Bayesian Hilbert Maps for Continuous Occupancy Mapping in Dynamic Environments

Ransalu Senanayake [1] **Fabio Ramos** [1]

Abstract

Building accurate occupancy maps is crucial for autonomous vehicles to make path planning safer. Hilbert maps (HMs) are used for building such occupancy maps in a continuous fashion from depth sensors such as LiDAR in static environments. However, HMs are highly dependent on coefficients of the regularization term of the objective function which needs to be tuned heuristically. In this paper, we take a Bayesian approach, thus getting rid of the regularization term. Further, we extend the proposed model, Bayesian Hilbert maps (BHMs), to learn long-term occupancy maps in dynamic environments. Comparing with the state-of-the art techniques, experiments are conducted in environments with moving vehicles to demonstrate the robustness against occlusions as well as various aspects of building long-term occupancy maps.

Video and code: `https://goo.gl/5yvHT4`

1. Introduction

Distinguishing occupied areas from unoccupied areas in previously unseen and unstructured environments is important for path planning in autonomous vehicles. The task becomes even more challenging in the presence of dynamic objects such as moving vehicles. Since almost all fully autonomous vehicles—commercial driverless cars such as Uber, Google, etc. and trucks used in the mining industry etc. (Scheding et al., 1999)—are equipped with depth sensors such as LiDAR, the objective is to build occupancy maps from such sparse laser or sound reflections.

Conventionally, occupancy grid maps have been used for modeling the occupancy state of the environment by dividing the world into a fixed-sized grid and then applying a Bayes filter for cells individually (Elfes, 1989). In order to alleviate its limitations such as, 1) the requirement of predefining a cell size, 2) completely ignoring the dependence between cells, Gaussian process occupancy maps (GPOM) (O'Callaghan & Ramos, 2012; Wang & Englot, 2016) were developed. The covariance function of the Gaussian process naturally considers neighborhood information making GPOMs robust against occlusions.

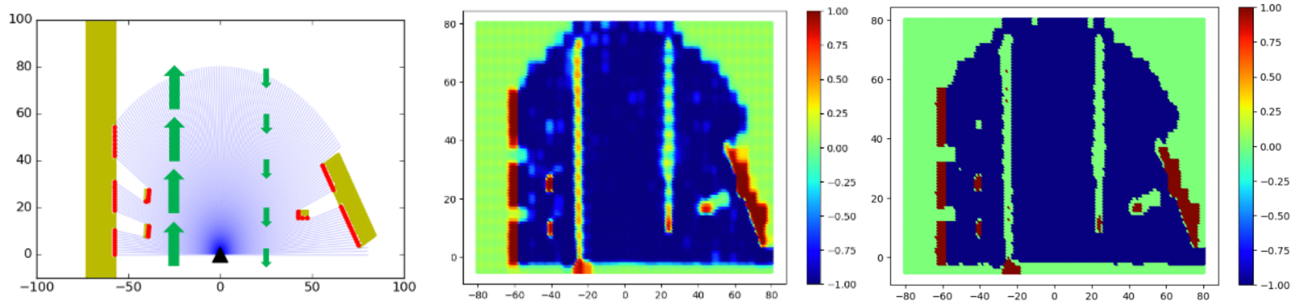
As opposed to occupancy grid maps, GPOMs model the occupancy as a function which can be easily queried to evaluate the occupancy probability of any point in the environment. Such probabilistic frameworks can be effectively used for path planning with safety in mind (Norouzii et al., 2016; Marinho et al., 2016), and perform simultaneous mapping and planing (Francis et al., 2017) under one framework thanks to kernelization. Nevertheless, being a Bayesian non-parametric model, the computational complexity of GPOM—$\mathcal{O}(N^3)$ for N data points—and its faster implementations (Senanayake et al., 2017) grows with the number of data points.

Bringing all advantages of GPOMs, (Ramos & Ott, 2015) proposed a parametric model based on another kernel method, and named it as Hilbert maps (HMs). It takes the form of a kernelized logistic regression classifier which attempts to minimize the regularized negative log-likelihood using stochastic gradient descent to estimate its parameters. Although HMs have been extended to predict short-term dynamics (Senanayake et al., 2016), they cannot be directly used for mapping long-term dynamics—areas which are occupied in general when observed over a period of time, as in the proposed model illustrated by the example in Figure 1. The reason is that it internally uses a Gaussian process regression model with a squared-exponential kernel in the time domain.

