Kok, Manon; Solin, Arno

Scalable Magnetic Field SLAM in 3D Using Gaussian Process Maps

Published in:
21st International Conference on Information Fusion

DOI:
10.23919/ICIF.2018.8455789

Published: 05/09/2018

Document Version
Peer reviewed version

Please cite the original version:
Kok, M., & Solin, A. (2018). Scalable Magnetic Field SLAM in 3D Using Gaussian Process Maps. In *21st International Conference on Information Fusion: FUSION 2018* (pp. 1353-1360). [8455789] IEEE. https://doi.org/10.23919/ICIF.2018.8455789

Scalable Magnetic Field SLAM in 3D Using Gaussian Process Maps

Manon Kok*
Delft University of Technology
Email: m.kok-1@tudelft.nl

Arno Solin*
Aalto University
Email: arno.solin@aalto.fi

***Abstract*—We present a method for scalable and fully 3D magnetic field simultaneous localisation and mapping (SLAM) using local anomalies in the magnetic field as a source of position information. These anomalies are due to the presence of ferromagnetic material in the structure of buildings and in objects such as furniture. We represent the magnetic field map using a Gaussian process model and take well-known physical properties of the magnetic field into account. We build local maps using three-dimensional hexagonal block tiling. To make our approach computationally tractable we use reduced-rank Gaussian process regression in combination with a Rao–Blackwellised particle filter. We show that it is possible to obtain accurate position and orientation estimates using measurements from a smartphone, and that our approach provides a scalable magnetic field SLAM algorithm in terms of both computational complexity and map storage.**

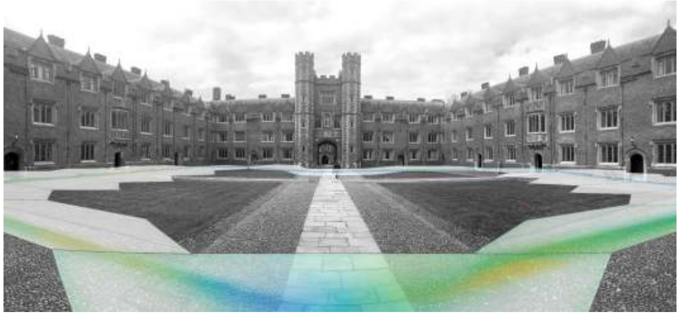


Fig. 1. Illustration of a magnetic field map built using our SLAM approach. Each hexagonal block (with radius of 5 m and height of 4 m) covers a Gaussian process magnetic map volume of $\sim$260 m^3 in a state dimension of 256. A more detailed map is provided in Fig. 5.

I. Introduction

The use of the magnetic field as a source of position information for indoor navigation is a promising novel approach that has gained interest in recent years [1]–[8]. It relies upon the spatial variation of the ambient magnetic field which is typically due to ferromagnetic material in the structures of buildings and to a lesser extent due to the presence of *e.g.* furniture. The advantage of using the magnetic field for positioning is that it can be measured by a small device, without additional infrastructure and without line-of-sight requirements. Furthermore, magnetometers are nowadays present in (almost) any smartphone. Crucial to this approach is the ability to build maps of the magnetic field that can be used for interpolation and extrapolation and the ability to localise inside this map. In this paper, we present a scalable and fully three-dimensional *simultaneous localisation and mapping (SLAM)* approach that builds a map of the magnetic field while simultaneously localising the sensor in the map.

Extending previous work [9], we build the magnetic field map using Gaussian process regression [10] and incorporate physical knowledge about the magnetic field known from Maxwell's equations in the Gaussian process prior. To overcome issues with computational complexity, we use the rank-reduced approach first presented in [11]. This results in a representation that fits perfectly in a *Rao–Blackwellised particle filter (RBPF)* [12] which can be used both for building the map and for localisation in the map, resulting in a tractable localisation algorithm. Example results can be found in Fig. 1.

The approach presented in [11] relies on a basis function expansion on a specific domain where the required number of basis functions scales with the size of the domain. To allow for mapping large 3D areas, we propose a map representation using hexagonal block tiling, which aims at providing a very compact (in terms of required storage per volume) representation of the magnetic field map (all three vector field components and associated uncertainties). Using this approach, hexagonal block tiles are created whenever particles enter unexplored areas, resulting in a growing number of local Gaussian process maps.

We present experimental results where we obtain magnetometer measurements as well as odometry from a smartphone. This odometry gives us information about the change in position and orientation but the resulting pose estimates drift over time. This drift is corrected by making use of the magnetometer measurements and the magnetic field map. We show that accurate 6D pose and 3D map estimates can be obtained in large scale and 3D experiments, as illustrated in Fig. 1.

Previous work on magnetic field SLAM typically assumes that the motion is in 2D or 2.5D [13]–[16]. To the best of the authors' knowledge, this is the first fully 3D approach to magnetic field SLAM. It is also the first time the reduced-rank approach from [11] has been used for SLAM and that the domain in this approach has been chosen to be hexagonal.

II. Problem formulation

We are interested both in estimating the position and the orientation of the sensor and in building a map of the magnetic

* This work was undertaken whilst MK was a Research Associate at the Department of Engineering, University of Cambridge, and AS was a Visiting Research Fellow at the Department of Engineering, University of Cambridge.

field. We denote the position of the sensor at time instance t by $\mathbf{p}_t^{\text{w}}$. The superscript w refers to the *world frame*. In our SLAM formulation, it is always possible to move or rotate the estimated map while simultaneously moving and rotating the position and orientation estimates. In other words, the absolute position and orientation are not observable. Because of this, without loss of generality we choose the origin of the world frame to be equal to the initial position of the sensor and its axes to be aligned with the coordinate system defined by the initial orientation.

