Hybrid Indoor Localization using GSM Fingerprints, Embedded Sensors and a Particle Filter

Ye Tian, Bruce Denby
Université Pierre et Marie Curie and SIGMA Laboratory
Paris, France
ye.tian@etu.upmc.fr, denby@ieee.org

Iness Ahriz
LAETITIA/CEDRIC Laboratory
Conservatoire National des Arts et Métiers
Paris, France
iness.ahriz@cnam.fr

Pierre Roussel, Gérard Dreyfus
SIGMA Laboratory, ESPCI ParisTech
Paris, France
pierre.roussel@espci.fr, gerard.dreyfus@espci.fr

***Abstract*—The article presents an indoor localization scheme for mobile devices based on GSM Received Signal Strength fingerprints combined with embedded sensor information and an area site map. Displacements of a mobile user are first estimated using a sensor dead-reckoning approach that adapts stride length to different users and environments, and a dynamically switched orientation estimation scheme responding to orientation changes of the mobile device. Positions derived from GSM fingerprints, along with constraints imposed by a site map, are then integrated using a particle filter in order to prevent the accumulation of dead-reckoning errors over time. The study demonstrates that a standard handset with a cellular network connection and embedded inertial sensors can provide a good solution for indoor localization.**

Keywords—indoor localization; fingerprinting; support vector machine; sensor dead-reckoning; particle filter

I. Introduction

Global Navigation Satellite Systems (GNSS), such as GPS and GLONASS, furnish navigation, tracking, and monitoring services in outdoor environments [1]; however, due to propagation effects, they are unable to operate effectively in indoor environments. This situation has led to intense activity in developing indoor localization techniques that provide seamless and ubiquitous services for mobile users [2].

Beacon-based approaches proposed over the past decade include such technologies as infrared [3], Bluetooth [4], Radio-Frequency Identification (RFID) [5], Wireless Local Area Networks (WLAN) [6], Ultra-wideband (UWB) [7], acoustic signals, etc. The need to deploy and maintain an underlying infrastructure unfortunately renders these methods somewhat less desirable. Indoor localization based on fingerprints from ambient radiotelephone networks, such as GSM and CDMA, has also been proposed [8][9]. Indoor localization based on classification of RSS fingerprints of very large number of GSM channels has been reported in [10][11][12], albeit providing a simple room label rather than a physical coordinate.

Beacon-free solutions relying only on sensors such as accelerometers, gyrometers, magnetometers, and barometer, can track users by continuously estimating their displacement from a known starting point. Many such studies, ([13] [14] [15] [16] [17] [18]), invoke dedicated Inertial Measurement Units (IMU) mounted at waist, leg, or head, or, Micro-Electro-Mechanical System (MEMS) embedded in smartphones, with few taking into account the orientation of the phone in practical scenarios, e.g., in a hand or a pocket. Furthermore, due to the drift and low precision of MEMS sensors, integration of sensor readings can result in an unacceptable accumulation of error.

The particle filter has become a powerful tool in location estimation and target tracking systems, that allows to combine localization data from a variety of sources – for example, beacon data, embedded smartphone sensors, and building layout constraints. In this paper, we present a particle filter based indoor localization system that uses room-level positioning results from GSM fingerprinting to correct for accumulated inertial dead-reckoning errors, while also accessing a building map to exclude inaccessible regions and forbid unreasonable movements such as traversing a wall. The approach furthermore incorporates a novel, adaptive stride-length step detection algorithm that can handle arbitrary positions changes of the mobile device. This system is to the best of our knowledge the first to use a particle filter to combine GSM fingerprints with inertial sensors and a site map, and to incorporate an adaptive stride/orientation model.

The article is organized as follows. The GSM fingerprinting algorithm is presented in section II, the sensor dead-reckoning algorithm in section III, and the particle filter data fusion in section IV. An evaluation of results is presented in section V, while conclusions and future perspectives appear in the final section.