As another limitation of Hilbert maps, its objective function contains a regularization term which needs to be prefixed. The regularization parameters are used to prevent over-fitting and maintain a consistent sparsity in the map, especially in areas where no data are available. This effect the regularization term will be further discussed in Section 2.3. In order to alleviate the issues in static Hilbert

[1] School of Information Technologies, The University of Sydney, Australia. Data61/CSIRO, Australia.. Correspondence to: Ransalu Senanayake <rsen4557@uni.sydney.edu.au>.

Workshop on Machine Learning for Autonomous Vehicles. 34^{th} International Conference on Machine Learning, Sydney, Australia, 2017.



(a) Motion field (b) Bayesian Hilbert Map (c) Tristate BHM

Figure 1. The long-term occupancy map produced using the proposed algorithm—Bayesian Hilbert maps (BHM)—after observing the field for several minutes The robot, indicated by the black arrow head, resides at the middle of the two roads. Its field of view is shown in blue laser beams with red laser hit points, when there are no moving vehicles. Static objects such as buildings and parked vehicles are shown in yellow and the traffic flow in green arrows. Vehicles moving in the upward direction are more frequent than that of downward. Therefore, after several laser observations, the occupancy probability of the left road shown in (b) is higher than that of the right road. More occupied areas are indicated by $+1$ while less occupied areas are indicated by -1. The occupancy level of the unseen outlying areas is almost 0, i.e. uncertain. (c) is the tristate BHM map showing occupied, unoccupied, and undecidable was obtained by rounding the occupancy levels to the nearest integers$\in \{-1, 0, +1\}$. Since the model captures neighborhood relationships, areas around $(-50, 15)$, $(-50, 35)$ and $(60, 20)$ are correctly mapped, regardless of occlusions due to the three parked vehicles.

maps we propose techniques to,

1. eliminate crucial parameter tuning in Hilbert maps
2. analyze long-term dynamics.

For the best of our knowledge, the other methods that concern about dynamics in the environment when building occupancy maps are either for extracting patterns (Saarinen et al., 2012; Meyer-Delius et al., 2012; Wang et al., 2015; Mitsou & Tzafestas, 2007; Krajnık et al., 2014) or eliminating dynamic objects from the environment in order to build robust static maps (Hahnel et al., 2003; Walcott, 2011). In contrast, as in (Senanayake et al., 2017), our method focused on developing long-term dynamic maps which can later be used for path planning. On the other hand, the other methods are prone to limitations of grid maps.

2. Hilbert Maps

The Hilbert maps framework (Ramos & Ott, 2015) is developed for building continuous occupancy maps in static environments. It makes use of regularized logistic regression to model occupied and unoccupied states, and optimizes its model parameters using stochastic gradient descent (SGD).

2.1. Data

It is assumed that data points are collected from a line-of-sight depth sensor such as LiDAR or sonar. The end point of each beam, when it hits an obstacle, is labeled as an occupied point $y = 1$, and samples drawn from a uniform distribution with a support between the sensor and the end point are labeled as unoccupied points $y = -1$. The spatial locations, latitude and longitude, correspond to each y are denoted by $\mathbf{x} \in \mathbb{R}^2$. N such input-output pairs $\{\mathbf{x}_n, y_n\}_{n=1}^N$ will be used for supervised learning.

2.2. The Hilbert maps model

Hilbert maps are based on an approximate kernel defined by the inner product $\mathrm{kern}(\mathbf{x}, \tilde{\mathbf{x}}) \approx \Phi(\mathbf{x})^\top \Phi(\tilde{\mathbf{x}})$ with features $\Phi(\cdot)$. Although three different features are suggested in (Ramos & Ott, 2015), our discussion will be based on hinged features defined by,

$$k(\mathbf{x}, \tilde{\mathbf{x}}) = \exp\Big(- \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|^2}{l} \Big), \tag{1}$$

as they have a physical meaning and, as also concluded by authors, they experimentally outperform other features. Here, l is the length scale which controls the width of the Gaussian-shaped curve, and $\tilde{\mathbf{x}}$ is a spatially fixed point in the environment. Having M such points hinged in different locations of the environment, the feature vector can be computed by,

$$\Phi(\mathbf{x}) = \big(k(\mathbf{x}, \tilde{\mathbf{x}}_1), k(\mathbf{x}, \tilde{\mathbf{x}}_2), ..., k(\mathbf{x}, \tilde{\mathbf{x}}_M)\big), \tag{2}$$

