

Efficient, Multifidelity Perceptual Representations Via Hierarchical Gaussian Mixture Models

Shobhit Srivastava and Nathan Michael

Abstract—This paper presents a probabilistic environment representation that allows efficient high-fidelity modeling and inference toward enabling informed planning (active perception) on a computationally constrained mobile autonomous system. The proposed approach exploits the fact that real-world environments inherently possess structure that introduces dependencies between spatially distinct locations. Gaussian mixture models are employed to capture these structural dependencies and learn a semiparametric, arbitrary resolution spatial representation. A hierarchy of spatial models is proposed to enable a multifidelity representation with the variation in fidelity quantified via information-theoretic measures. Crucially for active perception, the proposed modeling approach enables a distribution over occupancy with an associated measure of uncertainty via incorporation of free space information. Evaluation of the proposed technique via a real-time graphics processing unit based implementation is presented on real-world data sets in diverse environments. The proposed approach is shown to perform favorably as compared to state-of-the-art occupancy mapping techniques in terms of memory footprint, prediction accuracy, and generalizability to structurally diverse environments.

Index Terms—Active perception, large-scale mapping, machine perception, multi-fidelity representation.

I. INTRODUCTION

A FUNDAMENTAL requirement for autonomous systems to plan and navigate in *a priori* unknown environments is the generation of a precise occupancy representation. A major challenge in the online generation of a high-fidelity representation is posed by the computational constraints on a mobile robot that include restricted processing power and limited memory. Other significant challenges include perceptual diversity in real-world environments [see Fig. 1(a) and (b)] in terms of scale, clutter, and type of structure (i.e., man-made versus natural) and varied sensor characteristics (i.e., dense versus sparse) [see Fig. 1(c) and (d)]. A computationally efficient perceptual modeling strategy is required to handle large environments and sparsity in sensor measurements. This paper presents an occupancy mapping approach that learns a multifidelity probabilistic representation based on a hierarchy of Gaussian mixtures. The procedure employed to learn the Gaussian mixtures allows

Manuscript received December 14, 2017; revised July 20, 2018; accepted September 24, 2018. Date of publication November 14, 2018; date of current version February 4, 2019. This paper was recommended for publication by Associate Editor D. Scaramuzza and Editor F. Chaumette upon evaluation of the reviewers' comments. This work was supported by the Westinghouse and ARL Grant W911NF-08-2-0004. (*Corresponding author: Shobhit Srivastava.*)

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: shobhits@cmu.edu; nmichael@cmu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2018.2878363

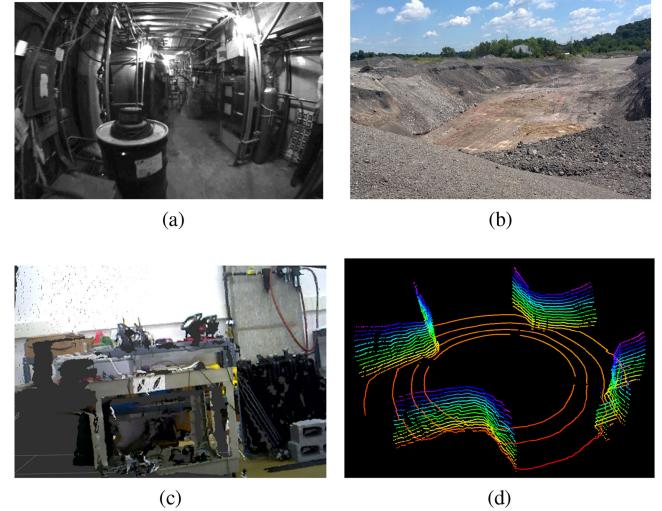


Fig. 1. Diversity in operating environments represented by a structured, and (a) cluttered power plant and (b) an unstructured open pit. (c) Variation in sensor characteristics represented by a dense RGBD point cloud and (d) a sparse LiDAR point cloud.

structural dependencies to be captured, enabling a compact, high fidelity, and robust occupancy representation. Additionally, informed planning in complex environments benefits from an uncertainty measure associated with model predictions [1], [2]. This is incorporated in the form of a variance estimate obtained from the occupancy distribution.

Occupancy grid maps, since their introduction by Elfes [3] and Moravec [4], have been widely used for occupancy representation throughout the mobile robotics community. These maps discretize the environment into cells of a predefined size, each classified as free or occupied. The likelihood of occupancy is stored in each cell and is updated via ray-tracing. The simplicity and computational efficiency associated with occupancy grid maps comes at the expense of certain restrictive assumptions that adversely affect model fidelity. Specifically, occupancy grids assume that the likelihood of cell occupancy depends only on the rays that pass through the cell and is independent of other measurements. This prevents the model from capturing spatial dependencies, resulting in incorrect classifications due to sparsity in sensor measurements. Also, a predefined cell size limits the generalizability of the model to environments exhibiting structural diversity. A high-fidelity occupancy grid for a complex environment requires a small cell size that in turn leads to a high memory footprint and increased vulnerability to measurement sparsity.

Several efforts have been made in literature to address problems associated with occupancy grids. Payeur *et al.* [5] proposed an approach based on octrees [6] to address memory concerns and achieve compactness via on-demand subdivision of cells. A similar approach was used by Fournier *et al.* [7] and Fairfield *et al.* [8] and extended by Hornung *et al.* [9], who incorporated online map compression. Octrees serve to reduce the memory footprint in environments with a varying degree of clutter. However, the leaf cell size of the tree needs to be specified as a parameter, which impacts representation fidelity in highly cluttered environments. Another approach to obtain relatively compact representation leverages elevation maps, where each cell of the two-dimensional (2-D) grid stores the height of the surface at that location. However, this approach has representation limitations due to a single height value per cell. Triebel *et al.* [10] extended this representation to incorporate multiple height values per cell. However, these approaches are vulnerable to discretization errors and make restrictive conditional independence assumptions.

Thrun [11] proposed an approach for map updates based on forward models to address the limitations of the conditional cell independence assumption. The approach employs forward sensor models to transform map updates into a latent-variable optimization, thus maintaining dependencies between cells. An unfortunate drawback of this approach is that a computationally expensive expectation maximization procedure [12] in a high-dimensional space, corresponding to the size of the maps, is required to realize map updates.

Another class of approaches leverages structural dependencies to learn low complexity surface models from sensor data. Thrun *et al.* [13] fitted a set of three-dimensional (3-D) planes to represent the environment, with the number of planes estimated via a Bayesian prior that penalizes complex maps. However, they make an assumption of planarity that may not always hold. Veeck and Veeck [14] fitted a model based on polylines as a continuous environment representation, with a set of heuristics to guide the optimization. However, the approach has been found to suffer from consistency issues [15]. The work of Veeck and Veeck was extended by Paskin and Thrun [16] to enable inference over occupancy via polygonal random fields. However, the computational cost of generating the map, as noted by the authors, is prohibitive for online operation.

The normal distribution transform (NDT) is a spatial representation proposed by Biber and Straber [17] for 2-D scan matching and extended to 3-D by Magnusson *et al.* [18]. This representation discretizes the environment into fixed size cells and fits a Gaussian distribution to each cell. The model is claimed to achieve a higher fidelity than occupancy grids at a coarser resolution. The representation was extended to an occupancy mapping framework (NDT-OM) by Saarinen *et al.* [19] via incorporation of a log-likelihood update for each cell. The representation assumes conditional cell independence that restricts the support of the Gaussian distributions and has been found to result in higher uncertainty at cell boundaries [20].

