026–1031.

[16] Y. Feng *et al.*, "An improved indoor localization of WiFi based on support vector machines," *Int. J. Future Gener. Commun. Netw.*, vol. 7, no. 5, pp. 191–206, 2014.

[17] D. Sánchez-Rodríguez, P. Hernández-Morera, J. M. Quinteiro, and I. Alonso-González, "A low complexity system based on multiple weighted decision trees for indoor localization," *Sensors*, vol. 15, no. 6, pp. 14809–14829, 2015.

[18] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Trans. Mobile Comput.*, vol. 11, no. 12, pp. 1983–1993, Dec. 2012.

[19] B. Viel and M. Asplund, "Why is fingerprint-based indoor localization still so hard?" in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Budapest, Hungary, 2014, pp. 443–448.

[20] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, 1st Quart., 2016.

[21] A. K. M. M. Hossain, Y. Jin, W.-S. Soh, and H. N. Van, "SSD: A robust RF location fingerprint addressing mobile devices' heterogeneity," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 65–77, Jan. 2013.

[22] C. Laoudias, D. Zeinalipour-Yazti, and C. G. Panayiotou, "Crowdsourced indoor localization for diverse devices through radiomap fusion," in *Proc. IEEE Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Montbéliard, France, 2013, pp. 1–7.

[23] Y. Chapre, P. Mohapatra, S. Jha, and A. Seneviratne, "Received signal strength indicator and its analysis in a typical WLAN system (short paper)," in *Proc. IEEE 38th Conf. Local Comput. Netw. (LCN)*, Sydney, NSW, Australia, 2013, pp. 304–307.

[24] F. D. Rosa *et al.*, "Hand-grip and body-loss impact on RSS measurements for localization of mass market devices," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, Tampere, Finland, 2011, pp. 58–73.

[25] F. D. Rosa, M. Pelosi, and J. Nurmi, "Human-induced effects on RSS ranging measurements for cooperative positioning," *Hindawi Int. J. Navig. Observation*, vol. 2012, pp. 1–13, Nov. 2012.

[26] M. A. Youssef, A. Agrawala, and A. U. Shankar, "WLAN location determination via clustering and probability distributions," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Fort Worth, TX, USA, 2003, pp. 143–150.

[27] S.-H. Fang and T. Lin, "Principal component localization in indoor WLAN environments," *IEEE Trans. Mobile Comput.*, vol. 11, no. 1, pp. 100–110, Jan. 2012.

[28] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: Indoor Wi-Fi fingerprinting system," in *Proc. 39th Annu. IEEE Conf. Local Comput. Netw.*, Edmonton, AB, Canada, 2014, pp. 202–209.

[29] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding WiFi localization," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Services*, Taipei, Taiwan, 2013, pp. 249–262.

[30] D. Chen *et al.*, "A fine-grained indoor localization using multidimensional Wi-Fi fingerprinting," in *Proc. 20th IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Hsinchu, Taiwan, 2014, pp. 494–501.

[31] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, New Orleans, LA, USA, 2015, pp. 1666–1671.

[32] X. Wang, L. Gao, and S. Mao, "PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

[33] R. Faragher and R. Harle, "An analysis of the accuracy of Bluetooth low energy for indoor positioning applications," in *Proc. 27th Int. Tech. Meeting Satellite Div. Inst. Navig. (ION)*, Tampa, FL, USA, Sep. 2014, pp. 201–210.

[34] S. Ashok and R. Krishnaiah, "Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 9, pp. 737–740, 2013.

[35] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does BTLE measure up against WiFi? A comparison of indoor location performance," in *Proc. Eur. Wireless 20th Eur. Wireless Conf.*, Barcelona, Spain, 2014, pp. 1–6.

[36] S. Liu, Y. Jiang, and A. Striegel, "Face-to-face proximity estimationusing Bluetooth on smartphones," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 811–823, Apr. 2014.