The probability that a point in the environment is not-occupied is defined by the sigmoid function,

$$p(y = -1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp\big(\mathbf{w}^\top \Phi(\mathbf{x})\big)} =: \sigma(-\mathbf{w}^\top \Phi(\mathbf{x})). \tag{3}$$

The parameters $\mathbf{w}$ are learned by minimizing the regularized negative log-likelihood,

$$\sum_{i=1}^N \log\Big(1 + \exp\big(- y_i \mathbf{w}^\top \Phi(\mathbf{x})\big)\Big) + \mathrm{Reg}(\mathbf{w}), \tag{4}$$

where,

$$\mathrm{Reg}(\mathbf{w}) = \underbrace{\alpha_1 \|\mathbf{w}\|_1}_{Lasso} + \underbrace{\alpha_2 \|\mathbf{w}\|_2^2}_{Ridge}, \tag{5}$$

with α_1 and α_2 regularization parameters.

2.3. Importance of regularization

The regularization term in (5) is commonly known as the elastic-net regularizer which can be set as a convex combination of Lasso and Ridge regularization. The L1 norm controls the sparsity while the L2 norm controls the convexity. As illustrated in Figure 2, the Hilbert map model heavily depends on the regularization and requires careful tuning.

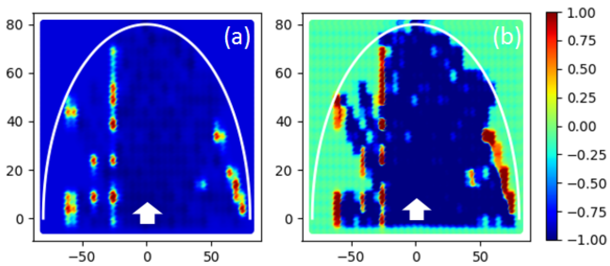


Figure 2. The white arrows show the position of the robot while the white arcs show the range the robot can see. (a) Hilbert maps "without regularization" consider areas where there are no data as unoccupied—majority dark blue, (b) Bayesian Hilbert maps which does not explicitly require a regularization term intrinsically identify such areas as unknown—green.

3. Bayesian Hilbert Maps

3.1. The model

As with HMs, data are collected as discussed in Section 2.1 and features $\Phi(\mathbf{x})$ are computed using (2). However, unlike HMs, we will take a Bayesian approach, effectively eliminating the requirement of regularization terms, rather than minimizing the regularized negative log-likelihood. It is not possible to obtain an analytical solution for the posterior because of the sigmoid likelihood, and hence, as indicated by (6), the posterior is approximated by another distribution Q.

$$\begin{aligned}\underbrace{\mathrm{P}(\mathbf{w},\alpha|\mathbf{x},\mathbf{y})}_{\text{posterior}} &= \frac{\overbrace{\mathrm{P}(\mathbf{y}|\mathbf{x},\mathbf{w})}^{\text{likelihood}} \times \overbrace{\mathrm{P}(\mathbf{w}|\alpha)}^{\text{prior}} \times \overbrace{\mathrm{P}(\alpha)}^{\text{hyper-prior}}}{\underbrace{\mathrm{P}(\mathbf{y})}_{\text{marginal likelihood}}} \\ &\approx \underbrace{\mathrm{Q}(\mathbf{w},\alpha)}_{\substack{\text{approx.}\\ \text{posterior}}} = \underbrace{\mathrm{Q}(\mathbf{w})\mathrm{Q}(\alpha)}_{\substack{\text{factorized approx.}\\ \text{posterior}}}\end{aligned} \tag{6}$$

Each term of (6) will be elaborated in the following sections.

3.1.1. The likelihood

As discussed in Section 2.1, LiDAR data points are independent from each other and hence the data likelihood can be written as,

$$\mathrm{P}(\mathbf{y}|\mathbf{x},\mathbf{w}) = \prod_{n=1}^{N} \mathrm{P}(y_n|\mathbf{x}_n,\mathbf{w}) = \prod_{n=1}^{N} \sigma(y_n \mathbf{w}^\top \mathbf{x}_n). \tag{7}$$

This sigmoid likelihood does not have a conjugate prior. Therefore, it will be locally approximated by the exponential of a quadratic form in such a way a standard prior distribution can be used to make the posterior evaluable using variational inference (Bishop, 2006).

Theorem 1 *(Jaakkola & Jordan, 1997) A sigmoid likelihood $\sigma(yr) := \mathrm{P}(t|r)$ can be lower bounded by,*

$$\sigma(yr) \geq \sigma(\xi) \exp\left(\frac{\xi - r}{2} - \lambda(\xi)\big(r^2 - \xi^2\big)\right), \tag{8}$$

where,

$$\lambda(\xi) = \frac{1}{2\xi}\left(\sigma(\xi) - \frac{1}{2}\right), \tag{9}$$

with ξ as a local parameter used to linearize the function using the Taylor expansion.