The orientation of the sensor is denoted by $\mathbf{q}_t^{\text{wb}}$, where b refers to the *body frame*. Its origin lies in the centre of the magnetometer triad and its axes are aligned with the axes of the magnetometer. We encode the orientation using a unit quaternion. There is a direct mapping from the quaternion $\mathbf{q}_t^{\text{wb}}$ to the rotation matrix $\mathbf{R}_t^{\text{wb}}$ and because of this we use these interchangeably. The double superscript wb denotes the rotation from body frame to world frame. The reverse rotation is denoted by bw and hence, $\mathbf{R}_t^{\text{wb}} = \left(\mathbf{R}_t^{\text{bw}}\right)^{\mathsf{T}}$.

We denote our estimate of the magnetic field map by $\mathbf{m}_t$. More details about the exact form of $\mathbf{m}_t$ can be found in Section III-A. Note that we assume that the map of the magnetic field does not change over time. The subscript t refers to the fact that our belief of the map changes over time because more data results in more information about the map.

The resulting state vector $\mathbf{x}_t$ is given by

$$\mathbf{x}_t = \left((\mathbf{p}_t^{\text{w}})^{\mathsf{T}} \quad (\mathbf{q}_t^{\text{wb}})^{\mathsf{T}} \quad \mathbf{m}_t^{\mathsf{T}} \right)^{\mathsf{T}}. \tag{1}$$

The dynamics of the position and orientation are modelled in terms of the change in position $\Delta \mathbf{p}_t$ and orientation $\Delta \mathbf{q}_t$ obtained from the odometry information. We consider these to be inputs to our dynamic model. Using the assumption that the magnetic field does not vary over time, our dynamic model can be written as

$$\mathbf{x}_{t+1} = \begin{pmatrix} \mathbf{p}_t^{\text{w}} + \Delta \mathbf{p}_t + \boldsymbol{\varepsilon}_{\text{p},t} \\ \Delta \mathbf{q}_t \odot \mathbf{q}_t^{\text{wb}} \odot \exp_{\text{q}}(\boldsymbol{\varepsilon}_{\text{q},t}) \\ \mathbf{m}_t \end{pmatrix}. \tag{2}$$

We assume that the noise on the position ($\boldsymbol{\varepsilon}_{\text{p},t}$) and on the orientation ($\boldsymbol{\varepsilon}_{\text{q},t}$) is Gaussian with $\boldsymbol{\varepsilon}_{\text{p},t} \sim \mathcal{N}(\mathbf{0}, \Delta T \sigma_{\text{p}}^2 \mathbf{I}_3)$ and $\boldsymbol{\varepsilon}_{\text{q},t} \sim \mathcal{N}(\mathbf{0}, \Delta T \sigma_{\text{q}}^2 \mathbf{I}_3)$. Here, ΔT is the (possibly time-varying) time between subsequent samples. In (2), $\odot$ denotes the quaternion multiplication and $\exp_{\text{q}}(\boldsymbol{\varepsilon}_{\text{q},t})$ refers to the representation of the noise vector $\boldsymbol{\varepsilon}_{\text{q},t} \in \mathbb{R}^3$ as a unit quaternion. The dynamics of the orientation $\mathbf{q}_t^{\text{wb}}$ can be interpreted as the rotation equivalent of the dynamics of the position $\mathbf{p}_t^{\text{w}}$. For more information about quaternion algebra and representations of orientation, see *e.g.* [17].

We use the magnetometer measurements to provide information about the magnetic field map as well as the position and orientation of the sensor. This can be modelled as

$$\mathbf{y}_{\text{m},t} = \mathbf{R}_t^{\text{bw}} \nabla \boldsymbol{\Phi}_t(\mathbf{p}_t^{\text{w}}) \mathbf{m}_t + \boldsymbol{\varepsilon}_{\text{m},t}. \tag{3}$$

Here, $\nabla \boldsymbol{\Phi}_t(\mathbf{p}_t^{\text{w}}) \mathbf{m}_t$ is the prediction of the magnetic field at the location $\mathbf{p}_t^{\text{w}}$. More details about the specific form of $\nabla \boldsymbol{\Phi}_t(\mathbf{p}_t^{\text{w}})$ can be found in Section III-A. Note that the magnetic field map is represented in world frame. Hence, the magnetometer measurements are modelled as the prediction $\nabla \boldsymbol{\Phi}_t(\mathbf{p}_t^{\text{w}}) \mathbf{m}_t$, rotated to body frame.

Together with a prior on the state at the first time instance, the models (2) and (3) constitute the state space model that we use in the RBPF to estimate the state defined in (1). Because the initial position and orientation fix the world coordinate frame w, $\mathbf{p}_1^{\text{w}}$ and $\mathbf{q}_1^{\text{wb}}$ are set to the initial position and orientation with zero uncertainty. The magnetic field map at the first time instance is set to be equal to the Gaussian process prior. More details about the Gaussian process map will be discussed in Section III-A. The RBPF implementation will subsequently be discussed in Section III-B.

III. Methods

A. Gaussian process magnetic field map

Building on previous work [9], we build a Gaussian process magnetic field map in which we encode physical knowledge about the magnetic field. First of all, we know that the measured magnetic field consists of the Earth's magnetic field and the magnetic field due to anomalies induced by small-scale variations and building structures. Furthermore, using Maxwell's equations and classical electromagnetism [18], we know that the magnetic field due to the anomalies can be modelled using a *latent scalar potential field* $\varphi(\mathbf{p})$. Here, $\varphi : \mathbb{R}^3 \to \mathbb{R}$, where $\mathbf{p} \in \mathbb{R}^3$ is the spatial coordinate in world frame and the magnetometer measures the *derivative* of this scalar potential. We assume the scalar potential field to be a realisation of a Gaussian process prior and the magnetometer measurements to be corrupted by Gaussian noise, resulting in the following model

$$\begin{aligned} \varphi(\mathbf{p}) &\sim \text{GP}(0, \kappa_{\text{lin.}}(\mathbf{p}, \mathbf{p}') + \kappa_{\text{SE}}(\mathbf{p}, \mathbf{p}')), \\ \mathbf{y}_{\text{m},t} &= \mathbf{R}_t^{\text{bw}} \nabla \varphi(\mathbf{p}) \big|_{\mathbf{p}=\mathbf{p}_t^{\text{w}}} + \boldsymbol{\varepsilon}_{\text{m},t}, \end{aligned} \tag{4}$$