[37] E. S. Lohan *et al.*, "Received signal strength models for WLAN and BLE-based indoor positioning in multi-floor buildings," in *Proc. IEEE Int. Conf. Localization GNSS (ICL-GNSS)*, Gothenburg, Sweden, 2015, pp. 1–6.

[38] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robot. Auton. Syst.*, vol. 57, no. 10, pp. 1028–1035, 2009.

[39] G. Berkovich, "Accurate and reliable real-time indoor positioning on commercial smartphones," in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Busan, South Korea, 2014, pp. 670–677.

[40] B. Gozick, K. P. Subbu, R. Dantu, and T. Maeshiro, "Magnetic maps for indoor navigation," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 12, pp. 3883–3891, Dec. 2011.

[41] M. J. Abadi, L. Luceri, M. Hassan, C. T. Chou, and M. Nicoli, "A collaborative approach to heading estimation for smartphone-based PDR indoor localisation," in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Busan, South Korea, 2014, pp. 554–563.

[42] Y. Shu *et al.*, "Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1443–1457, Jul. 2015.

[43] J. Torres-Sospedra *et al.*, "Providing databases for different indoor positioning technologies: Pros and cons of magnetic field and Wi-Fi based positioning," in *Proc. Mobile Inf. Syst.*, vol. 2016. Jan. 2016, pp. 1–22.

[44] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?" in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Sydney, NSW, Australia, 2012, pp. 1–9.

[45] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," in *Proc. Ubiquitous Position. Indoor Navig. Location Based Service (UPINLBS)*, Kirkkonummi, Finland, 2010, pp. 1–10.

[46] K. B. Anonsen and O. Hallingstad, "Terrain aided underwater navigation using point mass and particle filters," in *Proc. IEEE/ION Position Location Navig. Symp.*, vol. 2006. Coronado, CA, USA, 2006, pp. 1027–1035.

[47] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1263–1278, Apr. 2015.

[48] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[49] W. Widyawan, M. Klepal, and S. Beauregard, "A backtracking particle filter for fusing building plans with PDR displacement estimates," in *Proc. 5th Workshop Position. Navig. Commun.*, Hanover, Germany, Mar. 2008, pp. 207–212.

[50] S. Beauregard, "Omnidirectional pedestrian navigation for first responders," in *Proc. 4th Workshop Position. Navig. Commun.*, Hannover, Germany, Mar. 2007, pp. 33–36.

[51] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proc. UbiComp*, Seoul, South Korea, Sep. 2008, pp. 114–123.

[52] B. Krach and P. Robertson, "Integration of foot-mounted inertial sensors into a Bayesian location estimation framework," in *Proc. 5th Workshop Position. Navig. Commun.*, Hannover, Germany, Mar. 2008, pp. 55–61.

[53] M. Khider, S. Kaiser, P. Robertson, and M. Angermann, "The effect of maps-enhanced novel movement models on pedestrian navigation performance," in *Proc. 12th Eur. Navig. Conf.*, Toulouse, France, Apr. 2008.

[54] P. Davidson, J. Collin, and J. Takala, "Map-aided autonomous pedestrian navigation system," in *Proc. 18th Int. Conf. Integr. Navig. Syst.*, Saint Petersburg, Russia, May 2011, pp. 314–318.

[55] C. Ascher, C. Kessler, R. Weis, and G. F. Trommer, "Multi-floor map matching in indoor environments for mobile platforms," in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, Sydney, NSW, Australia, Nov. 2012, pp. 1–8.

[56] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[57] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization," in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, Montbéliard, France, Oct. 2013, pp. 1–10.

[58] M. Khider, S. Kaiser, P. Robertson, and M. Angermann, "Maps and floor plans enhanced 3D movement model for pedestrian navigation," in *Proc. ION GNSS*, Savannah, GA, USA, Sep. 2009, pp. 790–802.

