Indoor Navigation with a Smartphone Fusing Inertial and WiFi Data via Factor Graph Optimization

Michał Nowicki(✉) and Piotr Skrzypczyński

Institute of Control and Information Engineering, Poznań University of Technology,
Ul. Piotrowo 3A, 60-965 Poznań, Poland
{michal.nowicki,piotr.skrzypczynski}@put.poznan.pl

Abstract. Mobile devices are getting more capable every year, allowing a variety of new applications, such like supporting pedestrian navigation in GPS-denied environments. In this paper we deal with the problem of combining in real-time dead reckoning data from the inertial sensors of a smartphone, and the WiFi signal fingerprints, which enable to detect the already visited places and therefore to correct the user's trajectory. While both these techniques have been used before for indoor navigation with smartphones, the key contribution is the new method for including the localization constraints stemming from the highly uncertain WiFi fingerprints into a graphical problem representation (factor graph), which is then optimized in real-time on the smartphone. This method results in an Android-based personal navigation system that works robustly with only few locations of the WiFi access points known in advance, avoiding the need to survey WiFi signal in the whole area. The presented approach has been evaluated in public buildings, achieving localization accuracy which is sufficient for both pedestrian navigation and location-aware applications on a smartphone.

Keywords: Navigation · Localization · Factor graph · Data fusion · WiFi · IMU · Smartphone · Android

1 Introduction

Nowadays smartphones became a viable computation platform to implement indoor localization in GPS-denied environments. Indoor localization functionality for mobile devices is commercially available using services provided by such companies as Skyhook [22], which maintain WiFi and cellular fingerprint databases for specific locations. However, such solutions require to survey the area in order to obtain a signal strength map, which is time consuming and expensive. Moreover, the user has to have a persistent Internet connection to the database provider. Therefore, we are interested in self-localization solutions for smartphones that do not require laborious surveying of the locations, but use opportunistic sensing of signals from the ubiquitous wireless networks to enable

S. Sigg et al. (Eds.): MobiCASE 2015, LNICST 162, pp. 280–298, 2015.
DOI: 10.1007/978-3-319-29003-4_16

pedestrian navigation in unknown, cluttered environments. One of the possible approaches is to use WiFi fingerprinting without a presurveyed database of places. This solution can be then combined with visual appearance-based location verification algorithm, as in [19]. However, such a system is able to tell the user only his discrete location, and cannot provide a continuous pose estimate to a pedestrian. Thus, we propose a solution that combines WiFi fingerprinting with the Pedestrian Dead Reckoning (PDR) for continuous localization.

The problem we have to deal with is how to combine the user's trajectory estimate obtained from a dead reckoning algorithm exploiting the inertial sensors and magnetometer of a smartphone, and the WiFi signal fingerprinting method, which allows for detection of already visited places. Thus, the system has the ability to close a loop, if the user re-visits a location that was previously associated with a fingerprint. The information coming from both sources is very different as to the spatial uncertainty characteristics, but both sources define useful constraints as to the current location of the user. We formulate the smartphone localization problem in a graphic model, as optimization of a graph of constraints that are related to the sensory observations. The resulting factor graph has a sparse structure and can be optimized in real-time on the smartphone using the Android port of the g^{2}o general graph optimization library [15] applying the Preconditioned Conjugate Gradient (PCG) algorithm.

Our solution assumes that only a small fraction of the WiFi networks existing at the given area has been pre-labeled and anchored to the floor plan. We do not assume a dense distribution of the WiFi Access Points (APs), and we can work with few APs in the area. The proposed system does not need an off-line learning or calibration stage. Although full simultaneous localization and mapping (SLAM) solutions based on WiFi signals that enable to obtain a map of the APs in the area are known from the literature [9], they usually assume an environment densely populated by the WiFi networks and are computation intensive, thus they are unsuitable for implementation on a smartphone. Thus, a solution which requires to know only few APs in the area is advantageous, as the APs can be easily anchored to a known floor plan uploaded to the smartphone (we use standard building blueprints digitized to images). The WiFi nodes may be attached to known locations in the global coordinates, or can be discovered opportunistically, and treated as labelled poses. All computations are performed on a smartphone.

2 Related Work

The rapid proliferation of high-end smartphones has brought a growing interest among the researchers in using these devices for indoor navigation. One of the possibilities is to use the camera of a smartphone and a monocular visual odometry algorithm implemented on Android platform to estimate user motion. However, existing research [6] demonstrated that real-time operation of a simple visual odometry pipeline is only possible if there is no significant motion blur and there are not sudden orientation changes, which makes it an impractical solution.

Therefore, we are interested in exploring other sensing modalities for smartphone-based navigation. Various forms of dead reckoning have been considered for mobile devices, e.g. as an aid to visually impaired users [21]. Reliable pedestrian dead reckoning may be obtained by combining robust attitude estimation of a smartphone [7] with a smartphone-based pedometer [25]. The literature is also rich in papers concerning using WiFi for indoor localization. A survey of the possible localization methods is given in [17]. The WiFi triangulation uses three or more line-of-sight AP positions to determine position of the receiver based on the measured signal strength of each network [1]. To localize in areas where the signal strength from particular APs may vary due to occlusions and attenuation the WiFi fingerprinting approach is more appropriate [2]. This technique was already demonstrated to be feasible in smartphones [16], and was applied to localize elderly patients by means of Android application [10].

