Terrain Navigation in the Magnetic Landscape: Particle Filtering for Indoor Positioning

Arno Solin
IndoorAtlas Ltd.
Helsinki, Finland
arno.solin@indooratlas.com

Simo Särkkä
Aalto University
Espoo, Finland
simo.sarkka@aalto.fi

Juho Kannala
Aalto University
Espoo, Finland
juho.kannala@aalto.fi

Esa Rahtu
IndoorAtlas Ltd.
Oulu, Finland
esa.rahtu@indooratlas.com

***Abstract*—Variations in the ambient magnetic field can be used as features in indoor positioning and navigation. We describe a technique for map matching where the pedestrian movement is matched to a map of the magnetic landscape. The map matching algorithm is based on a particle filter, a recursive Monte Carlo method, and follows the classical terrain matching framework used in aircraft positioning and navigation. A recent probabilistic Gaussian process regression based method for modeling the ambient magnetic field is employed in the framework. The feasibility of this terrain matching approach is demonstrated in a simple real-life indoor positioning example, where both the mapping and positioning is done using a smartphone device.**

I. Introduction

Indoor positioning is an important prerequisite for a wide range of applications, for example, in robotics, health-care, security, business, and transportation. Location information can be used to monitor patients and staff in hospitals, to find injured people on accident sites, for targeted advertising, and to guide people in shopping malls, among other uses. Recent advances in smartphone technology have made indoor positioning practical also for consumer use, because most smartphones nowadays contain a wide range of sensors which are required and which are also enough for accurate indoor positioning.

Indoor positioning is considerably harder than outdoor positioning. For finding the location outdoors, satellite based positioning systems (*e.g.* GPS [1]) provide good accuracies for most purposes, but not for indoor use. This article is concerned with magnetic field based positioning [2, 3], where the basic idea is to create a map of magnetic anomalies inside a building and use it to find the current position. The advantage of this methodology over other commonly used methods such as Wi-Fi and Bluetooth based methods is that no infrastructure needs to be installed—the magnetic field is an inherent property of the indoor environment. Furthermore, magnetometers are nowadays present in (almost) any smartphone.

For indoor positioning using the ambient magnetic field one requirement is that accurate maps of the magnetic field need to be created before positioning is possible—another approach is to form them simultaneously during positioning via SLAM methods (see, *e.g.*, [4]), which is not considered in this paper. In this work we follow the approach of [5] and use Gaussian process models from machine learning [6] to create a magnetic map from pre-collected smartphone measurements.

Magnetic field positioning methods (and Wi-Fi/Bluetooth based methods) can be significantly improved by combining them with inertial navigation, which keeps the system informed about movement in the local coordinate frame. Inertial navigation [7] is a class of methods, which is also applicable to indoor positioning. Because pure inertial navigation is hard or even impossible with smartphone sensors, in smartphones, one typically uses a limited form of inertial navigation which combines step-counting based speed measurements (see, *e.g.*, [8, 9]) with gravitation tracking and gyroscope-based orientation estimation (see, *e.g.*, [10]). Another commonly used idea is to use zero-velocity updates to compensate for the inertial sensor errors, but the use of this in practice requires that the sensors are attached to shoes [11].

The locally accurate, but long-term inaccurate inertial measurements can be combined with global measurements such as magnetic field measurements or Wi-Fi/Bluetooth measurements using Bayesian state-estimation (or filtering) methods (see, *e.g.*, [12]). In this work we use particle filtering for this purpose. This kind of general methodology is often referred to as terrain navigation (see, *e.g.*, [13–15]). However, the use of magnetic fields for terrain navigation has a long history. Tyrén (1982) [16] argued that the heterogeneous character of the intensity of the Earth's magnetic field could be used as a potential basis for a ground-speed measurement system in vehicle and aircraft localization. Ideas of magnetic terrain navigation have been considered in naval applications to submarine positioning and tracking [17].

Previously, in indoor environments, the anomalies in the magnetic field have been successfully used in positioning for example by [2, 18]. More recent contributions include, for example, [19–21]. In this article the contribution is to use terrain navigation particle filtering methods together with probabilistic physics-aided Gaussian process generated maps [5] for accurate magnetic field based indoor positioning.

This paper is structured as follows. In the next section we present how the concept of particle filtering based terrain navigation can be combined with a probabilistic map of the ambient magnetic field, which also accounts for the uncertainties related to the magnetic map. Section III presents an empirical study where the positioning algorithm is employed in indoor localization using a handheld smartphone. Finally, the results are discussed in Section IV.

Fig. 1: Principle of magnetic terrain navigation. Here a pre-generated magnetic map is overlaid on top of a picture of the space. The map depicts a vector field with both a direction (the arrows indicate the direction based on the x and y components) and magnitude (warm colours indicate stronger values, cool colours weaker). During positioning, the vector valued (three-component) measurement track obtained by the smartphone magnetometer is matched to the magnetic landscape.