where $\boldsymbol{\varepsilon}_{\text{m},t} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{m}}^2 \mathbf{I}_3)$, for each observation $t = 1, 2, \ldots, N_T$. The local Earth's magnetic field contributes *linearly* to the scalar potential which we model in terms of a linear covariance function [19] as

$$\kappa_{\text{lin.}}(\mathbf{p}, \mathbf{p}') = \sigma_{\text{lin.}}^2 \, \mathbf{p}^{\mathsf{T}} \mathbf{p}', \tag{5}$$

where $\sigma_{\text{lin.}}^2$ is the magnitude scale hyperparameter. The magnetic field anomalies are captured by the squared exponential covariance function

$$\kappa_{\text{SE}}(\mathbf{p}, \mathbf{p}') = \sigma_{\text{SE}}^2 \, \exp\left(-\frac{\|\mathbf{p} - \mathbf{p}'\|^2}{2\ell^2} \right), \tag{6}$$

where σ_{SE}^2 is a magnitude and ℓ the characteristic length-scale hyperparameter. Including the additional prior information introduced by the scalar potential in (4) results in a coupling between the three components of the magnetic field according to physical laws. This has been shown in [9] to improve prediction accuracy over simply modelling each magnetic field component separately.

Using the model (4), it is possible to predict the magnetic field at previously unseen locations. In practice, however, this quickly becomes computationally intractable because of the large amount of magnetometer measurements we collect and the fact that the computational complexity scales as $\mathcal{O}(N_T^3)$. A computationally tractable solution was presented in [9] where the problem was projected on the eigenbasis of the (negative) Laplace operator in a confined domain $\Omega \subset \mathbb{R}^3$. In this domain, the eigendecomposition of the Laplace operator subject to Dirichlet boundary conditions can be solved as

$$\begin{cases} -\nabla^2 \phi_j(\mathbf{p}) = \lambda_j^2 \phi_j(\mathbf{p}), & \mathbf{p} \in \Omega, \\ \phi_j(\mathbf{p}) = 0, & \mathbf{p} \in \partial\Omega. \end{cases} \tag{7}$$

Using this eigendecomposition, it is possible to approximate the covariance function in (4) as

$$\begin{aligned} \kappa(\mathbf{p}, \mathbf{p}') &= \kappa_{\text{lin.}}(\mathbf{p}, \mathbf{p}') + \kappa_{\text{SE}}(\mathbf{p}, \mathbf{p}') \\ &\approx \kappa_{\text{lin.}}(\mathbf{p}, \mathbf{p}') + \sum_{j=1}^{m} S_{\text{SE}}(\lambda_j)\, \phi_j(\mathbf{p})\, \phi_j(\mathbf{p}'), \end{aligned} \tag{8}$$

where $\kappa_{\text{SE}}(\mathbf{p}, \mathbf{p}')$ is approximated using m basis functions ϕ_j and their corresponding eigenvalues λ_j. The exact form of ϕ_j and λ_j depends on the shape of the domain Ω. Note that $S_{\text{SE}}(\cdot)$ denotes the spectral density function of the squared exponential kernel $S_{\text{SE}}(\cdot)$ for which a closed form solution exists, see [9]. Using the approximation (8) with the boundary conditions defined in (7) implies that our model of the magnetic field reverts back to the Earth's magnetic field at the boundary of the domain.

A fixed domain Ω is not desirable in SLAM, as we do not know *a priori* the spatial extent of the final magnetic field map. Furthermore, the number of basis functions that is needed for a good approximation scales with the size of the domain and for large scale SLAM problems will hence become intractable. We therefore propose a scalable representation where we split the magnetic map into a three-dimensional grid of hexagonal blocks with radius r and height $2L_{\text{z}}$, such that each subdomain d is given by

$$\begin{aligned} \Omega^{(d)} = \{\mathbf{p} \mid (p_1, p_2) &\in \text{hexagon}(r, \mathbf{p}_{1,2}^{(d)}), \\ p_3 &\in [p_3^{(d)} - L_{\text{z}}, p_3^{(d)} + L_{\text{z}}]\}. \end{aligned} \tag{9}$$

Here, $\mathbf{p}^{(d)}$ is the center point of the dth hexagon cell in the grid representation. The hexagonal grid provides an efficient way of maximising the area per number of basis functions for representing the magnetic field map, thus requiring less memory for storing the map representation.

To compute the basis functions ϕ_j and their corresponding eigenvalues λ_j, we need to solve the eigendecomposition of the Laplace operator (7) in a two-dimensional hexagon. This can not be solved in closed form, but instead it can be solved numerically. We set up the eigenvalue problem by composing the sparse stencil matrix corresponding to the Laplacian using a nine-point finite difference approximation. The solution itself is given by the Lanczos algorithm (see, *e.g.*, [20], callable in Matlab through `eigs`) that we use for solving the m largest real eigenvalues $[\lambda_n^{\text{hex}}]^2$—together with the corresponding eigenfunctions $\phi_n^{\text{hex}}(\mathbf{p})$. Fig. 2 shows the 40 first eigenfunctions of the negative Laplacian in a unit hexagonal domain with respect to Dirichlet boundary conditions.

The eigendecomposition is extended to cover the three-dimensional hexagonal grid cell $\Omega^{(d)}$ by considering the numerically solved horizontal eigendecomposition and a closed-form solution to the separable vertical dimension. This choice is motivated by the assumption that paths typically exhibit most variation in the horizontal direction while exhibiting less variation in the vertical direction. In other words, we assume most variation to happen in horizontal cross sections since human motion tends to be along two-dimensional planes with less-frequent ascending and descending. None of the computational benefits rely on this assumption, and it is also a sensible choice for other (non-bipedal) movement.