Also systems employing more than one sensor type and localization method have been investigated. Some researchers used sensors that were equivalent to those employed in mobile devices, but did not perform the experiments on actual smartphones/tablets, like Quigley *et al.* [20], who investigated combination of vision, accelerometer, and WiFi signal-strength measurements for localization. This approach required also to build a priori map of the environment using a mobile robot equipped with a laser scanner. The WiFi fingerprinting technique is a basis for the multimodal localization solution for smartphones presented in [18]. Dead reckoning from smartphone-embedded inertial sensors is used together with an independent position estimate from WiFi fingerprinting and a pre-built environment map to localize the user in [14]. In this system a particle filter is applied to obtain the final pose estimate. Similarly, Wu *et al.* [24] applied off-line a particle filter to fuse WiFi and inertial data from a smartphone for indoor localization. To implement SLAM relying on the WiFi strength signals also other off-line optimization frameworks have been applied. Ferris *et al.* [5] used Gaussian processes to obtain a map of WiFi signal strength in a given area, avoiding to model explicitly the radio signal propagation. A GraphSLAM-like algorithm is employed in [9] for localization using only the WiFi signal strength. However, the off-line approaches cannot be implemented on a smartphone and used for real-time indoor navigation. A more computation efficient approach for the fusion of WiFi and inertial data from a smartphone was demonstrated in [4]. This approach employs Kalman filtering and estimates the user location with respect to the WiFi APs by computing the distances upon a signal propagation model. Unfortunately, using such a model in real indoor environments usually yields highly inaccurate distance measurements, which renders this approach rather impractical.

Recently, factor graphs became an increasingly popular framework for solving real-time SLAM and similar navigation problems [8]. Although most of the applications of this framework to navigation assume a single sensor type, there are notable works demonstrating that factor graphs are also a viable solution for multi-sensor integration. Indelman *et al.* [11] used factor graph formulation for information fusion in IMU-based navigation systems with constrained computation resources. Research within the DARPA All Source Positioning and

Navigation project demonstrated that the extended, sliding-window factor graphs can be used to integrate data from many sensor types in a plug-and-play manner [3]. However, none of these works tackled the problem of modelling WiFi fingerprinting constraints in the factor graph framework, neither demonstrated real-time performance on such a low-power and resource constrained device as a smartphone.

3 Indoor Localization in a Smartphone

Modern smartphones provide a variety of sensors, but the proposed graph-based localization scheme focuses on the use of inertial sensors (accelerometer, magnetometer and gyroscope) and data from the WiFi signal scanning. The overall structure of the processing pipeline is presented in Fig. 1. The main modules include: the stepometer that uses data from accelerometers, orientation estimation based on the Adaptive Extended Kalman Filter (AEKF) that fuses data from inertial sensors and mangnetometer, and WiFi fingerprints matching that exploits WiFi signal scans provided by the Android OS. The user poses and localization-relevant constraints estimated by these modules are then used to build a factor graph, which is optimized providing to the user the current pose estimate in real-time.

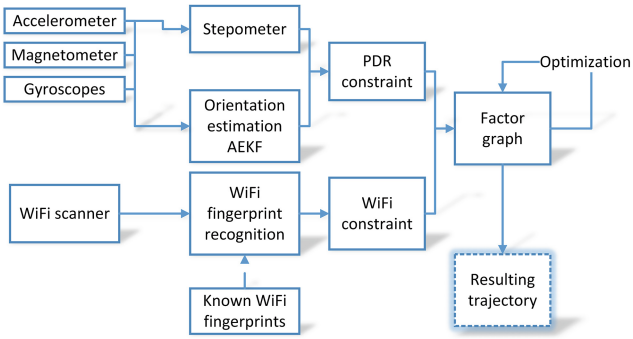

Fig. 1. Processing pipeline of the proposed navigation system

3.1 Stepometer

The stepometer, called also pedometer, is a subsystem that measures the user movement by detecting user's steps. Smartphones are a perfect platform to run stepometer processing, as the modern devices contain accelerometers and enough processing power, while the users keep them close to their bodies. In the presented system, the step detection is performed by computing the FFT of the accelerometer signal in a moving window, which is a simplified version of the

algorithm presented in [12]. To reduce the influence of the device orientation on the step detection, the magnitude of the accelerometer signal is processed. The steps are considered to be detected if the dominant frequency f found by the FFT is between the values of 1.3Hz and 2.0Hz. Knowing the person's step length s (default value measured in tests for an adult male person was 0.65 m), it is possible to compute the covered distance using:

$$d_i = s \cdot f_i \cdot \frac{n}{n_w}, \tag{1}$$

where d_i denotes the covered distance in i-th iteration, f_i is the found dominant frequency for the moving window in i-th iteration, n is the number of new accelerometer measurements since the last stepometer output computation, and n_w is the number of accelerometer measurements in the processing window. If the total covered distance value d is sought, it is computed by summing up the partial distances d_i from each processing iteration.

The precision of the stepometer defined by (1) depends on the precision of step size s estimate and the moving window size n_w. Larger window sizes result in more precise frequency estimation, but with greater latency in the detection of user movement or even inability to detect sudden motion. Smaller window size gives an ability to detect those rapid movements, but results in more false-positive step detections. In the proposed system the window size is equal to 512 measurements. On the Android-based device this value allowed to easily utilize DFT computations, as the window size is a power of 2, and resulted in the computation time-window of 2.5 sec. The settings allowed to detect sudden movements, while providing satisfactory precision in step recognition.

3.2 Orientation Estimation

The orientation estimation of the smartphone is performed by fusing data from the accelerometers, magnetometer and gyroscope, which are present in any modern mobile device. The method used for orientation estimation is based on the AEKF algorithm implemented using quaternions, and has been presented in [7]. The state of the AEKF subsystem consist of four quaternion values and estimates of three gyroscope biases. The magnetometer and accelerometer data are combined to estimate the smartphone coordinate system in the ENU global reference system (X – East, Y – North, Z – Up). The covariance matrices used in the system were estimated by the PSO optimization algorithm [13], and the AEKF parameter values used here are consistent with the optimal parameters presented in [7].

3.3 WiFi Fingerprinting

In the proposed system, the WiFi fingerprint matching approach is utilized [1]. It is possible to: (i) compare the current WiFi scan to a pre-existing database of fingerprints taken at known locations, (ii) compare the current WiFi scan to

the WiFi fingerprints recorded earlier during the system operation. The former method assumes that before the start of a navigation task a small set of WiFi scans is taken in known poses of the user, and these fingerprints are stored in the memory of the smartphone. The procedure to obtain these WiFi fingerprints is straightforward and fast, as only WiFi scans in few selected locations are needed. The latter operation mode assumes no prior knowledge of the environment as only the WiFi scans taken during the system operation are compared. The matching of WiFi fingerprints provides localization constraints equivalent to the loop closure mechanism in SLAM, as it detects previously visited locations (either known a priori or discovered) and allows to reduce the dead reckoning drift.