[59] P. Gilliéron, D. Büchel, I. Spassov, and B. Merminod, "Indoor navigation performance analysis," in *Proc. Eur. Navig. Conf. (GNSS)*, Rotterdam, The Netherlands, 2004, pp. 17–19.

[60] M. Koivisto, H. Nurminen, S. Ali-Löytty, and R. Piché, "Graph-based map matching for indoor positioning," in *Proc. Int. Conf. Inf. Commun. Signal Process.*, Singapore, 2015, pp. 1–5.

[61] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Seattle, WA, USA, 2014, pp. 147–158.

[62] I. Spassov, "Algorithms for map-aided autonomous indoor pedestrian positioning and navigation," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2007.

[63] P. Davidson, J. Collin, and J. Takala, "Application of particle filters to a map-matching algorithm," *J. Gyroscopy Navig.*, vol. 2, no. 4, pp. 286–293, 2011.

[64] K. Abdulrahim, C. Hide, T. Moore, and C. Hill, "Integrating low cost IMU with building heading in indoor pedestrian navigation," *J. Glob. Position. Syst.*, vol. 10, no. 1, pp. 30–38, 2011.

[65] K. Abdulrahim, C. Hide, T. Moore, and C. Hill, "Aiding low cost inertial navigation with building heading for pedestrian navigation," *J. Navig.*, vol. 64, no. 2, pp. 219–233, 2011.

[66] J. Borenstein, L. Ojeda, and S. Kwanmuang, "Heuristic reduction of gyro drift for personnel tracking systems," *J. Navig.*, vol. 62, no. 1, pp. 41–58, 2009.

[67] J. Borenstein and L. Ojeda, "Heuristic drift elimination for personnel tracking systems," *J. Navig.*, vol. 63, no. 4, pp. 591–606, 2010.

[68] A. R. Jiménez, F. Seco, F. Zampella, J. C. Prieto, and J. Guevara, "Improved heuristic drift elimination (iHDE) for pedestrian navigation in complex buildings," in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, Guimarães, Portugal, Sep. 2011, pp. 1–8.

[69] P. Davidson, M. Kirkko-Jaakkola, J. Collin, and J. Takala, "Navigation algorithm combining building plans with autonomous sensor data," *Gyroscopy Navig.*, vol. 6, no. 3, pp. 188–196, 2015.

[70] Z. Syed *et al.*, "Methods of attitude and misalignment estimation for constraint free portable navigation," U.S. Patent 0 245 839 A1, Sep. 27, 2012.

[71] V. Renaudin, C. Combettes, and F. Peyret, "Quaternion based heading estimation with handheld MEMS in indoor environments," in *Proc. IEEE/ION Position Location Navig. Symp. (PLANS)*, Monterey, CA, USA, 2014, pp. 645–656.

[72] C. Combettes and V. Renaudin, "Comparison of misalignment estimation techniques between handheld device and walking directions," in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Banff, AB, Canada, 2015, pp. 1–8.

[73] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1281–1293, 3rd Quart., 2013.

[74] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Zürich, Switzerland, 2013, pp. 225–234.

[75] J. Käppi, J. Syrjärinne, and J. Saarinen, "MEMS-IMU based pedestrian navigator for handheld devices," in *Proc. ION GPS*, Salt Lake City, UT, USA, Sep. 2001, pp. 1369–1373.

[76] H. Leppäkoski, J. Käppi, J. Syrjärinne, and J. Takala, "Error analysis of step length estimation in pedestrian dead reckoning," in *Proc. ION GPS*, Portland, OR, USA, Sep. 2002, pp. 1136–1142.

[77] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," Analog Devices, Norwood, MA, USA, Application note AN-602, Jul. 2002, pp. 1–6.

[78] J. Saarinen, "A sensor-based personal navigation system and its application for incorporating humans into a human-robot team," Ph.D. dissertation, Dept. Autom. Syst. Technol., Helsinki Univ. Technol., Espoo, Finland, 2009.

