Gray-box probabilistic occupancy mapping

Anonymous Author(s)
Affiliation
Address
email

Abstract

In order to deploy robots in previously unseen and unstructured environments, the robots should have the capacity to learn on their own and adapt to the changes in the environments. For instance, in mobile robotics, a robot should be able to learn a map of the environment from data itself without the intervention of a human to tune the parameters of the model. To this end, leveraging the latest developments in automatic machine learning (AutoML) and probabilistic programming, under the Hilbert mapping framework which can represent the occupancy of the environment as a continuous function of locations, we formulate a Bayesian framework to learn all parameters of the map. Crucially, this way, the robot is capable of learning the optimal shapes and placement of the kernels in Hilbert maps by merely embedding high-level human knowledge of the problem by means of prior probability distributions. A direct consequence of this is the ability to enable improved risk management through more robust perception and planning in complex environments. Experiments conducted on simulated and real-world datasets demonstrate the importance of incorporating prior information.

1 Introduction

Modeling the environment a robot operates in is fundamental to safer decision-making such as path planning in safety and infrastructure critical situations. To this end, discerning occupied areas from unoccupied areas of the environment using depth measurements is required. Typically, occupancy states exhibit highly nonlinear and spatially correlated patterns that cannot be captured with a simple linear classification model. Furthermore, because it is required to learn the occupancy level using very few sparse sensor measurements in a reasonable time, kernel methods have been the *de jure* choice in recent occupancy mapping [1, 2].

One of the major challenges in employing kernel methods in occupancy mapping is the requirement of choosing parameters and hyperparameters of the model [3]. In order for mobile robots to maneuver fully autonomously in unknown environments or to interact with humans and other agents, the robots should have the capability to automatically learn their model parameters from data. Only the most simple environments contain spatially homogenous features, however this is typically not the case in real-world mapping - e.g. walls and furniture may contribute to sharp features while open spaces and large hills may contribute to spatially smooth features. To better understand the significance of representing nonstationarity in terms of kernels, first consider the SE kernel which is parametrized with lengthscale and position hyperparameters. As seen in Figure 1, with large lengthscales it is possible to capture smoother changes across the space, while small lengthscales allow one to capture sharp changes in the space. Hyperparameter optimization is critical for almost all machine learning methods and the best values are almost always dependent on the dataset. Often,

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

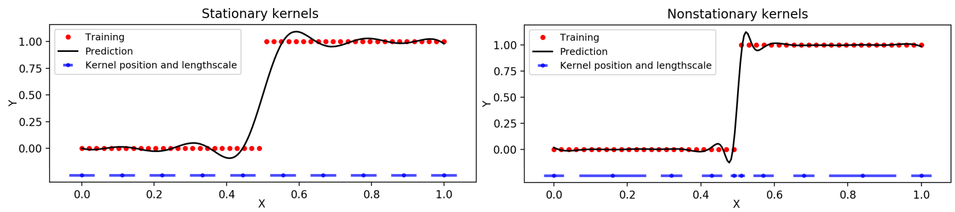

Figure 1: Comparison of stationary and nonstationary kernels, $\exp(-\|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2/2l^2)$ with bivariate Gaussian distributions $\tilde{\mathbf{x}}$ hinged on the environment and lengthscales l, and their ability to represent sharp spatial changes. Note that both examples have the same number of kernels, however in the non-stationary case the kernels have different positions and lengthscales to account for abrupt changes in the training data.