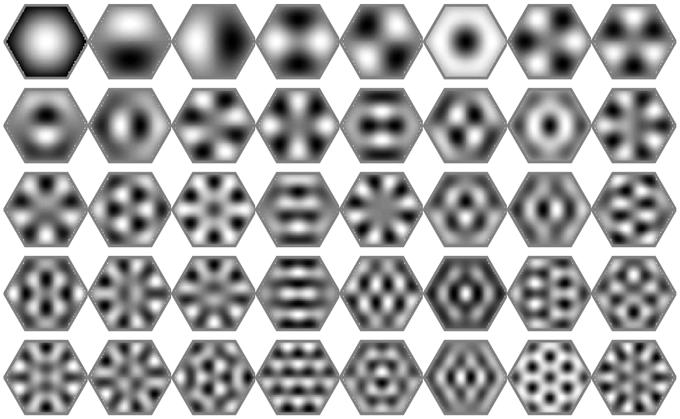


Fig. 2. The first 40 eigenfunctions of the negative Laplace operator in a hexagonal domain subject to Dirichlet boundary conditions. The eigenbasis can be effectively solved by a numerical solver, and the calculations are only required once.

The final eigenbasis then becomes the combination:

$$\phi_j(\mathbf{p}) = \phi_{n_{j,1}}^{\text{hex}}(\mathbf{p}_{1,2}) \left[\frac{1}{\sqrt{L_{\text{z}}}} \sin\!\left(\frac{\pi n_{j,2}(p_3 + L_{\text{z}})}{2L_{\text{z}}} \right) \right], \tag{10}$$

$$\lambda_j^2 = [\lambda_{n_{j,1}}^{\text{hex}}]^2 + \left(\frac{\pi n_{j,2}}{2L_d} \right)^2, \tag{11}$$

where the matrix $\mathbf{n} \in \mathbb{R}^{m \times 2}$ consists of an index set of permutations of integers $\{1, 2, \ldots, m\}$ (*i.e.*, $\{(1,1), (2,1), \ldots, (1,2), \ldots, (2,2), \ldots\}$) corresponding to the m largest eigenvalues.

The numerical solution for the two-dimensional hexagon is fast and takes less than a second even with a tight discretisation on an average laptop computer. Furthermore, because the basis functions (10) are independent of the hyperparameters and the input locations, they only need to be evaluated once, not causing any computational overhead in our SLAM algorithm.

The basis function expansion described above allows us to write the prior model for each hexagonal tile at time $t = 0$ in terms of a mean $\mathbf{m}_0 = \mathbf{0}_{m+3}$ and a covariance

$$\mathbf{P}_0 = \operatorname{diag}\left(\sigma_{\text{lin.}}^2, \sigma_{\text{lin.}}^2, \sigma_{\text{lin.}}^2, S_{\text{SE}}(\lambda_1), S_{\text{SE}}(\lambda_2), \ldots, S_{\text{SE}}(\lambda_m) \right). \tag{12}$$

The posterior can be updated sequentially [21] as new magnetometer data arrives at $t = 1, 2, \ldots, N_T$ and can be written as

$$\begin{aligned} \mathbf{S}_t &= \mathbf{C}_t \mathbf{P}_{t-1} \mathbf{C}_t^{\mathsf{T}} + \sigma_{\text{m}}^2 \mathbf{I}_3, \\ \mathbf{K}_t &= \mathbf{P}_{t-1} \mathbf{C}_t^{\mathsf{T}} \mathbf{S}_t^{-1}, \\ \mathbf{m}_t &= \mathbf{m}_{t-1} + \mathbf{K}_t(\mathbf{y}_{\text{m},t} - \mathbf{C}_t\, \mathbf{m}_{t-1}), \\ \mathbf{P}_t &= \mathbf{P}_{t-1} - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^{\mathsf{T}}, \end{aligned} \tag{13}$$

with $\mathbf{C}_t = \mathbf{R}_t^{\text{bw}} \nabla \mathbf{\Phi}_t$ and

$$\nabla \mathbf{\Phi}_t = \left(\nabla \mathbf{p}_t^{\mathsf{T}}, \nabla \phi_1(\mathbf{p}_t), \nabla \phi_2(\mathbf{p}_t), \ldots, \nabla \phi_m(\mathbf{p}_t) \right). \tag{14}$$

Note that (13) is a Kalman filter measurement update of the state $\mathbf{m}_t$ representing the magnetic field map in a specific tile. This fact will be exploited in our magnetic field SLAM algorithm presented in Section III-B.

B. Magnetic field SLAM

As discussed in Section II, we are interested in estimating the position and the orientation of the sensor while building a map of the magnetic field. We represent the map in terms of a varying number of hexagonal block tiles, each using the basis function expansion presented in Section III-A. We use a nonlinear filtering technique to estimate the position, the orientation and the map of each tile that has been visited. The state vector (1) has a relatively large dimension due to the representation of the magnetic field map $\mathbf{m}_t$. For instance, in Section IV we set the dimension of the state $\mathbf{m}_t$ to 256 per hexagonal block. However, $\mathbf{m}_t$ enters *conditionally linearly* into the state space model. It is possible to exploit this by using an RBPF [12] which uses a Kalman filter for the conditionally linear states $\mathbf{x}_t^{\text{l}}$ and a particle filter for the nonlinear states $\mathbf{x}_t^{\text{n}}$ where

$$\mathbf{x}_t^{\text{n}} = \begin{pmatrix} (\mathbf{p}_t^{\text{w}})^\mathsf{T} & (\mathbf{q}_t^{\text{wb}})^\mathsf{T} \end{pmatrix}, \qquad \mathbf{x}_t^{\text{l}} = \{\mathbf{m}_t\}_{d=1}^{N_{D,t}}. \tag{15}$$

Here, $N_{D,t}$ is the number of hexagonal block tiles that have been created up to time t.

In the RBPF, $i = 1, \dots, N_P$ particles having a position $\mathbf{p}_t^{\text{w},(i)}$ and an orientation $\mathbf{q}_t^{\text{wb},(i)}$ are used to represent the state at time t. Each of the particles also contains an estimate of the map $\mathbf{m}_t^{(i)}$ and its covariance $\mathbf{P}_t^{(i)}$. Note that $\mathbf{m}_t^{(i)}$ consists of the magnetic field map of each hexagonal block that particle i has visited. We assume that each tile is independent. The covariance of the linear state can therefore be described as $\mathbf{P}_t = \{\mathbf{P}_t\}_{d=1}^{N_{D,t}}$.

