Bayesian Generalized Kernel Inference for Occupancy Map Prediction

Kevin Doherty, Jinkun Wang, and Brendan Englot

***Abstract*— We consider the problem of building accurate and descriptive 3D occupancy maps of an environment from sparse and noisy range sensor data. We seek to accomplish this task by constructing a predictive model online and inferring the occupancy probability of regions we have not directly observed. We propose a novel algorithm leveraging recent advances in data structures for mapping, sparse kernels, and Bayesian nonparametric inference. The resulting inference model has several desirable properties in comparison to existing methods, including speed of computation, the ability to be recursively updated without approximation, and consistency between batch and online inference. The method also reverts to the use of a specified prior state when insufficient relevant training data exist to predict the occupancy probability of a query point, a property which is attractive for motion planning and exploration applications with mobile robots.**

I. Introduction

When a range-sensing mobile robot's perceptual information is sparse and noise-corrupted, planning and decision-making may prove challenging unless the contents of the surrounding environment can be intelligently inferred. Standard occupancy grid mapping makes the conservative assumption that the states of all discrete cells in an occupancy map are independent, and an individual cell's occupancy depends only on sensor rays that pass directly through it [2], [9]. As a result, the method is ideally suited to situations in which a robot's sensor data is dense. A sparse, discontinuous occupancy map could potentially be restrictive during robot navigation. For example, gaps between occupied grid cells may be interpreted by a path planner as unoccupied map regions, but considering correlation between grid cells would reveal that the perceived gap is likely to be occupied. This problem has also been noted in the context of exploration using aerial vehicles with limited perceptual capabilities [19].

At present, several methods exist which infer the occupancy of unobserved map regions to mitigate the effects of sparsity and noise, and produce dense and predictive occupancy maps. In particular, Gaussian process (GP) occupancy mapping [12] has shown promising results due to its predictive capability. GP occupancy mapping demonstrated the use of Gaussian processes [16], a nonparametric Bayesian learning method, for probabilistic inference in unobserved regions of maps. The ability to express correlation between neighboring cells of an occupancy grid map makes GPs an attractive solution, however the $\mathcal{O}(N^3)$ time complexity of GP regression in training (for N training points) and $\mathcal{O}(N^2M)$ time complexity in testing (for M query points), has limited the ability of this method to scale to large datasets with real-time computation.

Fusion of several GPs in mapping with Bayesian Committee Machines (BCMs) [21] has been found to be useful for approximating GP occupancy maps [6]. Using a divide and conquer approach, a combination of nested BCMs, the extended block feature of "GPmap" [5], and test-data octrees has enabled real-time computation of GP occupancy maps [24]. The BCM, however, is known to be an approximate update to Gaussian process regression [21]. Successive application of approximate updates may lead to unreliable prediction in long-term mapping scenarios, and can lead to poor inference in unobserved areas of the map.

For this reason, we instead opt for a simpler model which is capable of exact recursive updates. Motivated by its recent success in applications spanning safe high-speed navigation [17] to state uncertainty estimation [13], we have chosen to apply the nonparametric Bayesian generalized kernel inference method in [23] in the context of mapping. Using this method, we are able to predict the occupancy probabilities and their corresponding variances for cells of the map not directly observed by a sensor. In contrast with previous work ([5], [24]), this method explicitly reverts to a selected prior when there is insufficient training data to draw conclusions about the occupancy state of a query cell.

Models in machine learning should perform well when queried with data that is similar to training data, but may offer poor or unpredictable performance for query points which are particularly dissimilar to training data. We desire a model which will recognize when there is insufficient training data near the query point to make an accurate prediction. Rather than attempt to make increasingly inaccurate predictions as query points become far from training data, we would like the model to smoothly transition to some prior with high variance, which may consist of a prediction representing an unknown state. This recognition is desirable in a variety of scenarios, but it is particularly is attractive for avoiding situations where we may aggressively predict the contents of unoccupied space, then plan a high-speed trajectory into areas which should instead be treated cautiously as unknown.

In summary, we provide an application of nonparametric Bayesian kernel inference to the mapping problem in order to relieve the independent cell assumption in occupancy grids, and combine this inference method with recent developments in sparse kernels and data structures for learning-aided mapping to achieve real-time viability, while retaining comparable inference accuracy to existing methods. In Section II, we present related work, outlining several tradeoffs between

K. Doherty is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology. J. Wang and B. Englot are with the Department of Mechanical Engineering, Stevens Institute of Technology, 1 Castle Point Terrace, Hoboken NJ 07030 USA {kdoherty, jwang92, benglot}@stevens.edu

existing map inference methods. In Section III, we present our application of a recent nonparametric Bayesian local kernel inference model to the occupancy mapping problem and discuss the properties of the model and the implications of those properties with respect to the mapping application. Section IV contains quantitative and qualitative evaluation of our method against other relevant methods.