II. GSM Fingerprinting Algorithm

GSM is the most widely available cellular telephony standard in the world, with networks of nearly 800 mobile operators deployed in over 220 countries [19]. Here, we benefit from this ubiquitous GSM coverage to carry out indoor localization without the necessity of deploying and maintaining a dedicated infrastructure. RSS values from all 548 carriers in both 900 MHz and 1800 MHz GSM bands are used to create fingerprints to be used for localization.

Room-level indoor localization is considered as a multi-class classification problem, which is usually carried out in two phases:

- An offline training phase, also known as the site survey. In this phase, GSM RSS fingerprints are recorded in each of the rooms and labeled with the corresponding room number. A discriminative model is then built using the training fingerprints and known labels, so as to best separate the training examples into their correct classes (i.e., the correct rooms).

- An online localization phase, in which new fingerprints are given to the localization system and a room number output based on the previously defined model.

Only the offline training phase entails a heavy computational load, whereas during localization, the system needs simply evaluate a small number of discriminant functions. The Support Vector Machine (SVM) classifiers used in our work are deemed appropriate for dealing with very large numbers of variables and training examples due to their built-in regularization mechanism [20]. Before introducing the multi-class SVM classifier used in our work, we first present a pairwise (2-class) SVM classifier.

A. Pairwise SVM classifier

Consider a set of M examples of items belonging either to class A or class B, each example being described by a p-dimensional vector $\mathbf{x}_i$. Further assume that the examples are linearly separable, i.e. that there exists a hyperplane of equation $f(\mathbf{x}) = 0$ that separate all examples without error: $f(\mathbf{x}_i) > 0$ for all examples i belonging to class A and $f(\mathbf{x}_i) < 0$ otherwise. It can be proved that $f(\mathbf{x})$ can be written under the form

$$f(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + \alpha_0 \qquad (0)$$

where the α_i ($i = 0 \cdots M$) are parameters whose values are estimated from the examples; $y_i = +1$ if example i belongs to class A and $y_i = -1$ otherwise.

If the examples are not linearly separable, a "soft-margin" approach can be used to reduce the complexity of the classifier by introducing slack variables ζ_i and performing a tradeoff between accuracy of classification of the training examples and ability to generalize; the price to pay is the introduction of a "regularization" constant C whose value must be chosen appropriately.

To summarize, a GSM environment described by the fingerprint $\mathbf{x}$ is assigned to room A or room B according to the sign of $f(\mathbf{x})$, defined by (). $\mathbf{x}_i$ is the fingerprint dataset entry i, i.e. row i of RSS fingerprint.

B. Decision Rule for Multiclass Classification

When the discrimination problem involves more than two classes, it is necessary, for pairwise classifiers such as SVM, to define a method that allows combining multiple pairwise classifiers into a single multiclass classifier. We applied one-vs-all multiclass classifiers in this paper.

The one-vs-all approach consists of dividing the n-class problem into an ensemble of n pairwise classification problems, each of which is specialized in separating one class from all others. In the first stage, each of the n classifiers is trained separately, and in the second stage, the following decision rule is applied : the outputs of all n classifiers are first calculated and, following the conventional procedure, the predicted class is taken to be that of the classifier with the largest magnitude of $f(x)$ (relation (1)). The one-vs-all technique is advantageous from a computational standpoint, in that it only requires a number of classifiers equal to the number of classes.

The SVM used in our study, were implemented using LIBSVM toolbox[21].

III. Sensor Dead-Reckoning Algorithm

Sensor dead-reckoning aims to estimate the displacement from a previous location, usually consisting of a step detection module, a stride length model, and an orientation estimator. The reliable and widely-used "peak/trough" technique was chosen for step detection [22]. Stride length determination and orientation estimate, however, are critical to estimating user displacement. In order to render our system user-independent and robust against orientation changes, a novel stride length adaptation approach and orientation estimate "switching" scheme were introduced, as described below.