O’Callaghan and Ramos [21] employed Gaussian Processes (GPs) as a non-parametric Bayesian learning technique to model occupancy in the environment. The predictive mean and

variance distributions are passed through a sigmoid function to obtain the probability of occupancy at a location. The major drawback of this approach is the high memory footprint resulting from the need to cache the training data and the computational complexity, which grows cubically with the size of training data. Kim *et al.* [22] propose an occupancy mapping formulation leveraging sparse local GPs. The training data is partitioned into grid blocks whose size is determined by the characteristic length of the covariance function and the predictive mean and variance is estimated based on an extended block of data points. A strategy to incorporate incremental updates via Bayesian Committee Machines [23] has been proposed by Kim *et al.* [24] and extended by Wang *et al.* [25] to further reduce the computational complexity via test-data octrees. The scalability of GP based mapping to potentially large environments remains a concern as training data needs to be cached to estimate mean and variance for a query location. Also, offline training of hyperparameters assumes domain knowledge of the operating environment that may not always be available.

Recently, Hilbert maps have been proposed by Ramos *et al.* [26] as an efficient alternative to GP based mapping. Occupancy is represented as a linear discriminative model operating on a high-dimensional feature vector obtained by projecting observations into a reproducing kernel Hilbert space. Advances in kernel approximations, including Random Fourier Features [27] and Nyström Approximations [28], are leveraged to obtain feature vectors used to train the model via stochastic gradient descent. The work has since been extended by Guizilini *et al.* [29] to make feature selection more principled via k -means clustering and local queries more efficient via a KD-tree. However, the model accuracy is sensitive to user-defined parameters including the number of clusters and the neighborhood size for queries. Also, extension of the approach to associate an uncertainty measure with model predictions is non-trivial.

The approach proposed in this paper learns a continuous spatial model similar to [14], [21]. Gaussian Mixture Models (GMMs) are employed as a semi-parametric learning technique to capture spatial dependencies and obtain a high-fidelity, compact representation of sensor observations. The proposed methodology is similar in spirit to [13] in that Expectation Maximization is used to maximize data likelihood. However, a mixture of coupled Gaussian distributions is learned instead of a set of decoupled 3D planes. Also, information-theoretic measures are employed to estimate the required number of components and a multi-fidelity representation is enabled via a hierarchy of Gaussian Mixture Models (HGMM).

An HGMM based approach is proposed in favor of other modeling techniques to represent the hierarchy of information acquired via sensors. The choice is justified considering the nonlinear nature of data, the computational constraints, and the need for a hierarchical representation. The high degree of inherent nonlinearity makes simplistic models like linear regression classifiers insufficient to represent surfaces in the environment. High degree polynomials, though representative, require environment specific tuning to obtain correct model complexity that is challenging in diverse environments.

Non parametric approaches do not assume a fixed functional form and enable high-fidelity approximations to arbitrary functions given enough data, as evidenced by GP Occupancy Maps [21], [22], [24]. However, nonparametric approaches are data driven, and consequently, pose a challenge in terms of scalability to large environments. Therefore, a modeling technique that approximates the fidelity of a nonparametric approach at a reduced computational footprint is motivated. A GMM enables a semi-parametric representation capable of approximating a GP given sufficient number of components. Also, the associated memory footprint is significantly smaller owing to the parametric form of a Gaussian distribution. The continuous GMM probability density function (pdf) further enables principled generation of a hierarchy based on its information content. Another noteworthy approach to a hierarchical representation is based on hierarchical Dirichlet processes (HDPs) [30]. However, HDPs are more suited to model data that exhibits a natural grouping of a basis set of shared elements (called atoms) across groups. The absence of such grouping in real-world environments makes the application of HDPs to surface modeling challenging. Hierarchical Gaussian mixture models have been proposed by Eckart *et al.* [31], [32] as a high fidelity representation for point cloud data and leveraged to enable robust point cloud registration. In prior work [33], the expectation-maximization approach of Eckart *et al.* is leveraged to develop a point cloud representation. The proposed methodology leverages prior work to enable a probabilistic framework for high fidelity occupancy representation and uncertainty characterization while providing evidence for real-time viability on a computationally constrained processor.

The proposed approach learns a hierarchy of Gaussian mixtures given a range-based sensor observation $\mathcal{R} = \{r_i\}$ with the i th observation of the surface, $r_i \sim \mathcal{N}(p_i, \sigma^2)$, where p_i represents the distance from the sensor to environment structure along the i th ray, with all space $p < p_i$ along the ray representing free-space. To simplify the presentation, we consider \mathcal{R} as a point-cloud $\mathcal{Z} = \{z_i \in \mathbb{R}^3\}, i = \{1, \dots, |\mathcal{Z}|\}$, under the appropriate sensor model transformation and treat the resulting distribution as unimodal (noting that more complex models are readily accommodated in the proposed framework). The organization of the paper is as follows. A generative point cloud model based on a hierarchy of Gaussian mixtures is developed in Section II. Section III augments this point cloud model into a spatial model that enables probabilistic reasoning over occupancy and incorporates free space information. A thorough evaluation of the proposed approach and comparison to state-of-the-art approaches on real-world data sets is provided in Section IV. Fig. 2 illustrates the sequence of steps involved in the generation of the spatial model given a stream of point clouds and sensor pose. Given the point cloud and sensor pose at time t , a novelty check is performed to determine the novel and redundant portions of the point cloud (see Section II-C). For the novel portion, the fidelity threshold is estimated (see Section II-B) followed by the training of the augmented point cloud model (see Section III-A). Free space information is incorporated to obtain the proposed spatial model (see Section III). For the non-novel portion, the parameters of the existing model are updated via bootstrapped EM (see Section II-C).

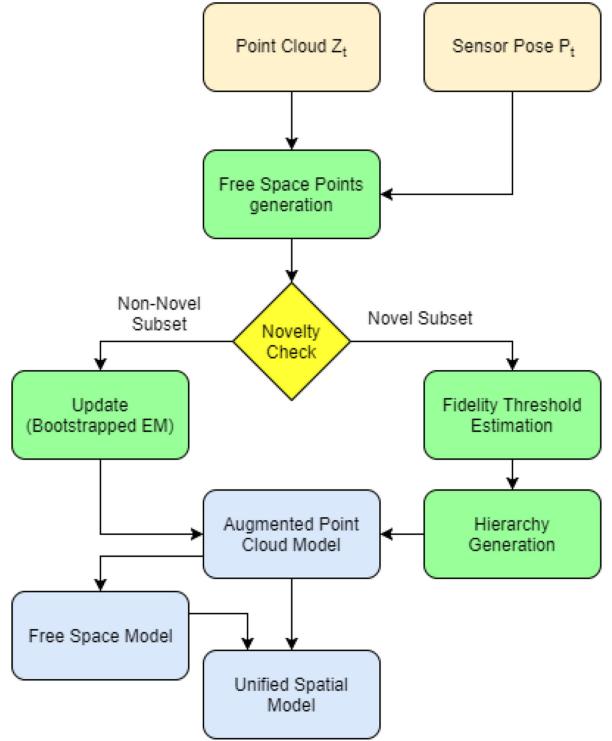


Fig. 2. Overview of the proposed methodology. The point cloud, \mathcal{Z}_t , at time instant, t , is divided into a novel and a redundant subset via a likelihood based novelty check. The redundant subset is used to update the parameters of the existing model while a fresh model is learned for the novel subset and merged with the existing model. This augmented point cloud model serves as a prior for the free space model and the two together form the unified spatial model.