II. Related Work

Several variants of GP occupancy mapping have been proposed to achieve reliable, computationally efficient inference in unobserved regions of occupancy maps. GPmap [5] partitions a map into "blocks" and "extended blocks", used in concert with sparse kernels to achieve offline inference with an overall complexity of $\mathcal{O}(\frac{N^3}{K^2} + \frac{N^2M}{K^2})$, where K is the number of blocks. This method was extended in [6] for online approximate updates using the BCM. By further segmenting grid blocks and making extensive use of BCM updates, the complexity of GP regression for occupancy mapping has been further reduced to $\mathcal{O}(\frac{N^3}{K^2E^2} + \frac{N^2M}{K^2E})$ (E represents additional training-data partitions) [24], with efficiency suitable for 3D real-time applications. This method, however, relies heavily on repeated approximate updates.

Other approaches have been proposed, including Hilbert mapping [15], which uses a logistic regression classifier in conjunction with approximate kernels to achieve comparable performance to standard GP occupancy mapping in less time. Since the logistic regression classifier can be trained with stochastic gradient descent, when combined with approximate kernel functions, training can be done in time linear in the size of the training data. In practice, however, for range data representing expansive environments, especially in 3D, the ability to achieve computational gains without significant loss of accuracy diminishes greatly. We have previously demonstrated a version of Hilbert maps which incrementally constructs a 3D map online [1], somewhat mitigating the difficulty in approximating kernel computation for large 3D environments, but this method provides only occupancy probability predictions without associated variances, which may provide deeper information about prediction uncertainty. This method also uses repeated approximate updates which may not be reliable for long-term mapping scenarios.

Our use of the generalized kernel inference model from [23] is motivated by its success in several other applications from safe high-speed navigation [17] to estimation of state uncertainty [13]. These applications emphasize the ability to apply prior information to the model as well as the model's predictable performance in scenarios where query data are dissimilar to training data. We are primarily interested in the model as a conservative estimator of occupancy which reverts to the prior occupancy probability for query points that lie sufficiently far from any training data.

III. Bayesian Generalized Kernel Occupancy Maps

In order to simplify the problem of maintaining a predictive distribution of occupancy states over potentially vast 3D maps, we make several assumptions. The proposed algorithm is intended for the case of *static* maps. In its current form, the algorithm presented is unable to support maps in which states other than that of the robot are allowed to vary. Furthermore, we assume that there is some distance l such that for all blocks, the occupancy state of a block is independent of the states of every block of distance l or further. This assumption is consistent with our choice of the sparse kernel presented in III-B. These assumptions allow modest improvements to computation time over existing methods, but more importantly allow us to perform exact inference and updates. One important consequence of these choices is that the model is unlikely to exhibit strong predictive performance when the data is prohibitively sparse. Instead, the proposed algorithm offers a more *conservative* approach to inference-based mapping in which, in absence of sufficient training data, we revert to some prior knowledge, rather than attempting to make predictions based on limited information.

A. Bayesian Nonparametric Inference

As in the standard formulation of occupancy grid mapping, a map cell m is occupied with probability $p(m = 1|x_*)$ and free with probability $p(m = 0|x_*) = 1 - p(m = 1|x_*)$ where x_* is the coordinate corresponding to the center of the grid cell. That is, occupancy is Bernoulli distributed, with parameter $\theta = p(m = 1|x_*)$. We seek to estimate the parameter θ of the cell centered at a query point x_*. To do so, we use the nonparametric Bayesian inference model for exponential families in [23]. With $\theta \sim Beta(\alpha, \beta)$, the predicted mean and variance of θ are as follows:

$$\mathbf{E}[\theta] = \frac{\alpha}{\alpha + \beta} \tag{1}$$

$$\mathbf{Var}[\theta] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}, \tag{2}$$

where α and β are hyperparameters. At each time step we obtain a range scan containing 3D points and their corresponding labels, i.e. occupancy probabilities constituting the training set $(x_i, y_i) \in \mathcal{D}_t$. Locations that the sensor hits are assigned occupancy probabilities of 1.0, and free-space points are interpolated along the sensor ray and assigned occupancy probabilities of 0.0. Given this training data, we apply exact recursive updates to the hyperparameters

$$\alpha_t = \alpha_{t-1} + \bar{y}_t \tag{3}$$

$$\beta_t = \beta_{t-1} + \bar{k}_t - \bar{y}_t, \tag{4}$$

in which $\bar{k}_t$ and $\bar{y}_t$ are the results of the following kernel computations:

$$\bar{y}_t = \sum_{i=1}^{N} k(x_i, x_*) y_i \tag{5}$$

$$\bar{k}_t = \sum_{i=1}^{N} k(x_i, x_*). \tag{6}$$

We can also solve for the log-odds represention of $\mathbf{E}[\theta]$, desirable for its numerical stability for probabilities near 0

and 1 [20], without directly computing $\mathbf{E}[\theta]$:

$$l_t = \log \frac{\alpha_t}{\beta_t}. \tag{7}$$

The recursive updates in Equations 3 and 4 rely only on the previous values of α and β and kernel computations based on the current scan. The additive nature of these updates suggests that we can generate equivalent maps offline in batch or incrementally while traversing the map. The quantities α_0 and β_0 represent prior pseudo-counts of the positive (occupied) and negative (free) classes respectively. We use α_0 and β_0 to apply a small uninformative prior over the possible values of θ, so that as the results of the kernel computation become very small, we gradually revert to the prior occupancy probability and variance. Due to the very small uninformative prior, when there is sufficient training data (i.e. $\bar{y} >> \alpha_0$ and $\bar{k} >> \alpha_0 + \beta_0$), $\mathbf{E}[\theta]$ approximates the Nadaraya-Watson estimator $\hat{m}(x_*)$ [10], [25]:

$$\hat{m}(x_*) = \frac{\sum_{i=1}^{N} k(x_i, x_*) y_i}{\sum_{i=1}^{N} k(x_i, x_*)}. \tag{8}$$

An extension of this method could leverage prior knowledge about the environment or sensor by making α_0 and β_0 functions of the query point x_*, as in [17].

We also take advantage of variance predictions in a similar fashion to previous work [24]. We use the following model of state for cells in the environment:

$$\text{state} = \begin{cases} \text{free}, & \text{if } p < p_{free}, \sigma^2 < \sigma^2_{th} \\ \text{occupied}, & \text{if } p > p_{occ}, \sigma^2 < \sigma^2_{th} \\ \text{unknown}, & \text{otherwise} \end{cases} \tag{9}$$

in which p corresponds to the occupancy probability, which in our case is the mean of the predictive distribution $\mathbf{E}[\theta]$, p_{free} is a threshold on the occupancy probability of cells deemed "free," and p_{occ} is a threshold on the occupancy probability of cells deemed "occupied". The variance σ^2, computed as $\mathbf{Var}[\theta]$, is thresholded by σ^2_{th} to filter out predictions with high variance as "unknown."

B. Sparse Kernel

Our choice of kernel will have important implications for the exactness of the update in Equations 3 and 4. We opt to use the sparse kernel presented in [8],

$$k(x, x') = \begin{cases} \sigma_0 \left[\frac{2+\cos(2\pi \frac{d}{l})}{3} (1 - \frac{d}{l}) + \frac{1}{2\pi} \sin(2\pi \frac{d}{l}) \right] & \text{if } d < l \\ 0 & \text{if } d \geq l \end{cases} \tag{10}$$

where $\sigma_0 > 0$ is a constant parameter of the kernel, $l > 0$ is the scale, and d is the distance between x and x'. By opting for a sparse kernel, we can efficiently and exactly compute $\bar{y}$ and $\bar{k}$ in $\mathcal{O}(\log N)$ time using a k-d tree. Simply by querying a k-d tree containing the training points for a scan with radius l about each query point x_*, we obtain all training points with nonzero contribution to the kernel computation in Eq. 10.

The overall computational complexity of the inference method is $\mathcal{O}(M \log N)$, where M is the number of test points and N is the number of training points. For other kernels, such as the radial basis function (RBF) kernel or Matérn kernel, only approximate updates can be obtained without using all of the training data. Though we use this sparse kernel, any kernel with finite support is viable, such as a polynomial approximation to the RBF kernel or the product of the sparse kernel and the Matérn kernel used in [5].

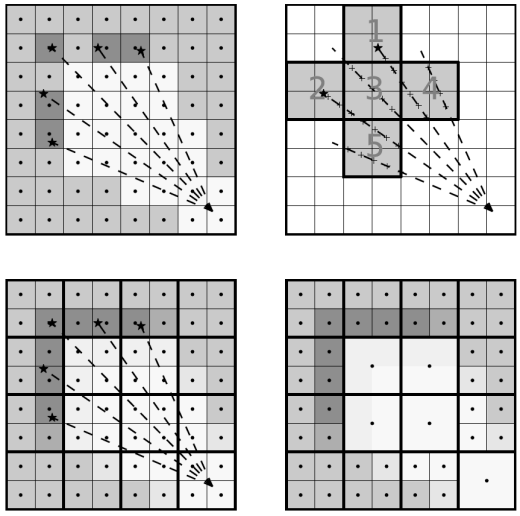


Fig. 1: A 2D illustration of the use of test-data octrees. The top-left image depicts standard occupancy grid mapping. In the top-right we show the setup for prediction of the occupancy probability of all cells in block 3. The extended block consists of all blocks within distance l, in this case the length of two grid cells, of block 3 that contain sensor data or sampled free-space points. For each block, the data from the corresponding extended block is aggregated and inference is performed, generating the image at bottom-left. Finally, at bottom-right, neighboring cells within a block with the same occupancy state are pruned. Image obtained from [24].