A. Adaptive Stride Length Model

It is necessary to use a value for stride length in order estimate the displacement due to a step. However, stride length varies significantly for different persons and walking styles. The literature contains a variety of models indicating that stride length is related to stride frequency and the "bounce" of the human hip, which we shall refer to as the "frequency model" and "bounce model", respectively [23][24].

To verify these, we performed an experiment to test the impact of stride length, in which subjects walked a specified path several times at different walking speeds and stride frequencies. Step numbers and durations were recorded, while the length of the path was known in advance. The results are shown in Fig. 1. As seen in the figure, stride length has a linear relationship with both step frequency and acceleration amplitude, indicating that the models are reasonable. Most dead-reckoning solutions using existing models must fix the coefficients or obtain them from training on different users and environments.

The "frequency model" necessitates an estimate of step frequency over a previous period of time, which has large errors, and can introduce delay if there is no step detected. In contrast, the accuracy of "bounce model" is easily influenced by environmental differences, such as going up and down stairs. As there is no stride length model that fits all subjects and environments, we propose to use a particle filter to adaptively select the best coefficients from a range of coefficients. The

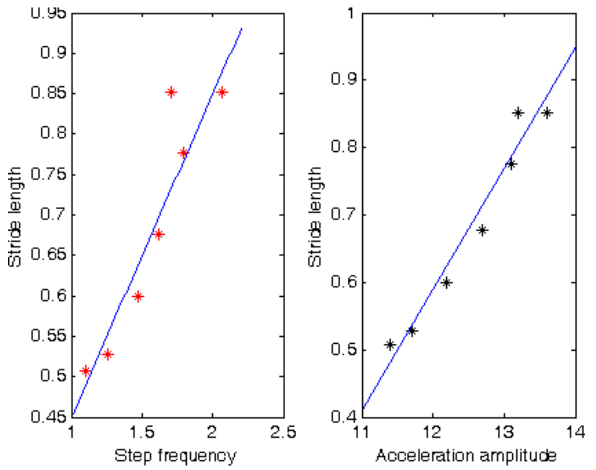

Fig. 1. Stride length with step frequency and acceleration amplitude, cooresponding to "frequency model" and "bounce model"

“bounce model” is used in our experiments and we define:

$$\varphi = p\left(a_{\max} - a_{\min}\right) + q \tag{0}$$

where φ is the stride length, and p and q are the two coefficients that will be automatically adjusted by the particle filter.

B. Orientation Estimate

The orientation of the mobile user in our system was estimated based on accelerometer, gyrometer and magnetometer. Both magnetometer and gyrometer can provide orientation information, but neither gives accurate and reliable moving orientation. Turn rate from a gyrometer can be integrated into an angle increment and the orientation obtained with a known initial azimuth angle, however; integration over a long period can introduce an unacceptable cumulative error. Orientation from a magnetometer is time independent, however the magnetometer has slow response rate and poor accuracy, especially in indoor environments where field disturbances always exist. For these reasons, a complementary filter was applied to combine these two orientation sources [25].

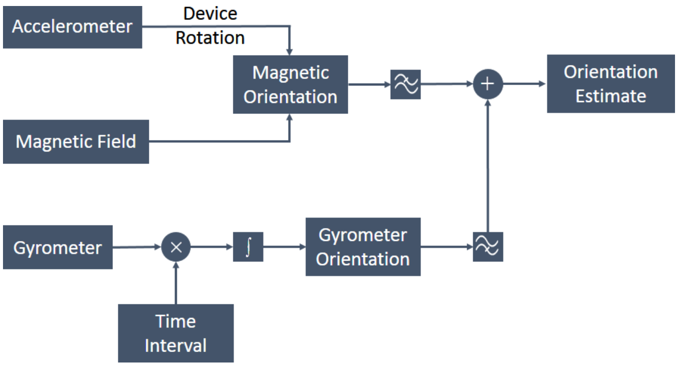

Fig. 2. Complementary filter for orientation estimate

The principle of the complementary orientation filter is shown in Fig. 2. A low pass filter is applied to the orientation obtained from the gyrometer, while a high pass filter is applied to that from the magnetometer and accelerometer. These high and low frequency orientation components are added at the input of the orientation estimate, which has a fast response time and minimizes drift over long periods. The orientation complementary filter in our experiments is based on Google Android sensor-related APIs [26].