II. POINT CLOUD MODEL

The proposed point cloud model consists of a hierarchy of Gaussian mixtures representing 3-D space, $X \in \mathbb{R}^3$, and the mixture at any given level of the hierarchy differs in size and fidelity from other levels. Expectation maximization [12] is employed to initiate model generation and information-theoretic measures are used to estimate the number of components required for a high-fidelity representation.

A. Model Definition

Let the l th level of the hierarchy ($l \in \mathbb{Z}, \forall l \in \{1 \dots L\}$) be given by the GMM, \mathcal{G}_l . Let \mathcal{G}_l contain J_l component Gaussian distributions specified by parameters, $\Theta_j = (\pi_j, \mu_j, \Sigma_j)$, where μ_j , Σ_j , and π_j represent the mean, covariance, and mixing weight for the j th component, with $j \in \{1 \dots J_l\}$. Given a 3-D point cloud, \mathcal{Z} , of size N , with points, $z_i \in \mathcal{Z}$, and assuming that the points are i.i.d. samples of the surface being observed, the likelihood of \mathcal{Z} to be generated by \mathcal{G}_l is

$$p(\mathcal{Z} | \Theta) = \prod_{i=1}^N p(z_i | \Theta) \quad (1)$$

$$= \prod_{i=1}^N \sum_{j=1}^{J_l} \pi_j p(z_i | \mu_j, \Sigma_j) \quad (2)$$

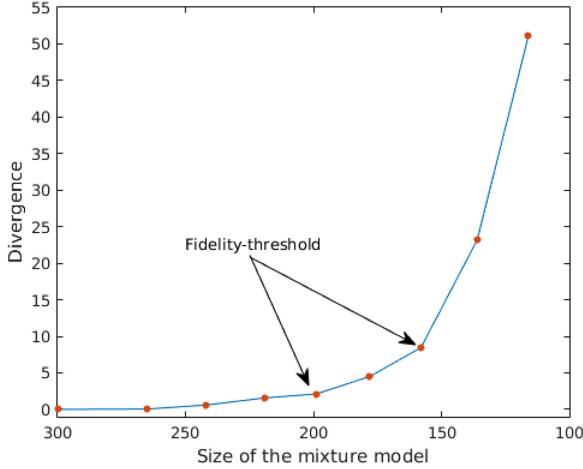


Fig. 3. Variation of KL-Divergence for GMMs of size varying from 300 to 116 with a reference GMM of size 300. The possible fidelity thresholds are highlighted. Increasing the size of the GMM beyond these thresholds does not significantly affect the fidelity of representation as indicated by the small decrease in divergence.

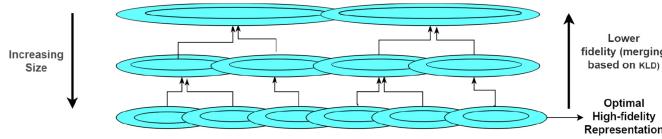


Fig. 4. Hierarchical model to represent 3-D space. The lowest level is the optimal high fidelity representation of size λ_f . Higher levels are generated by recursive merging of components based on KL Divergence to obtain lower fidelity and lower complexity models.

where

$$p(z_i | \mu_j, \Sigma_j) = \mathcal{N}(z_i | \mu_j, \Sigma_j) \quad (3)$$

and the corresponding log-likelihood of the data is

$$\ln p(\mathcal{Z} | \Theta) = \sum_{i=1}^N \ln \sum_{j=1}^J \pi_j p(z_i | \mu_j, \Sigma_j). \quad (4)$$

B. Model Generation

The number of components in a GMM directly impacts its ability to represent a complex function. An iterative approach is proposed to estimate the GMM size that enables precise modeling of the point cloud \mathcal{Z} . A key observation is that there exists a threshold on the GMM size (hereafter referred to as fidelity threshold, λ_f) beyond which model fidelity does not vary significantly on addition of components. Here, variation in model fidelity is quantified via divergence between the GMM density functions (see Fig. 3). This observation is leveraged to develop an iterative bottom-up approach that, when initialized with a GMM of size greater than λ_f , is able to estimate the fidelity threshold. Fig. 4 provides an overview of the resultant hierarchical model.

Algorithm 1 outlines the proposed bottom-up approach. Expectation maximization [12] is employed to initialize the

Algorithm 1 HGMM Generation.

Result: HGMM \mathcal{G}

```

1  $\lambda_d \leftarrow \text{Input};$  // Similarity threshold
2  $\lambda_{fo} \leftarrow \text{Input};$  // Fidelity threshold overestimate
3  $L \leftarrow \text{Input};$  // Number of hierarchy levels
4  $\mathcal{Z} \leftarrow \text{Input};$  // Point-cloud
5  $l \leftarrow 0;$  // Index of level in the hierarchy
6  $\mathcal{G} \leftarrow \{\emptyset\};$  // The HGMM
7  $\mathcal{G}_0 \leftarrow \text{EM}(\mathcal{Z}, \lambda_{fo});$  // Hierarchy Generation
8 while true do
    /* Merge components in current level  $\mathcal{G}_l$  to
     * generate the next level GMM  $\mathcal{M}$  */
    9  $\mathcal{M} \leftarrow \{\emptyset\};$ 
10 for  $i \leftarrow 1 \dots |\mathcal{G}_l|$  do
    11    $\theta_i \leftarrow \mathcal{G}_l, i;$  //  $i^{\text{th}}$  component in  $\mathcal{G}_l$ 
    12   for  $j \leftarrow i+1 \dots |\mathcal{G}_l|$  do
    13      $\theta_j \leftarrow \mathcal{G}_l, j;$  //  $j^{\text{th}}$  component in  $\mathcal{G}_l$ 
        // Merge if similar (Eq. 9)
    14     if  $\text{KLDivergence}(\theta_i, \theta_j) < \lambda_d$  then
    15       |  $\{\theta_i\} \leftarrow \text{Merge}(\theta_i, \theta_j);$ 
    16     end
    17   end
    18    $\mathcal{M} \leftarrow \mathcal{M} \cup \{\theta_i\};$  // Add component to GMM
    19 end
20  $\mathcal{G} \leftarrow \mathcal{G} \cup \mathcal{M};$  // Add level to hierarchy
    // Divergence between GMMs (Eq. 10)
21  $\text{div} \leftarrow \text{KLDivergence}(\mathcal{M}, \mathcal{G}_0);$ 
    // Check if knee point is reached (Fig. 3)
22 if  $\neg \text{Pruned}$  AND  $\text{IsKneePoint}(\text{div})$  then
    // Fidelity threshold estimated
23    $\lambda_f \leftarrow |\mathcal{G}_{l-1}|;$ 
    // Prune levels of size more than  $\lambda_f$ 
24    $\text{Prune}(\mathcal{G}, \lambda_f);$ 
25 end
    // End if desired number of levels generated
26 if  $|\mathcal{G}| < L$  then
27   | break;
28 end
29  $l \leftarrow l + 1$ 
30 end