Letting $r = \mathbf{w}^\top \Phi$ in Theorem 1 provides a lower bound for the data likelihood defined in (7). ξ parameters that needs to be learned from data.

3.1.2. The prior distribution

Rather than having a pre-defined hyperparameter α in $\mathrm{P}(\mathbf{w}|\alpha)$, the objective is to learn α from the data itself. Hence, a Gaussian prior $\mathrm{P}(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$ will be used with a Gamma hyper-prior $\mathrm{P}(\alpha) = \Gamma(\alpha|a_0, b_0)$, where constants a_0 and b_0 are shape parameter and scale parameter, respectively. For automatic relevance determination, each weight can have separate α values.

3.1.3. The posterior distribution

Considering the mean-field approximation (Bishop, 2006), the posterior distribution is factorized as $\mathrm{Q}(\mathbf{w},\alpha) = \mathrm{Q}(\mathbf{w})\mathrm{Q}(\alpha)$, where $\mathrm{Q}(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mu, \mathbf{\Sigma})$ and $\mathrm{Q}(\alpha) = \Gamma(a, b)$. In the learning phase, it is required to learn parameters $\mu, \mathbf{\Sigma}, a$, and b to accurately model occupancy states.

3.2. Learning parameters

The marginal likelihood,

$$\mathrm{P}(\mathbf{y}|\mathbf{x}) = \int\int \mathrm{P}(\mathbf{y}|\mathbf{x},\mathbf{w})\mathrm{P}(\mathbf{w}|\alpha)\mathrm{P}(\alpha)\mathrm{d}\mathbf{w}\mathrm{d}\alpha, \tag{10}$$

is intractable. Therefore, in variational inference, the log-marginal likelihood is decomposed as,

$$\ln \underbrace{\mathrm{P}(\mathbf{y}|\mathbf{x})}_{\substack{\text{marginal}\\\text{likelihood}}} = \mathcal{L}\big(\underbrace{\mathrm{Q}(\mathbf{w},\alpha)}_{\substack{\text{approx.}\\\text{posterior}}}\big) + \mathbb{KL}\big(\underbrace{\mathrm{Q}(\mathbf{w},\alpha)}_{\substack{\text{approx.}\\\text{posterior}}} \,\|\, \underbrace{\mathrm{P}(\mathbf{w},\alpha|\mathbf{x},\mathbf{y})}_{\text{posterior}}\big), \tag{11}$$

where,

$$\mathcal{L} = \int\!\!\int \mathrm{Q}(\mathbf{w},\alpha) \ln\left(\frac{\mathrm{P}(\mathbf{w},\alpha,\mathbf{y})}{\mathrm{Q}(\mathbf{w},\alpha)}\right) \mathrm{d}\mathbf{w}\mathrm{d}\alpha, \tag{12}$$

is a lower bound, and,

$$\mathbb{KL} = -\int\!\!\int \mathrm{Q}(\mathbf{w},\alpha) \ln\left(\frac{\mathrm{P}(\mathbf{w},\alpha|\mathbf{y})}{\mathrm{Q}(\mathbf{w},\alpha)}\right) \mathrm{d}\mathbf{w}\mathrm{d}\alpha, \tag{13}$$

is the Kullback-Leibler divergence. Typically, the objective is to find $\mathbf{w}$ and α that minimize the distance, i.e. KL-divergence, between the approximate posterior and true posterior. However, since computing KL-divergence requires access to the true posterior which we do not have, instead of minimizing the KL term, the lower bound $\mathcal{L}$ is maximized, considering the fact that marginal likelihood does not depend on parameters.

