Article

The International Journal of
Robotics Research
2016, Vol. 35(14) 1717–1730

DOI: 10.1177/0278364916684382
ijr.sagepub.com

Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent

Fabio Ramos and Lionel Ott

Abstract

The vast amount of data robots can capture today motivates the development of fast and scalable statistical tools to model the space the robot operates in. We devise a new technique for environment representation through continuous occupancy mapping that improves on the popular occupancy grip maps in two fundamental aspects: (1) it does not assume an a priori *discrimination of the world into grid cells and therefore can provide maps at an arbitrary resolution; (2) it captures spatial relationships between measurements naturally, thus being more robust to outliers and possessing better generalization performance. The technique, named Hilbert maps, is based on the computation of fast kernel approximations that project the data in a Hilbert space where a logistic regression classifier is learnt. We show that this approach allows for efficient stochastic gradient optimization where each measurement is only processed once during learning in an online manner. We present results with three types of approximations: random Fourier; Nyström; and a novel sparse projection. We also extend the approach to accept probability distributions as inputs, for example, due to uncertainty over the position of laser scans due to sensor or localization errors. In this extended version, experiments were conducted in two dimensions and three dimensions, using popular benchmark datasets. Furthermore, an analysis of the adaptive capabilities of the technique to handle large changes in the data, such as trajectory update before and after loop closure during simultaneous localization and mapping, is also included.*

Keywords

Cognitive robotics, learning and adaptive systems, mapping, mobile and distributed robotics SLAM, range sensing, sensing and perception computer vision

1. Introduction

Representing the physical properties of three-dimensional (3D) space is central to robotics, from manipulation and grasping to autonomous navigation. Amongst the many physical properties characterizing the environment the likelihood that a particular point is occupied by a solid object which the robot needs to interact with is certainly one of the most important. Traditional techniques to create a map of occupancy rely on the discrimination of an area into regular sized cells to form a fixed grid on which a sensor model or likelihood function is applied to estimate the posterior of occupancy given some sensory data, for example, laser scans or sonars (Elfes, 1987, 1989). One of the main limitations of such techniques is the assumption that each cell in the grid is independent of each other and the posterior computation for the entire map is performed separately for each cell. This assumption disregards important spatial relationships between cells and leads to maps with a series of "gaps" between cells. For example, cells with no observations have a 0.5 likelihood of being occupied even though they are next to cells with high likelihood of being occupied. The problem becomes more severe in 3D maps where the number of cells necessary to represent the environment with the same resolution grows exponentially as does the number of required observations. In indoor environments when the area to be mapped is relatively small and the density of observations is large, occupancy grids are generally sufficient to provide a representation that is both fast and compact. However, representing large 3D outdoor regions with sparse observations still remains a challenge.

In an attempt to resolve some of these issues, the Gaussian processes occupancy map (GPOM) (O'Callaghan and Ramos, 2012; O'Callaghan et al., 2009) was proposed. The idea is to use a Gaussian processes prior over the space of functions mapping locations to the occupancy class. The

School of Information Technologies, The University of Sydney, Australia

Corresponding author:
Fabio Ramos, University of Sydney, SIT Building, J12 Sydney, New South Wales 2006, Australia.
Email: f.ramos@acfr.usyd.edu.au

method is continuous, i.e. it does not require a prior discrimination of the space, and nonparametric; the complexity of the representation grows with the number of data points. The final Gaussian process classifier model possesses many of the advantages we would like to have in a spatial representation as it directly captures spatial relationships through a parameterized covariance function and produces principled probabilistic posteriors naturally encoding the uncertainty of the process. The main drawback is the computational complexity that without approximations or division of the data into smaller sets scales cubically with the size of the data.

We propose a simpler and faster approach to continuous occupancy mapping in this paper. By utilizing recent advancements in optimization (Zhang, 2004) and efficient kernel approximations, we represent the occupancy property of the world with a linear discriminative model operating on a high-dimensional feature vector that projects observations into a reproducing kernel Hilbert space (RKHS) (Scholkopf and Smola, 2001). The objective function for training the model is convex in the parameters and therefore the global optimum can be found. Furthermore, the model can be trained and updated using *stochastic gradient descent* (SGD) making the computation theoretically independent of the number of observations. The key to our approach is to quickly generate a large number of features whose dot product approximates the well-known *radial basis function kernel* (RBF) (Scholkopf and Smola, 2001). The RBF kernel can be seen as a feature mapping into a infinite dimensional space that can asymptotically represent the complexity of the physical world. We present three solutions to approximate the kernel. (1) The first is based on the recently proposed *Random Kitchen Sinks* (RKSs) by (Rahimi and Recht, 2008, 2009). (2) The second is based on the Nyström approximation which is very popular in kernel machines (Williams and Seeger, 2000b). (3) Finally we introduce a novel feature mapping that generates sparse features, better capturing local information. We also show how to generalize these features to accept probability distributions as inputs which are more robust to handle localization errors or sensor noise. As opposed to GPOM, our method can be updated in linear time and scales well with large amounts of data. It can be used to address several tasks in robotics, for example, grasping and path planning. The model provides a function that represents the occupancy property of the environment, thus enabling the computation of gradients of surfaces for grasping. Additionally, it can be seamlessly integrated into optimization procedures for path planning, minimizing the likelihood of collisions.

The technical contributions of the paper[1] are:

1. Hilbert maps—a novel continuous occupancy map technique scalable to large datasets and updated in linear time;
2. a novel sparse Hilbert space feature that better preserves local information and leads to faster SGD iterations;
3. a generalization of the method to receive probability distributions as inputs to accommodate, in a principled manner, the uncertainty in the position of the measurements.

The paper is organized as follows. We first introduce the method, the features and the extensions to probabilistic inputs in Section 2. The objective function and optimization for online learning through SGD is introduced in Section 3. We discuss relationships between Hilbert maps, GPOMs and recent results in machine learning related to important aspects of this technique in Section 4. Experiments on benchmark datasets and comparisons are presented in Section 5, and conclusions with ideas for future work are in Section 6.

2. Hilbert maps

We begin the presentation of the method by first introducing notation. We assume a robot captures a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $\mathbf{x_i} \in \mathbb{R}^D$ is a point in two-dimensional (2D) or three-dimensional (3D) space and $y_i \in \{-1, +1\}$ is a categorical variable corresponding to the occupancy property of $\mathbf{x}_i$. The dataset is obtained while the robot moves in the environment with a range sensor such as a laser scanner. Randomly selected points in the line segment of a laser beam between the sensor and an object are labeled as unoccupied. The final point in the beam generating a return is labeled as occupied. The length of the beam determines the number of unoccupied points. Due to the amount of beams in laser scans, one point for every one to two meters of beam length is typically sufficient. Selecting the position along the beam at random creates a more uniformly distributed dataset over the free space compared to fixed distance interval sampling. We also assume that the dataset is incrementally built as the robot collects more data as it moves in the environment.

Given the dataset, our objective is to incrementally learn a discriminative model $p(y|\mathbf{x}, \mathbf{w})$ parameterized by a vector $\mathbf{w}$ to predict the occupancy property for new query points $\mathbf{x}_*$. In this work we adopt a very simple *logistic regression classifier* that is simple and fast to learn while being directly amenable to online learning through SGD. The probability of nonoccupancy for a point $\mathbf{x}_*$ can be easily computed as

$$p(y_* = -1|\mathbf{x}_*, \mathbf{w}) = \frac{1}{1 + \exp(\mathbf{w}^{\mathrm{T}}\mathbf{x}_*)} \tag{1}$$

while $p(y_* = +1|\mathbf{x}_*, \mathbf{w}) = 1 - p(y_* = -1|\mathbf{x}_*, \mathbf{w})$ is the probability of occupancy.

The model can be seen as the sigmoid *logit* function applied to a linear projection of the input $\mathbf{x}$. However, how can this simple linear model be able to represent the complexity of the physical world? The key to this problem is to apply the discriminative model not directly to the inputs $\mathbf{x}$ but to a large number of features computed from $\mathbf{x}$, denoted as $\hat{\Phi}(\mathbf{x})$.[2] As we shall see next, the dot product of these features can approximate popular kernels commonly used in

kernel machines for nonlinear classification. These kernels define a Hilbert space and can represent a nonlinear mapping of the inputs to a space of potentially infinite dimension (for example in the case of the RBF kernel (MacKay, 1998)) with sufficient complexity to represent the environment. Note, however, that the advantage of using the feature approximation to the kernel rather than the kernel itself is that we can learn the model using fast primal procedures rather than expensive quadratic programming approaches as demonstrated in Singer and Srebro (2007) for the case of support vector machines (SVMs).

In the next three sections, we show approaches to generate features $\hat{\Phi}(\cdot)$ that efficiently approximate particular kernels: $k(\mathbf{x}, \mathbf{x}') \approx \hat{\Phi}(\mathbf{x})^{\mathrm{T}} \hat{\Phi}(\mathbf{x}')$.