While walking, a mobile device can be held in the hand or pocket in a variety of different orientations, which is very challenging for accurate orientation estimation, a point largely ignored in most previous studies. Unlike foot-mounted or head-mounted IMUs affixed to the body, the orientation of a mobile device is not always consistent with the user’s orientation, depending on the relative motion of mobile device and user. In this article, a “switching” scheme is introduced to handle arbitrary position changes of mobile devices, considering the following three situations:

- When the principal component of gravity changes with respect to the mobile device, the mobile device is assumed to be changing its orientation (Fig. 3). In this situation, the orientation estimate stops and the particle filter draws random orientations for each particle to estimate the location.
- When the mobile device is held with the screen upwards, typically meaning the user is checking content, the orientation of the mobile device and the mobile user are assumed to be consistent. In this case, the orientation of the mobile user is estimated using the complementary filter as introduced above.
- When the mobile device is not held with screen upward and the principal component of gravity is not changing, the mobile device is assumed to be held stationary or placed in a pocket. In this situation, the complementary filter stops and the orientation is estimated only using the gyrometer: a rotation that is orthogonal with the direction of gravity indicates the orientation of the mobile device and user changes.

$$\omega = R^T G \tag{0}$$

where $R = [R_x, R_y, R_z]^T$ is the turn rate readings from gyrometer, and $G = [G_x, G_y, G_z]^T$ is the gravity vector obtained from the acceleration data by filtering.

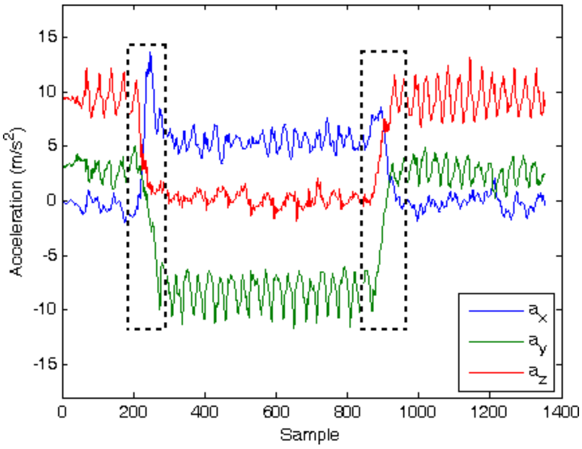

Fig. 3. Acceleration changes reflect mobile device position changes

IV. Particle Filter Data Fusion

We aim to obtain the best position estimate of a mobile user at each current time step by combining both the GSM fingerprinting result and the sensor dead-reckoning result. The state space filtering approach is used for sequentially estimating the variables of interest.

A. System model

In our system, at time step t the state of the system is $s_t = [x_t, y_t, p_t, q_t]^T$, where x_t and y_t are the position coordinates of the mobile user, and p_t and q_t are the coefficients of the stride length model explained in section III-B. The observation at time step t in our case is the fingerprint classifier output h_t, i.e. the room number. The state transition model can be characterized in terms of a state transition density $p(s_t \mid s_{t-1})$.