```

algorithm via parameter estimation for the bottom most GMM \mathcal{G}_0 (Line 7). EM has been established as a way to iteratively maximize the joint likelihood of the data and an associated set of latent variables. For estimation of GMM parameters, a set C of correspondence variables, $c_{ij} \in \mathbb{R}$, that represents the associations between the points, z_i , and the mixture components, Θ_j , is used (notation borrowed from Eckart and Kelly [34]). *E-step*, for the k th iteration, involves calculation of the posterior for all c_{ij} given Θ

$$E[c_{ij}] = \frac{\pi_j p(z_i | \mu_j, \Sigma_j)}{\sum_{j'=1}^J \pi_{j'} p(z_i | \mu_{j'}, \Sigma_{j'})}. \quad (5)$$

M-step involves a maximum likelihood estimation of the mixture parameters considering $E[c_{ij}] \stackrel{\text{def}}{=} \gamma_{ij}$ to be a constant

$$\mu_j^{k+1} = \frac{\sum_i^N \gamma_{ij} z_i}{\sum_i^N \gamma_{ij}} \quad (6)$$

$$\Sigma_j^{k+1} = \frac{\sum_i^N \gamma_{ij} z_i z_i^T}{\sum_i^N \gamma_{ij}} - \mu_j^{k+1} \mu_j^{k+1 T} \quad (7)$$

$$\pi_j^{k+1} = \sum_i^N \frac{\gamma_{ij}}{N}. \quad (8)$$

An overestimate of the fidelity threshold, λ_{fo} , is required as a parameter in Algorithm 1 and is passed as input to EM. The implications of the parameter value are discussed in Section IV-G.

Lines 9–19 generate a reduced size GMM, \mathcal{M} , given the most recent GMM in the hierarchy, \mathcal{G}_l . This is achieved by merging *similar* Gaussian components where the measure of similarity is given by Kullback–Leibler (KL) divergence [35]. The closed-form solution for KL divergence between two Gaussian distributions, $f = \mathcal{N}(\mu_f, \Sigma_f)$ and $g = \mathcal{N}(\mu_g, \Sigma_g)$, for D -dimensional data is given as

$$D_{\text{KL}}(f||g) = \frac{1}{2} \left(\log \frac{|\Sigma_g|}{|\Sigma_f|} + \text{trace}(\Sigma_g^{-1} \Sigma_f) + (\mu_f - \mu_g)^T \Sigma_g^{-1} (\mu_f - \mu_g) - D \right). \quad (9)$$

A pairwise calculation of KL divergence (Line 14) reveals components, which are similar (divergence less than similarity threshold, λ_d) and these are merged into a single Gaussian distribution (Line 15) based on the approach proposed by Saarinen *et al.* [19].

KL divergence between GMMs is employed for the estimation of λ_f (Lines 21–25). A closed-form approximation for KL divergence between GMMs has been proposed by Goldberger *et al.* [36]. For two GMMs, p and q , with M and K components, respectively, and parameters $(\pi_m, \mu_m, \Lambda_m)$ and $(\tau_k, \nu_k, \Omega_k)$, it is given as

$$D_{\text{KL}}(q||p) = \sum_{i=1}^M \pi_i \min_{j \in \{1, K\}} \left(D_{\text{KL}}(p_i || q_j) + \log \frac{\pi_i}{\tau_j} \right). \quad (10)$$

KL divergence between the current level, \mathcal{G}_l , and \mathcal{G}_0 , (Line 21) is used to estimate the knee point and thus the fidelity threshold (Lines 23–25) based on the work of Satopää *et al.* [37]. Once estimated, all levels of the hierarchy with size more than λ_f are pruned (Line 24) and the GMM with size λ_f now forms the bottom most level of the hierarchy \mathcal{G}_0 . The algorithm terminates when the desired number of levels, L , have been generated (Line 26).

C. Model Update

A consequence of the sequential motion of mobile robots is that a portion of the point cloud, \mathcal{Z}_t , obtained at time in-

stant t , contains information that has already been modeled based on $\mathcal{Z}_{1 \dots t-1}$ and the remaining portion contains novel information. The novel and redundant portions of \mathcal{Z}_t are estimated via calculation of the log-likelihood of $z_i \in \mathcal{Z}_t$ with respect to the existing model, \mathcal{G}_0 . Inspired by the work of Engel and Heinen [38], an empirically determined novelty threshold, λ_n , is used to categorize points as novel (\mathcal{Z}_{tn}) versus redundant (\mathcal{Z}_{tr}) based on (1) as

$$\mathcal{Z}_{tn} = \{z \in \mathcal{Z}_t \mid p(z \mid \Theta) < \lambda_n\} \quad (11)$$

$$\mathcal{Z}_{tr} = \{z \in \mathcal{Z}_t \mid p(z \mid \Theta) \geq \lambda_n\} \quad (12)$$

where Θ represents the parameters of \mathcal{G}_0 .

The update of \mathcal{G}_0 with \mathcal{Z}_{tr} proceeds via bootstrapping of EM with Θ . *M-step* is slightly modified to incorporate the posterior from the previous point clouds as well as \mathcal{Z}_{tr} . Let the support size of \mathcal{G}_0 be N . Then, following entities are defined at time instant $t-1$ considering $\{\Theta_j = (\pi_j, \mu_j, \Sigma_j)\}$

$$\begin{aligned} S_{\pi_j} &= \sum_i^N \gamma_{ij} = N\pi_j \\ S_{\mu_j} &= \sum_i^N \gamma_{ij} z_i = S_{\pi_j} \mu_j \\ S_{\Sigma_j} &= \sum_i^N \gamma_{ij} z_i z_i^T = S_{\pi_j} (\Sigma_j + \mu_j \mu_j^T). \end{aligned}$$

The updated mean, covariance, and weights, given $|\mathcal{Z}_{tr}| = N'$ are

$$S'_{\pi_j} = S_{\pi_j} + \sum_i^{N'} p_{ij} \quad (13)$$

$$\pi'_j = \frac{S'_{\pi_j}}{N + N'} \quad (14)$$

$$\mu'_j = \frac{S_{\mu_j} + \sum_i^{N'} p_{ij} z_i}{S'_{\pi_j}} \quad (15)$$

$$\Sigma'_j = \frac{(S_{\Sigma_j} + \sum_i^{N'} p_{ij} z_i z_i^T)}{S'_{\pi_j}} - \mu'_j \mu'^T_j. \quad (16)$$

Higher levels of the hierarchy are regenerated based on the procedure outlined in Algorithm 1 (Lines 9–19).

A fresh HGMM, \mathcal{G}' , is learned for \mathcal{Z}_{tn} following the same procedure as outlined in Algorithm 1. The levels of the existing HGMM, \mathcal{G} , are then merged with the components of the corresponding levels of \mathcal{G}' via weight normalization. The required insight is that the points in \mathcal{Z}_{tn} are minimally influenced by the components in \mathcal{G} as evidenced by the log-likelihood based novelty check. Thus, a naive augmentation closely approximates the distribution otherwise learned by training over the collective set $\mathcal{Z}_{1 \dots t}$. The updated weight vector for level l , $\pi_{\mathcal{G}_l}$, with a support set of size $N_{\mathcal{G}_l}$ is

$$\pi_{\mathcal{G}_l} = \frac{\pi_{\mathcal{G}_l} N_{\mathcal{G}_l}}{N_{\mathcal{G}_l} + N_{\mathcal{G}'_l}} \quad (17)$$

where $N_{\mathcal{G}'_l}$ is the support set size of the \mathcal{G}'_l .

The computational cost of log-likelihood calculation (11) grows with the size of \mathcal{G}_0 . This cost is capped by leveraging the observation that most components in \mathcal{G}_0 have a negligible influence on \mathcal{Z}_t due to spatial distance. An approximate local GMM is maintained to enable real-time updates by retaining those components of \mathcal{G}_0 that have a non-negligible maximum pdf over \mathcal{Z}_{t-1} (4).