[79] Q. Ladetto, "On foot navigation: Continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering," in *Proc. ION GPS*, Salt Lake City, UT, USA, Sep. 2000, pp. 1735–1740.

[80] L. Fang *et al.*, "Design of a wireless assisted pedestrian dead reckoning system: The NavMote experience," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2342–2358, Dec. 2005.

[81] T. Judd, "A personal dead reckoning module," in *Proc. ION GPS*, Kansas City, MO, USA, 1997, pp. 47–51.

[82] H.-H. Lee, S. Choi, and M.-J. Lee, "Step detection robust against the dynamics of smartphones," *Sensors*, vol. 15, no. 10, pp. 27230–27250, 2015.

[83] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, Istanbul, Turkey, 2012, pp. 293–304.

[84] E. M. Diaz and A. L. M. Gonzalez, "Step detector and step length estimator for an inertial pocket navigation system," in *Proc. IEEE Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Busan, South Korea, 2014, pp. 105–110.

[85] M. Susi, V. Renaudin, and G. Lachapelle, "Motion mode recognition and step detection algorithms for mobile phone users," *Sensors*, vol. 13, no. 2, pp. 1539–1562, 2013.

[86] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. 2nd Int. Conf. Pervasive Comput.*, Linz, Austria, Apr. 2004, pp. 1–17.

[87] L. Rong, D. Zhiguo, Z. Jianzhong, and L. Ming, "Identification of individual walking patterns using gait acceleration," in *Proc. IEEE 1st Int. Conf. Bioinformat. Biomed. Eng. (ICBBE)*, Wuhan, China, 2007, pp. 543–546.

[88] Y. Makihara *et al.*, "Phase registration of a single quasi-periodic signal using self dynamic time warping," in *Proc. 10th Asian Conf. Comput. Vis. (ACCV)*, Queenstown, New Zealand, Nov. 2010, pp. 667–678. [Online]. Available: http://dl.acm.org/citation.cfm?id=1966049.1966102

[89] M. Susi, D. Borio, and G. Lachapelle, "Accelerometer signal features and classification algorithms for positioning applications," in *Proc. Int. Tech. Meeting Inst. Navig.*, San Diego, CA, USA, 2001, pp. 158–169.

[90] M.-S. Pan and H.-W. Lin, "A step counting algorithm for smartphone users: Design and implementation," *IEEE Sensors J.*, vol. 15, no. 4, pp. 2296–2305, Apr. 2015.

[91] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.

[92] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket—An experimental study," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mannheim, Germany, 2010, pp. 162–170.

[93] N. Wang, E. Ambikairajah, S. J. Redmond, B. G. Celler, and N. H. Lovell, "Classification of walking patterns on inclined surfaces from accelerometry data," in *Proc. 16th Int. Conf. Digit. Signal Process.*, Jul. 2009, pp. 1–4.

[94] I. Bylemans, M. Weyn, and M. Klepal, "Mobile phone-based displacement estimation for opportunistic localisation systems," in *Proc. IEEE Int. Conf. Mobile Ubiquitous Comput. Syst. Services Technol. (UBICOMM)*, Sliema, Malta, 2009, pp. 113–118.

[95] J. Jahn, U. Batzer, J. Seitz, L. Patino-Studencka, and J. G. Boronat, "Comparison and evaluation of acceleration based step length estimators for handheld devices," in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, Zürich, Switzerland, Sep. 2010, pp. 1–6.

[96] D. Gusenbauer, C. Isert, and J. Krösche, "Self-contained indoor positioning on off-the-shelf mobile devices," in *Proc. IEEE Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Zürich, Switzerland, 2010, pp. 1–9.

[97] A. Ali and N. El-Sheimy, "Low-cost MEMS-based pedestrian navigation technique for GPS-denied areas," *J. Sensors*, vol. 2013, pp. 1–10, Jun. 2013.