In our system, the state transition density $p(s_t \mid s_{t-1})$ is determined by both multiple sensor information and map layout information. As for state $s_t = [x_t, y_t, p_t, q_t]^T$, when a step is detected, based on multiple sensor dead-reckoning we have:

$$\begin{aligned} x_t &= x_{t-1} + \varphi_{t-1} \cdot \cos(\theta_{t-1}) + w_{t-1}^x \\ y_t &= y_{t-1} + \varphi_{t-1} \cdot \sin(\theta_{t-1}) + w_{t-1}^y \\ p_t &= p_{t-1} + w_{t-1}^p \\ q_t &= q_{t-1} + w_{t-1}^q \end{aligned} \quad (0)$$

where φ_{t-1} is the stride length (relation (4)), θ_{t-1} is the orientation estimation, and w_{t-1}^x, w_{t-1}^y, w_{t-1}^p and w_{t-1}^q represent system noise.

Not all the state transitions based on sensor dead-reckoning are reasonable; for example, the state transition may suggest that the mobile user passed through a wall, usually due to a false orientation estimate. Map layout information can be used to eliminate this kind of state transitions, as explained in the second step of the particle filter, in subsection 3.

The observation density $p(h_t \mid s_t)$ represents the likelihood, given the state s_t , that room number h_t is obtained from the GSM fingerprinting result. Given the available data, the required probability can be estimated from the SVM confusion matrix, whose element C_{ij} is the number of examples that are assigned to class *i* while the target is actually in room *j*. Then

$$P\left(h_t = i \middle| room(s_t) = j\right) \approx \frac{C_{ij}}{\sum_{k=1}^{7} C_{ik}} \quad (0)$$

where $room(s_t)$ is the number of the room to which the components x_t and y_t of state s_t belong.

B. Bayesian filter

The task here is to obtain the belief of the state s_k given the observations, which is the *a posteriori* density of the system state $p(s_t \mid h_t)$. A general approach to estimating the state over time from observations is the Bayesian filter. For a system with a state transition density $p(s_t \mid s_{t-1})$ and observation density $p(h_t \mid s_t)$, the Bayesian filter recursively computes posterior density of the state at time *t* $p(s_t \mid h_t)$ based on posterior density of the previous state $p(s_{t-1} \mid h_{t-1})$ and the most recent observation h_t , in two steps, prediction and update:

$$p(s_t \mid h_{t-1}) = \int p(s_t \mid s_{t-1}) \cdot p(s_{t-1} \mid h_{t-1}) ds_{t-1} \quad (0)$$

$$p(s_t \mid h_t) = \frac{p(h_t \mid s_t)}{p(h_t \mid h_{t-1})} p(s_t \mid h_{t-1}) \quad (0)$$

where $p(h_t \mid h_{t-1})$ is the normalizing constant

$$p(h_t \mid h_{t-1}) = \int p(h_t \mid s_t) \cdot p(s_t \mid h_{t-1}) ds_t .$$

C. Particle filter

The particle filter is a technique for implementing a recursive Bayesian filter by Monte Carlo sampling, which uses a finite number of random samples (called particles) with associated weights that provide a discrete approximation of the posterior density [27]. If the number of particles is very large, the discrete approximation approaches the true posterior density with arbitrary accuracy. The particle filter uses a set of particles taken from the previous time step ($pt_{t-1}^1,\ pt_{t-1}^2,\ \cdots pt_{t-1}^n \sim p(s_{t-1} \mid h_{t-1})$) and the most recent observation h_t to produce a set of particles approximately following the distribution of $p(s_t \mid h_t)$. The particle filter algorithm used in our system is realized in the following four steps:

- Initialize the particles: Since sensor dead-reckoning requires a starting position for integration, we sample *N* particles based on the output of the SVM classifier, which have the same weight that is equal to $1/N$. The coordinates (*x*, *y*) of the particles are distributed uniformly in each room, while the number of particles in each room depends on the observation distribution $p(h_0 \mid s_0)$. The stride model coefficients p_0 and q_0 are drawn uniformly over a suitable range.
- Make predictions based on the system model: When a step is detected, each particle makes a movement using sensor dead-reckoning. Map information is used in this step to remove unreasonable movements. As shown in Fig. 4, if a particle made a movement that crosses a wall or enters an inaccessible region, it will be removed, which, in filtering, is realized by giving the weight of the particle value 0.
- Update the weights based on the observation model: When the SVM classifier result is received, the likelihood $p(h_t \mid s_t)$ is applied to each particle to update the weight.
- Resample particles based on the weights: Through the previous steps, the number of particles diminishes and the weights of the particles change considerably. To better represent the posterior density $p(s_t \mid h_t)$, particles are resampled to the same number *N* with probabilities equal to their weights.