C. Test Data Octrees

We adopt the test-data octrees proposed in [24] with slight adaptations to the use of the "extended block". The method proposed suggests training several separate GP regressions for a group of query points. With the kernel inference method used, we need not explicitly train at all. When a ray is cast, we sample free-space points linearly along the ray at a fixed resolution, then aggregate all "free" and "hit" point data from the extended block (i.e. all blocks within distance l of a center block) to update the predictions at the query points in the center block. Figure 1 shows a 2D illustration of the algorithm, as well as a depiction of standard occupancy grid mapping for comparison. By considering all blocks within distance l of the query block to comprise the extended block, we are able to quickly retrieve all points with nonzero contribution to the occupancy probability of the cells in the query block, which maintains the exactness of the inference.

Test-data octrees enable dynamic allocation as a robot explores, avoiding the need for large, finely-discretized grids to be initialized. Test-data octrees are pruned to lower resolution

Dataset	Dimensions (m)	Scans	Points/Scan	Sampled Points/Scan	Method	Avg. Time/Scan (s)	Time (s)
Structured Simulation	$10.0 \times 7.0 \times 2.0$	12	3500	1506	BGKOctoMap	**0.021**	**0.25**
					GPOctoMap-NBCM-P	0.091	1.1
					OctoMap	0.027	0.32
Unstructured Simulation	$10.0 \times 7.0 \times 2.0$	12	3500	1506	BGKOctoMap	0.019	0.23
					GPOctoMap-NBCM-P	0.075	0.90
					OctoMap	**0.018**	**0.22**
Freiburg Corridor FR-079	$43.8 \times 18.2 \times 3.3$	66	89445	7601	BGKOctoMap	**0.15**	**9.6**
					GPOctoMap-NBCM-P	0.28	18.4
					OctoMap	0.15	10.1

TABLE I: Computation times for the three maps used in testing.

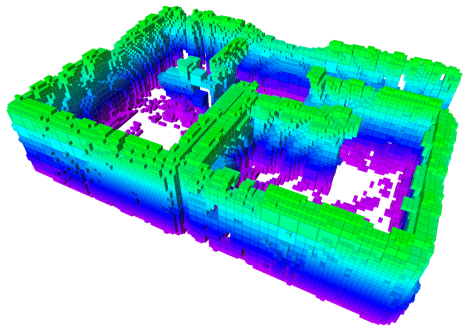


Fig. 2: Pruning test data octrees in a map inferred using Bayesian generalized kernel inference. The octrees have depth three; pruned blocks are especially prevalent along the bottom wall, where high-resolution cells have been merged into lower-resolution cells. Figure 3 shows maps of the same environment without pruning. The map is colored by height.

when all of the children of a particular node achieve the same state [24]. In such a situation, all of the children are removed, and the parent remains. This allows us to refine the number of query points needed for areas of space that are likely to be highly correlated, further decreasing the time needed for computation. Pruning also has the benefit of reducing the map's memory consumption. We show the effects of pruning in 3D on a map generated from simulated data in Figure 2.

IV. Computational Results

We evaluated the inference method on two synthetic datasets representing "structured" and "unstructured" environments, as well as the corridor dataset from the University of Freiburg [22]. The mapping algorithm is our own C++ implementation[1], and we compare our method quantitatively to GPOctoMap with test-data octrees [24]. We also compare our method to OctoMap [3], which provides an efficient multi-resolution occupancy grid. We use the Robot Operating System (ROS) [14] as well as the Point Cloud Library (PCL) [18] in our tests. The synthetic examples were made using the Gazebo simulator [7]. We apply the parameters $\alpha_0 = \beta_0 = 0.001$ to enforce a weak uninformative prior on grid cells, and the remaining parameters $\sigma_0 = 10.0$ and $l = 0.3$ were hand-tuned on one synthetic dataset and applied consistently throughout. Generally the desired locality of inference will depend on the resolution of the range sensor being used. Free space samples are taken linearly along each ray at 0.5m resolution. All computations were performed on an HP EliteBook 8570w with a 2.40 GHz Intel i7 CPU.