The dynamics of the state is chosen as in (2) and the measurement model is given in (3). It is important to note that, as discussed in Section II, the absolute position and orientation are not observable. Because of this, we set the initial position $\mathbf{p}_0^{\text{w},(i)}$ and orientation $\mathbf{q}_0^{\text{wb},(i)}$ of all particles $i = 1, \dots, N_P$ equal to an initial position and orientation $\mathbf{p}_0^{\text{w},(i)} = \mathbf{0}_3$, $\mathbf{q}_0^{\text{wb},(i)} = \mathbf{q}_0^{\text{wb}}$. We initialise the magnetic field according to the Gaussian process prior (see also Section III-A).

The state space model presented in Section II results in a fairly straightforward RBPF implementation. The major difference from a standard RBPF is that each particle only updates the local magnetic field map at its current location. Note that the computational complexity of the algorithm scales linearly with the number of particles and linearly with the number of hexagonal tiles per particle. The algorithm is summarised in Alg. 1.

In a practical magnetic field SLAM implementation care must be taken of the following in terms of updating the map:

- To avoid boundary effects due to the Dirichlet boundary conditions, the actual tile extends slightly outside of the hexagonal block domain for updating.
- To allow for a smooth transition from one tile to the next, particles close to the border of a tile also update the map of the neighbouring tiles.

Alg 1: Magnetic field SLAM

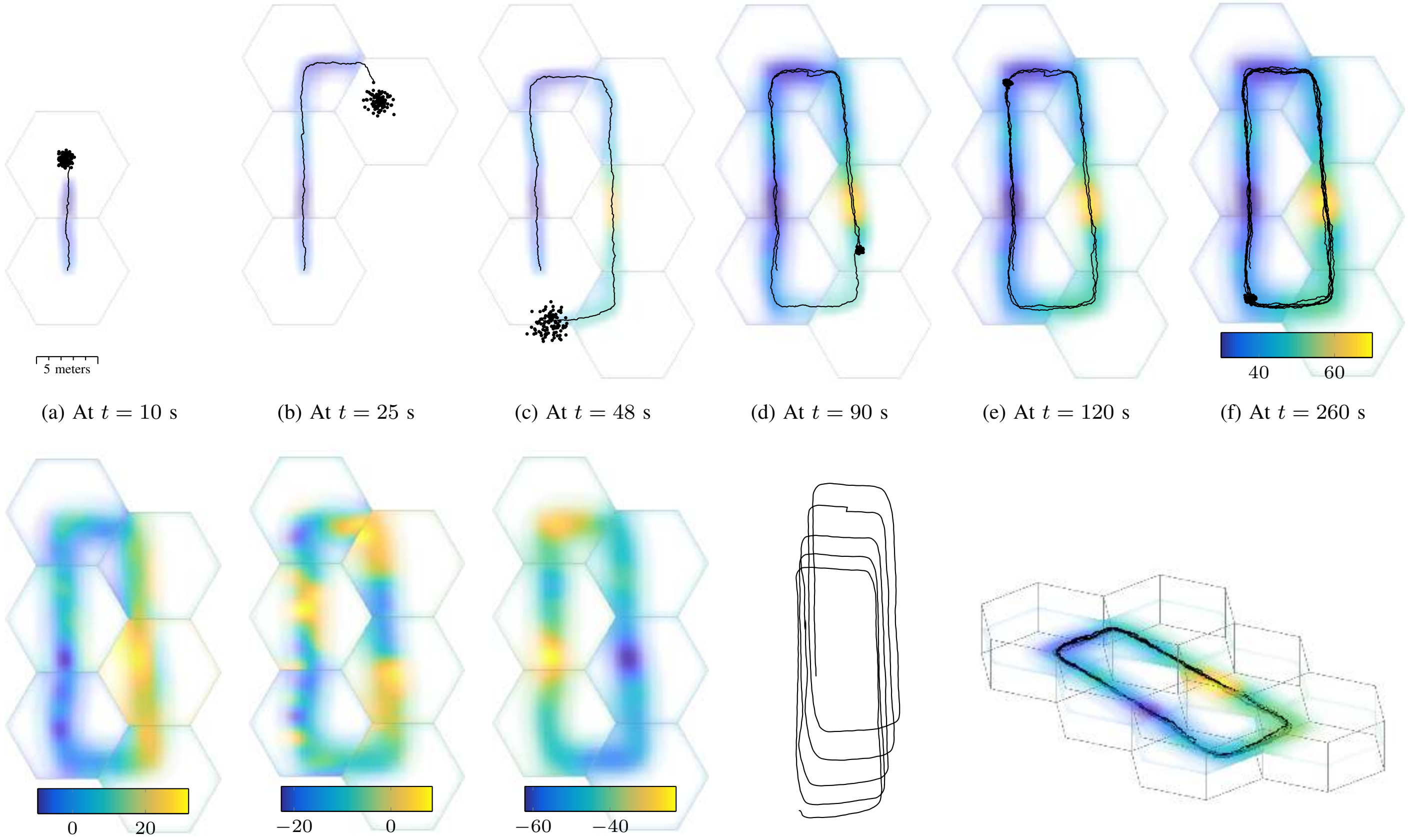
```
Input: ${\mathbf{y}_{\text{m},t}, \Delta\mathbf{p}_t^{\text{w}}, \Delta\mathbf{q}_t}_{t=1}^{N_T}$, $\mathbf{q}_0^{\text{wb}}$, $\Sigma_{\text{p}}$, $\Sigma_{\text{q}}$, $m$.
Output: For each $t$, $\mathbf{p}_t^{\text{w}}$, $\mathbf{q}_t^{\text{wb}}$, $\mathbf{m}_t$ of the highest-weight particle at $t$.
  Initialisation: Initialise the particles as $\mathbf{p}_0^{\text{w},(i)} = \mathbf{0}_3$, $\mathbf{q}_0^{\text{wb},(i)} = \mathbf{q}_0^{\text{wb}}$, set $\mathbf{m}_0^{(i)} = \mathbf{0}_{m+3}$, $\mathbf{P}_0^{(i)}$ as given in (12) and initialise the weights of the particles as $w_0^{(i)} = \frac{1}{N_P}$, for $i = 1, \dots N_P$.
  For $t = 1, \dots, N_T$,
  Create new hexagonal block tiles: If $\mathbf{p}_t^{\text{w},(i)}$ is in a new hexagonal block tile $d$, create this tile for particle $i$ and initialise $\mathbf{m}_{d,t}^{(i)} = \mathbf{0}_{m+3}$ and $\mathbf{P}_{d,t}^{(i)}$ as given in (12).
  Evaluate the importance weights: For $i = 1, \dots, N_P$, evaluate the importance weights $w_t^{(i)}$ based on the measurement model (3) and normalise.
  Resampling: If $90\%$ of the particles is in a tile that they are revisiting, resample those particles with replacement.
  Kalman filter measurement update: For $i = 1, \dots, N_P$, update the magnetic field map $\mathbf{m}_{d,t}^{(i)}$ and its covariance $\mathbf{P}_{d,t}^{(i)}$ of the hexagon $d$ that the particle is in using the nonlinear states $\mathbf{p}_t^{\text{w},(i)}$ and $\mathbf{R}_t^{\text{wb},(i)}$ according to (13). If the particle is close to the border of a tile, also update the map of the neighbouring tiles.
  Obtain a point estimate: Output the position, orientation, and estimated map of the particle with the highest weight.
  Particle filter time update: For $i = 1, \dots, N_P$, predict new particles $\mathbf{x}_{t+1}^{\text{n},(i)}$ as
      $$\mathbf{x}_{t+1}^{\text{n},(i)} = \begin{pmatrix} \mathbf{p}_t^{\text{w},(i)} + \Delta\mathbf{p}_t^{\text{w},(i)} + \boldsymbol{\varepsilon}_{\text{p},t}^{(i)} \\ \Delta\mathbf{q}_t \odot \mathbf{q}_t^{\text{wb},(i)} \odot \exp_{\text{q}}(\boldsymbol{\varepsilon}_{\text{q},t}^{(i)}) \end{pmatrix}, \tag{16}$$
  where $\boldsymbol{\varepsilon}_{\text{p},t}^{(i)}$ and $\boldsymbol{\varepsilon}_{\text{q},t}^{(i)}$ are drawn from $\mathcal{N}(\mathbf{0}, \Sigma_{\text{p}})$ and $\mathcal{N}(\mathbf{0}, \Sigma_{\text{q}})$, respectively.
  Set $t := t + 1$.
```