Both of the proposed modes rely solely on comparing the two WiFi scans $\mathcal{X}$ and $\mathcal{Y}$. One of the scans $\mathcal{X}$ is the currently scanned, and the second scan $\mathcal{Y}$ comes from the stored database, either created in advance, or discovered during the system operation. The Euclidean norm between signal strength values of networks found in both scans is used to compare the fingerprints:

$$d(\mathcal{X}, \mathcal{Y}) = \sqrt{\frac{\sum_{i=1}^{N} (\mathcal{X}_i - \mathcal{Y}_i)^2}{N}}, \tag{2}$$

where $\mathcal{X}_i$ and $\mathcal{Y}_i$ are the strengths of i-th shared network between both scans, $\mathcal{X}$ and $\mathcal{Y}$, while N is the number of shared networks found in both scans. It is assumed that if the Euclidean distance $d(\mathcal{X}, \mathcal{Y})$ is less than a threshold d_{WiFi} then the fingerprints match. In the proposed solution $d_{\text{WiFi}} = 8$ dBm was used. If the Euclidean distance test was passed, the number of networks N shared between the two scans $\mathcal{X}$ and $\mathcal{Y}$ is compared to the number of WiFi networks detected in each scan, N_X and N_Y, respectively. If $N > p \times N_X$ and $N > p \times N_Y$ then the WiFi fingerprint matching is considered to be correct, where p denotes an experimentally set parameter, which may vary between environments. In the proposed system it is set to 0.75.

If the WiFi scans taken during system operation are added to the database, the number of necessary comparisons grows linearly with the number of records in the database. We deal with this problem by having a dedicated thread responsible for WiFi place recognition, which compares WiFi scans prioritized in a queue according to the difference in their IDs. If the database has grown to a size that prevents real-time operation, the fingerprints with lowest priority are omitted in the comparison.

4 Factor Graph Representation

The main contribution of this paper is the data fusion scheme using the factor graph representation and non-linear least-squares optimization. This formulation of the SLAM problem is considered the state of the art in mobile robotics [8], but is novel in the context of pedestrian navigation with sparse WiFi fingerprints, and in the context of real-time, Android-based implementation.

To find the most plausible sequence of the nodes (user poses or map positions with known WiFi scans) $\mathbf{p}_i \in \mathrm{SE}(2)$, $i = 1 \dots n$ satisfying k constraints existing in the factor graph the following function is minimized:

$$\underset{\mathbf{p}}{\operatorname{argmin}} F = \sum_{i=1}^{n} \sum_{j=1}^{k} e_j(\mathbf{p}_i, \mathbf{m}_{ij})^T \mathbf{\Omega}_{ij} e_j(\mathbf{p}_i, \mathbf{m}_{ij}), \tag{3}$$

where $e_j(\mathbf{p}_i, \mathbf{m}_{ij})$ is the error function of j-th constraint $j = 1 \dots k$, evaluated for the estimated pose of the node and the measured pose of the node stemming from the j-the measurement $\mathbf{m}_{ij}$ related to this pose. The poses $\mathbf{p}$ for known WiFi scans are anchored to the map coordinate system and cannot be moved by optimization process. The measurement means either the user motion estimate (from PDR) or the measurement resulting from two matching WiFi fingerprints. The information matrix $\mathbf{\Omega}_{ij}$ models uncertainty of the computed error.

4.1 PDR Motion Constraints

To represent the motion constraints imposed by the PDR-estimated motion of the user, it was decided to use the *EDGE:SE2* factor graph edge defined in the g^{2}o library. This edge represents the measurement between two 2D poses as: $\mathit{EDGE{:}SE2} = (\Delta x, \Delta y, \Delta\Theta)$, where Δx and Δy are the measured distances along the X and Y axis w.r.t. the local coordinate system. The $\Delta\Theta$ is the difference between the orientations of both connected user poses. Because the PDR employs two independent subsystems to estimate the covered distance and the orientation change, the information matrix $\mathbf{\Omega}^{\mathrm{pdr}}$ was assumed to have the form:

$$\mathbf{\Omega}^{\mathrm{pdr}} = \begin{bmatrix} I_{2\times 2} & 0_{2\times 1} \\ 0_{1\times 2} & k \end{bmatrix}, \tag{4}$$

where the k parameter was determined experimentally, to allow the g^{2}o optimization software to trust more in the pedometer distance than in the orientation estimate, which can degrade due to very sharp turns or in presence of magnetic fields in the vicinity. We got best results with $k \geq 10$.

4.2 WiFi Fingerprint Constraints

The WiFi fingerprint information represents some belief that the user is located in a vicinity of an already visited or previously mapped location, but such a constraint cannot be easily converted into an existing g^{2}o edge, due to its topological relation rather than metric measurement nature. Therefore, we propose a novel factor graph edge, which directly represents any information that can be understood as vicinity measurement. Such vicinity measurement is represented as an edge with the error function:

$$\mathrm{Err}(x) = \begin{cases} 0 & d_{\mathrm{err}}(x) < d_{\min} \\ \mathrm{Err}_{\max} & d_{\mathrm{err}}(x) > d_{\max} \\ \mathrm{Err}_{\max} \frac{d_{\mathrm{err}}(x) - d_{\min}}{d_{\max} - d_{\min}} & \text{otherwise.} \end{cases} \tag{5}$$

The error function (5) is presented in Fig. 2. The edge formulated this way is deactivated in case of small distances, which means that two nodes are close enough to be indistinguishable by means of WiFi fingerprint matching. Then, when the error considered as the Euclidean distance between the nodes is greater then the $d_{\min}$ threshold, the edge is activated. The error increases and the optimization procedure tries to reduce that error by moving the two considered nodes closer. As the uncertainty in matching WiFi fingerprints is isotropic, we set the information matrices of the WiFi-related constraints $\mathbf{\Omega}^{\text{wifi}}$ to identity.