In this section, we refer to the GPOctoMap implementation with nested BCM updates as GPOctoMap-NBCM and GPOctoMap-NBCM-P when we apply pruning [24], while we refer to our method as Bayesian generalized kernel OctoMap or BGKOctoMap. We show that our approach offers comparable performance to GPOctoMap-NBCM-P when there is sufficient data. A comparison of computation time is given in Table I. Generally the proposed method achieves map inference in time comparable to OctoMap, which suggests applicability to real-time tasks. In maps with sparser coverage, such as the "unstructured" map, our more conservative method exhibits decreased predictive performance, since more of the query points are far from training data. We additionally provide an experimental demonstration of a robot performing station-keeping in the simulated "structured" environment while scanning one region repetitively. We show that our method is reliable over many scans of the same area, whereas GPOctoMap predictions gradually become overly aggressive, over-predicting occupancy.

A. Simulated Data

The simulated environments each span $10.0 \times 7.0 \times 2.0$ meters. The structured simulation used is significantly more open than the unstructured simulation, allowing for more complete sensor coverage. Qualitatively, we observe in Figure 3 that BGKOctoMap fills in several walls where OctoMap does not, drawing useful conclusions about regions such as the far corner. In Figure 4 we again show that BGKOctoMap closes many of the gaps in the OctoMap. The receiver operating characteristic (ROC) curves for the structured map in Figure 5 show comparable performance between BGKOctoMap and GPOctoMap when even though data is sparse, overall coverage of the map is good. On the other hand, the ROC curve for the "unstructured" map in Figure 6 shows that the limited sensor coverage affects the performance of our method, while GPOctoMap-NBCM-P performs well in both cases. The ROC curves plot the true positive rate versus the false positive rate. Here we use the predicted occupancy probabilities and compare to ground truth occupancy probabilities of 1.0 for an occupied cell

[1]The code for this paper is implemented in LA3DM, a C++ library for learning-aided 3D mapping, available at `https://github.com/RobustFieldAutonomyLab/la3dm`.

and 0.0 for an unoccupied cell, so that the comparison of inference accuracy is independent of our choice of thresholds for the state model in Equation 9. Along the curve, the probability threshold we use to choose the positive or negative class varies from 1.0 to 0.0. This can be seen as a plot of predictive performance as we change the occupancy probability threhsold. The area under the curve (AUC) is also provided in each case for comparison of inference accuracy.

In the case of the "unstructured" map, BGKOctoMap often reverts to a prior occupancy probability of 0.5 with high variance in regions where data is sparse. In some ways this is a desirable attribute of the model, since it captures the uncertainty inherent in prediction with limited data. On the other hand, GPOctoMap is capable of producing accurate predictions even when training data is far from query points. In both cases, GPOctoMap and OctoMap achieve better coverage of the floor than BGKOctoMap. This is an artifact of the 2.5D nature of the simulated environment coupled with our naiïve representation of free space used for inference. Interpolation of many free-space samples along the sensor ray increases support for the free-space class in the more open regions of the environment, causing many areas of the floor to be misclassified as free or unknown, not passing the occupancy and variance thresholds in Equation 9.

B. Real Data

The Freiburg corridor pointcloud dataset [22] has dimensions $43.8 \times 18.2 \times 3.3$ m. It represents a more expansive environment than the simulated data, and accordingly, the data requires more computation time. Since the point cloud is dense and we are primarily concerned with the application of our algorithms to sparse data, we down-sample the 89445 points per scan on average to a resolution of 0.1m, amounting to 7601 points per scan, which provides an artificially sparsified dataset. In this case, with substantially more data spanning a large environment, BGKOctoMap continues to perform comparably to OctoMap in computation time required. The map produced by BGKOctoMap from the Freiburg corridor pointclouds is shown in Figure 7.

C. Comparison of Long-Term Behavior

Here we demonstrate the stable long-term performance of the proposed mapping algorithm, one of its most desirable and useful features. We provide a station keeping scenario in which a robot repeatedly scans a single location. Using the structured environment simulation, we repeatedly input the same point cloud to both BGKOctoMap and GPOctoMap-NBCM. The effects of this demonstration are provided in Figure 8, where we show the output of each method after 1, 15, 30, and 60 scans. We observe that while our method does experience some slight changes due to the contributions from the new data (particularly in areas where $\alpha_0 + \beta_0 \approx \bar{k}$ after one scan), the change is mild in comparison to that of GPOctoMap. Repeated application of the BCM update approximations cause GPOctoMap to gradually predict that the walls and floor of the map are thicker, even though we update it with the same point cloud.