2.1. Random Fourier features

This kernel approximation method is based on the work of Rahimi and Recht (2008) with approximation bounds presented in Rahimi and Recht (2009) for general learning problems. Formally, a kernel $k(\mathbf{x}, \mathbf{x}')$ defines a Hilbert space with inner product $\langle \cdot, \cdot \rangle$ from a feature vector $\Phi(\mathbf{x})$ such that

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^D : k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle. \tag{2}$$

If a kernel is shift invariant (also called stationary) it can be written as $k(\tau)$ where $\tau = \mathbf{x} - \mathbf{x}'$ and Bochner's Theorem (Gihman and Skorohod, 1974) can be applied to create a representation in terms of its Fourier transform.

Theorem 1. *(Bochner's Theorem) Any shift invariant kernel $k(\tau)$, $\tau \in \mathbb{R}^D$, with a positive finite measure $d\mu(\mathbf{s})$ can be represented in terms of its Fourier transform as*

$$k(\tau) = \int_{\mathbb{R}^D} e^{-i\mathbf{s}\cdot\tau} d\mu(\mathbf{s}). \tag{3}$$

The proof can be found in Gihman and Skorohod (1974). If μ has a density $S(\mathbf{s})$, the measure $d\mu(\mathbf{s})$ can be represented as $S(\mathbf{s})\, d\mathbf{s} = d\mu(\mathbf{s})$ and $S(\mathbf{s})$ is called the *spectral density* of k. We can then write

$$k(\tau) = \int_{\mathbb{R}^D} e^{-i\mathbf{s}\cdot\tau} S(\mathbf{s})\, d\mathbf{s} = E_{S(\mathbf{s})}\left[e^{-i\mathbf{s}\cdot\tau}\right]$$

where $E_{S(\mathbf{s})}[\cdot]$ denotes the expectation w.r.t. the density $S(\mathbf{s})$. The expected value can thus be approximated as

$$k(\tau) \approx \frac{1}{n} \sum_{k=1}^{n} e^{-i\mathbf{s}_{\mathbf{k}}\cdot\tau} = \langle \hat{\Phi}(\mathbf{x}), \hat{\Phi}(\mathbf{x}') \rangle \tag{4}$$

where $\mathbf{s}_1, \ldots, \mathbf{s}_n$ are samples from $S(\mathbf{s})$ and

$$\hat{\Phi}(\mathbf{x}) = \frac{1}{\sqrt{n}} \left[e^{-i\mathbf{s}_1\cdot\mathbf{x}}, \ldots, e^{-i\mathbf{s}_n\cdot\mathbf{x}}\right] \tag{5}$$

is the Fourier feature map approximating $k(\mathbf{x}, \mathbf{x}')$. In the case of the RBF kernel defined as

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}'\|_2^2\right), \tag{6}$$

where $\|\cdot\|$ is the Euclidean distance, the approximation is obtained in two steps: (1) we generate n samples from $S(\mathbf{s}) \sim \mathcal{N}(0, 2\sigma^{-2} I)$ and $b \sim \text{uniform}[-\pi, \pi]$; (2) for each sample i compute the feature approximation as $\cos(\mathbf{s}_i\mathbf{x} + b_i)$. The approximation is thus given by

$$\hat{\Phi}^{\text{Random}}(\mathbf{x}) = \frac{1}{\sqrt{n}} \left[\cos(\mathbf{s}_1\mathbf{x} + b_1), \ldots, \cos(\mathbf{s}_n\mathbf{x} + b_n)\right]. \tag{7}$$

In equation (7) we used the relation $e^{-i\mathbf{s}\cdot\mathbf{x}} = \cos(\mathbf{s}\cdot\mathbf{x}) - i\sin(\mathbf{s}\cdot\mathbf{x})$ and noted that the imaginary part must be zero for real kernels. Also note that $S(\mathbf{s})$ and $k(\tau)$ are duals and thus $S(\mathbf{s})$ can be obtained by calculating the inverse Fourier transform of $k(\tau)$. b is symmetric about 0 and is introduced to rotate the projection into the real axis by a random amount. This is known to produce better results in several practical problems (Rahimi and Recht, 2008).

2.2. Nyström features

The Nyström method (Williams and Seeger, 2000b) approximates a kernel matrix K by projecting it into a set of m inducing points, denoted by $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_m$. Then, $K \approx K_b \hat{K}^{\dagger} K_b^{\mathrm{T}}$, where $K_b = [k(\mathbf{x}, \hat{\mathbf{x}})]_{N\times m}$ is a kernel matrix computed between all points in the dataset and the inducing points, $\hat{K} = [k(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)]_{m\times m}$ is a kernel matrix between the inducing points and $\hat{K}^{\dagger}$ is the pseudo inverse of $\hat{K}$. Factorizing the approximation into a feature vector yields

$$\hat{\Phi}^{\text{Nyström}}(\mathbf{x}) = \hat{D}^{-1/2} \hat{V}^{\mathrm{T}} \left(k(\mathbf{x}, \hat{\mathbf{x}}_1), \ldots, k(\mathbf{x}, \hat{\mathbf{x}}_m)\right)^{\mathrm{T}} \tag{8}$$

where $\hat{D} = \text{diag}(\lambda_1, \ldots, \lambda_r)$ are the r nonnegative eigenvalues of $\hat{K}$ in decreasing order and $\hat{V} = (\mathbf{v}, \ldots, \mathbf{v}_r)$ are the corresponding eigenvectors. It can be shown (Williams and Seeger, 2000a) that the Nyström approximation minimizes the functional

$$\mathcal{E}(\hat{\Phi}) = \int \left(k(\mathbf{x}_i, \mathbf{x}_j) - \langle \hat{\Phi}(\mathbf{x}_i), \hat{\Phi}(\mathbf{x}_j) \rangle\right)^2 p(\mathbf{x}_i) p(\mathbf{x}_j)\, d\mathbf{x}_i \mathbf{x}_j \tag{9}$$

where $p(\mathbf{x}_i)$ and $p(\mathbf{x}_j)$ are approximated by a set of r samples from the data. Therefore, the Nyström approximation is *nested*; i.e. it depends on the particular dataset being used. This is in contrast to *random Fourier features* which are dataset independent and can be computed *a priori*, once a specific kernel is defined.

2.3. Sparse random features

The two approaches above can be used to approximate a RBF kernel but do not produce sparse features. With the goal to produce a sparse set of features that can be more easily optimized with SGD, we explore the properties of the sparse kernel introduced in Melkumyan and Ramos (2009).

The sparse kernel is defined as

$$k_{\text{sparse}}\left(\mathbf{x}, \mathbf{x}'\right) = \begin{cases} \left[\frac{2+\cos(2\pi r)}{3}(1-r) + \frac{1}{2\pi}\sin(2\pi r)\right] & \text{if } r < 1 \\ 0 & \text{if } r \geq 1 \end{cases} \tag{10}$$

where the matrix Ω is positive semi-definite and

$$r = \sqrt{(\mathbf{x} - \mathbf{x}')^{\mathrm{T}}\,\Omega\,(\mathbf{x} - \mathbf{x}')}, \quad \Omega \geq 0. \tag{11}$$

This kernel has an important property that for distances $r \geq 1$ it returns 0. It also approximates the smoothness of a RBF kernel being four times differentiable. With this result, we define the sparse feature as

$$\hat{\Phi}^{\text{Sparse}}(\mathbf{x}) = \left(k_{\text{sparse}}(\mathbf{x}, \hat{\mathbf{x}}_1), \ldots, k_{\text{sparse}}(\mathbf{x}, \hat{\mathbf{x}}_m)\right)^{\mathrm{T}} \tag{12}$$

where, as with the Nyström feature, $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_m$ is a set of inducing points where the kernel is centered on. These inducing points can be uniformly sampled in the area the robot explores or can be placed in a grid.

2.4. Feature approximation of kernels on distributions

In realistic mapping tasks there is typically uncertainty associated with the robot's position and imperfect sensor measurements. This uncertainty needs to be taken into account to accurately reflect the likelihood of occupancy of a given point in the map. In simultaneous localization and mapping problems, the sequence in which observations are made and the uncertainty in data association and motion estimates all play a significant role in determining the absolute position of laser returns. We show how these uncertainties can be incorporated in Hilbert maps by deriving feature approximations to kernels over distributions.