II. Methods

An illustration of the general concept of magnetic terrain navigation is shown in Figure 1. The magnetic terrain navigation setup in this paper boils down to three distinctive parts:

- The positioning is overseen by a *particle filter*, which is a sequential Monte Carlo approach for proposing different state histories and finding which one matches the data the best.
- The *magnetic terrain* which the observations are matched against. The map is constructed by a Gaussian process model which is able to return a magnetic field estimate and its variance for any spatial location in the building.
- A model for the movement of the person being tracked, often referred to as a *pedestrian dead reckoning* model.

The following sections will explain these components of the map matching algorithm in detail.

A. Particle filtering

Particle filtering [12, 22, 23] is a general methodology for probabilistic statistical inference (*i.e.*, Bayesian filtering and smoothing) on state space models of the form

$$\begin{aligned} \mathbf{x}_{k+1} &\sim p(\mathbf{x}_{k+1} \mid \mathbf{x}_k), \\ \mathbf{y}_k &\sim p(\mathbf{y}_k \mid \mathbf{x}_k), \end{aligned} \tag{1}$$

where $p(\mathbf{x}_{k+1} \mid \mathbf{x}_k)$ defines a vector-Markov model for the dynamics of the state $\mathbf{x}_k \in \mathbb{R}^{d_\mathrm{x}}$, and $p(\mathbf{y}_k \mid \mathbf{x}_k)$ defines the model for the measurements $\mathbf{y}_k \in \mathbb{R}^{d_\mathrm{y}}$ in the form of conditional distribution of the measurements given the state. For example, in (magnetic) terrain navigation, the dynamic model tells how the target moves according to a (pedestrian) dead reckoning and the (Markovian) randomness is used for modeling the errors and uncertainty in the dynamics. In conventional terrain navigation, the measurement model tells what distribution of height we would measure at each position, and in magnetic terrain navigation it tells what is the distribution of magnetic field measurements we could observe at a given position and orientation.

A particle filter aims at computing the (Bayesian) filtering distribution, which refers to the conditional distribution of the current state vector given the observations up to the current time step $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$. Particle filtering uses a weighted Monte Carlo approximation of n particles to approximate this distribution. The approximation has the form

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \approx \sum_{i=1}^{n} w_k^{(i)} \, \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}), \tag{2}$$

where $\delta(\cdot)$ stands for the Dirac delta distribution and $w_k^{(i)}$ are non-negative weights such that $\sum_i w_k^{(i)} = 1$. Under this

Alg. 1: Algorithm for particle filter based (abstract) terrain navigation. The recursion defines a sequential Monte Carlo method.

Initialization: Draw n samples $\mathbf{x}_0^{(i)}$ from the prior

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0), \qquad i = 1, \ldots, n,$$

and set $w_0^{(i)} = 1/n$, for all $i = 1, \ldots, n$.

For each time step $k = 1, 2, \ldots$

1) **Prediction**: Draw samples $\mathbf{x}_k^{(i)}$ from the importance distributions

$$\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_{1:k}), \qquad i = 1, 2, \ldots, n.$$

This propagates the particles according to the PDR model.

2) **Map matching**: Calculate new weights according to

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{y}_k \mid \mathbf{x}_k^{(i)})\, p(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_{1:k})}$$

and normalize them to sum to unity.

3) **Resampling**: According to the resampling strategy: Take n samples with replacement from the set $\{\mathbf{x}_k^{(i)}\}_{i=1}^n$ where the probability to take sample i is $w_k^{(i)}$, after which let $w_k^{(i)} = 1/n$. The resampling is only done when necessary, that is, when the effective number of particles is too low (see the body text).

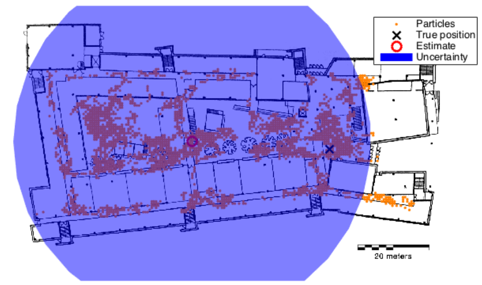
framework, the expectation of an arbitrary function $\mathbf{g}(\cdot)$ can be approximated as

$$\mathrm{E}[\mathbf{g}(\mathbf{x}_k) \mid \mathbf{y}_{1:k}] \approx \sum_{i=1}^{n} w_k^{(i)}\, \mathbf{g}(\mathbf{x}_k^{(i)}). \tag{3}$$

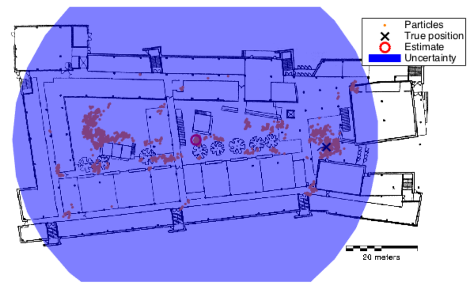
When the state contains the position, as is typical in terrain navigation, the filtering distribution gives the posterior distribution of the position given all the observations obtained so far. A particle filter estimate of this kind is illustrated in Figure 2. The experiment setup is described in detail in Section III.