During the "exploration phase" when particles start building a map of previously unseen locations, very little information is available to distinguish between the particles. Because of this, we would like to ensure that the particle cloud keeps spreading during this phase. To this end, we adapt the RBPF implementation in two ways:

- To maintain the particle spread during the exploration phase, we delay the updating of the map by one lengthscale ℓ.
- To not collapse the particle cloud too quickly, we only resample the particles when they revisit a tile. Resampling is further delayed until at least 90% of the particles has arrived in a tile that they revisit.

IV. Results

In the empirical experiments, we use the built-in magnetometer and odometry data captured by a smartphone for simultaneously building the magnetic field map and localising on it.



(a) At $t = 10$ s (b) At $t = 25$ s (c) At $t = 48$ s (d) At $t = 90$ s (e) At $t = 120$ s (f) At $t = 260$ s

(g) Magnetic x (μT) (h) Magnetic y (μT) (i) Magnetic z (μT) (j) Odometry (k) Tilted 3D view

Fig. 3. The SLAM solution for walking in a square for several loops for around 250 meters. Subfigures (a)–(f) show the norm of the magnetic field map for the highest weight particle. The transparency of the maps scales with the marginal variance of the GP map. Subfigures (g)–(i) show the corresponding vector field components for the final map in (f). The odometry trajectory is visualised in (j), and (k) shows the 3D hexagonal tiles with a cross-section of the magnetic field maps.

We use an Apple iPhone 6s (published in September 2015), a previous flagship product of Apple which by current standards represents more of a standard smartphone.

The interest in this paper is purely on building of the map and on localisation, so we choose to leverage on the recent developments in visual-inertial pedestrian dead-reckoning (PDR) methods and use the built-in ARKit PDR provided by Apple. ARKit (released September 2017, used version running in iOS 11.2.5) is using the IMU and camera of the phone to provide a 6-DoF (position and orientation) PDR movement trajectory. ARKit does visual relocalisation under the hood, which aims to correct for long-term drift, resulting in discontinuities in the PDR track. These discontinuities turned out to be more harmful than useful for our approach, and we implemented a heuristic filtering approach for removing the relocalisation jumps. The PDR—albeit being surprisingly robust—suffers from long-term drift both in position and orientation (see Fig. 3).

The data was captured by a customised data capture application running on the phone. It collects the IMU (at 100 Hz) and three-axis magnetometer (100 Hz) data time-locked to the ARKit PDR estimates (60 Hz). Furthermore, the captured data was complemented with video frames of the back-facing phone camera (resolution 1280×720 at 60 fps), the barometric air pressure (10 Hz), and GNSS locations captured by the phone, none of which were used in the SLAM implementation, but captured for validation and reference. All data was stored on the device, downsampled to 10 Hz before use, and the SLAM implementation was run off-line in a Mathworks MATLAB implementation.

For simplicity and generalisability, we use the same model parameters in all our experiments. We choose the settings for the RBPF and the hexagonal blocks to be:

- We set the size of the hexagonal blocks to $r = 5$ m, $L_\mathrm{z} = 2$ m and the number of basis functions per tile to $m = 256$. The radius and height used in solving the eigenbasis of the Laplacian extends one meter outside the hexagon.
- The number of particles in the RBPF is $N_P = 100$.
- Neighbouring tiles are updated if samples are closer than 10 cm to the hexagonal block boundary.
- The process noise parameters are set to $\Sigma_\mathrm{p} = \mathrm{diag}(0.1^2, 0.1^2, 0.02^2)$ (values specified in meters drift per second) and $\Sigma_\mathrm{q} = \mathrm{diag}(0.01^2, 0.01^2, 0.24^2)$ (values specified in degrees drift per second).