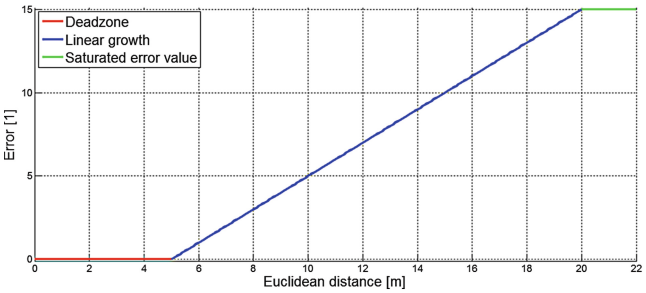

Fig. 2. The error function used for WiFi edge in graph-based optimization

4.3 Implementation Details

The proposed system was implemented for mobile devices with the Android OS (versions above 4.0), and finally tested on a Samsung Galaxy Note 3 with Android 4.4.4. The program is divided into 4 threads: (i) processing inertial data for orientation estimation and stepometer; (ii) fingerprints matching processing the WiFi scans and finding corresponding scans in the database and in the queue of scans taken in-motion; (iii) factor graph management polling data from subsystems and performing optimization; (iv) user interface. The inertial sensors data used in orientation estimation are processed with the maximal available frequency of 200 Hz whereas stepometer processing is performed at 5 Hz. The WiFi scans are performed as fast as possible (0.75 Hz). The implementation of the system is divided between JAVA and C++ (NDK) parts as it allowed us to combine C++ efficiency and possibility to use existing code with Java ease to create GUI and access available sensors.

5 Experiments

5.1 Experimental Setup

The experiments to evaluate the proposed fusion scheme were performed in two public buildings at the Poznań University of Technology (PUT) campus – the

Lecturing Center and the Mechatronics Center, shortly abbreviated to "LC" and "MC", respectively. The user equipped with a smartphone was asked to move around a building and the Android-based system was estimating the trajectory. It should be noted that the user was not stopping while moving along the trajectory to obtain the data, and therefore the WiFi scans taken in-motion cannot be easily associated with a single user position. However, the experiments conducted this way closely simulate real-life use cases of the pedestrian navigation system. The resulting trajectories are presented against the building floor plans, which allows for easy visual assessment of the correctness of the trajectories.

5.2 Pedestrian Dead Reckoning Results

The experiments started with tests focusing on evaluating the accuracy of the proposed PDR subsystem. The first test was performed inside the LC building, which hosts a large lecturing auditorium surrounded by an open space (lobby area) at the ground floor. The planned user path made a semi-loop around the auditorium, which however could not be closed by the user due to the restricted access to the area of auditorium's backstage. The user moved almost from the staircase on the left side, to the exit of the building on the right side of the floor plan (Fig. 3A). The trajectory obtained using the PDR subsystem (stepometer with orientation estimate) is presented in red, whereas the approximate ground truth trajectory, obtained by referencing to the known objects in the vicinity is depicted in yellow. The total covered distance was approximately 92 m. The trajectory obtained from the PDR subsystem is relatively good, although it can be observed that the user motion estimate is getting worse with the increasing operation time. This drift is caused by the nature of the dead reckoning principle, which accumulates small errors in estimation of the relative displacements along the trajectory. It is also evident, that even a small angle error can have a significant impact on the trajectory, and leads to large error in the Cartesian position of the user.

Due to the limited scale of the experiments in the LC building, it was decided to perform an experiment inside the MC building, which has a considerably different structure, with narrow corridors and less amount of open space. The user was asked to move twice along a rectangular trajectory and finish the experiment at the starting position, thus closing a loop. The obtained trajectory (red) is presented in Fig. 3B whereas the approximate ground truth path is denoted by the thicker yellow line. The total covered distance was approximately 178 m. This time the trajectory yielded by the PDR alone is much worse than in the LC case, due to inaccurate orientation estimation at sharp turns, and perhaps more noisy magnetometer readouts that were influenced by various electric equipment in the labs surrounding the area. The resulting PDR trajectory suffers from large drift and the second loop of the rectangular trajectory apparently does no match the first one.

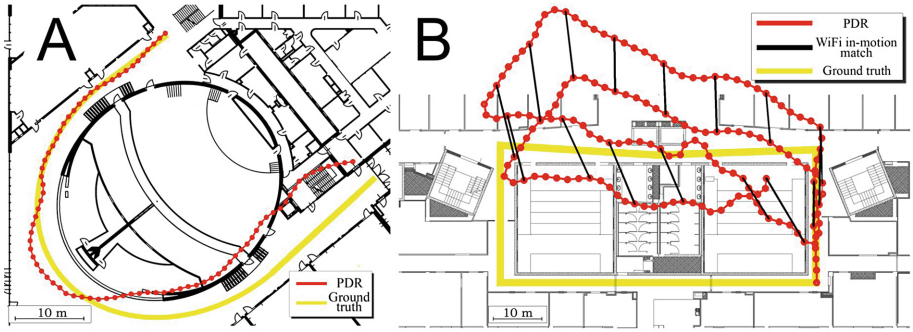

Fig. 3. PDR trajectories (red) estimated in the LC (A) and MC (B) buildings compared to the estimated pedestrian movement (yellow). The discovered WiFi-fingerprint matches (black lines) are shown for the MC trajectory (Color figure online).

5.3 PDR Supported by Matching of Discovered WiFi Fingerprints

To alleviate the odometry drift, the proposed system introduces the factor graph representation of the trajectory, which can accommodate additional information from WiFi fingerprints. The constraints stemming from WiFi fingerprints are discovered between the poses where WiFi scans were taken while the user was in motion. Unfortunately, those scans cannot be precisely associated to unique poses on the trajectory due to the long scanning time on the smartphone used in experiments. The Samsung Galaxy Note 3 scans 2.5 GHz and 5 GHz WiFi frequencies looking for WiFi APs, which takes about 4 sec. for a full scan. Therefore, for the discovered WiFi edges the deadzone parameter $d_{\min}^{\text{disc}}$ of the error function (5) was increased to 6 meters. This value captures the additional uncertainty in the location of the pose to which the fingerprint is anchored on the trajectory. Also, a discovered WiFi constraint can be spawned only to an already existing user pose. Therefore, those constraints cannot reduce the trajectory estimation drift that has mounted before the reference (revisited) pose was added to the trajectory (and the factor graph). However, the discovered constraints can keep a multi-loop trajectory more consistent, reducing the drift for motion along already covered paths. The discovered WiFi links are presented as black lines in Fig. 3B whereas the trajectory obtained after optimization is presented in Fig. 4.