The final location estimate is taken to be the centroid of all particles.

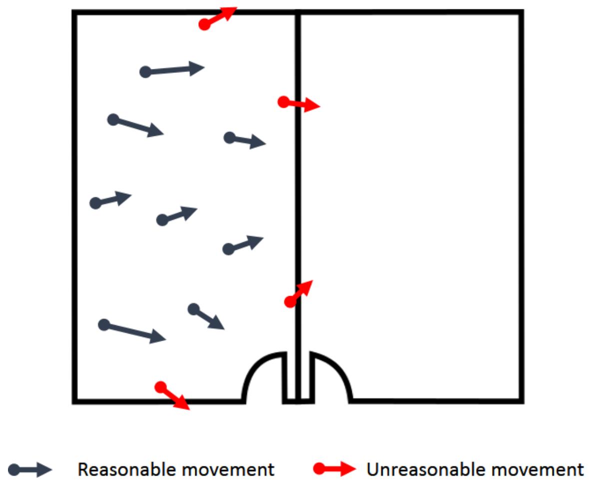

Fig. 4. Elimination of particles moving unreasonably

V. EVALUATION

A. Testbed

The experimental site tested is located on the fourth floor of a laboratory building (steel frame with concrete and plaster walls) in central Paris, France (Fig. 6). The rooms are numbered from one to seven, while the corridor is divided into three sections numbered from eight to ten. Map layout information of doors, walls and fixed-position obstacles and their orientations is stored in a map database for later use.

B. Data Acquisition Devices

In our experiments, two types of data were recorded, for GSM fingerprinting, and multiple sensor dead-reckoning, respectively. GSM RSS fingerprints were collected using the GSM trace mobile "TEMS Pocket", a standard Sony Ericsson W995 mobile phone to which network investigation software has been added by the manufacturer [28]. Using the TEMS investigation software package, this device is able to obtain a scan of the entire GSM900 and GSM 1800 bands in about 300 milliseconds. The multiple sensor readings were obtained using a commercial Google Nexus 7 tablet, containing an accelerometer, gyrometer, and magnetometer. Android open source operating system is embedded in order to program the recording of the sensor readings.

Although in our experiments, two different devices were used for recording GSM fingerprints and sensor readings respectively, there exists a newer smartphone-based generation of "TEMS Pocket" that can record all the necessary data on the same device. In addition, all GSM mobile phones are required by the GSM standard to be able to scan all channels of the GSM bands, so that our approach is potentially applicable to any commercial device supporting GSM.

C. Datasets

Training data and test data were recorded separately, for building the SVM classifier and testing the localization system, respectively. The training data was taken with the TEMS Pocket and manually labeled with the corresponding room numbers. In the testing phase, a TEMS Pocket and a Nexus 7 tablet were bundled together and held in the hand, recording GSM fingerprints and multiple sensor readings simultaneously. The TEMS Pocket and a Nexus 7 tablet were held either facing the user or swinging along with arm. The trajectory began at the south-west corner of room 7, continuing through the corridor, 8 and 9, into room 3, and finally stopping at the door of room 6, as shown in Fig. 6.

D. Results

GSM fingerprinting results are shown in Fig. 5, where the x axis gives the sample number along the trajectory and the y axis the SVM classifier outputs. As seen in the figure, the SVM classifier gives the correct room numbers for most of the test examples, but there are still misclassifications especially in adjacent rooms. The percentage of overall correct classification is 70%. (This result was obtained in a short time with a very small training set and is not considered to be optimized [11]).