D. Note On Inputs and Parameters

The similarity threshold, λ_d , regulates the difference in fidelity between subsequent levels of the hierarchy by regulating rate of merging of components. λ_d is environment agnostic (see Section IV) and should be tuned depending on the application. The divergence between Gaussian components increases as lesser fidelity GMMs are generated. This is addressed by iteratively increasing λ_d for higher levels. The overestimate of fidelity threshold, λ_{fo} , affects model accuracy if it is not a strict overestimate. Conversely, a very large value affects computational complexity (Section IV-G). A voxel grid filter is applied to the point cloud and the size of the filtered point cloud is used as the overestimate for λ_f . The number of levels, L , is selected based on preliminary evaluation in similar environments to achieve sufficient model fidelity while ensuring viable real-time performance (with application specific online computation of L as future work). The novelty threshold, λ_n , is observed to be environment agnostic (see Section IV). Implications on accuracy and computational cost are discussed in Section IV-G.

III. PROBABILISTIC REPRESENTATION OF OCCUPANCY

The point cloud model presented in Section II is extended in this section into a probabilistic representation of occupancy. An uncertainty measure, in the form of a variance estimate associated with model predictions, is also incorporated.

A. Augmented Point Cloud Model

The 3-D point cloud model is augmented to a four-tuple HGMM to enable reasoning over occupancy. The additional variable, W , is a function of the location in space, X , the 3-D point cloud \mathcal{Z} , and the sensor pose, P , and is called the distance function. The distance function maps a location on the surface to zero, and locations in free space to the distance from the surface along a ray emanating from the sensor. Given a point on the surface, Q , the distance for a point in free space, F , along the ray QF is $\|QF\|$. Thus, W is zero for all points on the surface and positive for the points in free space.

The additional variable W enables expression of occupancy as a conditional distribution over spatial location. Specifically, considering occupancy as a binary variable occ , the probability of occupancy at a 3-D location $X = x$ is

$$P(\text{occ} = 1 \mid X = x) = P(W \in [-\epsilon, \epsilon] \mid X = x) \quad (18)$$

and the probability of x to be in free space is

$$P(\text{occ} = 0 \mid X = x) = P(W > \epsilon \mid X = x) \quad (19)$$

where ϵ defines a small neighborhood around the surface where the distance function is zero. The conditional pdf for W over

X is derived as follows based on the work of Sung [39]. The joint density, $p_{X,W}$, can be represented as a GMM with J components specified by parameters $\Theta_j = (\pi_j, \mu_j, \Sigma_j)$ where π_j , μ_j , and Σ_j represent the mixing weight, mean, and covariance matrix as

$$p_{X,W}(x, w) = \sum_{j=1}^J \pi_j \phi(x, w; \mu_j, \Sigma_j) \quad (20)$$

where

$$\sum_{j=1}^J \pi_j = 1, \quad \mu_j = \begin{bmatrix} \mu_{jX} \\ \mu_{jW} \end{bmatrix}, \quad \Sigma_j = \begin{bmatrix} \Sigma_{jXX} & \Sigma_{jXW} \\ \Sigma_{jWX} & \Sigma_{jWW} \end{bmatrix}$$

and $\phi(x, w; \mu_j, \Sigma_j)$ is the four-tuple Gaussian distribution $\mathcal{N}(x, w; \mu_j, \Sigma_j)$ representing the pdf of the j th component. The joint density is decomposed by partitioning each Gaussian component as proposed by Mardia *et al.* [40]

$$\begin{aligned} p_{X,W}(x, w) &= p_{W|X}(w|x) p_X(x) \\ &= \sum_{j=1}^J \pi_j \phi(w|x; m_j(x), \sigma_j^2) \phi(x; \mu_{jX}, \Sigma_{jXX}) \end{aligned} \quad (21)$$

where

$$m_j(x) = \mu_{jW} + \Sigma_{jWX} \Sigma_{jXX}^{-1} (x - \mu_{jX}) \quad (22)$$

$$\sigma_j^2 = \Sigma_{jWW} - \Sigma_{jWX} \Sigma_{jXX}^{-1} \Sigma_{jXW}. \quad (23)$$

The marginal density of X is obtained from (21) as

$$\begin{aligned} p_X(x) &= \int p_{X,W}(x, w) dw \\ &= \sum_{j=1}^J \pi_j \phi(x; \mu_{jX}, \Sigma_{jXX}). \end{aligned} \quad (24)$$

The conditional density $p_{W|X}(w|x)$ follows from (21)

$$p_{W|X}(w|x) = \sum_{j=1}^J w_j(x) \phi(w; m_j(x), \sigma_j^2) \quad (25)$$

with the mixing weight

$$w_j(x) = \frac{\pi_j \phi(x; \mu_{jX}, \Sigma_{jXX})}{\sum_{k=1}^K \pi_k \phi(x; \mu_{kX}, \Sigma_{kXX})}. \quad (26)$$

Generation of the four-tuple HGMM follows the procedure outlined in Algorithm 1 with a minor perturbation to \mathcal{Z} . \mathcal{Z} is augmented with points in free space at a small distance (≈ 5 cm) from the surface to enable estimation of the correlation between X and W . It is observed that an augmentation of \mathcal{Z} by 20% is sufficient to learn this correlation.

B. Free Space Model

Structural sparsity in free space is leveraged to develop a constant time algorithm for learning a separate four-tuple HGMM, \mathcal{F} , to model observed free space. The augmented point cloud model, \mathcal{G} , provides a prior on the size of the free space model.

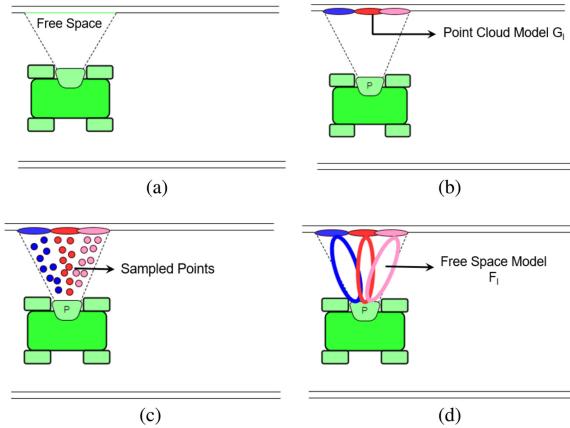


Fig. 5. Graphic illustration of the methodology to generate a free space model. Given a point cloud model, \mathcal{G}_l , (b) the observed free space (a) is modeled by uniformly sampling a set of points corresponding to each component of \mathcal{G}_l (c) and learning a Gaussian component per sample set (d). The weight vector for the resulting GMM, \mathcal{F}_l , is the same as that of \mathcal{G}_l .

Specifically, for every Gaussian component in the GMM \mathcal{G}_l , there is a corresponding component in \mathcal{F}_l .

The parameter estimation of the j th component of \mathcal{F}_l proceeds by sampling a set of points, S_j , from the j th component of \mathcal{G}_l . The set, S_j , together with the sensor pose, P , are used to sample points in free space, at a fixed resolution, along the rays originating at P and terminating at the points in S_j . The values for W are obtained by calculating the distance function from the sampled points and the sensor-pose. The mean and covariance matrix for the j th component in \mathcal{F}_l are then estimated from the set of sampled free space points. The weight of the j th component is set equal to the weight of the corresponding component in \mathcal{G}_l . The methodology is illustrated graphically in Fig. 5.