The BCM update for Gaussian process regression does not perform well in this scenario because of the approximation:

$$p(\mathcal{D}_i|\mathcal{D}_{i-1}, f_q) \approx p(\mathcal{D}_i|f_q) \tag{11}$$

where $\mathcal{D}_i$ is comprised of a single set of training points and corresponding outputs. In our case this is a single range scan with "occupied" hit points and "free" points interpolated along the sensor ray. The vector f_q consists of the unknown response variables corresponding to a set of query inputs. The assumption made in the BCM update is that the two datasets used to train seperate GP regression models are conditionally independent given the response variables to the query input. The assumption is violated in this experiment; our repeated observations create a situation that generates highly correlated observations. The consequences of this are relevant, since it is often a priority in simultaneous localization and mapping (SLAM) scenarios to seek out such correlated observations for loop closures. With the BCM update, those observations would produce errors in the map.

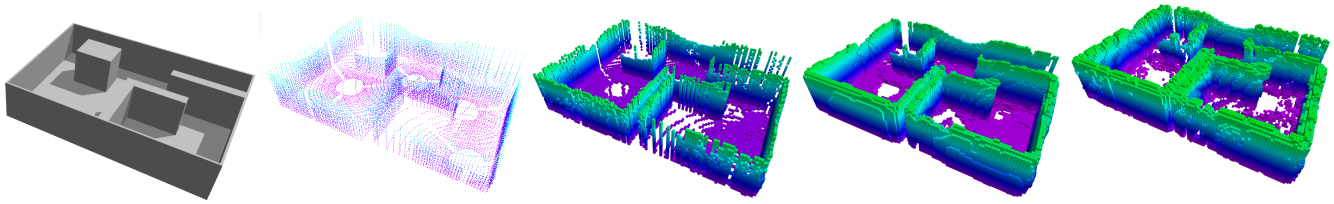
V. Conclusions and Future Work

We have proposed a novel inference-based occupancy mapping algorithm, BGKOctoMap, that leverages Bayesian kernel inference, sparse kernels, and test-data octrees. We have demonstrated that the method provides accurate predictions in areas where there is sufficient training data, and smoothly transitions to a prior occupancy probability with high variance when there is not. The method also shows promise for applications where multiple scans of the same areas are captured, since it integrates new data more reliably than existing inference-based mapping methods, and updates can be performed exactly. For the same reason, this method is also useful in situations where we may want to update a single map exactly over multiple sessions.

We have implicitly assumed certainty of pose knowledge in these experiments. This assumption can be relaxed somewhat by incorporating the use of the "expected kernel" used by Jadidi et al. [4] and formulated independently using reproducing kernel Hilbert spaces by Ramos and Ott [15]. Extension of this method with the "expected kernel" would enable the use of this inference technique in mapping scenarios where there is known pose uncertainty.

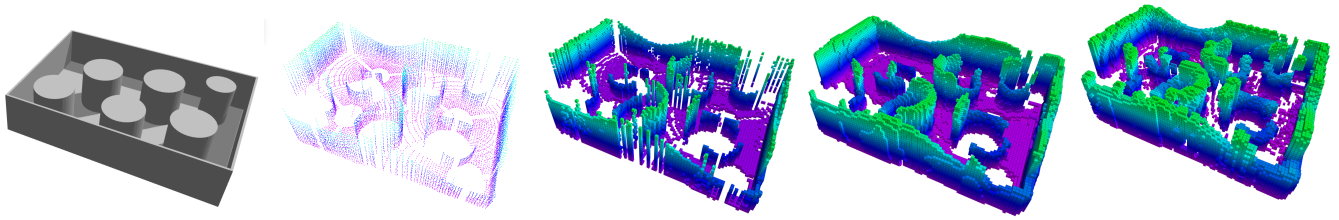
We do not make full use of all of the capabilities of the nonparametric Bayesian inference model used. It is possible with this method to incorporate more informed prior knowledge in a principled way by making α and β functions of the query point, as in [17]. Here we simply use a small uninformative prior to maintain an "unknown" state ($p = 0.5$) in areas with very little training data. The use of the sparse Matérn kernel in [5] may also benefit this method, since exact inference could still be performed, but the ability of the Matérn kernel to capture sharp changes in occupancy probability could improve predictive performance.

Finally, an important issue we wish to explore in greater detail is that of free-space sampling. In this work we sample linearly along sensor rays at a fixed resolution. Clearly, this can have significant effects on the nonparametric inference



(a) The Gazebo model (b) Raw sensor data (c) OctoMap (d) GPOctoMap (e) Our method, BGKOctoMap

Fig. 3: Structured environment simulation, with (a) simulated environment in Gazebo, (b) simulated raw sensor data, (c) a standard OctoMap, (d) map produced by prior method GPOctoMap, and (e) the result of the proposed method, BGKOctoMap.