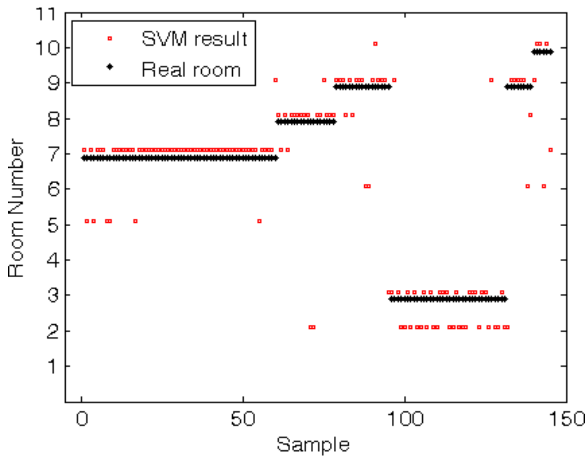

Fig. 5. GSM fingerprinting results

Fig. 6 shows the dead-reckoning results and particle filter results, where the actual trajectory is provided for comparison. It can be seen that multiple sensor dead-reckoning, even given a correct starting position, makes many mistakes. In a more realistic case, of course, the starting position is unknown. Furthermore, the obtained trajectory penetrates walls, which is not reasonable. The dashed line in Fig. 6 shows the particle filter results, in which the localization errors are seen to be corrected by combining GSM fingerprinting, sensor dead-reckoning and map layout restrictions. Only a few mistakes occurred at the beginning of the trace, due to the unknown starting position, since the SVM classifier only outputs room-level location, not a precise position.

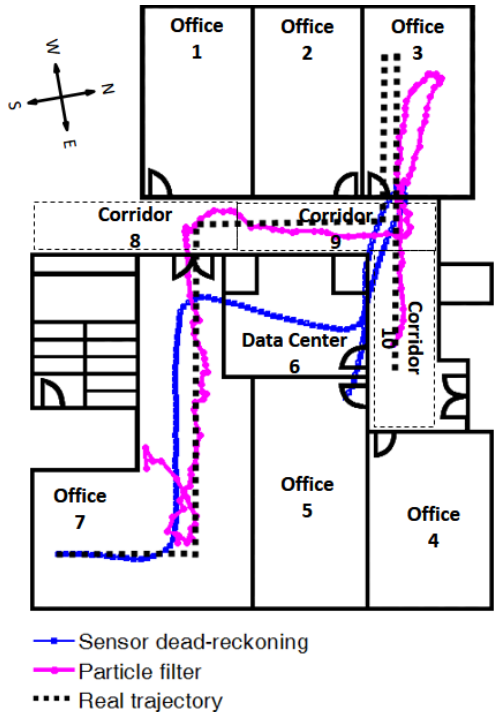

Fig. 6. Testbed and sensor dead-reckoning and particle filter results

VI. CONCLUSIONS AND FUTURE PERSPECTIVES

We have presented a hybrid approach for indoor localization using a particle filter to combine GSM fingerprinting results, sensor dead-reckoning and map layout information, which has been tested on a test trajectory acquired in a laboratory building under realistic conditions. Experimental results show that this approach can determine a mobile user's trajectory with good accuracy. The approach, which uses GSM fingerprints and multiple sensors that are easily obtainable due to the growing popularity of smart phones, is potentially ready for a practical implementation.

REFERENCES

[1] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more*. Springer, 2007, p. 516.

[2] A. Küpper, *Location-Based Services: Fundamentals and Operation*, vol. 45. 2005, p. 365.

[3] E. Aitenbichler and M. Muhlhauser, "An IR local positioning system for smart items and devices," *23rd Int. Conf. Distrib. Comput. Syst. Work. 2003. Proceedings.*, 2003.