Our terrain matching algorithm is a special case of a particle filter (see [13] for a good introduction to particle filtering in positioning). A general particle filter and hence an abstract form of our terrain matching algorithm is shown in Algorithm 1. In practice, the performance of the particle filtering algorithm is determined by the underlying state space model, the number of particles n, the selected importance distribution $\pi(\cdot)$, the resampling method as well as the schedule of resampling operations. For more comprehensive discussion on these issues the reader is referred to [12].

In principle, a particle filter is a sequential importance sampling algorithm, whose performance is determined by the choice of the importance distribution. A practical problem in particle filtering is sample depletion which means that over time the weights between particles become unevenly distributed. This can slowly lead to a situation where all

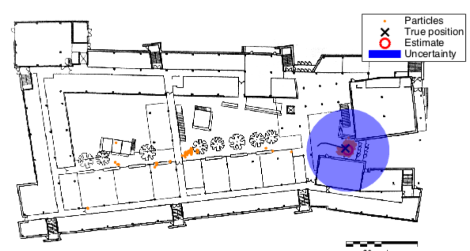


(a) $t = 0$ s

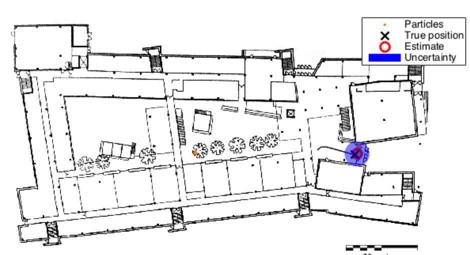


(b) $t = 2$ s

(c) $t = 4$ s



(d) $t = 6$ s



(e) $t = 8$ s

Fig. 2: Example evolution of the particle filtering estimate from initialization to convergence. See the body text for further explanation (Sec. III).

weight might be concentrated to a single particle, and the filter estimate is thus only dependent on that one particle.

Sample depletion is avoided by resampling (step 3 in Alg. 1). By resampling new (representative) particles are created to replace those particles which have become negligible. Resampling however increases uncertainty and therefore it is avoided until needed, and therefore resampling is done when the number of effective particles drops below a given threshold.

The effective number of particles (see [12] for discussion) gives a summary for sample depletion:

$$n_{\text{eff}} \approx \frac{1}{\sum_i [w_k^{(i)}]^2}. \tag{4}$$

The number of effective samples is between $1 \leq n_{\text{eff}} \leq n$, where the upper bound indicates that all particles are equally weighted and the lower bound that one particle has all the weight. Thus the resampling threshold can be determined, for instance, to be $n_{\text{eff}} < \frac{2}{3}n$.

If the terrain matching algorithm completely looses track of the position, the effective number of particles tends to drop drasticly at once. Therefore, on occasions when this happens, we propose a reinitialization strategy which resets the particle filter when the number of effective particles drop below $\frac{1}{3}n$ on two consecutive steps.

B. Pedestrian dead reckoning

The knowledge of the movement of the target is encoded into the dynamical model in the terrain matching algorithm (step 1 in Alg. 1). In pedestrian positioning this model is usually referred to as the ‘pedestrian dead reckoning’ (PDR) component. In theory, this PDR information can originate from various sources, such as device-provided odometry, wheel encoders, pure inertial navigation, or step-detector based movement indications.

In this study the interest is in matching the user-acquired magnetic trajectory with the magnetic terrain. Therefore we put less interest in the PDR model, and employ a simple baseline model for the user movement. The model assumes that the attitude and heading reference system (AHRS, see, *e.g.*, [10, 24]) estimating the orientation of the mobile device is able to return decent (but noisy and drifting) estimates of relative heading for each step. The AHRS operates on gyroscope and accelerometer measurements only.

In our terrain navigation setup, the state variables are $\mathbf{x}_k = (\mathbf{p}_k, \theta_k)$, where $\mathbf{p}_k$ stands for the metric position at time t_k, and θ_k is the current heading estimate at that time instance. The directed random walk model is as follow:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{u}_k, \tag{5}$$

where $\mathbf{u}_k = 1.5\,\Delta t_k\,(\cos\theta_k, \sin\theta_k)$, with a probability of 0.95, and $\mathbf{u}_k = \mathbf{0}$, with a probability of 0.05. The stand-still model allows the user to stop, and the default walking speed parameter 1.5 m/s corresponds to normal walking at approximately 5.4 km/h.