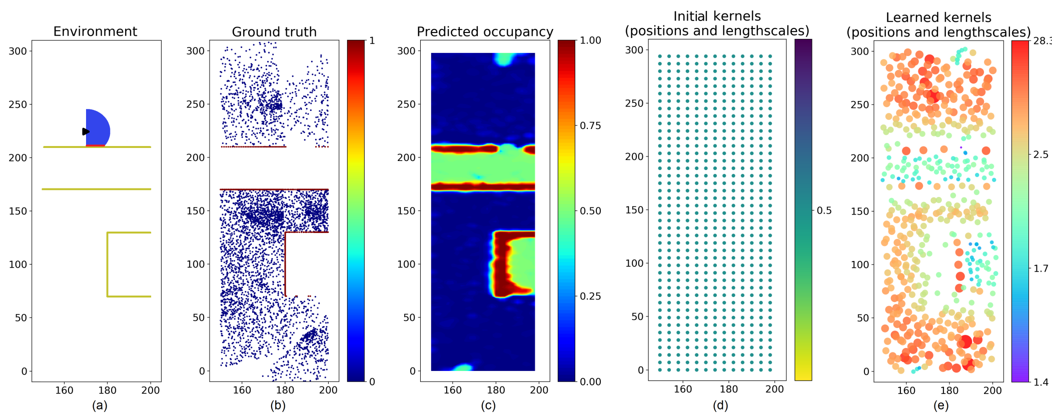

Figure 2: (a) A 50×300 m section of a simulated environment with obstacles in yellow. A robot shown as a black arrow has a lidar with beams shown in blue and the laser hit points in red. (b) The robot moves around and collects data. Red points are laser hit points and blue points are samples taken from lidar beams between the robot and laser hit points (c) The occupancy probability map. Red indicates occupied space, blue indicates free space, and colors in-between indicates the uncertainty of occupancy. (d) The map is built based on a set of squared-exponential kernels. The mean of the initial bivariate Gaussians is shown here—Gaussians are in a grid. (e) The proposed algorithm can learn both kernel parameters l and positions $\tilde{\mathbf{x}}$ alongside other model parameters. Both the color and the size of the marker indicates the size of the learned lengthscales. For instance, larger lengthscales are shown in a bigger marker size and in red.

a single best lengthscale is chosen that performs, on average, the best for the entire dataset. In our contribution, we address where to place kernels and what lengthscales they should have.

Another important aspect that should be taken into account when designing robot models is the uncertainty inherent to all levels of all interacting systems—from sensor and actuator imperfections to model misspecifications. Another incentive to use Bayesian models is that they provide an interface to incorporate high-level human knowledge about the system into the model through prior probability distributions. Furthermore, they inherently allow us to capture uncertainty about the perceived world as well as uncertainty about the model parameters themselves. This is an essential part of developing more *explainable* models which are particularly relevant to the future of intelligent systems. Such an approach where data, alongside prior knowledge or structure, are injected into the model is known as *gray-box modeling* [4].

In this paper, we use stochastic gradient descent with the reparameterization trick [5] to solve a challenging learning problem of automatically determining all parameters for Hilbert maps - which have traditionally used human-designed kernel hyperparameters. We demonstrate the importance of using more involved Bayesian formulations for uncertainty representation and learning thousands of parameters (Figure 2) in both small and bigdata settings without laborious mathematical derivations.