Recent work in kernel embeddings has shown how to map probability distributions to a RKHS (Smola et al., 2007; Song et al., 2013). We follow this idea to derive our approximations. First, we assume that each point $\mathbf{x}$ is distributed as $\mathbb{P}$ in a probability space $\mathcal{P}$ in $(\mathcal{X}, \mathcal{A})$, where $\mathcal{X}$ is the input space and $\mathcal{A}$ is an associated σ-algebra. Let $\mathcal{H}$ denote a RKHS of functions $f : \mathcal{X} \to \mathbb{R}$ with a reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The mean map μ from $\mathcal{P}$ into $\mathcal{H}$ can be obtained as

$$\mu : \mathcal{P} \to \mathcal{H}, \quad \mathbb{P} \mapsto \int_{\mathcal{X}} k(x, \cdot)\, d\mathbb{P}(\mathbf{x}) \tag{13}$$

We can produce an empirical estimate of μ by drawing independent samples from $\mathbb{P}$ and creating a set $W = \{\mathbf{x}^{(1)} \ldots, \mathbf{x}^{(n)}\}$ such that

$$\hat{\mu}(\hat{\mathbb{P}}) = \frac{1}{n}\sum_{i=1}^{n} k(\mathbf{x}^{(i)}, \cdot)\,. \tag{14}$$

This mean map estimator has been shown to converge to the mean map at a rate of $\mathcal{O}(n^{-\frac{1}{2}})$ in Smola et al. (2007).

With the mean map estimator $\hat{\mu}$, a general positive semi-definite kernel $k(\mathbb{P}_i, \mathbb{P}_j)$ on distributions $\mathbb{P}_i$ and $\mathbb{P}_j$ can be approximated as follows

$$k(\mathbb{P}_i, \mathbb{P}_j) = \int\int \langle k(\mathbf{x}_i, \cdot), k(\mathbf{x}_j, \cdot)\rangle_{\mathcal{H}}\, d\mathbb{P}_i(\mathbf{x}_i)\, d\mathbb{P}_j(\mathbf{x}_j) \tag{15}$$

$$= \int\int k(\mathbf{x}_i, \mathbf{x}_j)\, d\mathbb{P}_i(\mathbf{x}_i)\, d\mathbb{P}_j(\mathbf{x}_j) \tag{16}$$

$$\approx \frac{1}{n}\frac{1}{m}\sum_{k=1}^{n}\sum_{l=1}^{m} k(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(l)}). \tag{17}$$

In the above we used the reproducing property of $\mathcal{H}$ and the fact that $k(\mathbb{P}_i, \mathbb{P}_j) = \langle \mu_{\mathbb{P}_i}, \mu_{\mathbb{P}_j}\rangle_{\mathcal{H}}$. Finally, using the random Fourier, the Nyström or the sparse feature approximations as detailed above, the feature mapping approximation for a distribution $\mathbb{P}$ in $\mathcal{H}$ is

$$\hat{\Phi}(\mathbb{P}) = \frac{1}{n}\sum_{i=1}^{n} \hat{\Phi}(\mathbf{x}^{(i)}) \tag{18}$$

where $\mathbf{x}^{(i)}$ are samples in W from $\mathbb{P}$.

3. Online learning

The logistic regression model described in Section 2 can be learnt as part of an online optimization procedure. However, in contrast to conventional logistic regression, the model operates on features $\hat{\Phi}(\mathbf{x})$ creating a nonlinear decision boundary.

3.1. Objective function

To estimate the parameters $\mathbf{w}$ we minimize the regularized negative log-likelihood (*NLL*) given by

$$NLL(\mathbf{w}) = \sum_{i=1}^{N} -\log p(y_i|\hat{\Phi}(\mathbf{x}_i), \mathbf{w}) + R(\mathbf{w}) \tag{19}$$

$$= \sum_{i=1}^{N} \log\left(1 + \exp(-y_i \mathbf{w}^{\mathrm{T}} \cdot \hat{\Phi}(\mathbf{x}_i))\right) + R(\mathbf{w}) \tag{20}$$

where $R(\mathbf{w})$ is a regularizer to prevent overfitting and to enforce sparseness in $\mathbf{w}$. In this work we use the elastic net regularizer that has been shown to produce better results than L_1 (LASSO) while preserving the same level of sparsity (Zou and Hastie, 2005). The elastic net regularizer is defined as

$$R(\mathbf{w}) = \lambda_1 \|\mathbf{w}\|_2^2 + \lambda_2 \|\mathbf{w}\|_1 \tag{21}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ are the L_2 and L_1 norms respectively, and λ_1 and λ_2 are parameters balancing the quadratic term (also called shrinkage parameter) and degree of sparseness, respectively.

The gradient of the objective function with respect to $\mathbf{w}$ can be computed as

$$\nabla NLL(\mathbf{w}) = \sum_{i=1}^{N} -y_i \hat{\Phi}(\mathbf{x}_i)\,(1 + \exp(y_i \mathbf{w}^{\mathrm{T}} \cdot \hat{\Phi}(\mathbf{x}_i)))^{-1} + \frac{\partial R(\mathbf{w})}{\partial \mathbf{w}} \tag{22}$$

Note that the L_1 term in $R(\mathbf{w})$ is nondifferentiable so its derivative is generally approximated using sub-differentials.

3.2. Stochastic gradient descent

One of the main advantages of utilizing logistic regression is that the negative objective function in equation (20) can be optimized using fast SGD methods. This is because the negative log-likelihood is the sum of the negative log-likelihoods of individual points. In contrast to batch algorithms, such as Newton's method that require the computation of gradients and Hessians for all the points in the dataset, SGD operates iteratively, taking a small step towards the goal with each data point.

To minimize equation (20), SGD iterates between randomly selecting a training point $\{\mathbf{x}_t, y_t\}$ from $\mathcal{D}$ and updating the parameters $\mathbf{w}$ as

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t A_t^{-1} \frac{\partial}{\partial \mathbf{w}} NLL(\mathbf{w}) \tag{23}$$

where $\eta > 0$ is known as the learning rate and matrix A can be seen as a preconditioner to accelerate the convergence rate. In many cases, A can be set to the identity matrix. This method is intrinsically online as new data points arriving from sensor measurements can be selected to update the parameters $\mathbf{w}$. Additionally, convergence analysis and generalization behavior have been extensively studied (Zhang, 2004; Bottou and Bousquet, 2008). It has been shown that even if SGD is applied to an unregularized version of equation (20), it achieves an implicit regularization effect with good generalization performance (Bottou, 2010). This facilitates the manual setting of the regularization parameter as we know that even if we set it to zero, the model will still retain some resilience to overfitting.

The learning rate η is either constant or asymptotically decaying with the number of iterations. To guarantee convergence the learning rate is required to satisfy the conditions $\sum_t \eta_t^2 < \infty$ and $\sum_t \eta_t = \infty$. In our implementation, we use the procedure proposed in Bottou (2012) and set it to $\eta_t = \frac{1}{\alpha(t_0+t)}$, where t_0 is determined empirically from a small training set sampled from the full dataset, and α is the regularization strength. This expression ensures a good compromise between the beginning

learning

rates, and the ideal asymptotically decrease dominated by $(\alpha t)^{-1}$.

Equation (23) is effectively an online update procedure of the parameters. If the dataset grows, we can effectively select new points and update the parameters directly to reflect the new information. Conversely, we can shuffle the data, pass through each data point once and repeat the process. This is known as the batch version of SGD (Zhang, 2004). Finally, we can average the parameters in the last T iterations to remove some of the oscillation commonly seen during the optimization. This is known as the averaged SGD (Bottou, 2012; Xu, 2011) and has been shown to improve on the convergence of conventional SGD for learning rates satisfying the convergence requirements above (Xu, 2011). Note that SGD has a constant cost per iteration and asymptotically converges to the expected risk (Bottou, 2012).

4. Relationship to other methods

SVMs: There are several classifiers in the machine learning literature that resemble the method described so far, each with pros and cons. Notably, Pegasos (Singer and Srebro, 2007) is a different SVM formulation where the expensive quadratic programming optimization is replaced by SGD applied to the primal problem. The authors show that this method scales much better with the number of training points with strong convergence properties. Pegasos was not, however, trained on kernel feature approximations as our method was and used a different loss function. The main reason we did not use a max-margin loss or hinge loss as with SVMs and chose the logistic regression formulation relates to the probabilistic interpretation of the results naturally obtained with logistic regression. An example can be seen in Figure 1 where we show the same continuous occupancy map produced by the two methods. As SVMs do not produce a probabilistic interpretation directly, artificial probabilistic outputs are generally obtained using the method in Platt (1999). However, as the figure shows, this introduces a problem: areas not explored, with no sensory information, are classified as either occupied or nonoccupied with high confidence. Conversely, with the logistic regression formulation, unexplored areas are classified with probability of 50% of being occupied as expected.

