



Graph-Based Map Matching for Indoor Positioning

Citation

Koivisto, M., Nurminen, H., Ali-Löytty, S., & Piche, R. (2015). Graph-Based Map Matching for Indoor Positioning. In *10th International Conference on Information, Communications and Signal Processing (ICICS)* IEEE. <https://doi.org/10.1109/ICICS.2015.7459983>

Year

2015

Version

Peer reviewed version (post-print)

Link to publication

[TUTCRIS Portal \(http://www.tut.fi/tutcris\)](http://www.tut.fi/tutcris)

Published in

10th International Conference on Information, Communications and Signal Processing (ICICS)

DOI

[10.1109/ICICS.2015.7459983](https://doi.org/10.1109/ICICS.2015.7459983)

Copyright

(c) 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Take down policy

If you believe that this document breaches copyright, please contact cris.tau@tuni.fi, and we will remove access to the work immediately and investigate your claim.

Graph-Based Map Matching for Indoor Positioning

Mike Koivisto, Henri Nurminen, Simo Ali-Löytty, and Robert Piché
Tampere University of Technology
Tampere, Finland
Emails: {mike.koivisto, henri.nurminen, simo.ali-loytty, robert.piche}@tut.fi

Abstract—This article presents a probabilistic motion model that is based on an economical graph-based indoor map representation, such that the motion of the user is constrained according to the floor plan of a building. The floor plan is modeled as a combination of links and open space polygons that are connected by nodes. In the authors' earlier work the link transition probabilities in this graph are proportional to the total link lengths that are the total lengths of the subgraphs accessible by choosing the considered link option, and this article extends this model to include open space polygons as well. A particle filter using the extended motion model in which all particles are constrained according to the map structure is presented. Furthermore, wireless local area network and Bluetooth Low Energy positioning tests show that the proposed algorithm outperforms comparison methods especially if the measurement rate is low.

Keywords—indoor positioning; particle filter; motion model; map matching; graph

I. INTRODUCTION

Wireless local area network (WLAN) and Bluetooth Low Energy (BLE) systems are commonly used for indoor positioning, and their accuracy can be improved by filtering measurements over time using statistical models of the user's motion. When inertial navigation is not available, the user's position or velocity is typically modeled as a random walk. However, these motion models neglect the motion constraints imposed by the walls in the building. Using the floor plan information in the motion model enables more efficient particle filtering than the conventional particle filter [1] that uses the random-walk motion model and treats wall constraints as measurements [2].

The floor plan information can be included in the motion model using a graph-based floor plan representation [3], [4], [5], [6], [2]. In the graph-based floor plan, the expected user paths are represented by links (edges). The links are undirected line segments that are connected by nodes (vertices) according to their real-world connectivity. It is assumed that the user can be anywhere on the links. A graph-based motion model is potentially more realistic than random-walk-based models because typical pedestrian movement is oriented towards a destination in a more determined way than random-walk models predict. A well-tuned graph-based motion model assumes more continuity for the user's direction than random-walk models, while allowing sharp turns at corridor junctions [2].

Graphs usually represent well corridors and small rooms. However, large open spaces, wide corridors, and outdoor spaces are in a more economical and natural way represented by two-dimensional polygons. Inside these polygons moving is free in two dimensions, so the positioning accuracy is not

limited by the map representation. Thus, the map used in this article contains two types of map objects (MO): links and open space polygons (OS). Such a combined map representation has been proposed by Ferris et al. [5], but they do not give an automatic method for assigning the MO transition rules.

This paper extends the authors' earlier work [2], where a particle filter using a graph-based motion model for WLAN positioning is proposed. The novelty of this paper is extending the link transition rule proposed in [2] to the combined map representation that includes the OSs. In the proposed algorithm, the MO transition probability is proportional to the total link length (TLL) that describes the total size of the area accessible by choosing the considered MO option.

This paper also presents real-data tests in open spaces and at indoor-outdoor transitions. The tests are done in a campus building using the existing WLAN infrastructure and a BLE network built for positioning research. The proposed method is found to outperform the comparison methods especially if measurement rate is low.