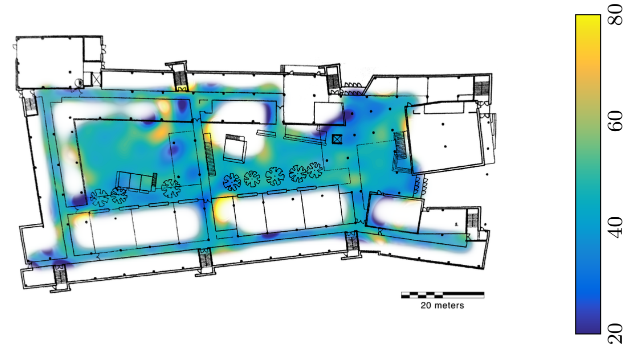


Fig. 3: The norm of the mapped and interpolated magnetic field. The opacity of the magnetic field estimate follows the certainty (marginal standard deviation) of the Gaussian process estimate.

The heading estimate θ_0 is initialized by matching the first magnetometer observation to the magnetic map, and it is updated from relative information provided by the AHRS system on each step: $\theta_k = \theta_{k-1} + \Delta\theta_k + q_k$, where $q_k \sim \mathrm{N}(0, \Delta t_k/2)$. The random walk models for the position and heading together define the dynamic (prediction) model

$$\mathbf{x}_{k+1} \sim p(\mathbf{x}_{k+1} \mid \mathbf{x}_k) \tag{6}$$

which is a probabilistic model for the transition between the previous state and the next state. This model is driven by the AHRS estimates of the orientation. We also use this model as the importance distribution in the particle filter.

C. Modeling the magnetic field by Gaussian processes

The modeling and interpolation of the magnetic field map is based on the methodology presented by Solin *et al.* [5], where additional knowledge from the physical properties of the magnetic field is encoded into a Gaussian process regression model.

The magnetic field is a vector field that obeys laws of physics known as Maxwell’s equations. When the spatial locations $\mathbf{x}$ at which the magnetic field is observed or estimated are far enough from any free currents (not inside building structures), we may consider a latent scalar potential field $\varphi(\mathbf{x})$ such that $\varphi : \mathbb{R}^3 \to \mathbb{R}$, where $\mathbf{x} \in \mathbb{R}^3$ is the spatial coordinate.

Gaussian processes (see, *e.g.*, [6]) are convenient and widely used tools in spatial statistics and machine learning. Their strength is the ease of encoding prior knowledge into the model through a covariance function structure. We assume the magnetic scalar potential field to be a realization of a Gaussian process prior and the observations (magnetic field readings) to be the gradients $\mathbf{y}_i \in \mathbb{R}^3$ of this field corrupted by Gaussian noise:

$$\begin{aligned} \varphi(\mathbf{x}) &\sim \mathcal{GP}(0, \kappa_{\text{lin.}}(\mathbf{x}, \mathbf{x}') + \kappa_{\text{SE}}(\mathbf{x}, \mathbf{x}')), \\ \mathbf{y}_i &= -\nabla\varphi(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_i} + \boldsymbol{\varepsilon}_i, \end{aligned} \tag{7}$$

where $\boldsymbol{\varepsilon}_i \sim \mathrm{N}(\mathbf{0}, \sigma^2_{\text{noise}}\mathbf{I}_3)$, for each observation $i = 1, 2, \ldots, n$.

The local Earth's magnetic field contributes linearly to the scalar potential as

$$\kappa_{\text{lin.}}(\mathbf{x}, \mathbf{x}') = \sigma^2_{\text{lin.}} \, \mathbf{x}^{\mathsf{T}} \mathbf{x}, \tag{8}$$

where $\sigma^2_{\text{lin.}}$ is a magnitude scale hyperparameter.

For the local variations in the magnetic field we use a squared exponential covariance function which allows for modeling anomalies induced by small-scale fluctuations and building structures:

$$\kappa_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma^2_{\text{SE}} \exp\left(- \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\,\ell^2_{\text{SE}}} \right), \tag{9}$$

where the hyperparameters σ^2_{SE} and ℓ_{SE} represent the magnitude scale and the characteristic length-scale, respectively.

The model now has four hyperparameters: two magnitude scale parameters ($\sigma^2_{\text{lin.}}$ and σ^2_{SE}), a length-scale parameter (ℓ_{SE}), and a noise scale parameter (σ^2_{noise}). These parameters can be learned from the data by maximizing the marginal likelihood, or fixed to sensible values describing typical variation.

The Gaussian process model can be used for modeling and interpolation of the local magnetic field by first collecting a batch of mapping data. This data consists of a set of input–output pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ at a discrete set of spatial inputs $\mathbf{x}_i$ and the (noisy) magnetic field observations $\mathbf{y}_i \in \mathbb{R}^3$ at those locations. In the following, we assume that the magnetic field readings collected during mapping are calibrated and corrected for rotation.