GPOM: This work also borrows ideas from GPOMs (O'Callaghan and Ramos, 2012; O'Callaghan et al., 2009) in that both attempt to represent the space in a continuous manner. Both produce probabilistic interpretations of occupancy, and both utilize kernels to represent data points in a high-dimensional space. However, there are also significant differences. GPOMs are nonparametric; i.e. the complexity of the model grows with the number of data points. Without approximations, GPOMs have a $\mathcal{O}(n^3)$ cost, where n is the number of data points, due to the need to invert a potentially large Gram matrix. GPOM assumes a Gaussian process prior over the function space mapping inputs to outputs. To obtain classification outputs (posterior), a sigmoid-type likelihood is combined with the prior, resulting in an expression with intractable integrals that also need to be approximated. GPOM resolves the

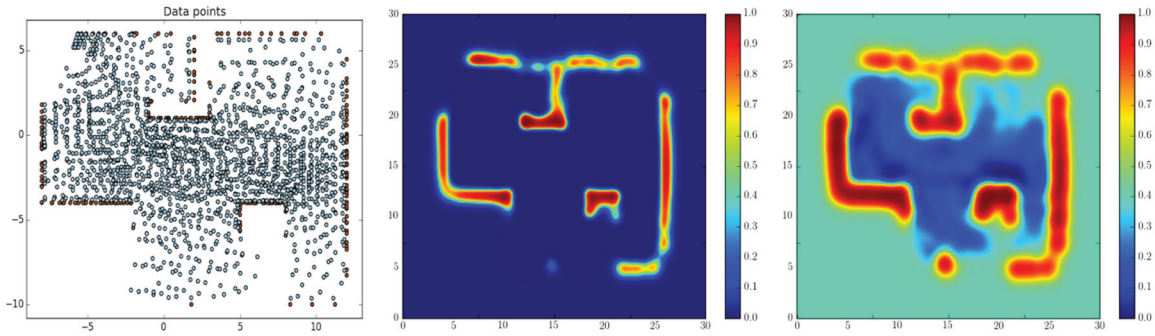

Fig. 1. Comparison between maps produced with logistic regression and SVMs. (a) Data points used for training, blue are non-occupied and brown are occupied. (b) Map produced with SVMs and Nyström features, with color indicating probability of occupancy. (c) The equivalent map generated with logistic regression. Note that the SVM map is over confident about the occupancy status in areas with no data points, assumed unoccupied. This does not occur with the logistic regression map. Axis units are in meters.

likelihood-prior marginalization with a point estimate around the Gaussian process predictive mean but this can result in inaccurate probability estimates for the posterior. In GPOMs, as a Bayesian method, the (hyper) parameters of the model can be obtained automatically by maximizing the marginal likelihood. In contrast to GPOMs, Hilbert maps are built around a much simpler discriminative model, based on the logistic regression classifier. Most of the power of the method in providing accurate interpolations is due to the combination of features approximating a Hilbert space embedding with the speed of SGD. Hilbert maps are parametric models and the complexity of both learning and inference is independent of the number of data points but rather depend on the number of features m. In its current form a Hilbert map is a frequentist technique and, as such, requires the specification of regularization parameters. However, we noticed in the experiments that the regularization parameters as well as the kernel parameters can be easily adjusted and the values do not need to change significantly for different environments. Hilbert maps offer significant computational advantages over GPOM at a reasonable cost: slightly inferior interpolation power due to the Hilbert space approximation. Note also that Hilbert maps can be extended to a variational Bayesian logistic regression formulation as discussed in future work. Table 1 summarizes the main differences between GPOM and Hilbert maps.

RKS: Another type of kernel approximation based on the RKSs but with better scalability properties to high-dimensional data was proposed in Le et al. (2013). The authors show how to improve the cost of computing the features from $\mathcal{O}(nmd)$ to $\mathcal{O}(nm \log d)$, where m is the number of features, n is the number of samples and d is the dimension of the inputs. The method is based on a fast factorized scheme to create the random matrices $\mathbf{s}$ in equation (7). Even though this extension is interesting for problems with high-dimensional data such as in computer vision, it is less effective in our occupancy mapping problem as the dimensionality of the data is at most 3 for the 3D case.

An interesting study comparing RKS and Nyström features on several regression and classification problems was presented in Yang et al. (2012). The authors show that when there is a large gap between the eigen-spectrum $\hat{D}$ in equation (8), the Nyström method can produce impressive results and outperform RKS. As we shall see in the experiments, this was observed in the occupancy mapping problem where the Nyström method required a much smaller set of features to achieve similar accuracy. Note however, that the Nyström method is data dependent and the features cannot be precomputed as with RKS.

Additive kernels: In computer vision, several kernels can be written as the sum of the kernel function applied to each dimension independently: $k(\mathbf{x}, \mathbf{x}') = \sum_{d=1}^{D} k(x_d, x'_d)$. Examples of these kernels include the intersection, χ^2, Hellinger's and Jensen–Shannon kernels. For this class of kernels, Vedaldi and Zisserman (2012) proposed a different approximation based on a discrete version of Bochner's theorem where the spectral representation of the kernel is approximated by a sum. Their approximation led to significant speed ups on popular computer vision tasks such as object and pedestrian recognition with similar accuracy. As with RKS and the Fastfood approximation in Le et al. (2013), Hilbert maps can also be implemented with additive kernels. However, the main benefit of such approximations are in tackling high-dimensional problems, and hence it is unclear which benefits these features will bring to low-dimensional problems such as occupancy representation.

5. Experiments

In the experiments unless stated otherwise we set the parameters of the model as follows: The kernel parameter σ was 1.0, the number of components m for each feature was Fourier = 10,000, Nyström = 1000 and sparse = 2000. The regularization parameters were $\lambda_1 = 0.0001$ and $\lambda_2 = 0.15$ (for the sparse case $\lambda_1 = 0.001$). These parameters were obtained through visual inspection of the results

Table 1. Main differences between GPOM and Hilbert maps.

Method	Type	Prior	Inference cost (per query)	Learning cost (per iteration)
Hilbert Maps	Parametric	Logistic	m	m
GPOM	Nonparametric	Gaussian	n^2	n^3

GPOM: Gaussian processes occupancy map.

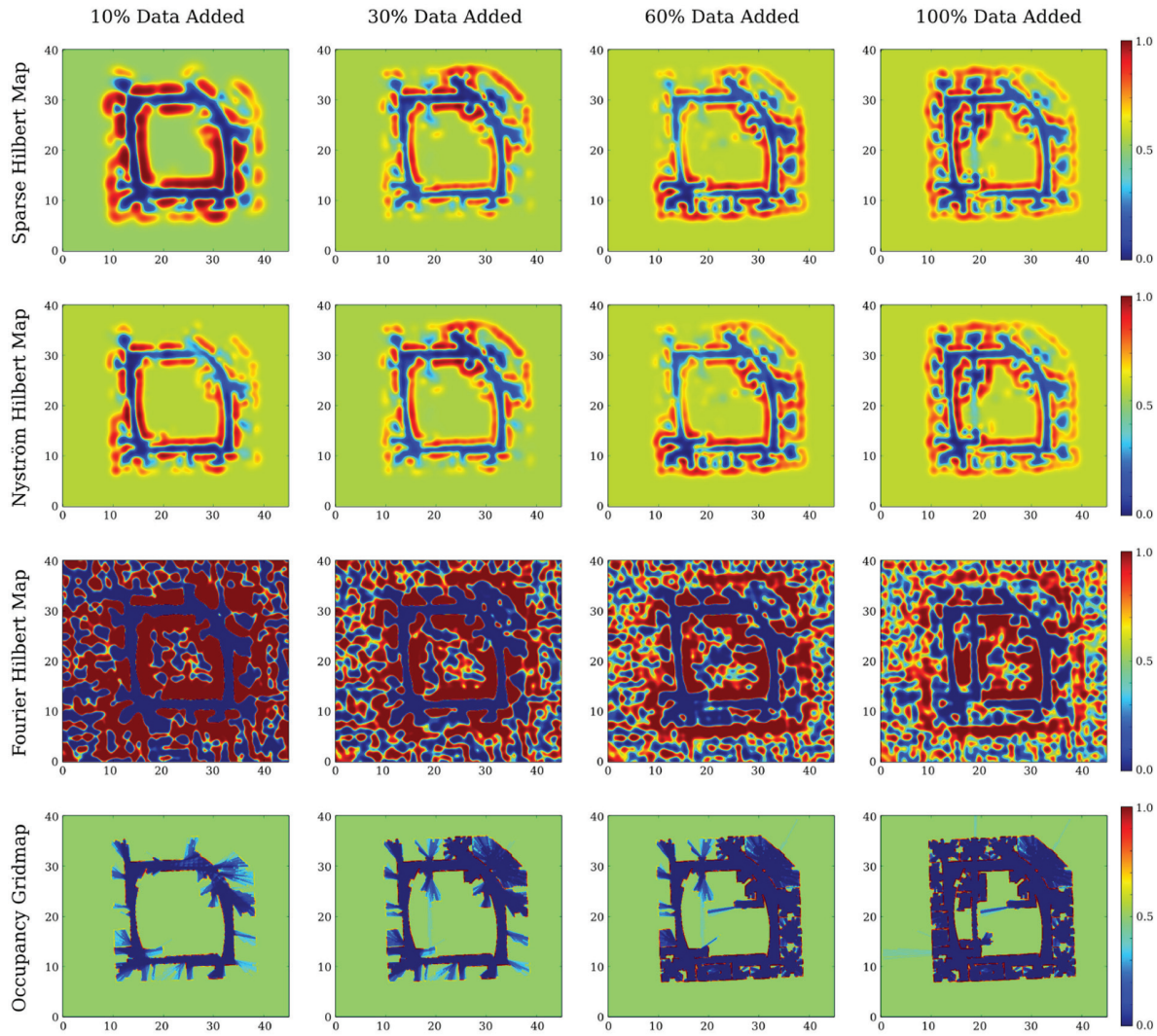

Fig. 2. Evolution of the different maps as observations are incrementally added. From top to bottom we have sparse RBF, Nyström, Fourier and occupancy grid maps with 10%, 30%, 60% and 100% (left to right) of the data incorporated. Axis units are in meters. Note that Fourier features introduce some artifacts, particularly in areas with no data points. Both Nyström and sparse features produce similar results. Occupancy grid maps are visually sharper but this does not necessarily translate to better accuracy as the method does not attempt to extrapolate or generalize to unobserved cells.

and remained unchanged for each of the maps we experimented with. A grid search can also be applied to set these parameters automatically.