2 Background

2.1 Bayesian Hilbert maps

With the advancement of depth sensors such as lidar and sonar, occupancy grid maps (OGM) developed in 1980s [6] became a popular choice for representing the environment. To alleviate the disadvantages of OGMs, Gaussian process occupancy maps (GPOMs) [7] were proposed. Eliminating the cubic run-time complexity in GPOMs, Hilbert maps (HMs) [1] and Bayesian Hilbert maps (BHMs) were proposed [8]. In BHMs, the map is learned on a reproducing kernel Hilbert space (RKHS) where kernel functions are used to characterize spatial relationships. A kernel $k(\mathbf{x}, \tilde{\mathbf{x}}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a function that measures the similarity between two multidimensional inputs $\mathbf{x}$, $\tilde{\mathbf{x}} \in \mathcal{X} \subset \mathbb{R}^2$. The pairwise similarities between the elements of the two sets of points $\{\mathbf{x}_n \in \mathbb{R}^2\}_{n=1}^N$ and $\{\tilde{\mathbf{x}}_m \in \mathbb{R}^2\}_{m=1}^M$ are computed. Here, $\mathbf{x}$ are longitude-latitude locations of either free or occupied $y \in \{0, 1\} = \{\text{free}, \text{occupied}\}$ data points sampled from lidar beams and $\tilde{\mathbf{x}}$ are points *hinged* on pre-defined locations of the space. A SE kernel $k(\mathbf{x}_n, \tilde{\mathbf{x}}_m; l) = \exp\big(-\|\mathbf{x}_n - \tilde{\mathbf{x}}_m\|_2^2/2l^2\big)$ with a heuristically determined lengthscale l is used to compute the the feature vector $\phi(\mathbf{x}_n; l) = (k(\mathbf{x}_n, \tilde{\mathbf{x}}_1; l), k(\mathbf{x}_n, \tilde{\mathbf{x}}_2; l), ..., k(\mathbf{x}_n, \tilde{\mathbf{x}}_M; l)) \in \mathbb{R}^M$ for all data points $\{\mathbf{x}_n\}_{n=1}^N$. In this sense, $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ is the dataset and $\{l, \{\tilde{\mathbf{x}}_m\}_{m=1}^M\}$ is the pre-defined parameter set. Once the feature vector is computed, it passes through a sigmoidal function to estimate the occupancy level $\hat{y} = p(y|\mathbf{x}_*, \mathbf{w}) = 1/(1 + \exp(\mathbf{w}^\top \phi(\mathbf{x}_n; l)))$ of a query point in the space $\mathbf{x}_*$, given the weights $\mathbf{w} \sim \mathcal{N}$. As this query point can be any longitude-latitude pair, as opposed to OGMs, BHMs can produce maps with arbitrary resolution at prediction time. In BHMs, the lengthscales of the kernel l and where to place them $\tilde{\mathbf{x}}$ are prefixed values.

2.2 Kernel learning

Kernels methods are used in robotics especially when the objective is to learn nonlinear patterns with a small amount of data [9–12]. Although only SE kernels with fixed lengthscales are used in robotic mapping [1–3], different kernel learning techniques have been previously discussed in machine learning, especially in the Gaussian process literature. The selection of kernels is typically done through expert human knowledge [13], a model selection criteria such as Bayesian information criteria [14], or expensive optimization procedures [15]. Alternatively, it is possible to combine kernels as a sum or a product of kernels [13] or as representing them as a spectral mixture in the frequency domain [16]. However, unlike in Gaussian process where optimizing the hyperparameters is well-studied and readily available through the log marginal likelihood, directly learning parameters online in a classification setting is not straightforward in HMs.

3 Nonstationary kernels for Hilbert mapping

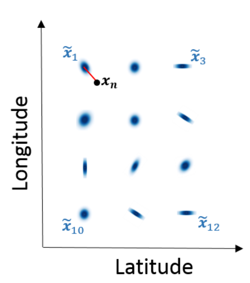

Figure 3: Feature vector computation. $\{\tilde{\mathbf{x}}\}_{m=1}^{M=12}$ are hinge distributions and $\mathbf{x}_n$ is the n^{th} data point.

Var.	Distributions
Prior distributions:	
w_m	$\mathcal{N}(\mu_m, \sigma_m^2)$
$\bar{l}_m$	$\mathcal{G}(\alpha_m, \beta_m)$ [1]
$\tilde{\mathbf{x}}_m$	$\mathcal{N}(\tilde{\mu}, \tilde{\Sigma}^2)$
Variational distributions:	
qw_m	$\mathcal{N}(\mu_m^{(q)}, \sigma_m^{2(q)})$
$q\bar{l}_m$	$\mathcal{LN}(\alpha_m^{(q)}, \beta_m^{(q)})$ [2]
$q\tilde{\mathbf{x}}_m$	$\mathcal{N}(\tilde{\mu}^{(q)}, \tilde{\sigma}^{2(q)})$