II. GRAPH WITH OPEN AREAS

A. Description and definition

In this paper, the map structure similar to [5] is expressed as a combination of MOs and nodes such that $G = (\Lambda, N)$. The set Λ contains the arbitrarily indexed MOs denoted by λ_k , which can be either links that represent small rooms and corridors, or OSs that represent larger open spaces. The other set N contains the arbitrarily indexed nodes ν_n that connect MOs and have a three-dimensional position.

Large OSs are defined as polygons inside which it is possible to place a circle with a 4-meter radius so that the circle does not cross any walls of a floor plan map. An OS λ_k contains information about the boundaries of the polygon and also accessor nodes (AN). These ANs are the only points where the user can enter the OS polygon λ_k from the graph or vice versa.

The rest of the map structure is represented as links such that each link λ_k has two end nodes. Different floors of a building are connected with vertical links in places where floor transitions are possible, such as in elevators and stair cases. The map structure for one floor is shown in Fig. 1.

B. Map object transition rule

Transitions between MOs can be modeled using different types of transition probabilities [3], [4]. To extend the TLL-based link transition rule of [2], corresponding MO lengths need to be defined for OSs.

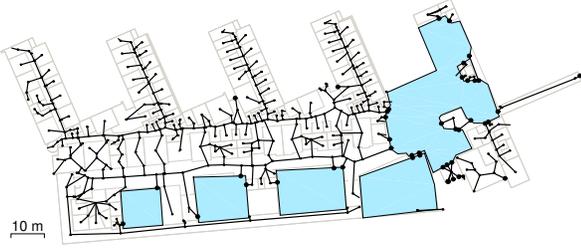


Fig. 1: Graph with OSs in the test building. Links and nodes as well as ANs are shown with black. OSs are modeled as polygons which are shown with light blue. The outdoor area is also considered as an OS.

The user can move to another destination through an OS, which affects the computation of the TLLs. Therefore, temporary links $\lambda_{k,1}, \dots, \lambda_{k,n}$ are created between the ANs of an OS λ_k to represent the shortest possible distances between different ANs. Such temporary links can be created in an off-line phase using Dijkstra's algorithm [7, Ch. 4.8], for instance, and they need not be stored in the permanent map data structure; the temporary links are used only to determine the TLLs in the off-line phase.

For adapting to the definition of TLL in [2], the parts of the OSs that are not covered by the temporary links are also given a size measure that is here referred to as MO length. The MO length should depend on the area of the OS, and the lengths of the temporary links which are parts of the shortest paths along the map structure are subtracted from the OSs' MO length. The MO length of an OS λ_k is thus defined as

$$\text{LENGTH}(\lambda_k) = \frac{\text{AREA}(\lambda_k)}{\text{CORRIDOR_WIDTH}} - \sum_{i \in \mathcal{A}} \text{LENGTH}(\lambda_{k,i}), \quad (1)$$

where $\text{AREA}(\lambda_k)$ is the area of the polygon, CORRIDOR_WIDTH is a configuration parameter that describes a typical corridor width, for example 3 m, and

$\mathcal{A} = \{j \mid \text{shortest path from initial node } \nu_m \text{ to or through } \lambda_k \text{ is shorter than } \ell_{\text{MAX}} \text{ and contains } \lambda_{k,j}\}.$

The MO transition probabilities $\mathbb{P}(\lambda_{k_i} | \lambda_k)$ are determined by extending the principles in the authors' earlier work [2] with an assumption that considers temporary links also as parts of the shortest paths. This assumption together with the principles in [2] imply that the user uses the shortest possible path to reach the destination. Thus, the old MO is not one of the possible destinations and U-turns are only allowed by an additional motion model as explained in section III-A. Using these assumptions the MO transition probability of arriving to node ν_m from link λ_k can be expressed by extending the transition rule as follows

$$\mathbb{P}(\lambda_{k_i} | \lambda_k) = \frac{\sum_{j \in \mathcal{I}_{k,i}} \text{LENGTH}(\lambda_j) + \sum_{j \in \mathcal{K}_{k,i}} \text{LENGTH}(\lambda_{p,j})}{\sum_{j \in \mathcal{I}_k} \text{LENGTH}(\lambda_j) + \sum_{j \in \mathcal{K}_k} \text{LENGTH}(\lambda_{p,j})}, \quad (2)$$

where

$\mathcal{I}_{k,i} = \{j \mid \text{shortest path from } \nu_m \text{ to } \lambda_j \text{ is shorter than } \ell_{\text{MAX}} \text{ and does not use } \lambda_k\},$