5.1. Comparisons between the features

In the first experiment, we compare the three approaches to construct features for Hilbert maps. The experiment was conducted using the data from Intel-Lab (available at http://radish.sourceforge.net/). To better understand the generalization power of the features, we created a series of occupancy maps where several beams from each observation were removed. The maps created were compared against test measurements retained for evaluation purposes and therefore not presented to the algorithm. Figure 2 shows the maps created by the three features and the conventional

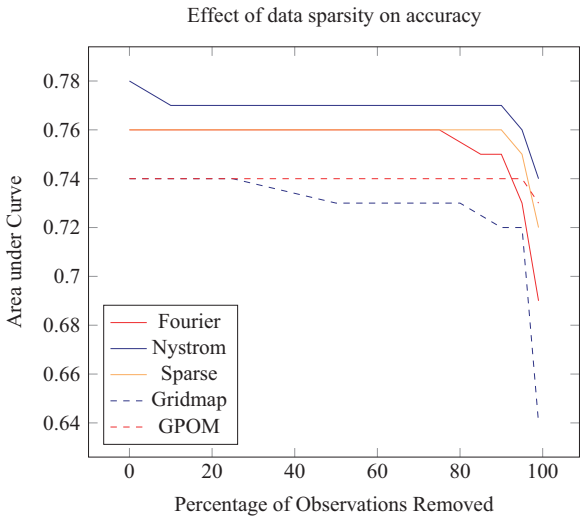

Fig. 3. Evolution of the area under curve for the five mapping methods when a variable amount of data is removed from each scan on the Intel-Lab dataset. Fourier, Nyström and Sparse refer to Hilbert maps using these features.

occupancy grid maps for 10%, 30%, 60% and 100% of the original data incorporated. It can be seen that for both methods, the maps generated with Hilbert maps are much smoother and represent the uncertainty in the occupancy status of the environment more clearly. The occupancy grid maps, despite being sharper, contain a series of artifacts originated from anomalous observations generated from laser returns hitting nonreflective objects or influenced by glass. The Fourier features exhibit artifacts resulting from the cosine approximation to the RBF kernel in areas without observations, but perform comparably to the other two features in areas with more observations.

Figure 3 shows the area under the receiver operating characteristic (ROC) curve for the four approaches (Hilbert maps with Fourier, Nyström, sparse features and occupancy grid maps) for several cases where a percentage of the observations is removed. It can be observed that the sparse features and the Nyström perform the best. We can also observe that occupancy grid maps have problems in representing the environment properly when more than 50% of the data are removed. This indicates that Hilbert maps are more robust and possess better generalization performance than occupancy grids.

In the second experiment we compare Hilbert maps with sparse features against occupancy grid maps for the outdoor dataset recorded at the University of Freiburg, also available at http://radish.sourceforge.net/. Figure 4 shows the maps produced by both methods when 75% of the laser data is removed from each scan. Removing data from the scans allows to better assess the generalization power of both methods. For this test the regularization parameter λ_2 was changed to 0.6 to exploit the sparsity of the data. It can be observed that the Hilbert map is significantly more resilient to outliers and noisy observations that naturally

Table 2. Area under the receiver operating characteristic (ROC) curve and runtime for occupancy grid maps and sparse Hilbert maps when evaluated on the outdoor dataset with 75% of each laser scan missing.

Method	Area under ROC curve	Runtime
Occupancy grid map	0.66	88 s
Sparse Hilbert map	0.84	850 s

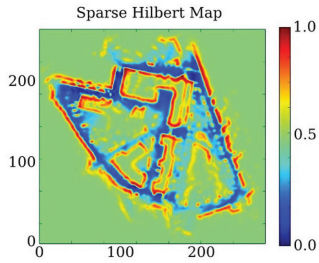

Fig. 4. Visualization of the map obtained using sparse feature Hilbert maps when 75% of the laser data is removed. The width of the roads and thickness of the obstacles are better represented than with occupancy grids.

exist when navigating outdoors. Roads are more clearly identifiable and the map better reflects the actual shapes as can be seen in the aerial photo. Note that the width of the roads is exaggerated in the occupancy grids mostly due to spurious observations on vegetation. Quantitatively, this is confirmed in Table 2 where area under ROC curve for both methods is presented. Also note that the overall uncertainty of the problem due to the noisy observations is much better handled. For this problem, due to the large number of observations, GPOM cannot be computed without sparse or nearest neighbor approximations that require storage of all the data.

Figure 5 shows the equivalent map produced with occupancy grid. Walls and other obstacles are represented as very thin lines. Compared to Hilbert maps, occupancy grids tend to overestimate the unoccupied areas in both size and confidence. An aerial view of the area is presented in Figure 6.

5.2. *Convergence of stochastic gradient descent*

In this experiment we compare the convergence of SGD for the full batch case where all the data is presented to the algorithm multiple times and the incremental version where the algorithm sees each datapoint only once. This experiment was conducted on the Intel-Lab dataset with Nyström and sparse features. Figure 7 shows the value of the SGD objective being minimized with more iterations. For each iteration of the full SGD, all points are presented to the algorithm while for the incremental version, only a small set is presented without repetition. As expected, the incremental version oscillates more than the full version however both cases achieve a similar final energy value. The incremental

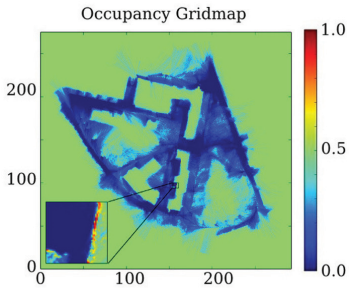

Fig. 5. Visualization of the map obtained using occupancy grid mapping when 75% of the laser data are removed. The zoomed area illustrates parts of a wall with high probability of occupancy, but represented as a very small edge. Compared to Hilbert maps, occupancy grids tend to overestimate the unoccupied areas in both size and confidence.

Fig. 6. Aerial view of the are the maps in Figure 4 and Figure 5.

version is significantly faster to execute; the full SGD takes approximately 21 s to complete 80 iterations with Nyström features and 1 s with sparse features, while the incremental version takes 0.3 s with Nyström and 0.02 s with sparse features. This result indicates the benefits of incremental SGD to quickly arrive at a solution even without observing the entire dataset. Also it shows that SGD can take advantage of sparse features to significantly speed up the computation.

5.3. Comparison with GPOM

We compare Hilbert maps with GPOM on two datasets. Results comparing Hilbert maps with the three features against GPOM for the map in Figure 1 are presented in Table 3. Both methods achieve remarkable classification results but as expected, GPOMs are much more costly to compute, typically $\mathcal{O}(n^3)$ in the number of datapoints. Hilbert maps can produce similar results while being an order of magnitude faster. Most of the computational cost in Hilbert maps is actually the feature computation rather than SGD. The Fourier features are the fastest followed by sparse

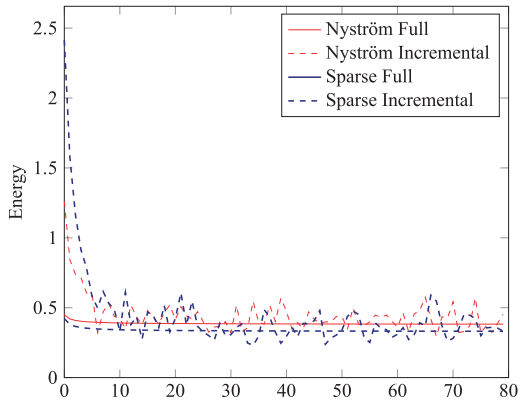

Fig. 7. Evaluation of the solution quality obtained when performing incremental SGD as opposed to full SGD with a fixed number of iterations using both Nyström and sparse features.

Table 3. Comparison of map quality and runtime of GPOM with the three proposed methods on the synthetic example shown in Figure 1. All methods outperform the GPOM by a large margin while obtaining identical or very similar results.