Table 1: Description of model parameters. Assuming independence, individual distributions are associated with all hinge points $m = 1...M$

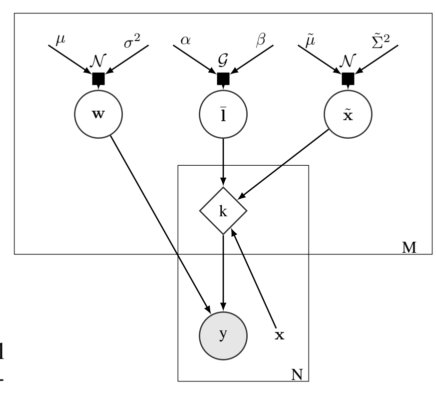

Figure 4: The graphical model. k represents the kernel which is evaluated $N \times M$ times.

In this section, we propose novel techniques for mapping unstructured environments without a human explicitly providing hyper-parameters. As the main contribution of this paper, we propose a

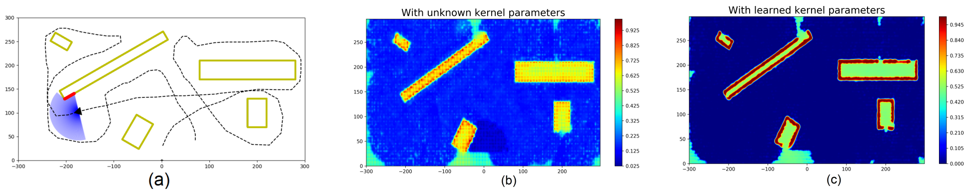

Figure 5: (a) Environment and entire dataset 1 (b) The averaged occupancy map of BHM with a random set of lengthscales (c) Predicted occupancy map using ABHM.

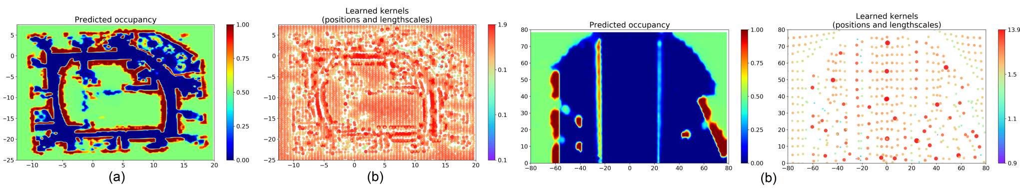

Figure 6: Predicted occupancy map and learned lengthscales. (a) Dataset 2 (b) Dataset 3

principled approach to learn weights, lengthscales, and positions of kernels. Individual lengthscales $\{l_m\}_{m=1}^{M}$ essentially model the nonstationary behavior and can easily acclimatize to local changes in the environment. To this end, in a sense of gray-box modeling, we start with possible locations for kernels as bivariate Gaussians and inverse length-scales as Gamma distributions, and then optimize them as the robot captures more data.

Since observed occupancy values are always binary and are independent of each other, we assume the likelihood follows a Bernoulli distribution $p(\mathbf{y}|\mathbf{x},\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})$ where $\log(\theta/(1-\theta))) = \mathbf{w}^\top\Phi(\mathbf{x};\mathbf{l},\tilde{\mathbf{x}})$. As shown in Figure 3, kernel functions are now implicitly evaluated between datapoints point and hinge distributions, naturally accounting for uncertainty. Our objective is to learn the posterior distribution. However, because of the Bernoulli likelihood, the posterior is intractable and hence is approximated using another distribution q. With the variables defined in Table 1, indicating longitude and latitude with lon and lat, respectively, the basic formulation with mean-field variational approximation is given in Figure 3 and the following equation,