From the presented trajectory, it is evident that the in-motion discovered WiFi constraints, which do not need any a priori knowledge of the environment nor the layout of the APs, reduce the trajectory estimation drift in case of longer operation with the smartphone. Unfortunately, these constraints do not guarantee a trajectory estimate that reasonably matches the ground truth path, due to the PDR errors that mounted before the first local loop closure between the fingerprints was discovered.

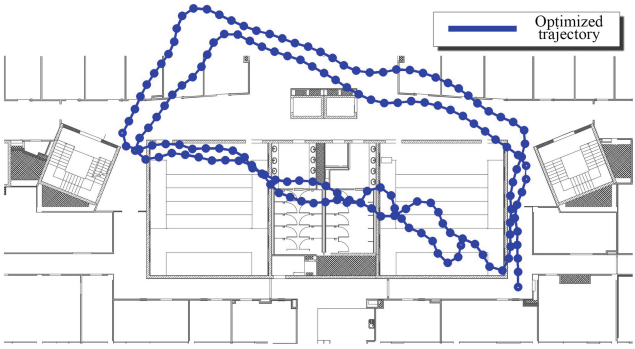

Fig. 4. Optimized trajectory estimate from PDR and WiFi links discovered in the MC experiment

5.4 Constraints from WiFi Fingerprints with Known Positions

Due to the inability of the in-motion discovered WiFi constraints to completely cancel the drift of the estimated trajectory other constraints related to the WiFi were considered. It is pretty evident that the drift in the PDR trajectory cannot be reduced significantly without prior knowledge of the locations of the APs in the environment. While these locations may be estimated by a SLAM algorithm, such an approach requires to survey the site prior to using the smartphone navigation system. We consider this too laborious and time consuming, and therefore we investigate an approach which uses only a minimal set of few APs at known positions. Those APs should be located in areas critical for PDR navigation, e.g. right-angle turns of corridors or crossroads. The few AP with known WiFi fingerprints and positions were determined prior to the experiment by a person who took 4 scans for each area of interest, and then simply pinpointed that location on the provided floor plan. The reference scans were taken with the user standing still, therefore they could be precisely associated to poses/places.

The experiments started with a trajectory taken inside the LC building as presented in Fig. 5A, where the PDR estimate (red) was presented with positions of the known WiFi fingerprints (blue crosses) and the discovered WiFi fingerprint matches (green lines) between scans taken in-motion and the known WiFi fingerprints stored in a database. The user was asked to keep moving continuously and therefore the WiFi constraints to known places could not always be detected, as in real-life scenarios.

When the factor graph of nodes and constraints is created, it is optimized on the smartphone by using the g^{2}o library. The resulting graph, representing the best estimate of the trajectory, is presented in Fig. 5B. To distinguish the graphs before and after optimization, the used colors were reversed – the blue circles represent estimated user position whereas red crosses correspond to the positions of the WiFi fingerprints stored in database.

The obtained, optimized trajectory is similar to the PDR estimate in the initial part, as from the beginning the odometry estimate is accurate and therefore,

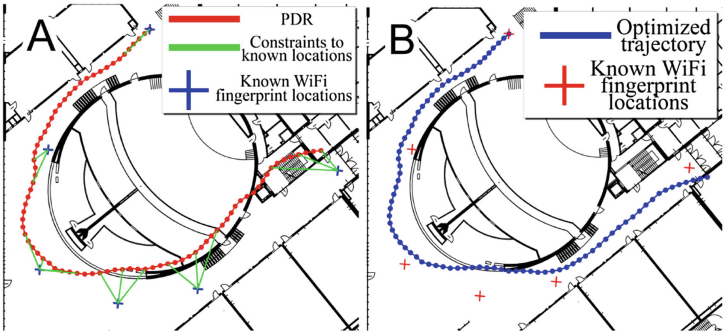

Fig. 5. PDR trajectory prior to optimization with found WiFi links (A) compared with resulting, and optimized trajectory with $d_{\min}^{map} = 6$ meters (B) in LC building

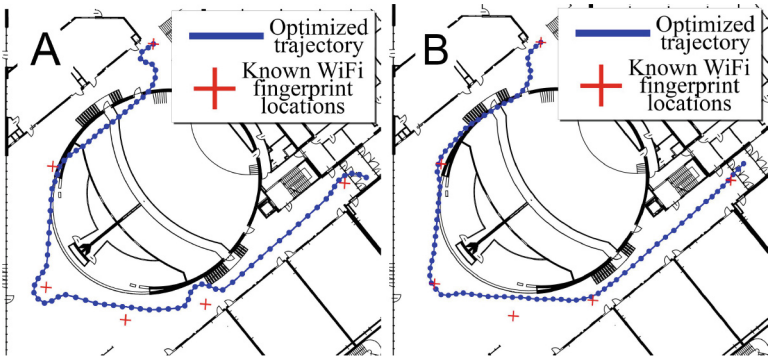

Fig. 6. Optimization trajectories obtained for differently tuned deadzone values: 3 meters (A) and 4.5 meters (B) in LC building

the WiFi constraints in the factor graph are not active. When the user moves further from the starting point, the odometry suffers from the accumulating errors, and the WiFi constraints get activated. The performed optimization results in a trajectory estimate which is close to the ground truth.