Method	Area under ROC curve	Runtime (s)
GPOM	1.00	38.0
Sparse Hilbert map	1.00	3.2
Fourier Hilbert map	0.98	1.2
Nyström Hilbert map	0.98	6.9

and Nyström which requires an eigen decomposition operation. Note, however, that in both cases the features can be computed in parallel, thus presenting significant speedups on GPUs. The overall cost of Hilbert maps is $\mathcal{O}(m)$ in the number of features per datapoint.

Figure 3 shows a comparison between GPOM, occupancy grids and Hilbert maps with different features on the Intel-Lab dataset as a proportion of the data is removed. As this dataset contains a very large number of data points, GPOM had to be implemented using sparse approximations. We utilized the state-of-the-art stochastic variational Gaussian process classifier as described in Hensman et al. (2015) for this comparison. Since this GPOM model relies on sparse variational Gaussian process approximations, it performs worse than the full GPOM model but becomes competitive as more data is removed. GPOM achieves a similar performance to Hilbert maps when 98% of the data is removed. However, when most of the data is added, its performance is about 3% worse than Hilbert maps.

5.4. Impact of parameters

In this section we evaluate the sensitivity of Hilbert maps to three of its parameters: kernel width $\gamma = \frac{1}{\sigma^2}$ and regularization constants λ_1 and λ_2. The experiments were conducted on the Intel-Lab dataset and the parameters that were not varying were set to the values indicated at the start of this section. In Figure 8, we plot L_1 ratio and strength

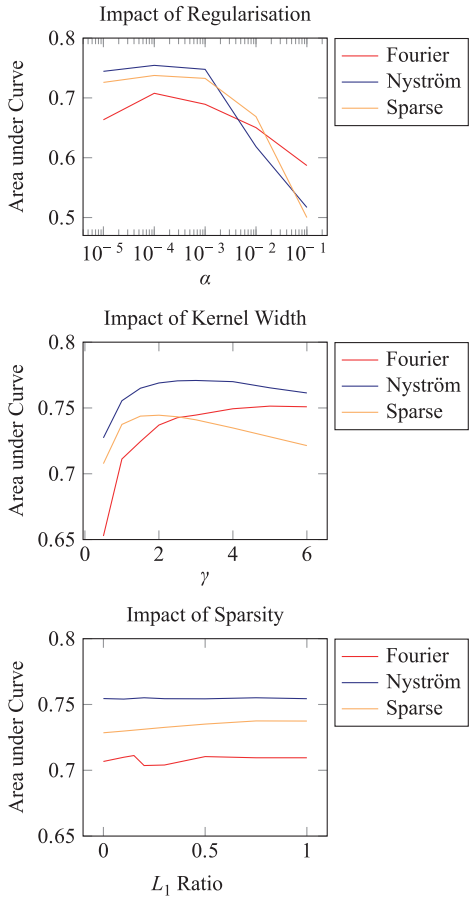

Fig. 8. Impact of parameters on accuracy for the Intel-Labs dataset. Only one parameter is varied at a time while the others are kept at their defaults, stated at the beginning of this section. (Top) Regularization strength. (Middle) Kernel width ($\gamma = \frac{1}{\sigma^2}$). (Bottom) L_1 ratio. In these plots, $\lambda_1 = \alpha \times (1 - L_1 - \text{ratio})$ and $\lambda_2 = \alpha \times L_1 - \text{ratio}$.

of the regularization (α) against accuracy such that $\lambda_1 = \alpha \times (1 - L_1 - \text{ratio})$ and $\lambda_2 = \alpha \times L_1 - \text{ratio}$.

Both features show reasonable robustness to all parameter variations. The sparse feature is more influenced by the choice of parameters, especially the regularizer. However, the range of values in which similar performance is obtained is quite large. All features exhibit similar behavior with regards to the regularization strength (α) in that there is a broad range of equal performance with a sharp drop when regularization becomes too large. The kernel width parameter has a similar impact on both Nyström and sparse in that after a certain value the performance stagnates or starts to decrease. Conversely, the Fourier feature seems to benefit from large values. Finally, the ratio between L_1 and L_2 used by the elastic net regularizer has little impact with the main change being observed for the sparse feature which naturally prefers sparser solutions, i.e. larger L_1-ratio.

5.5. Noisy inputs

In this experiment we demonstrate the ability of the features on distributions described in Section 2.4 to deal with errors in the position of observations. This also addresses the case where observations are partially observable and provided as distributions over the location. We used the synthetic dataset presented in Figure 1 but added noise to the position of the points. This simulates errors in localization commonly present in real problems but allows us to compare against the ground truth. Figure 9 shows the original Hilbert map obtained when there is no noise in the data (top left). The data is then corrupted by Gaussian noise with 20 cm standard deviation (as a reference the size of the map is about 30 m). The Hilbert map with Nyström features obtained on the corrupted dataset is displayed in the top right. It can be observed that walls and shapes are much less defined. This is also the case for the occupancy grid maps (bottom right) where not even the walls can be properly identified. The Hilbert map result with Nyström features on distributions generates the best result (bottom left) which resembles closely the result obtained when no noise is added to the data. This demonstrates the ability of the kernel approximation on distributions to handle highly noisy data.

5.6. Map adaptation

This experiment demonstrates the ability of Hilbert maps to quickly adapt to changes in the position of laser scans. This is typically required when performing simultaneous localization and map building after a loop closure is detected as the estimated positions of the robot during navigation can change significantly after loop closure leading to an inaccurate map. The experiment was conducted on the Intel-Lab dataset (available at http://radish.sourceforge.net). An initial Hilbert map with sparse features was created using simultaneous localitzation and mapping (SLAM) to compute the positions of the robot before a loop closure was detected. Once a loop closure is detected, the positions of the robot are re-estimated resulting in new locations for the laser scans. The updated data is then presented to the online learning algorithm and several passes over the corrected data are performed to assess how quickly a new map, with the updated data, can be generated from the initial, prior to loop closure, map. Figure 10 shows the estimated trajectory of the robot before and after loop closure.

A visualization of the maps after several passes over the corrected data is shown in Figure 11. From the initial map on the top left, it can be seen that after only one pass over the corrected data, the resulting map already contains most of the structure of the after loop closure map. In areas where no information had been available the corrected map is recovered almost instantly. In areas of conflicting information the first pass gives already a good result while the second and fifth pass are both equally accurate with the main difference being the sharpness of the final map.

A quantitative evaluation in terms of area under the ROC curve is presented in Table 4. The evaluation is performed against a test set comprised of 5% of the corrected data removed from training. The result obtained before loop

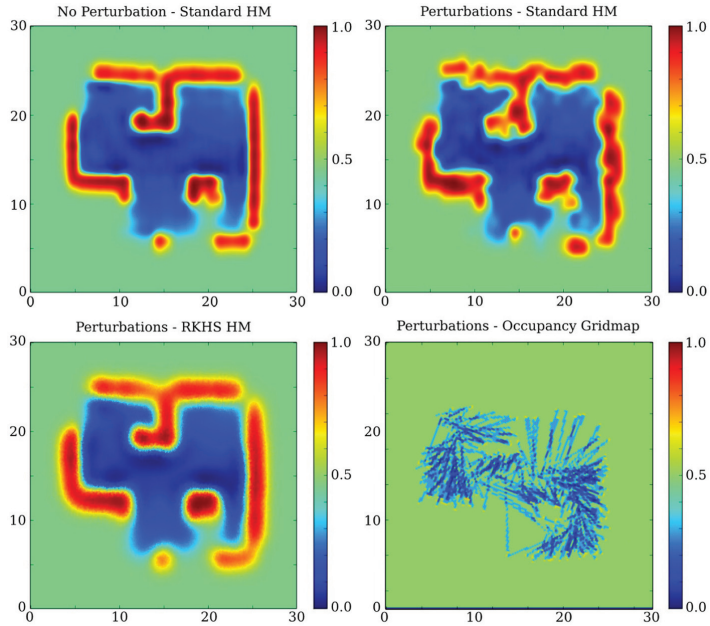

Fig. 9. Visualization of the impact of noisy data. The top left image shows the result obtained when no perturbations are present using Hilbert maps. In the top right the data is perturbed with no compensation performed in the mapping. The bottom left image shows how the RKHS allows us to recover the map. The bottom right shows the effect data sparsity and noise has on occupancy grid maps which would not be suitable for actual use. Axis units are in meters.

Table 4. Area under the ROC curve for the Intel-Lab map evaluated before loop closure, with the previous map and corrected data after loop closure and several passes over the corrected data, and the results using only the corrected data.