[4] J. Hallberg, M. Nilsson, and K. Synnes, "Bluetooth Positioning," *10th Int. Conf. Telecommun. 2003 ICT 2003*, vol. 2, pp. 954–958, 2003.

[5] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," *Wirel. Networks*, vol. 10, no. 6, pp. 701–710, Nov. 2004.

[6] H. W. H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. D. Hanebeck, "WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors," *2007 4th Work. Positioning, Navig. Commun.*, 2007.

[7] S. Gezici and H. V. Poor, "Position Estimation via Ultra-Wide-Band Signals," *Proc. IEEE*, vol. 97, 2009.

[8] A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca, and V. Otsason, "GSM indoor localization," *Pervasive Mob. Comput.*, vol. 3, pp. 698–720, 2007.

[9] E. De Lara and S. Saroiu, "CILoS : A CDMA Indoor Localization System," *Proc. 10th Int. Conf. Ubiquitous Comput. - UbiComp '08*, 2008.

[10] B. Denby, Y. Oussar, I. Ahriz, and G. Dreyfus, "High-Performance Indoor Localization with Full-Band GSM Fingerprints," *2009 IEEE Int. Conf. Commun. Work.*, 2009.

[11] Y. Tian, B. Denby, I. Ahriz, P. Roussel, and G. Dreyfus, "Fast, handset-based GSM fingerprints for indoor localization," *2012 Int. Symp. Wirel. Commun. Syst.*, pp. 641–645, Aug. 2012.

[12] Y. Tian, B. Denby, I. Ahriz, P. Roussel, R. Dubois, and G. Dreyfus, "Practical indoor localization using ambient RF," in *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2013, pp. 1125–1129.

[13] B. Krach and P. Robertson, "Integration of foot-mounted inertial sensors into a Bayesian location estimation framework," *2008 5th Work. Positioning, Navig. Commun.*, 2008.

[14] S. Beauregard, "A Helmet-Mounted Pedestrian Dead Reckoning System," *Appl. Wearable Comput. (IFAWC), 2006 3rd Int. Forum*, pp. 1–11, 2006.

[15] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Comput. Graph. Appl.*, vol. 25, 2005.

[16] A. Serra, D. Carboni, and V. Marotto, "Indoor pedestrian navigation system using a modern smartphone," in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, 2010, pp. 397–398.

[17] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," *Proc. 10th Int. Conf. Ubiquitous Comput. - UbiComp '08*, p. 114, 2008.

[18] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," *Proc. 2012 ACM Conf. Ubiquitous Comput. - UbiComp '12*, p. 421, 2012.

[19] GSM Association, "GSM Roaming and Coverage Maps." [Online]. Available: http://maps.mobileworldlive.com/. [Accessed: 17-Mar-2014].

[20] A. Mammone, M. Turchi, and N. Cristianini, "Support Vector Machines," *Challenge*, vol. 1, pp. 1–8, 2009.

[21] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. …*, vol. 2, pp. 1–39, 2011.

[22] R. W. Levei and T. Judd, "Dead reckoning navigational system using accelerometer to measure foot impacts," 1996.

[23] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," *Analog Devices AN-602 Appl. note*, pp. 1–8, 2002.

[24] W. Zijlstra, "Assessment of spatio-temporal parameters during unconstrained walking.," *Eur. J. Appl. Physiol.*, vol. 92, no. 1–2, pp. 39–44, Jun. 2004.

[25] J. Calusdian, X. Yun, and E. Bachmann, "Adaptive-gain complementary filter of inertial and magnetic data for orientation estimation," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1916–1922.

[26] "Android Sensor APIs." [Online]. Available: https://developer.android.com/reference/android/hardware/Sensor.html. [Accessed: 14-Mar-2014].

[27] 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, 2002.

[28] "Test Mobile System ." [Online]. Available: http://www.ascom.com/nt/en/index-nt/. [Accessed: 14-Mar-2014].