The critical parameter that needs to be properly tuned is the deadzone radius of the constraint related to the known WiFi fingerprint $d_{\min}^{\text{map}}$. Results for alternative deadzone values are presented in Fig. 6. The choice of $d_{\min}^{\text{map}}$=3 m (Fig. 6A) results in an optimized trajectory that is artificially curved around positions of the known APs. If a too strict deadzone value is selected, the WiFi constraint returns a non-zero error value even though there is no need to additionally correct the PDR estimate. A similar effect can be observed for $d_{\min}^{\text{map}} = 4.5$ meters where a single WiFi fingerprint match curves the trajectory at the beginning resulting in a trajectory estimate worse than the one using only the PDR. The proper choice of the deadzone radius $d_{\min}^{\text{map}}$ should depend on the environment

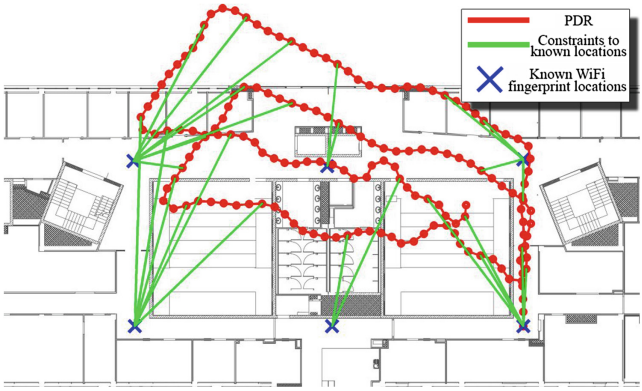

Fig. 7. Trajectory obtained with PDR and discovered WiFi links inside the MC building while moving twice along a rectangular trajectory

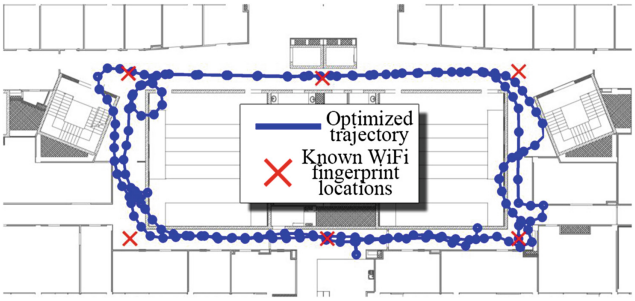

Fig. 8. Optimized trajectory with PDR and WiFi database edges inside MC building

characteristics, as in more cluttered environments smaller deadzone radius values result in better trajectories.

A similar experiment with the known WiFi fingerprints was performed in the MC building. The found WiFi constraints are visualized in Fig. 7. On the trajectory of the user, six places were chosen to record WiFi scans that can be used by the WiFi fingerprint localization system. In this experiment the in-motion discovered WiFi constraints between poses along the trajectory were neglected, in order to clearly present the benefits due to the constraints between the user pose and the WiFi fingerprints at known locations. The post-optimization trajectory is presented in Fig. 8. Again, the colors were reversed to represent optimized trajectory.

The optimized trajectory obtained from the proposed system almost perfectly resembles the real trajectory of the user. The WiFi fingerprint measurements with known positions placed on the crossroads allowed to properly constrain the trajectory. Even though the presented results are sufficient when it comes to pedestrian navigation, it is also possible to constraint the trajectory with

additional in-motion discovered WiFi edges. Such a system should combine all of the possible sources of information and therefore provide a better trajectory. The trajectory recorded in the previous experiment repeated with the additional discovered WiFi links (denoted using black color) is presented in Fig. 9. In this experiment, it was decided to reduce the number of known WiFi fingerprints to four places and therefore create a more challenging environment for the proposed system.

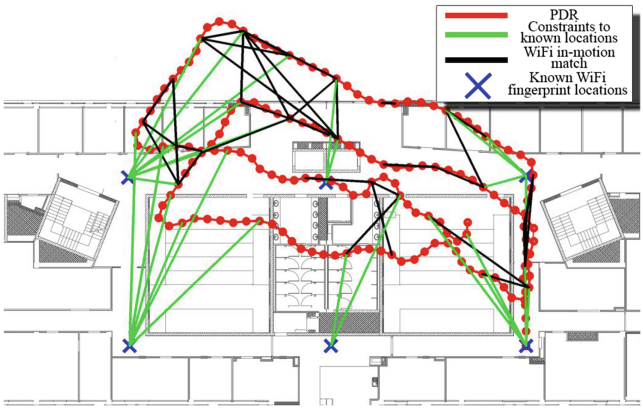

Fig. 9. Created graph with PDR edges (red), discovered WiFi edges (black), and WiFi edges to places found in the floor plan (green) (Color figure online).

Before performing the optimization, it is also important to set proper, different deadzone values of the in-motion discovered constraints and constraints to WiFi fingerprints of known position. Different parametrization has to be made due to the fact that both WiFi scans establishing a discovered constraint are taken in-motion, whereas one scan in the constraint related to a known WiFi location is static, and thus located more precisely. Therefore, we assumed and experimentally tested that the ratio of the discovered WiFi deadzone $d_{\min}^{\text{motion}}$ to the WiFi of known location deadzone $d_{\min}^{\text{map}}$ should be equal to 2. The resulting optimized trajectory with both types of WiFi constraints is presented in Fig. 10. The resulting system performs slightly better than the one using only the known WiFi locations. The discovered WiFi constraints allowed to improve the resulting trajectory, which is believed to come from the fact that the trajectory is more constrained between the repeated loops of similar shape.

The precision of the proposed system depends on the number and the location of the known WiFi positions used for smartphone localization. To demonstrate the difference in trajectory estimate, the user was asked to move for 119 meters in MC building and different configurations on known WiFi APs were tested in four experiments. The first trajectory with 15 known WiFi positions is presented in Fig. 11A. The obtained trajectory is very close to the real trajectory as there

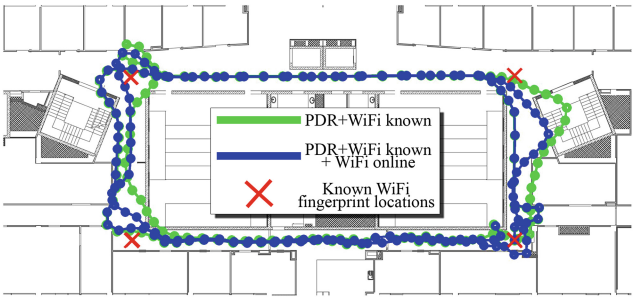

Fig. 10. Comparison of a trajectory obtained with a factor graph created using PDR and WiFi edges to known places (green), and trajectory obtained with a factor graph including constraints from PDR, WiFi edges to known places, and discovered WiFi edges (blue) (Color figure online).