[98] P. Davidson and J. Takala, "Algorithm for pedestrian navigation combining IMU measurements and gait models," *J. Gyroscopy Navig.*, vol. 4, no. 2, pp. 79–84, 2013.

[99] Q. Ladetto and B. Merminod, "Digital magnetic compass and gyroscope integration for pedestrian navigation," in *Proc. 9th Int. Conf. Integr. Navig. Syst.*, Saint Petersburg, Russia, May 2002, pp. 27–29.

[100] A. Ali, H.-W. Chang, J. Georgy, Z. Syed, and C. Goodall, "Method and apparatus for determination of misalignment between device and pedestrian," WO Patent PCT/CA2014/000 040, Jul. 24, 2014. [Online]. Available: https://www.google.com/patents/WO2014110672A1?cl=ja

[101] N. Mohssen, R. Momtaz, H. Aly, and M. Youssef, "It's the human that matters: Accurate user orientation estimation for mobile computing applications," in *Proc. 11th Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Services*, London, U.K., 2014, pp. 70–79.

[102] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, "Which way am I facing: Inferring horizontal device orientation from an accelerometer signal," in *Proc. IEEE Int. Symp. Wearable Comput. (ISWC)*, Linz, Austria, 2009, pp. 149–150.

[103] M. Chowdhary, M. Sharma, A. Kumar, S. Dayal, and M. Jain, "Method and apparatus for determining walking direction for a pedestrian dead reckoning process," U.S. Patent 13 682 684, May 22, 2014. [Online]. Available: http://www.google.com/patents/US20140142885

[104] S. A. Hoseinitabatabaei, A. Gluhak, and R. Tafazolli, "uDirect: A novel approach for pervasive observation of user direction with mobile phones," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Seattle, WA, USA, 2011, pp. 74–83.

[105] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.

[106] M. Susi, V. Renaudin, and G. Lachapelle, "Detection of quasi-static instants from handheld MEMS devices," in *Proc. IPIN*, Guimarães, Portugal, 2011, pp. 1–9.

[107] Z. Yang *et al.*, "Mobility increases localizability: A survey on wireless indoor localization using inertial sensors," *ACM Comput. Surveys*, vol. 47, no. 3, p. 54, 2015.

[108] Z. Syed, J. Georgy, A. Ali, H.-W. Chang, and C. Goodall, "Showing smartphones the way inside: Real-time, continuous, reliable, indoor/outdoor localization," *Glob. Position. Syst. World*, vol. 24, no. 3, pp. 30–35, 2013.

[109] N. El-Sheimy and C. Goodall, "Everywhere navigation: Integrated solutions on consumer mobile devices," *Inside Glob. Navig. Satellite Syst.*, vol. 6, no. 5, pp. 74–82, Sep. 2011.

Pavel Davidson received the M.Sc. degree in aerospace engineering from the Technion-Israel Institute of Technology in 1996, and the D.Sc. (Tech.) degree in computer science from the Tampere University of Technology, Finland, in 2013. From 1996 to 2006, he was with Israel Aerospace Industries Ltd., Lod, Israel, Tricom Technologies Inc., Mississauga, Canada, and Spirit DSP Ltd., Moscow, Russia. He is currently a Senior Researcher with the Tampere University of Technology. His research interests include non-GNSS navigation, sensor fusion for personal navigation, geographic information system, computer vision, and wearable electronics for sports and wellbeing.

Robert Piché (M'10–SM'16) received the B.A.Sc., M.A.Sc., and Ph.D. degrees from the University of Waterloo, Canada, in 1981, 1982, and 1986, respectively, all in civil engineering. He was an Assistant Professor of Mathematics with École polytechnique de Montréal from 1986 to 1988. Since 1988, he has been with the Tampere University of Technology, Finland, where he has been a Professor since 2004. His scientific interests include mathematical modeling, numerical analysis, estimation theory, and positioning technology.