The Gaussian process regression model in Equation (7) provides a means of estimating the predictive magnetic field observation $\mathbf{y}_*$ at an unseen test input $\mathbf{x}_*$. Thus returning the following marginal predictions for the vector field components x, y, and z ($j = 1, 2, 3$):

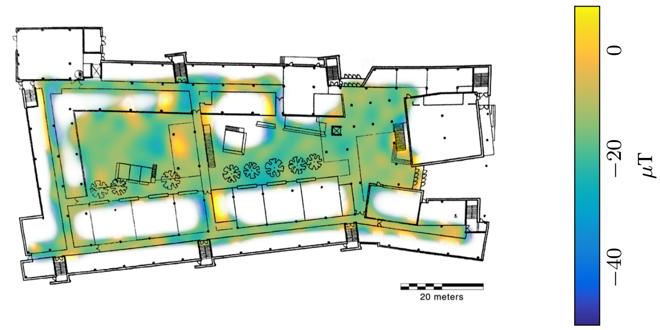
$$\mathbf{y}_{j,*} \sim \mathrm{N}(\mathbf{y}_{j,*} \mid \mathrm{E}[\mathbf{y}_{j,*} \mid \mathbf{x}_*, \mathcal{D}], \mathrm{var}[\mathbf{y}_{j,*} \mid \mathbf{x}_*, \mathcal{D}]). \tag{10}$$

This information can be utilized in the terrain matching algorithm (step 2 in Alg. 1) during the update step, as it directly gives the required probability formulation $p(\mathbf{y} \mid \mathbf{x})$ required during the weight calculation:

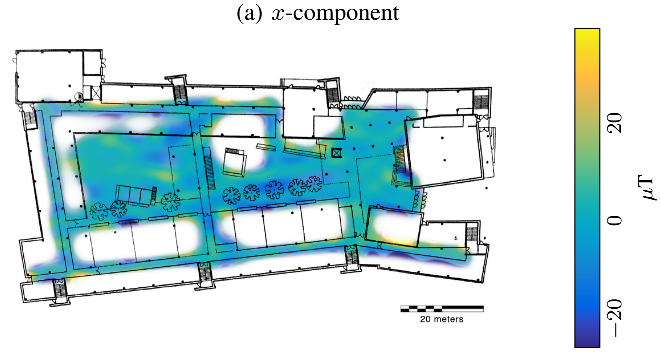
$$p(\mathbf{y} \mid \mathbf{x}_*) = \sum_{j=1}^{3} \mathrm{N}(\mathbf{y}_{j,*} \mid \mathrm{E}[\mathbf{y}_{j,*} \mid \mathbf{x}_*, \mathcal{D}], \mathrm{var}[\mathbf{y}_{j,*} \mid \mathbf{x}_*, \mathcal{D}]), \tag{11}$$

where $\mathbf{y}$ denotes the observed magnetic field, and $\mathbf{y}_*$ the predicted magnetic field from the GP regression model. Orientation correction was accounted for by using an AHRS algorithm for matching the z-component direction, and the heading θ in the state variable $\mathbf{x}$. As the variance grows outside mapped areas, the measurement model also implicitly restricts the particle movement to traversable areas defined during mapping.

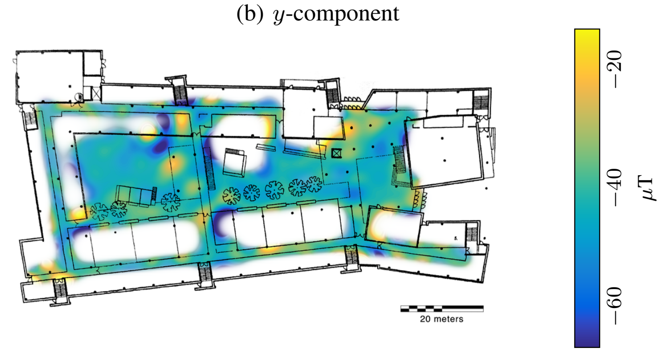
In the particle filter, the initialization strategy was also based on the magnetic field map. However, during initialization only the z-component and the magnitude of the perpendicular xy-component were used due to the lack of orientation information.



(a) x-component



(b) y-component



(c) z-component

Fig. 4: The components of the mapped and interpolated magnetic field. The opacity of the magnetic field estimate follows the certainty (marginal standard deviation) of the Gaussian process estimate.

TABLE I: Results for the 100 test paths (68 converged) each 30 s in length.

	Median	Mean	Standard deviation
Time-to-convergence	11.79 s	14.18 s	7.97 s
Distance-to-convergence	13.42 m	17.90 m	10.22 m
Error after convergence	4.87 m	9.28 m	10.86 m
Total error	18.49 m	19.42 m	11.47 m

III. Results

We illustrate the feasibility of the terrain navigation setup which combines a baseline PDR model, the a probabilistic map of the magnetic field, and a particle filter based map matching algorithm.