Unlike in the generic setting, $\mathcal{L}$ is also not explicitly computable here because of the sigmoid likelihood. Therefore, combining the bound (8) with the decomposition (11), a new lower bound $\tilde{\mathcal{L}}(\mathrm{Q},\xi) \leq \mathcal{L}(\mathrm{Q})$ is obtained.

$$\begin{aligned}\tilde{\mathcal{L}}(\mathrm{Q},\xi) &= \frac{1}{2}\log|\mathbf{\Sigma_n}| + \frac{1}{2}\mu_\mathbf{n}\mathbf{\Sigma_n^{-1}}\mu_\mathbf{n}{}^\top \\ &+ \sum_{t=0}^{n}\left(\log\sigma(\xi_t) + \xi_t^2\lambda(\xi_t) - \frac{\xi_t}{2}\right) \\ &+ \log\frac{\Gamma(a_n)}{\Gamma(a_0)} + \log\frac{{b_0}^{a_0}}{{b_n}^{a_n}} + a_n\left(1 - \frac{b_0}{b_n}\right)\end{aligned} \tag{14}$$

Algorithmically, although it is not required to compute (14) to learn parameters, it is useful to evaluate it once every few steps to guarantee the convergence. The variational parameters can be learned using the following iteratively as an Expectation-Maximization (EM) procedure.

E-step : In the expectation step, ξ values are fixed and $\mu, \mathbf{\Sigma}, a$, and b are updated.

M-step: In the maximization step, $\mu, \mathbf{\Sigma}, a$, and b are fixed and ξ values are updated.

3.3. Online learning

For each data point in each sequential scan $\mathbf{x}_*$, except for the very first scan, $f(\mathbf{x}_*)$ is calculated and if each data point satisfies the criterion $|f(\mathbf{x}_*) - y_{\text{true}}| \geq \eta$ such data points are used for learning the map. Here, $f(\cdot)$ is the function that is used to query from the map that has been learned before incorporating the data from the new scan, and y_{true} are the actual occupancy state $\in \{+1, -1\}$ of each point in the current scan. As illustrated in Figure 3, this filters points that can provide new information and η is the threshold for the filter. For instance, if a vehicle is moving into a new area, new information around that area will be higher than stationary areas. Although we attempted to use cross entropy, the aforementioned criterion experimentally gave better results.

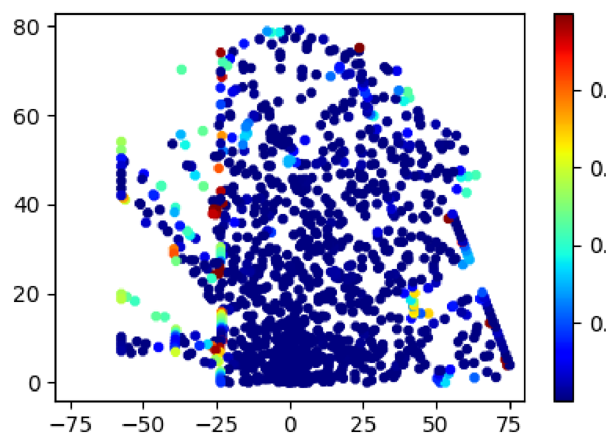


Figure 3. Having learned the map for $t-1$ time steps, the information gain is calculated for each point in the t^{th} scan. The higher values, i.e. red, indicates new information such as an area a vehicle has entered. Therefore, adding data points with smaller values, say values < 0.3, does hardly improve the accuracy.

3.4. Querying

Simply, in order to evaluate the occupancy level of any point in the environment $\mathbf{x}_*$, it is required to evaluate the map function. Therefore, the predictive distribution is obtained by marginalizing over the posterior distribution. Therefore, the log-probability that a given point is occupied $\log \mathrm{P}(y=1|\mathbf{x}_*,\mathbf{x},\mathbf{y})$ can be approximated as,

$$\begin{aligned}&\frac{1}{2}\left(\mu_*\mathbf{\Sigma}_*^{-1}\mu_*{}^\top - \mu_\mathbf{n}\mathbf{\Sigma_n^{-1}}\mu_\mathbf{n}{}^\top\right) \\ &\quad + \frac{1}{2}\log\frac{|\mathbf{\Sigma}_*|}{|\mathbf{\Sigma_n}|} + \log\sigma(\xi_*) + \xi_*^2\lambda(\xi_*) - \frac{\xi_*}{2},\end{aligned} \tag{15}$$

where,

$$\mathbf{\Sigma}_*^{-1} = \mathbf{\Sigma_n^{-1}} + 2\lambda(\xi_*)\mathbf{x}_*\mathbf{x}_*^\top, \tag{16}$$

$$\mu_* = \mathbf{\Sigma}_*\left(\mathbf{\Sigma_n^{-1}}\mu_n + \frac{1}{2}\mathbf{x}_*\right), \tag{17}$$

and

$$\xi_* = \mathbf{x}_*^\top(\mathbf{\Sigma}_* + \mu^\top \mu)\mathbf{x}_* \tag{18}$$

Note that ξ_* depends on $\mu_*, \mathbf{\Sigma}_*$ and vice versa. Hence, (16)-(17) and (18) have to be estimated iteratively to find the best values. Note that these computations are straightforward and inexpensive. There are alternative approaches as well.