$\mathcal{K}_k = \{j \mid \text{shortest path from } \nu_m \text{ to temporary link } \lambda_{p,j} \text{ is shorter than } \ell_{\text{MAX}} \text{ and does not use } \lambda_k\},$

$\mathcal{K}_{k,i} = \{j \mid \text{shortest path from } \nu_m \text{ to temporary link } \lambda_{p,j} \text{ is shorter than } \ell_{\text{MAX}} \text{ and uses } \lambda_{k_i} \text{ but does not use } \lambda_k\}.$

The transition probabilities defined in (2) can be computed efficiently by using Algorithm 1 of [2] and considering temporary links as ordinary links and open spaces as destinations with lengths determined in Eq. (1).

At MO transitions, this model gives most weight to corridors and major open areas, and a high TLL can be considered as an indication of a link being such a major pathway [2]. The assumptions limit the probability of the main routes with the variable ℓ_{MAX} such that options with shorter lengths get large enough probabilities. Although the variable ℓ_{MAX} ensures that less probable options get small probabilities, some lower limit for the probabilities can be also set. In the real-data tests, the limiting variable ℓ_{MAX} is set to 40 m and the lower limit for the transition probabilities is set to 5%.

The proposed TLL algorithm does not use any information about the functions of different building parts; the TLLs are computed off-line using only the map structure G as input. However, if real data of people's behavior are available, the MO transition probabilities can be updated using the TLLs as a prior. This learning process is similar to the one described in [2].

III. POSITIONING ALGORITHM

A. Motion model

Instead of being a random-walk, the motion of a user usually contains sharp turns and it tends to be oriented towards some destination. The graph-based motion models take these sharp turns into account at junction points as well as tendencies in velocity when the direction of the user is known.

In this paper, a two-mode motion model is used to model the constrained motion of the user as in [3], [5] and [2]. The mode m_k is either a combination of the MO transition probabilities and random-walk speed ($m_k = \text{'motion'}$) or is static ($m_k = \text{'static'}$) in both OSs and links.

The state vector \mathbf{x}_k at time instant t_k depends on the MOs such that

$$\mathbf{x}_k = \begin{cases} [I_k, p_k, d_k, v_k, m_k]^T & \mathbf{x}_k \text{ is on a link} \\ [I_k, x_k, y_k, d_k, v_k, m_k]^T & \mathbf{x}_k \text{ is in an OS,} \end{cases} \quad (3)$$

where I_k is the MO index, $p_k \in [0, 1]$ is the one-dimensional location on the link and $[x_k, y_k]^T$ is the two-dimensional location in the OS. Furthermore, $d_k \in \{-1, 0, 1\}$ indicates the direction of the user on the link, $d_k \in [0, 2\pi]$ denotes the heading in the OS, and v_k the speed (magnitude of velocity).

The motion model of the user contains four parts: motion in links and in OSs, and transitions between links and OSs. The motion model used in links is presented in [2]. Because

the user can go through OSs, straight motion is preferred in the motion model inside OSs as in [5]. Thus, using the probabilistic notation and assuming that the heading of the user is random-walk in OSs, the motion model in an OS is

$$p(s_k, v_k, d_k | v_{k-1}, d_{k-1}, m_k) = \begin{cases} \mathcal{N} \left(\begin{bmatrix} s_k \\ v_k \\ d_k \end{bmatrix} \middle| \begin{bmatrix} (\Delta t)_k v_{k-1} \\ v_{k-1} \\ d_{k-1} \end{bmatrix}, \mathbf{P}_k \right), & \text{if } m_k = \text{'motion' } \\ \text{DIRAC}(s_k, v_k, d_k), & \text{if } m_k = \text{'static' } \end{cases}, \quad (4)$$

where s_k is the distance travelled by the user within the time interval $[t_{k-1}, t_k]$ and DIRAC denotes the multi-dimensional Dirac delta function. Furthermore, $(\Delta t)_k = t_k - t_{k-1}$ is the length of the discretisation interval and the covariance matrix of the process noise is now