In terms of memory usage, these choices mean that for each initialised map tile, the particle state dimension grows by 256. For fast look-up, our implementation uses a hashmap (per particle) for storing and keeping track of the map tiles. Each tile stores the GP mean and covariance matrix.

We use the same Gaussian process hyperparameter values for each of the experiments. These hyperparameters are typically learned from data. However, in sequential algorithms, hyperparameter values need to be available already before any data is processed. We therefore choose values that seem physically reasonable and that fit the data in the different scenarios that we consider reasonably well. We set the Gaussian process hyperparameters defined in Section III-A to $\sigma^2_{\text{lin.}} = 650$, $\sigma^2_{\text{SE}} = 200$, $\ell = 1.3$, and $\sigma^2_{\text{n}} = 10$ (the length-scale units are meters, the magnitude parameters in μT). This assumes that the lengthscale of the magnetic field anomalies is in the order of 1.3 m. The measurement noise standard deviation σ_{n} is rather high to also account for model discrepancies.

A. Illustrating magnetic field SLAM in 2D

In a first experiment, a data set is collected in the Design & Project Office (DPO) at the Engineering Department of the University of Cambridge. Significant excitation in the magnetic field is present for instance due the presence of radiators and a large number of computers. The trajectory is around 250 meters long and covers an area of around 20 × 5 m. The same circular trajectory is traversed several times. Because of this, it is easy to visually inspect the quality of the estimates from our SLAM algorithm which are shown in Fig. 3. The different subfigures illustrate the workings of our algorithm. As a supplementary file to this paper, there is also a video[1] demonstrating the workings of the SLAM algorithm.

Figs. 3(a) – 3(f) show how the estimates progress over time. For visualisation purposes, the norm of the magnetic field is predicted by the map of the highest weight particle at a large number of locations in each hexagon. The transparency in the map visualises the uncertainty of the map. It can be seen that the uncertainty is large in unexplored regions (no map is actually visible) and that the uncertainty of the map decreases over time in the re-visited regions.

The black line displays the trajectory of the highest weight particle. In Figs. 3(a) and (b), the deliberate slight delay in updating the map can clearly be seen. As discussed in Section III-B, we wait with resampling until at least 90% of the particles is in a tile that they have previously updated and left. Because of this, no resampling is done in the first loop, until around 50 seconds and the spread in the particles in Fig. 3(c) is therefore fairly large. This allows the algorithm to properly close the loop as visualised in Fig. 3(d). Another aspect to note about our algorithm is that in Fig. 3(c), most particles are on the border between the tile in which the trajectory was started and a new tile below. To allow for smooth transitions between these two tiles, many of the particles will in this case update the local magnetic field maps in both tiles.

The map that is visualised in Figs. 3(a) – (f), shows the norm of the predicted magnetic field. Our model (4), however, explicitly models the magnetic field as a three-dimensional vector. Instead of visualising the norm of the predicted magnetic field, it is therefore also possible to visualise the predicted magnetic field in three directions. This is shown in Figs. 3(g) – (i). Because the dip angle of the Earth's magnetic field is fairly large in Cambridge, UK, the z-component of the magnetic field is of significantly larger magnitude than the x- and y-components.

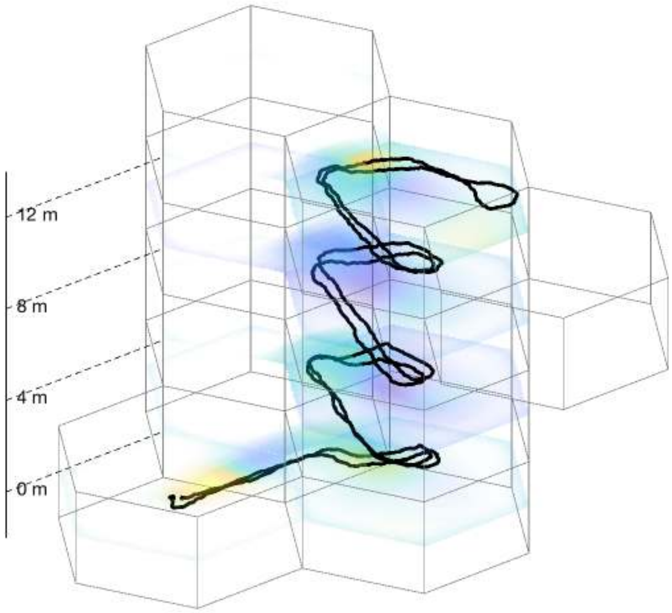


Fig. 4. Example showing the three-dimensional nature of our SLAM algorithm. The path covers a flight of stairs in the University of Cambridge Engineering Department. The complete path length is 125 meters. The cross-sections of the tiles show the magnetic map norm (the color scale is the same as in Fig. 3(f)).

Based on Fig. 3(f) it can be concluded that our SLAM algorithm estimates the trajectory with high accuracy since the same trajectory can be seen to be traversed multiple times. For comparison, we also show the odometry in Fig. 3(j) which clearly shows a drift in the position. Our magnetic SLAM approach is able to correct for this drift. To illustrate the fact that even though the movement for this data set is in a 2D plane the estimation is still done in 3D, the tilted 3D view is shown in Fig. 3(k).

B. Fully 3D magnetic field SLAM

In Section IV-A, the movement was close to two-dimensional. However, exactly the same procedure also applies to movement and block tile structures in three dimensions. In Fig. 4 we show the estimated trajectory and the magnetic field map for an empirical experiment in which we first walked up and subsequently down a set of stairs at the Engineering Department of the University of Cambridge.