$$\underbrace{\prod_{m=1}^{M} q(w_m)q(l_m^{\text{lon}})q(l_m^{\text{lat}})q(\tilde{\mathbf{x}}_m)}_{\text{factorized variational distribution}} = \underbrace{q(\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})}_{\substack{\text{variational}\\ \text{distribution}}} \approx \underbrace{p(\mathbf{w},\mathbf{l},\tilde{\mathbf{x}}|\mathbf{x},\mathbf{y})}_{\text{posterior}} \propto \underbrace{p(\mathbf{w})p(\mathbf{l})p(\tilde{\mathbf{x}})}_{\text{priors}}\underbrace{p(\mathbf{y}|\mathbf{x},\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})}_{\text{likelihood}}.$$

3.1 Experiments

We conducted experiments on four different datasets given in Table 2. These datasets contain both static and dynamic environments. As with [3, 8], our model will estimate the average long-term occupancy which is different to mapping short-term occupancy [17] or removing dynamics to build a static occupancy map [18, 19].

Table 2: Description of the datasets.

Dataset	Real	Dynamic	Description
1	✗	✗	A $600 \times 300\text{m}^2$ area [3]. This is a simple but large environment.
2	✓	✗	Intel lab dataset: a complex indoor environment.
3	✗	✓	Vehicles move in two directions and the robot sits in the middle [3].
4	✓	✓	Lidar dataset in a busy intersection [3].

This experiment was designed to validate the main contribution of the method—learning lengthscales and hinge locations. The learned environments for different datasets are shown in 6. To understand the full effect of the proposed model it is not enough to look at the predicted occupancy map—we must consider the underlying distributions. Figure 7 provides a visual map of the means and variances of a learned model's variational posteriors. Accounting for a large part of the upper and lower parts of the map, the position variance in Figure 7b shows that in areas of dense laser scans where no walls exist, a larger but uniform variance for each spatial dimension is learned. For

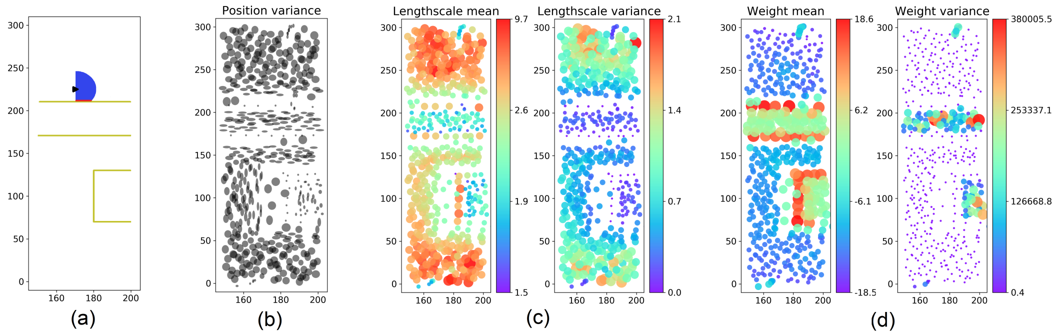

Figure 7: Uncertainty plots (a) A portion of the environment (b) Positions of hinge kernels $\tilde{\mathbf{x}}$ (c) Lengthscales (d) Weights

Table 3: Losses on all real datasets. The higher the area under curve (AUC) or the lower the mean negative log loss (MNLL), the better the model is.

Method	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	AUC	MNLL	AUC	MNLL	AUC	MNLL	AUC	MNLL
ABHM	**0.999**	**0.015**	**0.994**	**0.093**	**0.993**	**0.175**	**0.889**	**0.477**
BHM	**1.000**	0.176	0.921	0.362	0.990	0.280	0.825	0.570
HM	0.992	0.226	0.938	0.666	0.920	0.903	0.778	0.677
VSDGPOM	0.801	0.372	0.794	0.530	0.990	0.233	0.788	0.886
DOGM	0.792	0.593	0.901	0.744	0.980	0.495	0.779	3.449