4. Experiments

4.1. Experimental setup and evaluation

Two datasets taken from (Senanayake et al., 2017) will be used for experiments. Dataset 1 is obtained from a simulator which resembles a 80 m LiDAR covering 180^0. As shown in Figure 1, this dataset has been designed in such a way that vehicles travel more often in the left lane than in the right lane. The walls in the sides of the roads are partially occluded by the parked vehicles. Datasets 2 is taken from a real world four way intersection with vehicles turning in different directions following traffic lights. Its LiDAR covers 180^0 in a 30 m radius.

The proposed model, BHM, will be compared against, 1) variational sparse dynamic against Gaussian process occupancy maps (VSDGPOM) (Senanayake et al., 2017) which is capable of building similar maps to BHM, 2) dynamic Gaussian process occupancy maps (DGPOM) (O'Callaghan & Ramos, 2014), and 3) dynamic Grid maps (DGrid), an extension of occupancy grid maps to dynamic environments by keeping memory in each cell individually (Arbuckle et al., 2002).

All experiments were run on a laptop with a 8 GB RAM. It is assumed that the robot is stationary and the localization is given. As in (Senanayake et al., 2017), the area under the receiver operating characteristic (ROC) curve (AUC) will be used as the fundamental measure for comparisons. As an additional metric, negative log-likelihood loss (NLL), also known as cross entropy, defined by $-\log p(y|y_*) = -y\log\left(y_*\right) + (1-y)\log(1-y_*)$ will be used for evaluating accuracy in the spatio-temporal setting.

Unless otherwise mentioned, the following parameter values were used for all experiments: $a_0 = 10^{-3}$, $b_0 = 10^{-4}$, $\eta = 0.3$, and $l = 1/0.15$. As discussed in Section 3.1.2, the method is less sensitive to a_0 and b_0 because they are initial values of the hyper-prior which is used to learn the parameters of the prior. The filtering threshold η can be thought as a parameter which controls the tread off between the speed and accuracy. The effect of l which is the only crucial parameter will be discussed in Section 4.2. The accuracy of predicting the occupancy level of a given location is discussed in Section 4.3. The experiments section concludes by analyzing spatio-temporal effects in Section 4.4.

4.2. Effect of the length scale

In this section, we demonstrate the effect of the length scale. As shown in Figure 4, intuitively, the length scale controls the smoothness of the map. For instance, small length scales tend to produce less smooth maps while capturing sharp edges. In contrast, large length scales produce smoother maps because each point has a very high influence on even farther neighbors making an average over a large local realm.

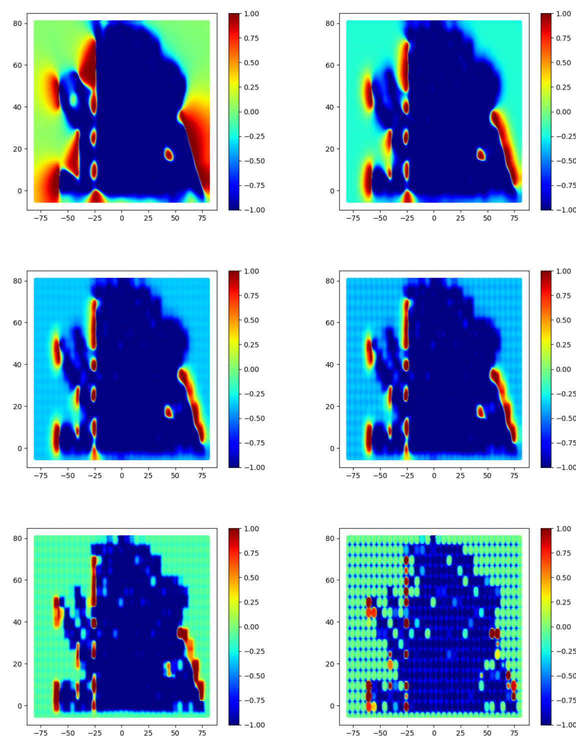


Figure 4. The effect of length scale for the LiDAR scan at $t = 0$ in dataset 1. Reading from left to write, θ values from top left to bottom right are 0.1, 0.5, 0.8, 1, 2, and 10, where length scale is $l = 1/(0.75 \times \theta)$ for a squared-exponential kernel $\exp\left(-\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|^2}{l}\right)$.