The procedure outlined above is repeated for every component in \mathcal{G}_l and for every layer in \mathcal{G} resulting in the generation of the free space HGMM, \mathcal{F} . Clearly, the value of distance-function for the free space model is always positive and varies up to the maximum distance of the sensor from the surface.

C. Unified Spatial Model

The augmented point cloud model, \mathcal{G} , and the free space model, \mathcal{F} , are merged to form a unified model of occupancy, \mathcal{H} . The merging proceeds via weight normalization as described in Section II-C. Separate training for the point cloud model and the free space model is preferred over direct training of the unified model in order to cap the computational complexity and retain the high-fidelity point cloud model. Given \mathcal{H} , the probability of occupancy at any location, x , is expressed as

$$\begin{aligned} P(\text{occ} = 1 \mid X = x) &= P(W \in [-\epsilon, \epsilon] \mid X = x) \\ &= \int_{W=-\epsilon}^{W=\epsilon} p_{W|X}(w|x) dw \\ &= \int_{W=-\epsilon}^{W=\epsilon} \sum_{j=1}^J w_j(x) \phi(w; m_j(x), \sigma_j^2) dw \end{aligned} \quad (27)$$

where ϵ is a small integration interval. The probability of a location to be in free space (19) is expressed as

$$\begin{aligned} P(\text{occ} = 0 \mid X = x) &= P(W > \epsilon \mid X = x) \\ &= \int_{W=\epsilon}^{W=\infty} p_{W|X}(w|x) dw \\ &= \int_{W=\epsilon}^{W=\infty} \sum_{j=1}^J w_j(x) \phi(w; m_j(x), \sigma_j^2) dw. \end{aligned} \quad (28)$$

A mean function and a variance estimate is regressed from the unified occupancy distribution, based on the work of Sung [39]. A mean function is obtained from (22) and (26) as the weighted average of component-wise means

$$m(x) = E[W|X = x] = \sum_{j=1}^J w_j(x) m_j(x). \quad (29)$$

A variance estimate associated with the regressed mean is obtained from (29) as

$$\begin{aligned} v(x) &= E[(W|X = x)^2] - E[W|X = x]^2 \\ &= \sum_{j=1}^J w_j(x)(m_j(x)^2 + \sigma_j^2) - \left(\sum_{j=1}^J w_j(x)m_j(x) \right)^2. \end{aligned} \quad (30)$$

IV. RESULTS AND ANALYSIS

The proposed approach is evaluated in this section to assess the fidelity of the point cloud model and occupancy representation along with the associated memory footprint. The correctness of the variance estimate is investigated and the viability of real-time operation on a computationally constrained processor is assessed. Three data sets are used for this evaluation. The first data set, **FR_ROOM**, is publicly available [41] and is collected using an RGBD sensor. The data set represents a small-scale cluttered environment with the average distance from the sensor to the nearest obstacle ranging up to 3.5 m. The second data set, **MINE**, represents a larger-scale environment. This data set is collected using a Velodyne VLP-32 LIDAR in an underground mine with the average distance to nearest obstacle ranging from 8 to 9 m. The data set extends over 1 km in length and contains significantly less structural detail. The third data set, **PIT**, is collected using a Velodyne VLP-16 LIDAR in an open pit and represents an unstructured environment with the average distance to nearest obstacle ranging from 15 to 17 m.

The proposed framework was implemented in C++ using the robot operating system (ROS) framework [42] and leveraging the ArrayFire library [43] for a graphics processing unit (GPU)-based parallelized implementation. A comparison to the implementations of GPOctoMap [25], NDT-OM¹ [19], and Octomap² [9] in terms of fidelity, memory footprint, and generalizability

¹NDT-OM software. https://github.com/OrebroUniversity/perception_oru-release [Accessed on: 20th July, 2018].

²Octomap software. <https://github.com/OctoMap/octomap> [Accessed on: 20th July, 2018].

is also provided. It is important to note that the same set of parameters is used for the HGMM approach for all experiments. The novelty threshold is set to -10.5 and the similarity threshold is initialized to 0.9 and iteratively incremented by 0.2 (see Section II-D). The integration interval, ϵ , and the number of levels, L , is set to 1 cm and 4 , respectively. The parameters for the competing techniques are tuned for each data set and stated when required in the following sections. For Octomap and NDT-OM, the optimal parameters are obtained by repeated trials while hyperparameter training is employed for GPOctoMap.

A. Fidelity of the Point Cloud Model

The augmented point cloud model developed in Section III-A is trained on a sequence of uniformly subsampled (1 cm resolution) point clouds from **FR_ROOM** and the highest fidelity level is queried for the likelihood of the test points to lie on the observed surface. The surface model corresponds to the distance function W being zero and the corresponding pdf is obtained similar to (25)

$$p_{X|W}(x|0) = \sum_{j=1}^J w_j(0) \phi(m_j(0), \Sigma_j) \quad (31)$$

where

$$m_j(W=0) = \mu_{jX} + \Sigma_{jXW} \Sigma_{jWW}^{-1} (0 - \mu_{jW}) \quad (32)$$

$$\Sigma_j = \Sigma_{jXX} - \Sigma_{jXW} \Sigma_{jWW}^{-1} \Sigma_{jWX} \quad (33)$$

$$w_j(W=0) = \frac{\pi_j \phi(0; \mu_{jW}, \Sigma_{jWW})}{\sum_{j'=1}^J \pi_{j'} \phi(0; \mu_{j'W}, \Sigma_{j'WW})}. \quad (34)$$

The test set consists of the set of points not made available to the algorithm during training, augmented with free space points sampled along the rays from the sensor. The performance is compared to the NDT [18] surface model (cell-size 5 , 10 , and 15 cm) via receiver operating characteristic (ROC) curves shown in Fig. 6. The proposed approach is observed to achieve a higher true positive rate and area under the curve (AUC) than the NDT representation. This can be attributed to the restricted support of the Gaussian distributions due to the cell independence assumption in NDT that leads to higher uncertainties at cell boundaries. Also, the accuracy of the NDT representation is observed to be sensitive to the resolution of the voxel grid.

Fig. 7 provides a qualitative evaluation of the model. Snapshots from the data set, **FR_ROOM**, are reconstructed via sampling from the highest fidelity level of the surface model (31). Fig. 7 shows the point cloud from the data set and the corresponding reconstruction via sampling.