We consider an example of mapping of and self-localization in an indoor environment. The test environment was chosen

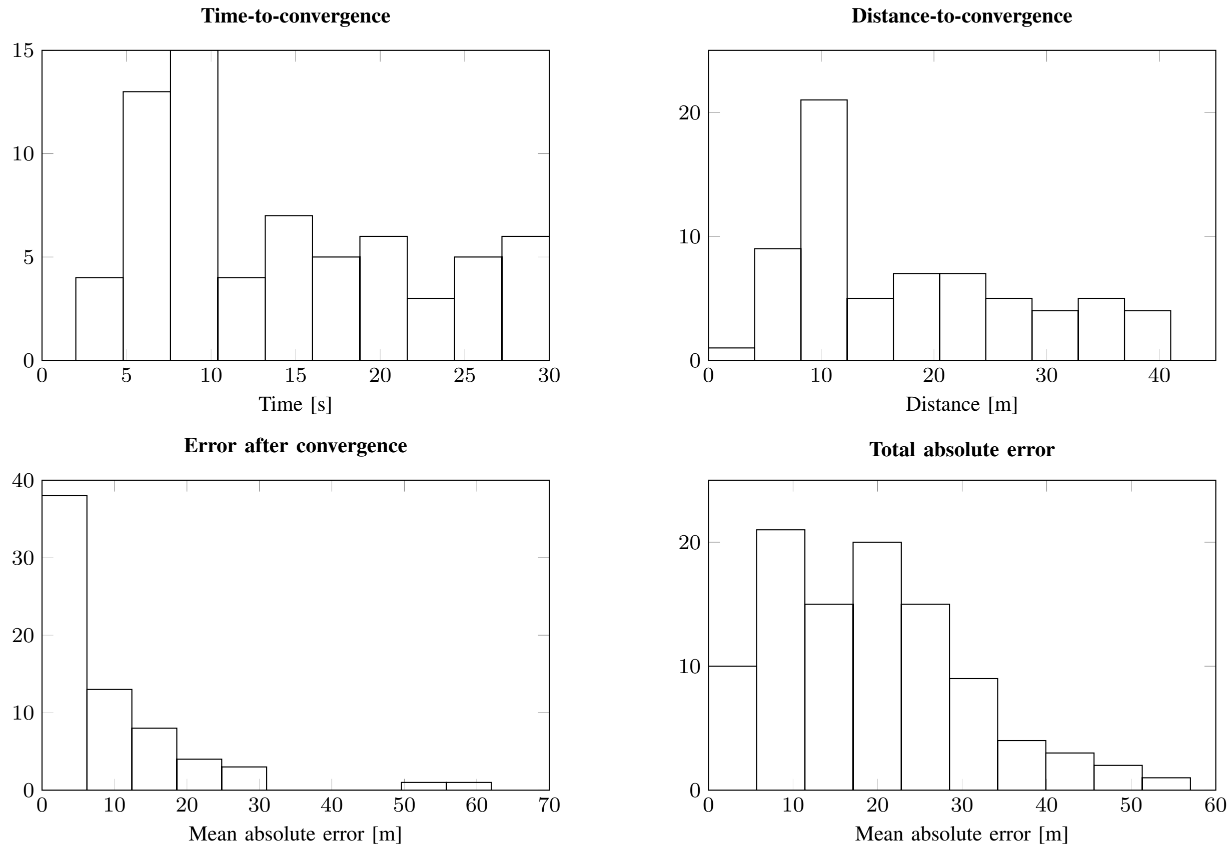


Fig. 5: Results for the 100 empirical test paths. The first plot shows a histogram of time-to-convergence for the 68 sessions which counted as converged within 30 seconds from starting the session. The second figure shows the distance travelled at convergence time for these sessions, and the third figure shows the mean absolute error after convergence. The last plot is included for comparison; it shows the mean absolute error over the whole paths for all 100 sessions.

so that it encloses both open areas and narrow corridors. The venue is located on the Aalto University campus in Espoo, Finland. This example covers the public space on the ground floor of the building. A floor plan sketch of the venue shown in Figure 3.

Mapping was performed on foot by using a Nexus 5 smartphone device (Google Inc., manufactured by LG, with a AKM AK8963 3-axis magnetometer). The mapping positions on the floorplan were matched using a foot-mounted sensor that internally uses short time-scale inertial navigation (GT Silicon Pvt Ltd.). The foot-path was manually aligned to the floor plan image. Slight drift in the foot-sensor paths was encountered.

In total, the mapping paths measured some 867 meters of walking and the total acquisition time was 13.74 minutes, during which 41,219 vector-valued magnetometer readings were obtained (sampling rate 50 Hz). The magnetometer was calibrated prior to mapping using standard spherical calibration.