Again, each hexagonal block has its own local magnetic field map. To visualise these, the norm of the predicted magnetic field for a hexagonal slice in the middle of each block is shown. The transparency visualises the uncertainty of

[1] The supplementary video is available on YouTube: https://youtu.be/pbwWLoh6mvI

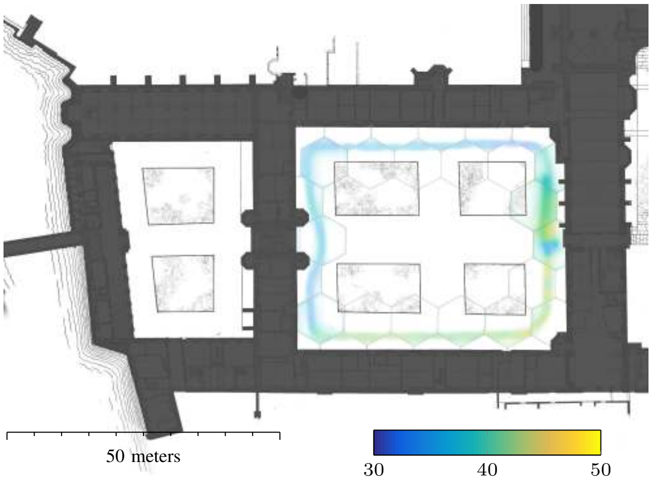


Fig. 5. The SLAM map solution for a large-scale outdoor–indoor example. The visualised color field is the norm of the magnetic field, and the opacity follows the uncertainty. The layout corresponds to that of Fig. 1.

the map. Just as in the results presented in Section IV-A, our algorithm only starts to resample particles on the way down the stairs. This results in an estimated path that is fairly similar in both directions.

C. Large-scale magnetic field SLAM

In a third empirical data set we walk around St. John's college in Cambridge. The college consists of several courts and even though the data set is captured outdoors, the close proximity of the buildings results in small magnetic field anomalies that can be used for localisation. The SLAM results were previously presented in Fig. 1 and are presented in more detail in Fig. 5. The map extends over 27 tiles covering the courtyard. For many algorithms the extent of the data set would therefore be prohibitively large. However, due to our computationally efficient model in terms of hexagonal tiling, our approximate Gaussian process model and the RBPF algorithm, the computations remain feasible. It can therefore be concluded from Fig. 5 that our approach scales to large-scale magnetic field SLAM.

V. Conclusions and future work

We have presented a scalable algorithm for magnetic field SLAM in 3D. It builds local Gaussian process maps where each map is an approximation to the local full GP solution in terms of a basis function expansion. This approximation allows us to map a magnetic map volume of $\sim$ 260 m^3 in a state dimension of only 256. Each local map is modelled in terms of a hexagonal block tile, which minimises the number of tiles that are needed to cover the total mapping volume. The formulation fits perfectly in an RBPF resulting in a computationally tractable algorithm. The computational complexity scales linearly both with the number of particles in the particle filter and with the number of hexagonal tiles of each particle. The algorithm is shown to perform well on three challenging empirical data sets.

One of the challenges in the data we used for testing, is that the odometry we obtain from the ARKit does not always obey our motion model. More specifically, for straight paths the odometry shows almost no drift, while fast turns can introduce sudden errors. This does not follow our motion model in which we model a constant process noise. In future work we would like to focus on using inertial PDR as odometry instead. Apart from avoiding these issues, this would also result in a positioning algorithm only based on widely available inertial and magnetometer sensors. Other directions of future work are in the direction of using a Rao–Blackwellised Particle Smoother and further reducing the computational complexity of the algorithm.

Acknowledgments

This research was financially supported by the EPSRC grant *Autonomous behaviour and learning in an uncertain world* (Grant number: EP/J012300/1) and the Academy of Finland grant *Sequential inference for real-time probabilistic modelling* (Grant number: 308640).

References

[1] 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.

[2] 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)*, 2012, pp. 1–9.

[3] M. Angermann, M. Frassl, M. Doniec, B. J. Julian, and P. Robertson, "Characterization of the indoor magnetic field for applications in localization and mapping," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012, pp. 1–9.

[4] E. Le Grand and S. Thrun, "3-axis magnetic field mapping and fusion for indoor localization," in *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2012, pp. 358–364.

[5] A. Solin, S. Särkkä, J. Kannala, and E. Rahtu, "Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning," in *Proceedings of the European Navigation Conference*, 2016.

[6] D. Hanley, A. B. Faustino, S. D. Zelman, D. A. Degenhardt, and T. Bretl, "MagPIE: A dataset for indoor positioning with magnetic anomalies," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017, pp. 1–8.

[7] C. Gao and R. Harle, "MSGD: Scalable back-end for indoor magnetic field-based GraphSLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3855–3862.

[8] J. Torres-Sospedra, D. Rambla, R. Montoliu, O. Belmonte, and J. Huerta, "UJIIndoorLoc-Mag: A new database for magnetic field-based localization problems," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–10.

[9] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä, "Modeling and interpolation of the ambient magnetic field by Gaussian processes," *IEEE Transactions on Robotics (accepted for publication)*, 2018, arXiv:1509.04634.

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

[11] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," *ArXiv preprint arXiv:1401.5508*, 2014.

[12] T. B. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, 2005.

[13] P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. J. Julian, M. Garcia Puyol, M. Khider, M. Lichtenstern, and L. Bruno, "Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2013, pp. 1–10.

[14] M. Frassl, M. Angermann, M. Lichtenstern, P. Robertson, B. J. Julian, and M. Doniec, "Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 913–920.

[15] 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)*, 2010, pp. 14–19.

[16] ——, "Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task," in *Proceedings of the 15th International Conference on Advanced Robotics (ICAR)*, 2011, pp. 198–203.

[17] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Foundations and Trends on Signal Processing*, vol. 11, no. 1–2, pp. 1–153, 2017.

[18] D. J. Griffiths and R. College, *Introduction to Electrodynamics*. Upper Saddle River, NJ: Prentice hall, 1999.

[19] D. K. Duvenaud, "Automatic model construction with Gaussian processes," Ph.D. dissertation, University of Cambridge, 2014.

[20] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 2012.

[21] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.