is enough known WiFi scans to correct the trajectory from PDR. The same trajectory with 6 known WiFi APs (Fig. 11B) still resembles the real path, although it is possible to observe increasing error drift in these parts where WiFi information is unavailable. The important factor is also the location of the WiFi known positions on the path. Randomly choosing 6 WiFi APs (as in Fig. 11C) results in a useless trajectory as the WiFi information is not available in situations when PDR has the greatest error (especially just after sharp turns at the junctions of the corridors). Therefore, it is important to have WiFi information on possible crossroads. It is also important to have enough locations with known WiFi scan as insufficient number of those positions results in imprecise trajectory as presented in Fig. 11D. The PDR system of the proposed solution can be used to estimate the trajectory between two locations with known WiFi scans, but the lack of WiFi information (as in the end of trajectory in Fig. 11D) results in accumulation of error, mostly due to the imprecise orientation estimation inside a building.

To enable quantitative comparison between the obtained trajectories we apply the translational trajectory error metric, which is similar in concept to the ATE (Absolute Trajectory Error) proposed in [23] and commonly used in robotics. Similarly to the ATE our translational trajectory error compares the absolute distances between the estimated $\mathbf{P}$ and the ground truth $\mathbf{Q}$ trajectory. At first we map the estimated trajectory onto the ground truth trajectory by computing the rigid-body transform $\mathbf{S}$ that is the least-square solution to the alignment problem [23]. Then, the trajectory error is computed as $\mathbf{F}_i = \mathbf{Q}_i^{-1}\mathbf{S}\mathbf{P}_i$, for each i-th trajectory node. We extract the translational component of $\mathbf{F}_i$ and compute the Root Mean Square Error (RMSE) or mean error, along with the standard deviation. However, while the ATE error is computed over all time indices of $\mathbf{F}_i$ for the matching (i.e. time-synchronized) nodes of the reference and the estimated trajectory, we compute $\mathbf{F}_i$ as the error between the given node of $\mathbf{P}$ and the closest (in the Euclidean sense) point on the $\mathbf{Q}$ trajectory. This difference is caused by the fact, that working over long paths in natural

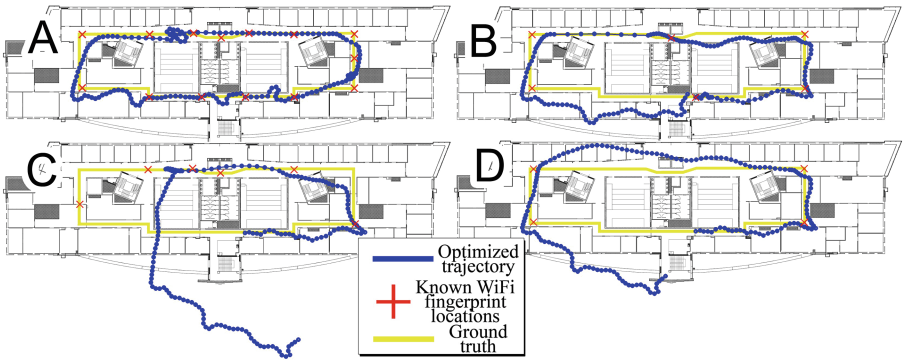

Fig. 11. Trajectories obtained with factor graphs created using PDR and WiFi edges: 15 known WiFi points (A), 6 known WiFi points anchored in critical locations of the floor plan (B), 6 randomly chosen known WiFi points (C), only 4 known WiFi points in critical locations (D)

indoor environments we cannot use an external motion capture system or GPS to obtain the ground truth trajectory, thus we cannot establish direct correspondence between the nodes of the estimated trajectory, and the ground truth path, which is surveyed manually by referencing to the walls and objects of known positions in the floor plan. The quantitative results for the experiment demonstrated in Fig. 11 are shown in Table 1.

Table 1. Accuracy of the recovered user paths with different choice of the number and location of the known WiFi Access Points

Choice of the known WiFi scan locations	Translational trajectory error			
	RMSE [m]	mean [m]	std. dev. [m]	max. [m]
15 (all known)	1.18	0.87	0.80	3.71
6 at critical places	2.33	1.74	1.54	5.38
6 chosen randomly	11.93	7.42	9.36	29.04
4 at critical places	5.60	3.88	4.05	14.92

The lowest RMSE error of 1.18 m is observed for the experiment with 15 know WiFi scans. Reducing the number of known WiFi scans to 6 yields a trajectory of RMSE error equal to 2.33 m, which is still acceptable for most of pedestrian navigation purposes. However, the maximal error increases to 5.38 m as the system depends for much longer periods on the noisy PDR estimate. The choice of 6 random locations of known WiFi scans results in the RMSE trajectory error of 11.93 m, which is useless for any application. More precise results are observed for a smaller number of known WiFi APs (only 4), but placed at locations critical for localization.

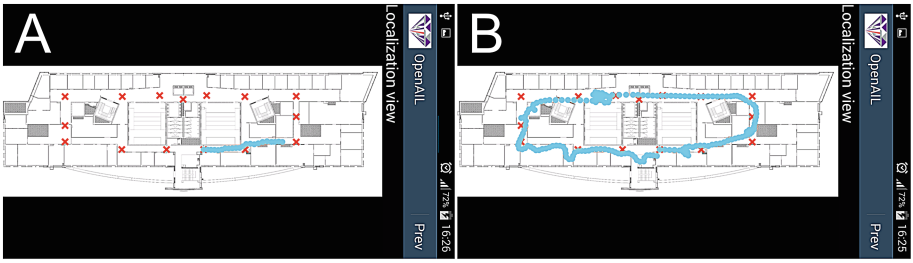

Fig. 12. Visualization of the user trajectory (cyan) on the Samsung Galaxy Note 3 in the MC experiment with places of known WiFi scans (red): during the experiment (A), and after g^2o optimization (B) (Color figure online).