(a) The Gazebo model (b) Raw sensor data (c) OctoMap (d) GPOctoMap (e) Our method, BGKOctoMap

Fig. 4: Unstructured environment simulation, with (a) simulated environment in Gazebo, (b) simulated raw sensor data, (c) a standard OctoMap, (d) map produced by prior method GPOctoMap, and (e) the result of the proposed method, BGKOctoMap.

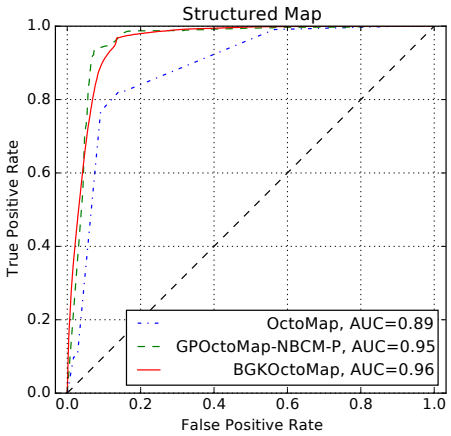


Fig. 5: Receiver operating characteristic curves for the 3 evaluated methods on the Structured Environment map.

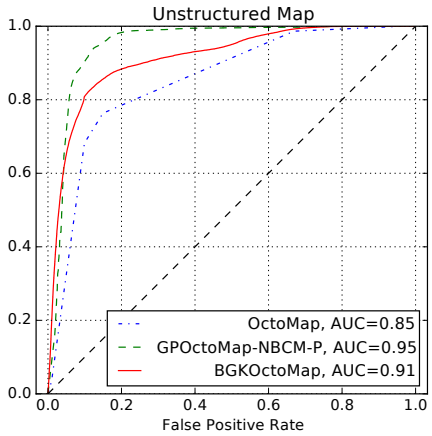


Fig. 6: Receiver operating characteristic curves for the 3 methods applied to the Unstructured Environment map.

model we use, since we can skew the model toward low occupancy probabilities by sampling free-space points very densely along a range beam. Other works have spanned a wide spectrum of parameterization choices, from representing a sensor ray with the single free point on the beam closest to the test point of interest [12] to employing free-space output values weighed by the entire continuous length of a range beam [11]. Further study of the impact of free-space sampling methodologies on the performance and accuracy of BGKOctoMap is a priority of future work.

Acknowledgments

We thank Prof. Philippos Mordohai for valuable feedback that has improved the quality of this manuscript. This research was supported in part by the National Science Foundation, grant number IIS-1551391.

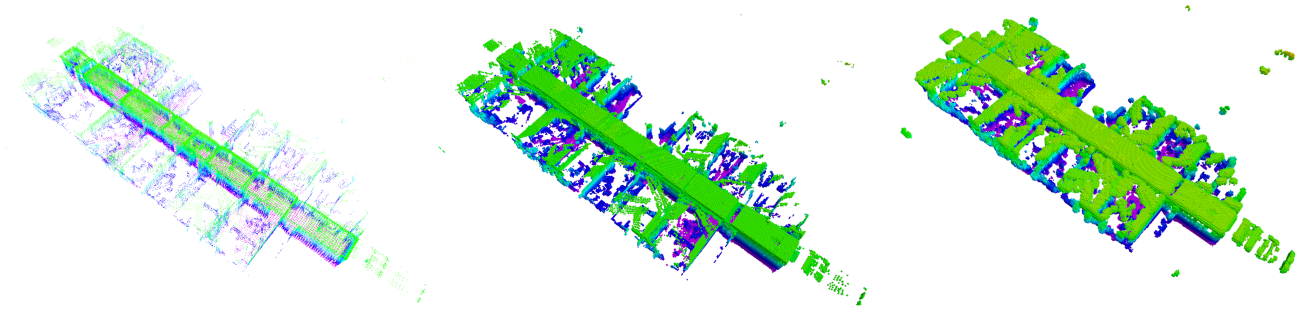
References

[1] K. Doherty, J. Wang and B. Englot, "Probabilistic map fusion for fast, incremental occupancy mapping with 3D Hilbert maps," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1011-1018, 2016.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation." *Computer*, vol. 22(6), pp. 46-57, 1989.

[3] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees." *Autonomous Robots*, vol. 34(3), pp. 189-206, 2013.

[4] M. Ghaffari Jadidi, J. V. Miro and G. Dissanayake, "Warped Gaussian Processes Occupancy Mapping With Uncertain Inputs," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 680-687, April 2017.



(a) Raw sensor data (b) OctoMap (c) Our method, BGKOctoMap

Fig. 7: BGKOctoMap applied to the Freiburg FR-079 corridor dataset [22].