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{Q}_k & 0 \\ 0 & \sigma_d^2 \end{bmatrix} \quad (5)$$

where the matrix \mathbf{Q}_k is similar to one in [2] and σ_d is another configuration parameter.

The transition from a link to an OS occurs when the user is in an accessor node coming from a graph. If the user chooses to enter the OS, the heading of the user is the heading of the latest link with some uncertainty. The transition from an OS to a graph occurs if the user is inside the OS and crosses a boundary of the OS close to an accessor node. Then the probability mass is set to the accessor node and the motion continues along the graph.

B. WLAN positioning

In this paper, most of the tests are based on WLAN positioning. Fingerprints that are collected beforehand from each floor of the test building are used in the experimental tests presented in section IV. The same measurement model based on the standard logarithmic path loss model with Gaussian shadowing noise is used as in [2]. A Gauss-Newton method is used to obtain a Gaussian approximation of the actual likelihood taking the parameter uncertainties into account as in [8]. Thus, the measurement model for the particle weighting is

$$p(\mathbf{y}_k | \mathbf{x}_k) \propto \mathcal{N}(\mathbf{r}_k(\mathbf{x}_k) | \boldsymbol{\mu}_k(\mathbf{y}_k), \boldsymbol{\Sigma}_k(\mathbf{y}_k)), \quad (6)$$

where \mathbf{y}_k is the vector of received signal strengths (RSS), and \mathbf{r}_k is the position of the user in Cartesian coordinates. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance matrix of the likelihood approximation, respectively.

In this paper, the current floor of the user is estimated by choosing the most probable floor in the building based on the measurement likelihood.

C. Bluetooth Low Energy positioning

The proposed method with the current map structure was also tested using measurements from StickNFind BLE beacons (Fig. 2). The beacons were placed inside the building such that the average number of heard beacons in one fingerprint location around the test track was 11, which is more than the threshold proposed in [9]. Since received measurements are RSS-values as in WLAN positioning, the BLE measurement model is similar to the measurement model (6).



Fig. 2: A BLE beacon

D. Particle filter

A particle filter is a Monte Carlo algorithm that approximates the posterior distribution of the state given the measurement history, when the measurement and the state transition models as well as prior information of the state is known, and certain Markovian assumptions hold. Although the continuity of the distributions is usually assumed, it is not necessary because the existence of the corresponding probability measures is a sufficient condition for convergence of a particle filter [10].

A particle filter approximates the posterior distribution $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ with a set of weighted particles $\{(\mathbf{x}_k^i, w_k^i) \mid i \in \{1, \dots, N\}\}$. Initially, particles are generated according to prior distribution $p(\mathbf{x}_0)$. In the prediction phase, the particles are generated according to the proposal distribution $q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{y}_{1:k})$, and in the update phase importance weights w_k^i are updated using the measurement likelihood $p(\mathbf{y}_k | \mathbf{x}_k^i)$. Finally, resampling ensures that the weight does not concentrate to one particle. [11]

In this paper, the state transition distribution $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is used as a proposal distribution, and the effective sample size is used to determine when resampling is needed. The particle filter is presented in detail in Algorithm 1, where the re-initialization method presented in [1] is also used.

The algorithm for moving a particle from an OS to a link is presented in detail in Algorithm 2. When a particle is inside an OS, it might reach the walls far away from any AN. Instead of bouncing the particle back by changing the heading as in [5], the weight of the colliding particle is set to zero. When the particle exits the OS close to an AN, the particle's weight is reduced by a coefficient that depends on the distance to the closest AN.