the areas where the laser scanner has detected walls one observes a stark contrast exhibited by the smaller spatial variances. In the walled area spanning the middle of the map the learned variances in the latitudinal direction are stretched out further relative to the longitudinal direction reflecting the narrow corridor-like shape of the wall. Concerning now the lengthscale mean and variance in Figure 7c we can observe the most significant effect in terms of the learned posteriors. At the top and the bottom open areas the largest lengthscales are observed signifying a minimal complexity of occupancy. Paralleling the learned position variances, the learned lengthscale means are clustered around either areas of detail or areas of uncertain occupancy. This effect is repeated in the lengthscale variance.

The kernel weights means and variances are depicted in Figure 7d where one can see the highest weights appear around areas associated with the smallest position and lengthscale variances. Contrastingly, the most negative weights appear in regions of highly confident predicted empty occupancy. The weights closest to zero occur in areas of the map the robot has no visual perception and these constitute the insides of walls. The effect of the weight means is reflected in the weight variance where areas of high observability, which include open spaces and walls, have a low uncertainty in their estimates. Areas of low observability, i.e. inner parts of walls, have extremely high variances. This underlying analysis of the learned posterior distributions not only substantiates the motivation for spatially adaptive kernel learning, but also gives an explainable and intuitive understanding of what the model has learned which is often critically important for robotic tasks that interact with real-world environments.

Using all four datasets, the area under curve (AUC) and mean negative log loss (MNLL) were calculated. As reported in Table 3, these metrics were also calculated for occupancy grid maps with dynamic updates (DOGM), variational sparse dynamic Gaussian process occupancy maps (VSDGPOM) [8], HMs, and Bayesian Hilbert Maps with sequential updates (BHM). The best lengthscales for previous Hilbert mapping techniques were determined using five-fold cross validation. Even when compared with hand-crafted features, ABHM outperforms. This is because it models nonstationarity and can capture subtle changes. For dataset 1 which has straight boundaries, the AUC value of both BHM and ABHM are comparable. However, ABHM outperforms in complex datasets such as in dataset 2 and dynamic environments such as in datasets 3 and 4. This is because only ABHMs can adjust the position and shape of kernels to locally adapt to environments.

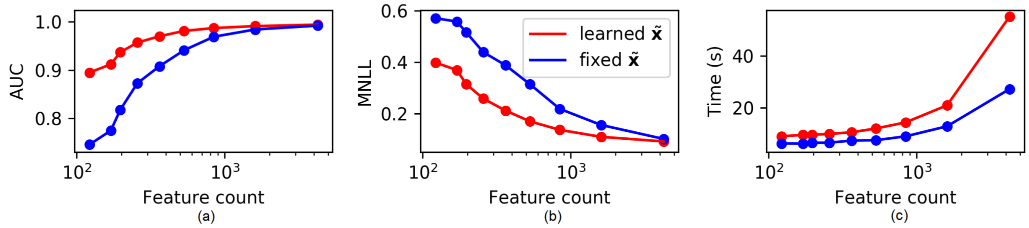

Figure 8: Performance vs. number of features for dataset 2. The blue lines show performance for fixed hinge positions while the red lines show the full ABHM model.

To further understand the relationship between the performance and number of hinge points, we analyzed the speed time and accuracy for dataset 2. We did this by, 1) learning both lengthscales and position, and 2) learning only the lengthscale keeping the kernels hinged on a grid. As shown in Figure 8, to achieve the same level of accuracy, only a smaller number of features is required when learning both the lengthscale and position.

Runtime: We conducted all experiments on a computer with a GTX1080 Ti 11 GB. For datasets 1 and 2, on average it takes around 10 minutes to learn all parameters. Note that this is to learn upwards of 57,600 parameters (8 parameters per hinge with more than 7200 hinges) and 300,000 data points. In contrast, [3] has an inevitable computational complexity $\mathcal{O}(M^3)$ while the proposed method uses stochastic gradient descent (SGD). Although analyzing the theoretical asymptotic complexity is not straightforward, it linearly increases with M and N empirically. In ABHM, we take the advantage of SGD to scalable for large datasets.