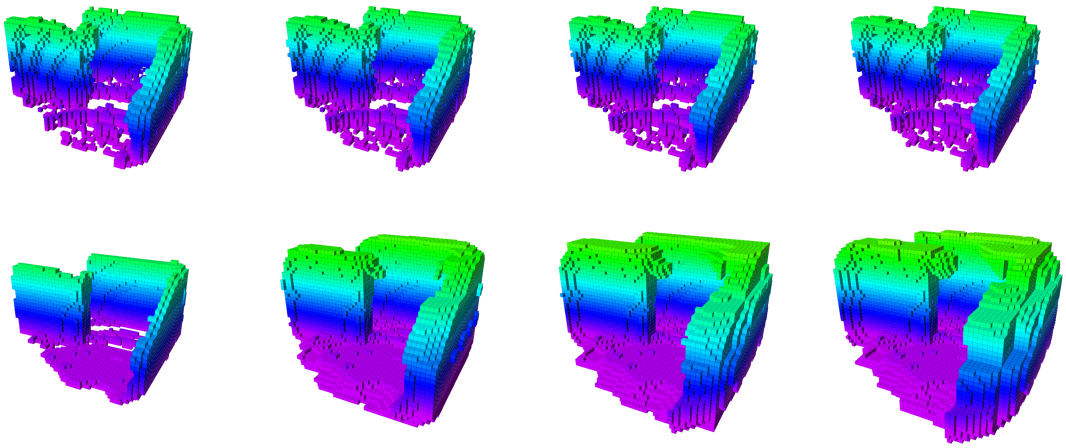


Fig. 8: In this simulated station-keeping demonstration, we show the results of updating both BGKOctoMap (Top) and GPOctoMap-NBCM (Bottom) after the introduction of 1, 15, 30, and 60 scans containing the same data (Left to Right).

[5] S. Kim and J. Kim, "GPmap: A unified framework for robotic mapping based on sparse Gaussian processes," *Proceedings of the 9th International Conference on Field and Service Robotics*, 2013.

[6] S. Kim and J. Kim,"Recursive Bayesian Updates for Occupancy Mapping and Surface Reconstruction," *Proceedings of the Australasian Conference on Robotics and Automation*, 2014.

[7] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2149-2154, 2004.

[8] A. Melkumyan and F. Ramos, "A Sparse Covariance Function for Exact Gaussian Process Inference in Large Datasets," *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*, vol. 9, pp. 1936-1942, 2009.

[9] H.P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 116-121, 1985.

[10] E.A. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9(1), pp. 141-142, 1964.

[11] S.T. O'Callaghan and F.T. Ramos, "Continuous occupancy mapping with integral kernels," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1494-1500, 2011.

[12] S.T. O'Callaghan and F.T. Ramos, "Gaussian process occupancy maps." *The International Journal of Robotics Research*, vol. 31(1), pp. 42-62, 2012.

[13] V. Peretroukhin, W. Vega-Brown, N. Roy, and J. Kelly, "PROBE-GK: Predictive robust estimation using generalized kernels," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 817-824, 2016.

[14] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," *ICRA workshop on open source software*, 2009.

[15] F. Ramos and L. Ott, "Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent" *Proceedings of Robotics: Science and Systems*, 2015.

[16] C.E. Rasmussen and C.K.I. Williams, *Gaussian Processes for Machine Learning*, Cambridge, MA: The MIT Press, 2006.

[17] C. Richter, W. Vega-Brown, and N. Roy, "Bayesian Learning for Safe High-Speed Navigation in Unknown Environments," *Proceedings of the 17th International Symposium on Robotics Research*, 2015.

[18] R.B. Rusu and S. Cousins, "3D is here: Point cloud library (pcl)," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011.

[19] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3D exploration with a micro-aerial vehicle." *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 9-15, 2012.

[20] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, Cambridge, MA: The MIT Press, 2005.

[21] V. Tresp, "A Bayesian Committee Machine, *Neural Computation*, vol. 12(11), pp. 2719-2741, 2000.

[22] University of Freiburg OctoMap 3D Scan Dataset http://ais.informatik.uni-freiburg.de/projects/datasets/octomap/

[23] W.R. Vega-Brown, M. Doniec, and N.G. Roy, "Nonparametric Bayesian inference on multivariate exponential families," *Advances in Neural Information Processing Systems*, pp. 2546-2554, 2014.

[24] J. Wang and B. Englot, "Fast, accurate Gaussian process occupancy maps via test-data octrees and nested Bayesian fusion," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1003-1010, 2016.

[25] G.S. Watson, "Smooth regression analysis," *Sankhya: The Indian Journal of Statistics, Series A*, vol. 26(4), pp. 359-372, 1964.