B. Fidelity of the Occupancy Representation

The accuracy of the unified spatial model is characterized in diverse environments represented by the three data sets (**FR_ROOM**, **MINE**, **PIT**) and compared to Octomap, NDT-OM, and GPOctoMap. Fig. 8 presents the ROC curves for the three data sets. For **FR_ROOM**, the cell-size for Octomap is set to 5 cm and that of NDT-OM is set to 10 cm. The hyperparameter training for GPOctoMap is performed on subsampled data. The

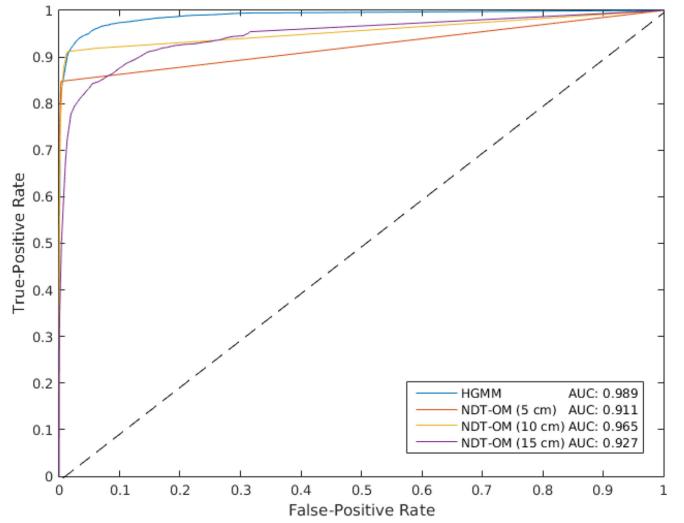


Fig. 6. ROC curves for the highest-fidelity level of the HGMM surface model and NDT representation (cell-size 5 , 10 , and 15 cm) for **FR_ROOM** data set. The proposed approach achieves a higher true-positive rate and AUC than NDT. The NDT model is also observed to be sensitive to cell size.

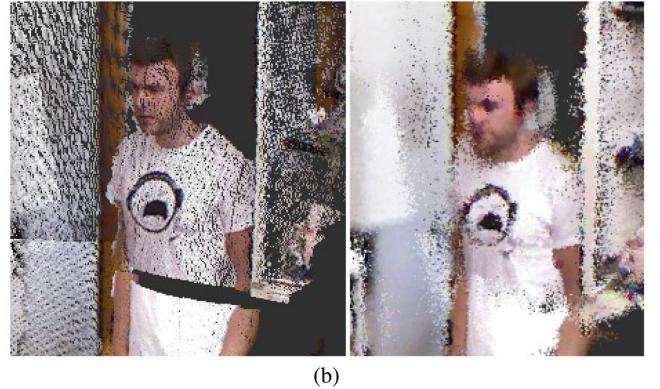
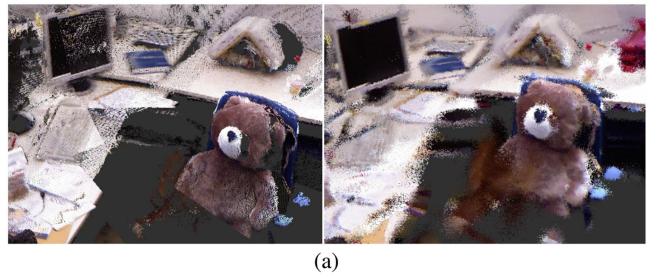


Fig. 7. Qualitative visual evaluation for the point cloud model. Snapshots from **FR_ROOM** (left) and the corresponding reconstruction via sampling from the lowest level of the model (right). With an average size of 116 components per point cloud, the proposed model provides a high-fidelity reconstruction and handles sparsity in sensor data (b). *RGB information is for illustration only and not obtained from the model.*

characteristic length, l , and signal variance, σ_f , are 0.1 m and 0.5 , respectively. It is observed [see Fig. 8(a)] that the proposed approach matches the performance of NDT-OM and Octomap in terms of AUC measure, but has a higher true-positive rate. The continuous nature of the HGMM approach makes it more robust to sensor sparsity leading to more correct classifications. GPOctoMap is observed to have the highest false-positive rate.

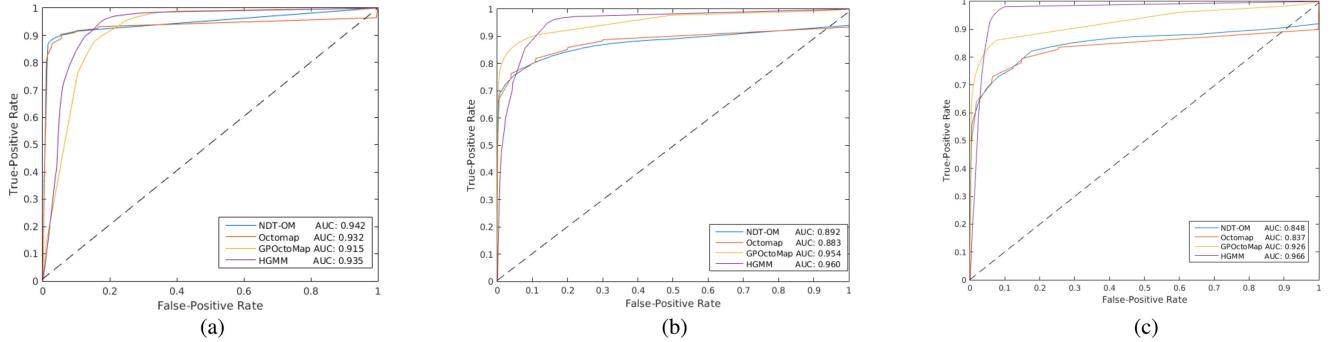


Fig. 8. ROC curves for the proposed spatial model (HGMM), GPOctoMap, NDT-OM, and Octomap. The HGMM approach is observed to maintain accuracy across all three data sets, (a) **FR_ROOM**, (b) **MINE**, and (c) **PIT**, while the competing techniques appear to be sensitive to environmental traits and sensor characteristics.

This is indicative of the fact that the fixed characteristic length used for GP Regression affects generalization of the approach to the whole environment. The HGMM approach has a smaller false positive rate as it is not restricted by a fixed characteristic length.

For **MINE**, the cell size for Octomap is increased to 15 cm and that of NDT-OM to 20 cm. The characteristic length for GPOctoMap is found to be 0.3 m. It is observed [see Fig. 8(b)] that sparsity of the data, induced by a larger-scale environment and the nature of the sensor, significantly affects the accuracy of both NDT-OM and Octomap while the proposed approach maintains its precision, matching that of GPOctoMap. The same set of parameters, as used for **MINE**, are used for **PIT** for all techniques. The performance of Octomap, NDT-OM, and GPOctoMap are observed to deteriorate while the HGMM approach maintains its level of fidelity, as shown in Fig. 8(c). It can be concluded from Fig. 8 that the proposed approach is able to generalize to diverse environments while the performance of the state-of-the-art is affected by the environment and sensor characteristics.

C. Multifidelity Representation

The implications of the multifidelity representation are quantitatively evaluated in Fig. 9 for **PIT**. It is observed that the reduction in memory footprint by 50% (from 320 to 160 bytes per point cloud) corresponds to a drop in AUC by 5% (from 0.95 to 0.9). A qualitative visualization of the effect of the hierarchy in terms of reduction in fidelity is shown in Fig. 10 via a plot of the probability of occupancy predicted by different layers of the hierarchy. The predictions corresponding to the levels, $l = 0$ and $l = 4$, are shown with the difference in predictions as a consequence of the drop in fidelity, highlighted with ellipses. It is observed that the model at $l = 4$ is slightly noisier than that at $l = 0$. Specifically, the probability distribution is less sharp in some sections (shown by white ellipses) and the model struggles to capture the observations corresponding to the vehicle (shown by red ellipses). A similar pattern is observed in Fig. 11 that demonstrates the generative capability of the model for **FR_ROOM** with the GMM at level $l = 4$ generating lower fidelity reconstructions, highlighted by red ellipses.