The magnetic Gaussian process regression map was generated by the batch processing approach presented in [5]. In this study, we used fixed hyperparameters: $\sigma^2_{\text{lin.}} = 800\ (\mu\text{T})^2$, $\ell_{\text{SE}} = 1$ m, $\sigma_{\text{SE}} = 600\ (\mu\text{T})^2$, and $\sigma^2_{\text{noise}} = 5\ (\mu\text{T})^2$. The mapped area corresponds approximately to 2330 m^2, and the magnitude of the generated magnetic field map is visualized in Figure 3. The individual vector components are shown in Figure 4. The opacity in the visualizations follow the marginal standard deviation of the predictions.

Validation data for testing was collected independently from the mapping data two weeks after the initial mapping data was collected. The test data was acquired with a similar foot sensor setup providing ground-truth positions, and later split into separate shorter validation paths. The collected validation data measured some 430 meters in length and covered largely the same open areas and corridors as the mapping data, but also leaving the mapped areas momentarily and exiting the building altogether. The total acquisition time for the validation data was 6.17 minutes, during which 18,536 vector-valued measurements of the magnetic field were obtained (sampling rate 50 Hz).

The terrain matching algorithm tends to converge rather quickly, and therefore we split the validation data into shorter test paths. We split the long validation set into 100 test paths, each 30 seconds in length and with a random initialization time. To generate more versatile test cases the paths are either extracted as is or flipped. The paths are finally downsampled

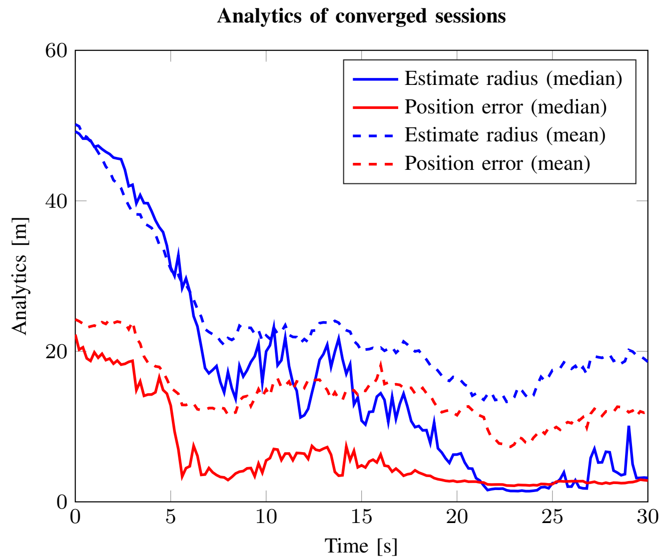


Fig. 6: Average behaviour over time for each of the test cases. The plot shows both the mean and median of the absolute positioning error and the evolution of the 95% certainty radius of the estimate.

from 50 Hz to 5 Hz. This all was done in order to provide a challenging enough test setup which would match real use cases.

The particle filter terrain matching algorithm was applied to each of the test paths with $n = 5000$ particles. The measurement noise parameters were tuned such that during initialization the magnetic field estimate is trusted more ($\sigma^2_{\text{noise}} = 3^2\ (\mu\text{T})^2$)—the initialization uses the heading-invariant magnetic field estimate. During the measurement updates, the mismatch in device and user heading cause additional uncertainty to the matching of the x- and y-components. Therefore the measurement noise levels were increased to $8^2\ (\mu\text{T})^2$ for those components.

We run the terrain matching algorithm for each of the 100 test sessions. We measure the performance of the terrain matching approach in terms of absolute error to the ground-truth position collected by the foot-sensor. Convergence of the algorithm is defined in terms of simultaneously following the 95% certainty radius and position (median of the particle swarm) estimate. The algorithm is regarded converged once the positioning error is less than 5 meters and the 95% certainty radius is at maximum 5 meters.

Figure 2 shows the evolution of the algorithm for one of the test sessions. The initialization based on the first magnetometer reading is shown in the first plot, where after the state of the algorithm is visualized with 2 second intervals until convergence at 8 seconds. The figures also show the ground-truth (black cross) and the true path walked since initialization.

Of the 100 test paths 68 reached convergence within the 30 second time-frame. Table I shows a set of summary results for these paths. In practice, the terrain matching algorithm requires that the user moves in order for the test trajectory to converge. Because the user did also stay stationary, we report both time-to-convergence and distance-to-convergence. As seen in the table, these values tend to peak around 10–15 seconds/meters. More granularity is provided in the histograms in Figure 5, which show the spread of the convergence time and distance. We also report the mean absolute error after convergence, and the absolute error calculated over all test paths (including non-converged cases).