4.3. Spatial accuracy

In order to determine how well the occupancy probability of a given location can be predicted accurately, always occupied areas such as walls, parked vehicles, and always unoccupied areas were labeled manually. Note that this test dataset contains both occluded and non-occluded areas of the environment. The AUC is calculated over a period of time as the map is built sequentially. The first two numerical columns of Table 1 indicate that spatial accuracy of BHMs is comparable to Gaussian process based techniques, while higher than grid maps.

The last column of Table 1 reports the accuracy of predicting occupancy state in occluded areas such as behind parked vehicles. As expected, the grid based method is not robust against occlusions as it does not consider neighborhood information unlike the other three kernel methods. However, on a different note, if the occlusions are large and the area is not visible at all, obviously, even kernel based methods are also not robust against occlusions, and the accuracy will not be close to one. Nevertheless, in comparison with DGrid, we demonstrate the advantage of considering neighborhood information.

Table 1. Average AUC ($\mu \pm 2\sigma$) for labeled spatial data. The first two columns show accuracy based on randomly selected points from the environment and the last column indicates accuracy of predicting occluded areas.

Method	Dataset 1	Dataset 2	Dataset 1 Occ
BHM	1.00 ± 0.01	1.00 ± 0.02	1.00 ± 0.00
VSDGPOM	0.99 ± 0.04	1.00 ± 0.00	1.00 ± 0.00
DGPOM	0.99 ± 0.02	0.98 ± 0.08	0.99 ± 0.02
DGrid	0.78 ± 0.04	0.84 ± 0.17	0.50 ± 0.00

4.4. Learning long-term maps

In this experiment, we demonstrate building spatio-temporal maps. Figures 1 and 5 illustrate such long-term occupancy maps—which areas of the environment are occupied in general.

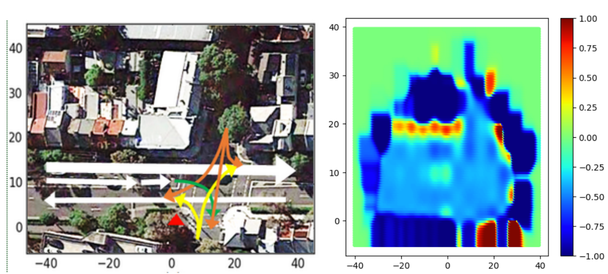


Figure 5. The satellite map of dataset 2 on left and its BHM on right. The arrows indicate the traffic flow of the busy four-way intersection.

In Figure 6, BHM is compared against other methods. In both datasets, the AUC of BHM is comparable to VSDGPOM at the beginning and AUC becomes slightly higher as it receives more data. The NLL measure of BHM is almost always slightly lower than that of VSDGPOM (the lower the NLL, the better the model is) while significantly lower than DGrid. Although DGPOM and DGPOM50% (GPOM with half of data removed randomly for speeding up) seems to have a better NLL, they cannot be run for more than approximately 50 time steps because of the unwieldy growing computational time which is also clear from the bottom row

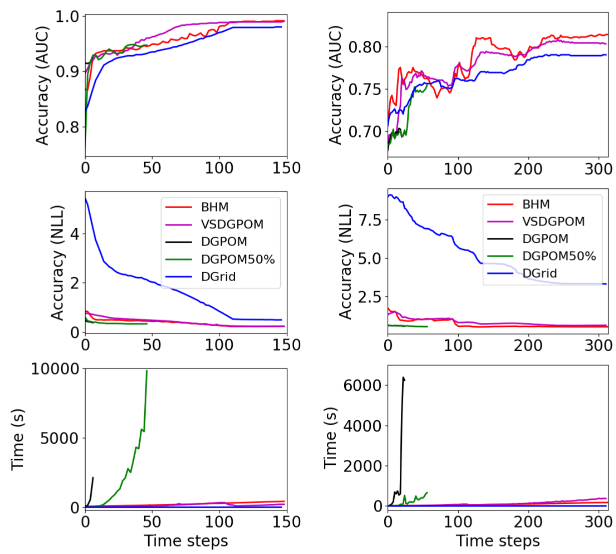


Figure 6. Left column: Dataset 1, Right column: Dataset 2. The first two rows show accuracy metrics while the bottom row shows the time performance. In general, BHMs have a significantly higher accuracy over grid based methods and have a slightly better accuracy over Gaussian process based methods.

of Figure 6. In summary, BHMs are faster, yet have a similar accuracy to that of Gaussian process based methods.