A natural choice for the point estimate would be the weighted mean of the particles. However, the estimate might be in a region that is inaccessible via the MOs even when all the particles are in the feasible region. Choosing the best particle [5] or using the MAP (*maximum a posteriori*) estimate [12] are possible options to constrain the estimate according to the map structure, but the estimate trajectories might display jumpiness due to the multimodality of the posterior distribution.

The continuous Constrained Mean algorithm [13] enforces the estimate to the map structure by minimizing the weighted sum of squares over all possible MOs. The Constrained Mean estimator for the graph-based model is equivalent to choosing

the MO point that is closest to the weighted mean. The detailed description of this point estimator can be found in [2].

It is still possible that the solution of the Constrained Mean algorithm is inside a MO that does not contain any particles. Such estimates can be avoided by constraining the set of possible MOs at each time step to include only specific accessible areas.

IV. TESTS

A. Test setup

The tests are carried out in Tietotalo building of Tampere University of Technology campus using Samsung Galaxy S5 phone. Three tests are done using only WLAN measurements from existing WLAN infrastructure and one test is done using only BLE measurements from a set of installed beacons. Actual positioning algorithms are computed offline using MATLAB software.

The test tracks are shown in Fig. 3. Track 1 (Fig. 3a) starts from a corridor, enters a lecture hall and stays inside for a while before leaving the lecture hall back to the corridor. Track 2 (Fig. 3b) starts from a narrow corridor and goes through an irregularly shaped open area ending up to the main corridor. Third WLAN track 3 (Fig. 3c) illustrates a movement from indoors to outdoors. Track 4 (Fig. 3d) is a BLE track that starts from an open area and goes to the main corridor by crossing an open area on its way.

Results from the experimental tests are compared with the Kalman filter (KF) and the similar particle filter with uniform MO transition probabilities. Since it is more convenient to model open areas such as outdoors with polygons instead of links, the proposed method is not compared with the authors' earlier method [2]. The KF uses a random-walk motion model.

B. Results and discussion

In the experimental tests, 400 particles are used for both particle filters. Each filter is run 100 times for all four test tracks and the measurement scanning intervals are set to 5 and 10 seconds. RMS-error (root mean square error) statistics from the tests are shown in Fig. 4.

The proposed TLL-based particle filter performs slightly better than the KF in many cases and better than the particle

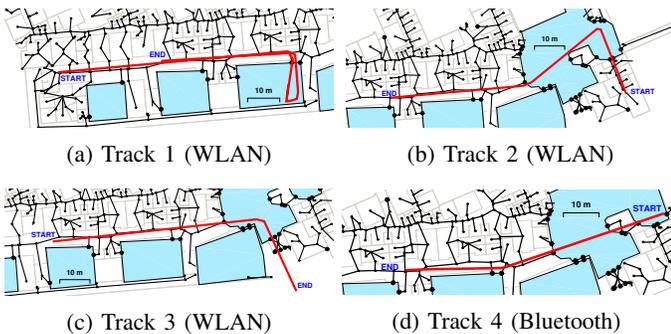


Fig. 3: The real-data test tracks with walls, links, nodes, and the ground truth of test tracks. Starting and ending points are labelled, and accessor nodes are shown as larger black points.

Algorithm 1 Particle filter for 2D indoor positioning

- 1) For each $i \in \{1, \dots, N\}$ set $w_0^i := \frac{1}{N}$, and generate $\mathbf{x}_0^i \leftarrow p(\mathbf{x}_0)$. Set the time index $k := 1$.
- 2) Generate motion states m_k^i based on the previous motion states m_{k-1}^i and the transition probabilities. For the particles $i \in \{i | m_{k-1}^i = \text{'static' and } m_k^i = \text{'motion'}\}$ generate $v_{k-1}^i \leftarrow p(v_o)$ and $d_{k-1}^i \leftarrow \text{cat}(0.5, 0.5)$. For each $i \in \{i | m_k^i = \text{'motion'}\}$ generate

$$\begin{bmatrix} s_k^i \\ v_k^i \end{bmatrix} \leftarrow N \left(\begin{bmatrix} (\Delta t)_k v_{k-1}^i \\ v_{k-1}^i \end{bmatrix}, Q_{k-1} \right),$$

set $v_k^i := \min(\max(v_k^i, v_{\min}), v_{\max})$ and set $d_k^i := d_{k-1}^i$. The matrix Q_{k-1} is the one in [2]. For each $i \in \{i | m_k^i = \text{'static'}\}$ set $s_k^i := 0$ and $v_k^i := 0$.