The absolute error after convergence appears slightly pessimistic in the table and histograms. This is due to the estimate still shrinking after reaching the convergence criterion. Figure 6 shows the median and mean of the evolution of both the absolute positioning error and the evolution of the 95% certainty radius of the estimate. The jitter in the mean is due to some test paths which diverged after the initial convergence and were reinitialized.

IV. Discussion and Conclusion

In this article we have presented a magnetic field based positioning algorithm which uses a physics-aided Gaussian process model for the magnetic field map and a terrain navigation particle filter for positioning on the map. The algorithm uses pedestrian dead reckoning and orientation tracking results as stochastic inputs. The algorithm was applied to smartphone-based indoor positioning at the Aalto University campus. Although the used particle filter model was very simple, the Gaussian process based magnetic map seems to be informative enough for the particle filter to give good results.

There are many ways how the algorithm can be improved. The PDR algorithm which we used was a fairly directed random walk model, and with more careful modeling of the movements its accuracy could be significantly improved. By careful design of the PDR, it is possible get rid of the orientation constraint: the current model assumes that the orientation and the direction of movement coincide, which might not reflect reality.

In this paper we only used magnetic field measurements, but similarly it is possible to create maps of Wi-Fi and BLE signals (such as beacons), and to use them in the particle filter as well. These measurements can also be used to initialize the positioning. This algorithm could be extended to simultaneous localization and mapping (SLAM) using magnetic fields such that localization is done while building the map [25, 26].

We also calibrated the magnetometer beforehand, although it is also possible to calibrate the magnetometer online during positioning. Similarly we can estimate the calibration parameters of the other sensors offline or online.

We also note that after the initial convergence, our simple baseline PDR based algorithm was indeed able to reach the 1–2 meter error region which is often considered as the feasibility limit for indoor positioning methods. Together with the improvements outlined above, the algorithm framework is likely to provide an accurate indoor positioning methodology with fast convergence and consistent one meter (or less) error in typical indoor spaces.

Acknowledgment

The authors would like to thank Academy of Finland and IndoorAtlas Ltd. for funding.

References

[1] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation and Integration*. Wiley, 2001.

[2] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1028–1035, 2009.

[3] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?" in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sydney, Australia, November 2012, pp. 1–9.

[4] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, "Simultaneous localization and mapping using ambient magnetic field," in *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Salt Lake City, UT, USA, September 2010, pp. 14–19.

[5] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä, "Modeling and interpolation of the ambient magnetic field by Gaussian processes," *arXiv preprint arXiv:1509.04634*, 2015.

[6] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[7] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*. The Institution of Electrical Engineers, 2004.

[8] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Robust pedestrian dead reckoning (R-PDR) for arbitrary mobile device placement," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Busan, Korea, October 2014, pp. 187–196.

[9] J. Bojja, J. Collin, S. Särkkä, and J. Takala, "Pedestrian localization in moving platforms using dead reckoning, particle filtering and map matching," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia: IEEE, April 2015, pp. 1116–1120.

[10] S. Särkkä, V. Tolvanen, J. Kannala, and E. Rahtu, "Adaptive Kalman filtering and smoothing for gravitation tracking in mobile systems," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Banff, Canada, October 2015, pp. 1–7.

[11] J.-O. Nilsson, A. K. Gupta, and P. Händel, "Foot-mounted inertial navigation made easy," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Busan, Korea, October 2014, pp. 24–29.

[12] S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[13] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010.

[14] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[15] N. Bergman, L. Ljung, and F. Gustafsson, "Terrain navigation using Bayesian statistics," *IEEE Control Systems*, vol. 19, no. 3, pp. 33–40, 1999.

[16] C. Tyrén, "Magnetic anomalies as a reference for ground-speed and map-matching navigation," *Journal of Navigation*, vol. 35, no. 02, pp. 242–254, 1982.

[17] ——, "Magnetic terrain navigation," in *Proceedings of the 1987 5th International Symposium on Unmanned Untethered Submersible Technology*, vol. 5. IEEE, 1987, pp. 245–256.

[18] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*. Kirkkonummi, Finland: IEEE, October 2010, pp. 1–10.

[19] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao, "Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 7, pp. 1443–1457, 2015.

[20] J. Menke and A. Zakhor, "Multi-modal indoor positioning of mobile devices," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Banff, Canada, October 2015.

[21] S. Shahidi and S. Valaee, "GIPSy: Geomagnetic indoor positioning system for smartphones," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Banff, Canada, October 2015, pp. 1–7.

[22] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[23] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[24] S. O. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," x-io Technologies and University of Bristol, Bristol, UK, Tech. Rep., 2010.

[25] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proceedings of the International Workshop of Intelligent Robots and Systems (IROS)*, vol. 3. Osaka, Japan: IEEE, November 1991, pp. 1442–1447.

[26] H. Durrant-Whyte and T. Bailey, "Simultaneous localiza-

tion and mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.