Method	Area under the ROC curve
Before loop closure	0.54
Before and after loop closure, 1 pass	0.86
Before and after loop closure, 2 passes	0.89
Before and after loop closure, 3 passes	0.90
Before and after loop closure, 5 passes	0.91
Only after loop closure, 5 passes	0.93

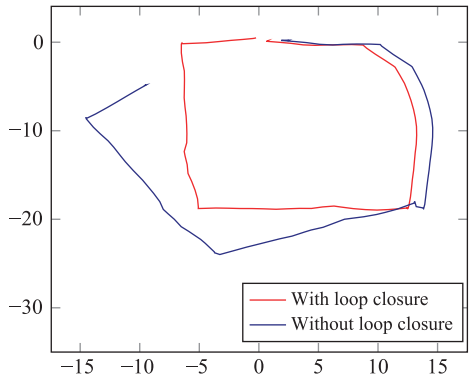

Fig. 10. Estimated trajectories taken by the robot before and after loop closure for the Intel-Lab dataset.

closure is poor as expected. Interestingly, after only one pass over the corrected data the area under the ROC curve already improves significantly compared to the before loop closure map, approaching the quality of the map with only the corrected data. Performing two, three and five passes over the data leads to modest improvements, approaching the maximum 93%. This experiment demonstrates that Hilbert maps can quickly adapt to changes in the environment even if previous observations contradict new information. Furthermore, the model obtained by such corrections has similar quality to a map built from the correct data only.

5.7. 3D Hilbert Maps

In this section we present experiments to demonstrate the ability of Hilbert maps to deal with 3D laser scanner data. We use data obtained from a stationary Riegl scanner, shown in Figure 12(top) which contains around 50,000 data points.

The Hilbert map was built using the sparse kernel with 2500 features initialized to locations found by clustering the data points with k-means++ (Arthur and Vassilvitskii, 2007). Figure 12(bottom) shows the representation obtained when querying the model in a uniform 3D grid. To evaluate the quality of the map representation we removed 5% of the training data containing both occupied and free points and used it for testing. We compare the results obtained with a Hilbert map with the commonly used Octomap representation (Hornung et al., 2013). Table 5 shows the are under the ROC curve scores obtained by the two methods from which we can see that the Hilbert map performs better. This stems from the method's ability to interpolate into areas where no observations have been made. In comparison, Octomaps, which are an effective representation of a fixed resolution 3D grid, have the same drawbacks as 2D occupancy

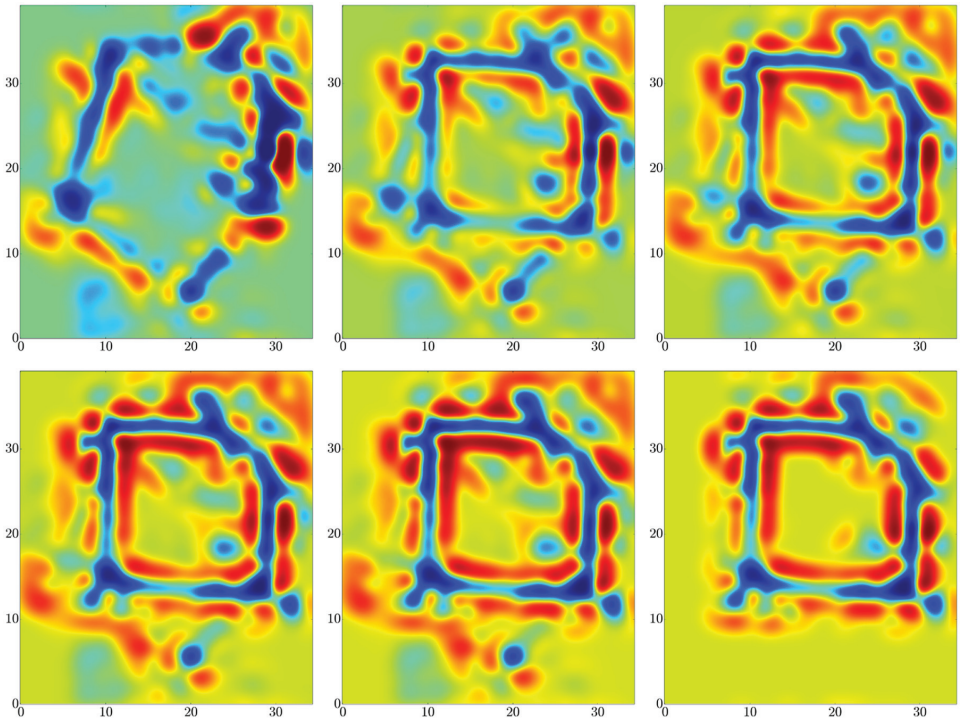

Fig. 11. Hilbert map adaptation after loop closure. Initially the map is constructed with the data before loop closure. Once loop closure is detected, multiple passes over the new data are performed. Top row, left to right: pre loop-closure, 1 pass and 2 passes. Bottom row, left to right: 3 passes, 5 passes and only data after loop closure.

Table 5. AROC for the 3D Riegl dataset when processed using Hilbert maps and Octomaps. The Hilbert map representation achieves a higher score due to the ability to interpolate in areas where data points are missing.

Method	AROC
Hilbert maps	0.92
Octomap	0.86

gridmaps in that they assume independence of neighboring cells causing the representation to have holes even with the high density data provided by the Riegl scanner.

Interestingly, the experiment also demonstrates another crucial advantage of representing space as Hilbert maps—the compression capabilities of the model. This relatively complicated 3D scene can be represented with only 2500 numbers corresponding to the weights of the features. This is in contrast to the 50,000 data points of the entire dataset. The compressed map representation can be exploited in multi-robot exploration and map building to reduce the costs of communicating maps between the robots. Similarly, it can be used for sequential Monte Carlo localization where each particle contains an instance of the map, reducing the memory requirements and computational costs.

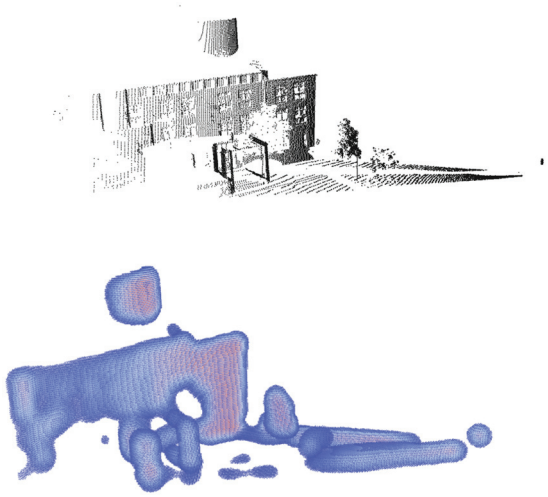

Fig. 12. Visualization of the raw Riegl sensor data used in the experiments (top) and the reconstruction obtained using Hilbert maps (bottom) where blue points have lower occupancy probability then those shown in red.

6. Conclusion and future work

This paper introduced a novel occupancy mapping technique, Hilbert maps. The technique improves over occupancy grid maps in several ways but notably it does not require discretization of the space, rendering maps at any resolution, and it captures spatial relationships to provide better generalization in areas with no measurements while being more robust to outliers. To address the case when the absolute position of observations is partially known or estimated as part of a SLAM procedure, we propose a method to embed the probabilistic distribution over the

spatial location of the observations. The technique explores recent advancements in kernel machines, in particular kernel approximations, to allow efficient learning through SGD where strong convergence guarantees exist even when each data point is visited only once during learning. Experimental results are very encouraging showing that the maps produced are less influenced by outliers and more accurate in representing the underlying uncertainty. The best results were achieved with Nyström and sparse features, both requiring the specification of inducing points. These inducing points can be placed on a grid or be obtained automatically with a clustering procedure such as k-means. Incrementally adding more features as more observations are captured or increasing the number of features in regions of the map where a higher accuracy is required are both interesting future directions to improve the adaptability of the model.

An important advantage of online training strategies based on SGD relates to the speed in which maps can be updated after major trajectory corrections during navigation such as loop closures. Upon loop closure, SGD can be executed on the updated data until convergence to create a new updated map. This procedure can be used in machine learning problems to deal with the problem of nonstationarity in sequential data (also known as "covariance shift" (Sugiyama et al., 2007)). The advantage of Hilbert maps is that SGD is very fast and inexpensive to run therefore retraining can be performed efficiently.

There are many uses for Hilbert maps as a continuous occupancy representation within a robot system. For example, they can be used to generate representations for grasping (Bohg et al., 2010) with an active visual segmentation and haptic exploration. The probabilistic outputs provided by Hilbert maps can guide exploration by informing areas with higher uncertainty in which additional observations are required. Path planning can also be performed with continuous occupancy representations as described in Yang et al. (2013). Popular planning algorithms such as rapidly-exploring random trees can use the occupancy property and associated uncertainty to reason about safety boundaries in the environment. This has been demonstrated for planning in cluttered natural environments with unmanned aerial vehicles. There is also potential for the development of path planning strategies based on functional optimization of trajectories over Hilbert maps. As a continuous representation, Hilbert maps can be directly utilized within optimization algorithms that attempt to find safer trajectories. The gradient of the occupancy function with respect to spatial coordinates can also be easily obtained to speed up the optimization process. Finally, recent results suggest that continuous occupancy maps can be utilized for Monte Carlo localization, achieving superior results to the equivalent formulation with grid maps (Hata et al., 2016). With an appropriate likelihood model that directly leverages the continuous representation of Hilbert maps, localization can be made more robust, particularly under higher levels of uncertainty.