- 3) **for** $i \in \{1, \dots, N\}$ **do**
Set $\tilde{w}_k^i := w_{k-1}^i$ and $\tilde{s} := s_k^i$.
if $\lambda_{I_{k-1}^i}$ is a link **then**
Move the i th particle to direction d_k^i at most to the end of $\lambda_{I_{k-1}^i}$ to distance $s' \leq s_k^i$.
Set $\tilde{s} := \tilde{s} - s'$.
end if
Set $I_k^i := I_{k-1}^i$.
while $\tilde{s} > 0$ **do**
Generate new MO index I_k^i from categorical distribution $\text{cat}(\mathbb{P}(\lambda_{k_1} | \lambda_{I_k^i}), \dots, \mathbb{P}(\lambda_{k_{n_\lambda}} | \lambda_{I_k^i}))$
if $\lambda_{I_k^i}$ is a link **then**
Update the direction d_k^i .
Move particle i to the direction d_k^i at most to the end of $\lambda_{I_k^i}$ to distance $s' \leq \tilde{s}$.
else
Generate direction d_k^i using Eq. (4) and move particle i to distance $s' \leq \tilde{s}$ not farther than the OS boundary.
Run the algorithm OS_to_link.
end if
Set $\tilde{s} := \tilde{s} - s'$
end while
Find the particle's position \mathbf{r}_k^i in Cartesian coord.
end for
4) If a measurement is obtained at time index k , run Gauss-Newton to obtain mean $\boldsymbol{\mu}_k$ and covariance Σ_k [8]. Set
$$\tilde{w}_k^i := N(\mathbf{r}_k^i | \boldsymbol{\mu}_k, \Sigma_k) \cdot \tilde{w}_k^i.$$
5) Normalize the weights by $w_k^i := \tilde{w}_k^i / \sum_{j=1}^N \tilde{w}_k^j$.
6) Perform divergence monitoring as in [2].
7) Compute the estimate $\hat{\boldsymbol{\mu}}_k$ using the Constr. Mean [13].
8) If $1 / \sum_{i=1}^N (w_k^i)^2 < N_{\text{eff,lim}}$, perform resampling and equalize the weights. Set $k := k + 1$, and go to phase 2.

filter with uniform transition probabilities in every test case. In track 1, which goes through a lecture hall, the TLL-based method is giving less probability to the lecture hall according to the TLLs. However, the proposed method performs well with both scan rates because less probable options have enough

Algorithm 2 OS_to_link

```
if The particle is on an OS boundary then
  distAN := distance to the closest AN
  if 0 ≤ distAN ≤ 0.5 · widthdoor then
    Move the particle to the AN.
  else if 0.5 · widthdoor < distAN ≤ 5 m then
    Move the particle to the AN and update the weight:
     $w_{k-1}^i := 2^{-4(\text{dist}_{\text{AN}} - 0.5 \cdot \text{width}_{\text{door}})^2} \cdot w_{k-1}^i$ .
  else
    Set  $s' := \tilde{s}$  and  $w_{k-1}^i := 0$ .
end if
end if
```

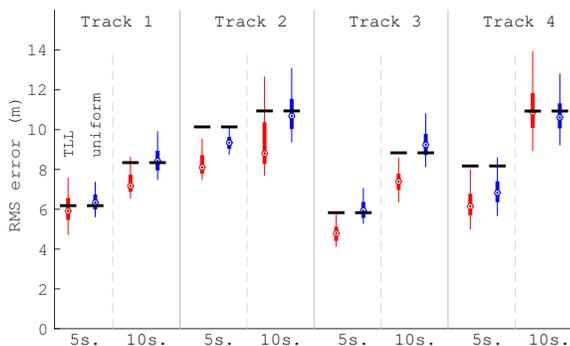


Fig. 4: Error statistics from the real-data tests for each method and track when the measurement scanning rates are 5 and 10 seconds. Black line segments are the KF results

probability to attract particles. The KF is almost as good as the TLL-method with the scanning interval of 5 seconds, but when the scans are made less frequently, the TLL-based method outperforms both comparison methods clearly. Despite the fact that track 2 has one sharp turn in the large OS, the proposed particle filter performs well because the variance of the random-walk direction is large enough. It also outperforms the comparison methods significantly.