4 Conclusion

With the intention of building continuous occupancy maps without the human intervention, we devised methods to learn all parameters of the Hilbert maps. We also demonstrated the use of the latest AutoML techniques to learn complex models without relying on tedious mathematical derivations. Since kernel methods have also been successfully used in a variety of nonlinear path planning methods [9, 20, 21] we plan to extend these ideas to path planning so that mapping and path planning can be performed simultaneously in real-world in an end-to-end fashion under one framework.

References

[1] F. Ramos and L. Ott, "Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent," in *Proceedings of Robotics: Science and Systems (RSS)*, (Rome, Italy), July 2015.

[2] K. Doherty, J. Wang, and B. Englot, "Probabilistic map fusion for fast, incremental occupancy mapping with 3d hilbert maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[3] R. Senanayake and F. Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping," in *Conference on Robot Learning*, pp. 458–471, 2017.

[4] P. Galliani, A. Dezfouli, E. Bonilla, and N. Quadrianto, "Gray-box inference for structured gaussian process models," in *Artificial Intelligence and Statistics*, pp. 353–361, 2017.

[5] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

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

[7] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 1, pp. 42–62, 2012.

[8] R. Senanayake, S. O'Callaghan, and F. Ramos, "Learning highly dynamic environments with stochastic variational inference," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[9] M. Mukadam, X. Yan, and B. Boots, "Gaussian process motion planning," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 9–15, IEEE, 2016.

[10] M. Deisenroth and C. E. Rasmussen, “Pilco: A model-based and data-efficient approach to policy search,” in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

[11] H. A. Kingravi, H. R. Maske, and G. Chowdhary, “Kernel observers: systems-theoretic modeling and inference of spatiotemporally evolving processes,” in *Advances in Neural Information Processing Systems*, pp. 3990–3998, 2016.

[12] H. A. Kingravi, G. Chowdhary, P. A. Vela, and E. N. Johnson, “Reproducing kernel hilbert space approach for the online update of radial bases in neuro-adaptive control,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1130–1141, 2012.

[13] D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, “Structure discovery in nonparametric regression through compositional kernel search,” *arXiv preprint arXiv:1302.4922*, 2013.

[14] D. Duvenaud, *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014.

[15] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the smo algorithm,” in *Proceedings of the twenty-first international conference on Machine learning*, p. 6, ACM, 2004.

[16] A. G. Wilson, E. Gilboa, A. Nehorai, and J. P. Cunningham, “Gpatt: Fast multidimensional pattern extrapolation with gaussian processes,” *arXiv preprint arXiv:1310.5288*, 2013.

[17] R. Senanayake, L. Ott, S. O’Callaghan, and F. Ramos, “Spatio-temporal hilbert maps for continuous occupancy representation in dynamic environments,” in *Neural Information Processing Systems (NIPS)*, 2016.

[18] D. Meyer-Delius, M. Beinhofer, and W. Burgard, “Occupancy Grid Models for Robot Mapping in Changing Environments,” in *AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[19] C. Stachniss and W. Burgard, “Mobile robot mapping and localization in non-static environments,” in *aaai*, pp. 1324–1329, 2005.

[20] Z. Marinho, B. Boots, A. Dragan, A. Byravan, G. J. Gordon, and S. Srinivasa, “Functional gradient motion planning in reproducing kernel hilbert spaces,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2016.

[21] G. Vallicrosa Massaguer and P. Ridao Rodríguez, “H-slam: Rao-blackwellized particle filter slam using hilbert maps,” *Sensors (Switzerland), 2018, vol. 18, núm. 5, p. 1386*, 2018.