5. Future Work

The current python implementation of the algorithm is a crude version programmed merely for pilot experiments and for demonstrating the feasibility of the method. Taking a different approach, sequential learning can be done by feedbacking the posterior in the current step into the prior in the next step. Additionally, we intend to learn the length scale of the kernel to satisfy a given accuracy criterion in a more theoretically sound manner. For instance, recall that we observed in Section 4.2 that smaller length scales are desirable close to edges. Therefore, learning these local length-scale in a more principled approach is certainly valuable.

6. Conclusions

We extended the Hilbert maps algorithm for mapping long-term dynamics. Furthermore, we eliminated some vital parameter tuning in conventional Hilbert maps, making it further easier to use. Additionally, we demonstrated that the maps are less susceptible to occlusions as they consider neighborhood information. These inherent properties in Bayesian Hilbert maps as well as the main workhorse of the algorithm—kernels—can be use for probabilistic path planing in autonomous vehicles.

References

Arbuckle, Daniel, Howard, Andrew, and Mataric, Maja. Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pp. 409–414, 2002.

Bishop, Christopher M. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

Elfes, Alberto. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.

Francis, Gilad, Ott, Lionel, and Ramos, Fabio. Stochastic functional gradient for motion planning in continuous occupancy maps. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Hahnel, Dirk, Triebel, Rudolph, Burgard, Wolfram, and Thrun, Sebastian. Map building with mobile robots in dynamic environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4270–4275, 2003.

Jaakkola, Tommi and Jordan, Michael. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, volume 82, 1997.

Krajnık, Tomas, Fentanes, Jaime Pulida, Cielniak, Grzegorz, Dondrup, Cristian, and Duckett, Tom. Spectral analysis for long-term robotic mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3706–3711, 2014.

Marinho, Zita, Boots, Byron, Dragan, Anca, Byravan, Arunkumar, Gordon, Geoffrey J, and Srinivasa, Siddhartha. Functional gradient motion planning in reproducing kernel hilbert spaces. In *Proceedings of Robotics: Science and Systems (RSS)*, 2016.

Meyer-Delius, D., Beinhofer, M., and Burgard, W. Occupancy Grid Models for Robot Mapping in Changing Environments. In *proc. AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2024–2030, 2012.

Mitsou, N. C and Tzafestas, C. Temporal Occupancy Grid for mobile robot dynamic environment mapping. In *Mediterranean Conference on Control & Automation (MED)*, pp. 1–8, 2007.

Norouzii, Mohammad, Miro, Jaime Valls, and Dissanayake, Gamini. Probabilistic stable motion planning with stability uncertainty for articulated vehicles on challenging terrains. *Autonomous Robots*, 40(2):361–381, 2016.

O'Callaghan, Simon T. and Ramos, Fabio T. Gaussian process occupancy maps. *The International Journal of Robotics Research (IJRR)*, 31(1):42–62, 2012.

O'Callaghan, Simon T. and Ramos, Fabio T. Gaussian process occupancy maps for dynamic environments. In *The 14th International Symposium on Experimental Robotics (ISER)*, Marrakesh, Morocco, June 2014.

Ramos, Fabio and Ott, Lionel. Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent. In *Proceedings of Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015. doi: 10.15607/RSS.2015.XI.002.

Saarinen, J., Andreasson, H., and Lilienthal, A. Independent Markov Chain Occupancy Grid Maps for Representation of Dynamic Environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3489–3495, 2012.

Scheding, Steven, Dissanayake, Gamini, Nebot, Eduardo Mario, and Durrant-Whyte, Hugh. An experiment in autonomous navigation of an underground mining vehicle. *IEEE Transactions on Robotics and Automation*, 15(1):85–95, 1999.

Senanayake, Ransalu, Ott, Lionel, O'Callaghan, Simon, and Ramos, Fabio. Spatio-temporal hilbert maps for continuous occupancy representation in dynamic environments. In *Neural Information Processing Systems (NIPS)*, 2016.

Senanayake, Ransalu, O'Callaghan, Simon, and Ramos, Fabio. Learning highly dynamic environments with stochastic variational inference. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Walcott, Aisha. *Long-term robot mapping in dynamic environments*. PhD thesis, Massachusetts Institute of Technology, 2011.

Wang, Jinkun and Englot, Brendan. Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1003–1010, 2016.

Wang, Zhan, Jensfelt, Patric, and Folkesson, John. Modeling spatial-temporal dynamics of human movements for predicting future trajectories. In *AAAI Conference on Artificial Intelligence*, pp. 42–48, 2015.