The KF as well as the particle filter with uniform transition probabilities are less accurate than TLL-method also in track 3 which contains an indoor–outdoor transition. In track 4 both particle filters outperform the KF with the 5-second scanning interval. The difference between the particle filters is smaller than with the other tracks, which is probably because most of the track is in an open area.

V. CONCLUSIONS

A novel statistical motion model for indoor positioning with the map structure that contains a graph with open area regions was proposed. The proposed model constrains the motion of the user based on the map structure such that larger and more branched areas in the map have more probability at map object transitions. A particle filter algorithm using the proposed model and WLAN or BLE measurements was presented and compared with two other methods.

The user’s motion typically takes place in corridors and major open spaces (OS) and is oriented towards some destination instead of being random-walk. The proposed motion model prefers motion in OSs and corridors but is also able to handle less probable areas such as small rooms. The motion model also prefers straight motion inside OSs where the motion of the particles is constrained by the OS boundaries.

The presented experimental tests indicate that the proposed motion model is advantageous especially if the measurement scanning rate is low. In addition to the good performance in the corridors, the particle filter with the proposed motion model performs well also in OSs and outdoor spaces, and outperforms the comparison methods in most of the cases.

ACKNOWLEDGMENTS

This research was partly funded by HERE, a Nokia business. Henri Nurminen receives funding from Tampere University of Technology Graduate School, Finnish Doctoral Programme in Computational Sciences, the Foundation of Nokia Corporation, and Tekniikan edistämissäätiö.

REFERENCES

- [1] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, “Particle filter and smoother for indoor localization,” in *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN2013)*, October 2013, pp. 137–146.
- [2] H. Nurminen, M. Koivisto, S. Ali-Löytty, and R. Piché, “Motion model for positioning with graph-based indoor map,” in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN2014)*, October 2014, pp. 646–655.
- [3] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, “Voronoi tracking: location estimation using sparse and noisy sensor data,” in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS 2003)*, October 2003, pp. 723–728.
- [4] F. Evennou, M. François, and E. Novakov, “Map-aided indoor mobile positioning system using particle filter,” in *2005 IEEE Wireless Communications and Networking Conference (WCNC 2005)*, vol. 4, March 2005, pp. 2490–2494.
- [5] B. Ferris, D. Hähnel, and D. Fox, “Gaussian processes for signal strength-based location estimation,” in *2006 Robotics: Sciences and Systems Conference (RSS 2006)*, August 2006.
- [6] M. I. Khan and J. Syrjärinne, “Investigating effective methods for integration of building’s map with low cost inertial sensors and wifi-based positioning,” in *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN2013)*, October 2013, pp. 884–891.
- [7] S. S. Skiena, *The Algorithm Design Manual*. New York, NY, USA: Springer-Verlag, 1998.
- [8] H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E.-S. Lohan, R. Piché, and M. Renfors, “Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks,” in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, November 2012, pp. 461–469.
- [9] R. Faragher and R. Harle, “Location Fingerprinting with Bluetooth Low Energy Beacons,” in *IEEE Journal on Selected Areas in Communications*, 2015.
- [10] M. Koivisto, “Graph-based particle filter in indoor positioning,” Master’s thesis, Tampere University of Technology, 2015.
- [11] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
- [12] H. Driessen and Y. Boers, “MAP estimation in particle filter tracking,” in *IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications*, April 2008, pp. 41–45.
- [13] R. Piché and M. Koivisto, “A method to enforce map constraints in a particle filter’s position estimate,” in *11th IEEE Workshop on Positioning, Navigation and Communication (WPNC’14)*, March 2014.