There are several avenues for future work. First it is easy to show that both occupancy grid maps and Octomaps are particular cases of Hilbert maps with specific features. In the case of grids, if the features are defined as 0 or 1 depending on whether the position of the input point is within a square or cube, Hilbert maps will produce similar outputs to gridmaps/Octomaps. This highlights the flexibility of the method and opens the perspective for more feature developments or combinations of features with very different characteristics. Second, the method is formulated as a frequentist technique and requires the manual tuning of kernel and regularization parameters. Even though recent techniques such as Bayesian optimization can be used for this purpose (Snoek et al., 2012), the more principled solution is to formulate the problem as a Bayesian learning task. Unfortunately, this will lead to nonanalytical solutions to the posterior and approximation techniques such as variational methods will need to be applied. Also, the objective function will no longer be convex and the SGD convergence guarantees in this case are less developed. Nevertheless this remains an interesting area for further investigation. Finally, occupancy mapping is just one example of many other problems where this technique can be applied. The fundamental idea of the algorithm which is to provide probabilistic predictions based on a stream of data captured by a moving robot in an online and efficient manner has a number of other applications. For example, the algorithm can be used to learn optimal policies in reinforcement learning, or to perform online object recognition.

Acknowledgments

The authors would like to acknowledge the anonymous reviewers who provided numerous suggestions and constructive criticism during the reviewing process for this extended version.

Funding

This research was supported by funding from the Faculty of Engineering & Information Technologies, The University of Sydney, under the Faculty Research Cluster Program.

Notes

1. A Python implementation of Hilbert maps is available for download at https://bitbucket.org/LionelOtt/hilbert_maps_rss2015.
2. The *hat* in $\hat{\Phi}$ is used to indicate that the feature approximates a kernel in expectation.

References

Arthur D and Vassilvitskii S (2007) k-means++: The advantages of careful seeding. In: Gabow H (ed.) *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, New Orleans, Louisiana, 07–09 January 2007, pp.1027–1035. Philadelphia, PA, US: Society for Industrial and Applied Mathematics.

Bohg J, Johnson-Roberson M, Björkman M, et al. (2010) Strategies for multi-modal scene exploration. In: Luo RC and Asama

H (eds) *IEEE/RSJ international conference on intelligent robots and systems*, Taipei, Taiwan, 18–22 October 2010, pp.4509–4515. Piscataway, NJ, US: IEEE.

Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Yves, Saporta, Gilbert (eds) *Proceedings of COMSTAT*, Paris, France, 22–27 August 2010, pp.177–186. Berlin, Heidelberg: Springer-Verlag.

Bottou L (2012) Stochastic gradient descent tricks. In: *Neural networks: Tricks of the trade*, pp.421–436.

Bottou L and Bousquet O (2008) The tradeoffs of large scale learning. In: Platt JC and Koller D, Singer Y, et al. (eds) *Neural information processing systems (NIPS)*, Vancouver, BC, Canada, 3–5 December 2007, pp.161–168. Red Hook, NY: Curran Associates, Inc.

Elfes A (1987) Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation* RA-3(3): 249–265.

Elfes A (1989) *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD Thesis, Carnegie Mellon University, USA.

Gihman II and Skorohod AV (1974) *The Theory of Stochastic Processes.* Vol. 1. Berlin, Germany: Springer Verlag.

Hata A, Ramos F and Wolf D (2016) Particle filter localization on continuous occupancy maps. In: *International symposium on experimental robotics*.

Hensman J, Matthews AGdG and Ghahramani Z (2015) Scalable variational Gaussian process classification. In: Lebanon G and Vishwanathan SVN (eds) *Proceedings of the eighteenth international conference on artificial intelligence and statistics*, San Diego, California, USA, 9–12 May 2015, pp.351–360. Brookline, MA: Microtome Publishing.

Hornung A, Wurm K, Bennewitz M, et al. (2013) OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots* 34(3): 189–206.

Le Q, Sarlos T and Smola A (2013) Fastfood - Approximating kernel expansions in loglinear time. In: Dasgupta S and McAllester D (eds) *International conference on machine learning (ICML)*, Atlanta, 17–19 June 2013, pp.244–252. Brookline, MA: Microtome Publishing.

MacKay DJC. Introduction to Gaussian processes. In: Bishop CM (ed.) *Neural Networks and Machine Learning*, volume 168 of NATO ASI Series, Springer, Berlin, 1998, pp.133–165.

Melkumyan A and Ramos F (2009) A sparse covariance function for exact Gaussian process inference in large datasets. In: Kitano H (ed.) *International joint conference on artificial intelligence (IJCAI)*, Pasadena, California, USA, 11–17 July 2009, pp.1936–1942. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

O'Callaghan ST and Ramos FT (2012) Gaussian process occupancy maps. *The International Journal of Robotics Research* 31(1): 42–62.

O'Callaghan ST, Ramos FT and Durrant-Whyte H (2009) Contextual occupancy maps using Gaussian processes. In: *IEEE international conference on robotics and automation (ICRA)*, Kobe, Japan, 12–17 May 2009, pp.3630–3636. Piscataway, NJ, USA: IEEE Press.

Platt JC (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Platt JC, Koller D, Singer Y, et al. (eds) *Advances in Large Margin Classifiers*. Cambridge, USA: MIT Press, pp.61–74.

Rahimi A and Recht B (2008) Random features for large-scale kernel machines. In: Koller D, Schuurmans D, Bengio Y, et al. (eds) *Neural information processing systems (NIPS)*, Vancouver, B.C., Canada, 3–5 December 2007, pp.1177–1184. Red Hook, NY: Curran Associates, Inc.

Rahimi A and Recht B (2009) Weighted sums of random kitchen sinks: Replacing minimization with randomisation in learning. In: Koller D, Schuurmans D, Bengio Y, et al. (eds) *Neural information processing systems (NIPS)*, Vancouver, B.C., Canada, 8–10 December 2008, pp.1313–1320. Red Hook, NY: Curran Associates, Inc.

Scholkopf B and Smola AJ (2001) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press.

Singer Y and Srebro N (2007) Pegasos: Primal estimated subgradient solver for svm. In: Ghahramani Z (ed.) *International conference on machine learning (ICML)*, Corvalis, Oregon, USA, 20–24 June 2007, pp.807–814. New York, NY: ACM.

Smola A, Gretton A, Song L, et al. (2007) A hilbert space embedding for distributions. In: Hutter M, Servedio RA, Takimoto E, et al.(eds) *International conference algorithmic learning theory (COLT)*, Sendai, Japan, 1–4 October 2007, pp.13–31. Berlin, Heidelberg: Springer-Verlag.

Snoek J, Larochelle H and Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In: Pereira F, Burges CJC, Bottou L, et al. (eds) *Neural information processing systems (NIPS)*, Harrahs and Harveys, Lake Tahoe, 03–08 December 2012, pp.2951–2959. Red Hook, NY: Curran Associates, Inc.

Song L, Fukumizu K and Gretton A (2013) Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine* 30(4): 98–111.

Sugiyama M, Krauledat M and Müller K (2007) Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research* 8: 985–1005.

Vedaldi A and Zisserman A (2012) Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(3): 480–492.

Williams C and Seeger M (2000a) The effect of the input density distribution on kernel-based classifiers. In: Langley P (ed.) *International conference on machine learning (ICML)*, Stanford University, Stanford, CA, USA, 29 June–2 July 2000, pp.1159–1166. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Williams CKI and Seeger M (2000b) Using the Nyström method to speed up kernel machines. In: Leen TK, Dietterich TG and Tresp V (eds) *Neural information processing systems (NIPS)*, Denver, CO, USA, 28–30 November 2000, pp.682–688. Cambridge, Massachusetts: MIT Press.

Xu W (2011) Towards optimal one pass large scale learning with averaged stochastic gradient descent. *CoRR* abs/1107.2490 (arXiv:1107.2490v1).

Yang K, Gan SK and Sukkarieh S (2013) A Gaussian process-based rrt planner for the exploration of an unknown and cluttered environment with a uav. *Advanced Robotics* 27(6): 431–443.

Yang T, Li Y, Mahdavi M, et al. (2012) Nyström method vs random fourier features: A theoretical and empirical comparison. In: Leen TK, Dietterich TG and Tresp V (eds) *Neural information processing systems (NIPS)*, Denver, CO, USA, 28–30 November 2000, pp.682–688. Cambridge, Massachusetts: MIT Press.

Zhang T (2004) Solving large-scale linear prediction problems using stochastic gradient descent algorithms. In: Greiner R and Schuurmans D (eds) *International conference on machine learning (ICML)*, Banff, Alberta, Canada, 04–08 July 2004, pp.116–124. New York, NY, USA: ACM.

Zou H and Hastie T (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B* 67: 301–320.