All of the presented results were obtained on the real smartphone as presented in Fig. 12 and the code of the solution is publicly available.[1]

6 Conclusions

The presented graph-based representation of data coming from inertial and WiFi sensors presents an alternative, flexible and computation efficient way of data fusion when compared to typical approaches based on EKF or particle filters. The factor graph representation allows to easily model the error function to suit the uncertainty characteristics of measurements. In our system this representation allows to model the deadzone in the localization of places by using the WiFi fingerprint matching technique. The experiments proved that the stepometer with orientation estimation make a reasonable PDR system on the Android platform, but the user's position estimate from dead reckoning inevitably has a drift. The WiFi fingerprint matching approach allows to alleviate the negative effects of odometry drift and therefore provides much more precise trajectory estimates, even for long paths in complicated indoor environments.

The proposed system was evaluated when merging inertial and WiFi information, but it can be easily extended to incorporate information from additional sources, such like the smartphone's camera, which will be the main focus of further research.

Acknowledgment. This work is financed by the Polish Ministry of Science and Higher Education in years 2013–2015 under the grant DI2012 004142.

References

1. Bahl, P., Padmanabhan, V.N.: RADAR: an in-building RF-based user location and tracking system. In: Proceedings of Joint Conference of the IEEE Computer and Communications Societies, pp. 775–784 (2000)

[1] https://github.com/LRMPUT/DiamentowyGrant.

2. Biswas, J., Veloso, M.: WiFi localization and navigation for autonomous indoor mobile robots. In: Proceedings of IEEE International Conference on Robotics & Automation, Anchorage, pp. 4379–4384 (2010)
3. Chiu, H.-P., Zhou, X., Carlone, L., Dellaert, F., Samarasekera, S., Kumar, R.: Constrained optimal selection for multi-sensor robot navigation using plug-and-play factor graphs. In: Proceedings of IEEE International Conference on Robotics and Automation, Hong Kong, pp. 663–670 (2014)
4. Chen, Z., Zou, H., Jiang, H., Zhu, Q., Soh, Y., Xie, L.: Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. Sensors **15**, 715–732 (2015)
5. Ferris, B., Fox, D., Lawrence, N.: WiFi-SLAM using Gaussian process latent variable models. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 2480–2485 (2007)
6. Fularz, M., Nowicki, M., Skrzypczyński, P.: Adopting feature-based visual odometry for resource-constrained mobile devices. In: Campilho, A., Kamel, M. (eds.) ICIAR 2014, Part II. LNCS, vol. 8815, pp. 431–441. Springer, Heidelberg (2014)
7. Gośliński, J., Nowicki, M., Skrzypczyński, P.: Performance comparison of EKF-based algorithms for orientation estimation on Android platform. IEEE Sens. J. **15**(7), 3781–3792 (2015)
8. Grisetti, G., Kümmerle, R., Stachniss, C., Burgard, W.: Tutorial on graph-based SLAM. IEEE Intell. Transp. Syst. Mag. **2**(4), 31–43 (2010)
9. Huang, J., Millman, D., Quigley, M., Stavens, D., Thrun, S., Aggarwal, A.: Efficient, generalized indoor WiFi GraphSLAM. In: Proceedings of IEEE International Conference on Robotics & Automation, Shanghai, pp. 1038–1043 (2011)
10. Husen, M.N., Lee, S.: Indoor human localization with orientation using WiFi fingerprinting. In: Proceedings of ACM International Conference on Ubiquitous Information Management and Communication, Siem Reap (2014)
11. Indelman, V., Williams, S., Kaess, M., Dellaert, F.: Information fusion in navigation systems via factor graph based incremental smoothing. Rob. Auton. Syst. **61**(8), 721–738 (2013)
12. Inoue, S., Hattori, Y.: Toward High-level activity recognition from accelerometers on mobile phones. In: Proceedings of IEEE International Conference on Internet of Things, and Cyber, Physical and Social Computing, Dalian, pp. 225–231 (2011)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth, pp. 1942–1948 (1995)
14. Kothari, N., Kannan, B., Dias, M.B.: Robust indoor localization on a commercial smart-phone. Technical report CMU-RI-TR-11-27, Carnegie-Mellon University, Pittsburgh (2011)
15. Kümerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g^2o: a general framework for graph optimization. In: Proceedings of IEEE International Conference on Robotics & Automation, Shanghai, pp. 3607–3613 (2011)
16. Liu, H., et al.: Accurate WiFi based localization for smartphones using peer assistance. IEEE Trans. Mob. Comput. **13**(10), 2199–2214 (2013)
17. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning: techniques and systems. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **37**(6), 1067–1080 (2007)
18. Martin, E., Vinyals, O., Friedland, G., Bajcsy, R.: Precise indoor localization using smart phones. In: Proceedings of ACM International Conference on Multimedia, pp. 787–790 (2014)
19. Nowicki, M.: WiFi-guided visual loop closure for indoor localization using mobile devices. J. Autom. Mob. Rob. Intell. Syst. (JAMRIS) **8**(3), 10–18 (2014)

20. Quigley, M., Stavens, D., Coates, A., Thrun, S.: Sub-meter indoor localization in unmodified environments with inexpensive sensors. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots & Systems, Taipei, pp. 2039–2046 (2010)
21. Riehle, T., Anderson, S., Lichter, P., Whalen, W., Giudice, N.: Indoor inertial waypoint navigation for the blind. In: Proceedings of International Conference on IEEE Engineering in Medicine and Biology Society, pp. 5187–5190 (2013)
22. Skyhook. http://www.skyhookwireless.com/
23. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots & Systems, Vilamoura, pp. 573–580 (2012)
24. Wu, D., Xia, L., Mok, E.: Hybrid location estimation by fusing WLAN signals and inertial data. In: Liu, C. (ed.) Principle and Application Progress in Location-Based Services. LNGC, pp. 81–92. Springer, Berlin (2014)
25. Wu, S.-S., Wu, H.-Y.: The design of an intelligent pedometer using Android. In: Proceedings of International Conference on Innovations in Bio-inspired Computing and Applications, pp. 313–315 (2011)