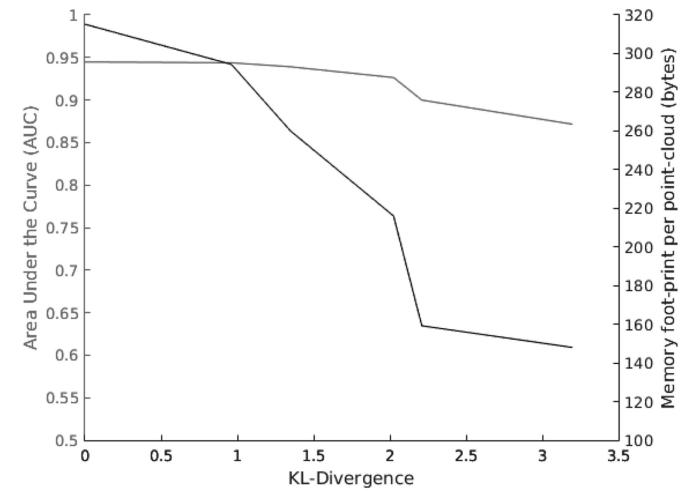


Fig. 9. Variation of the accuracy and memory footprint against variation in KL divergence corresponding to the levels of the hierarchy. A reduction in AUC of 0.05 is observed corresponding to a reduction in memory footprint by 50%.

D. Memory Footprint

A comparison of the memory footprint of the HGMM approach to the state-of-the-art approaches is shown in Table I. The same set of parameters, as mentioned in Section IV-B, are used. For all techniques, the footprint increases with the scale of the environment. However, the memory footprint of the proposed approach is observed to be significantly less (up to 40 times less for **PIT**) than all the other techniques for all data sets. The proposed approach is thus able to provide a high-fidelity representation at significantly reduced memory footprint.

E. Variance Estimate Characterization

A comparison of the proposed technique with GP regression is provided to assess correctness of the variance estimate associated with the model predictions. For this, a data set is generated consisting of samples, $x_i \in \mathbb{R}$, and the target function value corrupted with noise given by

$$y_i = \sin(3x_i) + \mathcal{N}(0, \sigma^2). \quad (35)$$

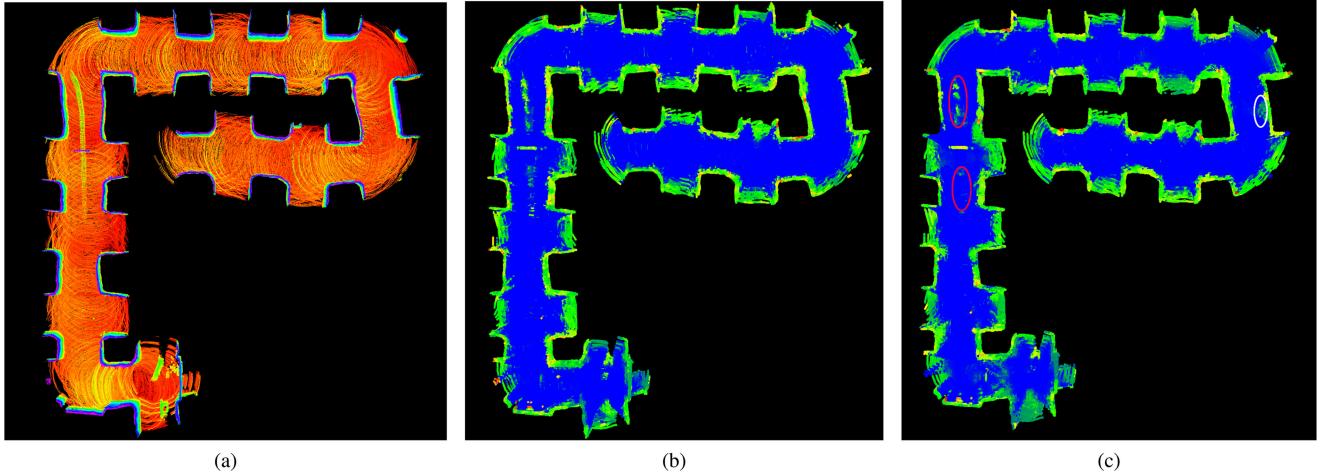


Fig. 10. Qualitative evaluation of the implications of the hierarchy on the occupancy distribution. The probability of occupancy, visualized via a heat-map (probability increases from blue to green), for the levels, $l = 0$ (b), and $l = 4$ (c) for the input point cloud data set, PIT (a). The lower fidelity model is observed to be less sharp (white ellipse) and tends to miss on the vehicle in the data set (red ellipse).

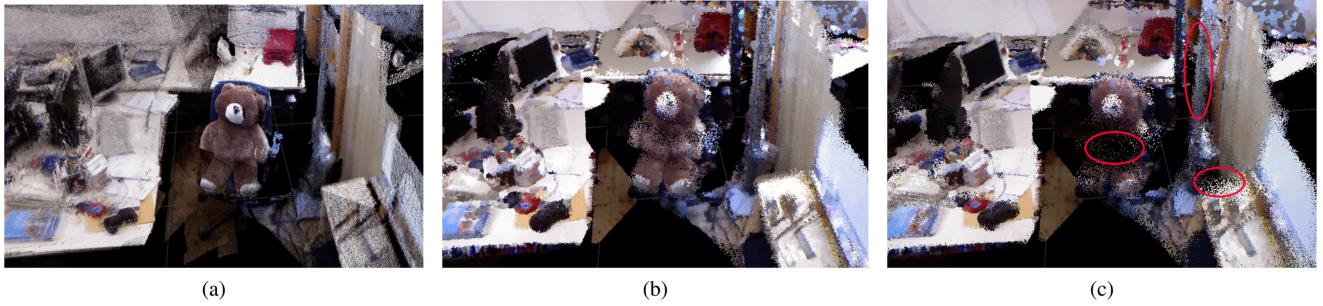


Fig. 11. Qualitative evaluation of the implications of the hierarchy on the point cloud model. The reconstruction of a snapshot from (a) FR_ROOM, (b) for the higher-fidelity level, $l = 0$, and (c) lower-fidelity level, $l = 4$. The lower fidelity model is observed to generate noisier reconstructions for complex surfaces (red ellipses) as compared to the higher-fidelity representation. *RGB information is for illustration only and not obtained from the model.*

TABLE I
COMPARISON OF THE MEMORY FOOTPRINT FOR THE LOWEST LEVEL (HIGHEST FIDELITY) OF THE SPATIAL MODEL WITH COMPETING TECHNIQUES

Dataset	Scans	Method	Memory (KB)	Memory / Scan (KB)
FR_ROOM	1361	Octomap	8987	6.60
		NDT-OM	1020	0.75
		GPOctoMap	3794	2.79
		HGMM	274	0.20
MINE	3043	Octomap	191126	62.41
		NDT-OM	41785	13.73
		GPOctoMap	145550	47.83
		HGMM	1792	0.59
PIT	2006	Octomap	646718	322.42
		NDT-OM	83263	41.53
		GPOctoMap	156218	77.88
		HGMM	2362	1.18

Both the HGMM model and GP Regression are trained with a sequence of 25 sample sets with each set containing 500 uniformly sampled values $x_i \in [-2, 2]$ and corresponding corrupted output with $\sigma = 0.05$. The models are then queried for mean and variance estimates for a fixed test set. Fig. 12 shows the results of this experiment.

The proposed approach estimates the fidelity threshold, λ_f , to be 14. It is observed that the proposed approach converges to a variance estimate same as the injected input noise. Also, given the same sequence of points, both approaches are observed to converge to a similar entropy value, even though the initial entropy for GP regression is lower than that of the HGMM.

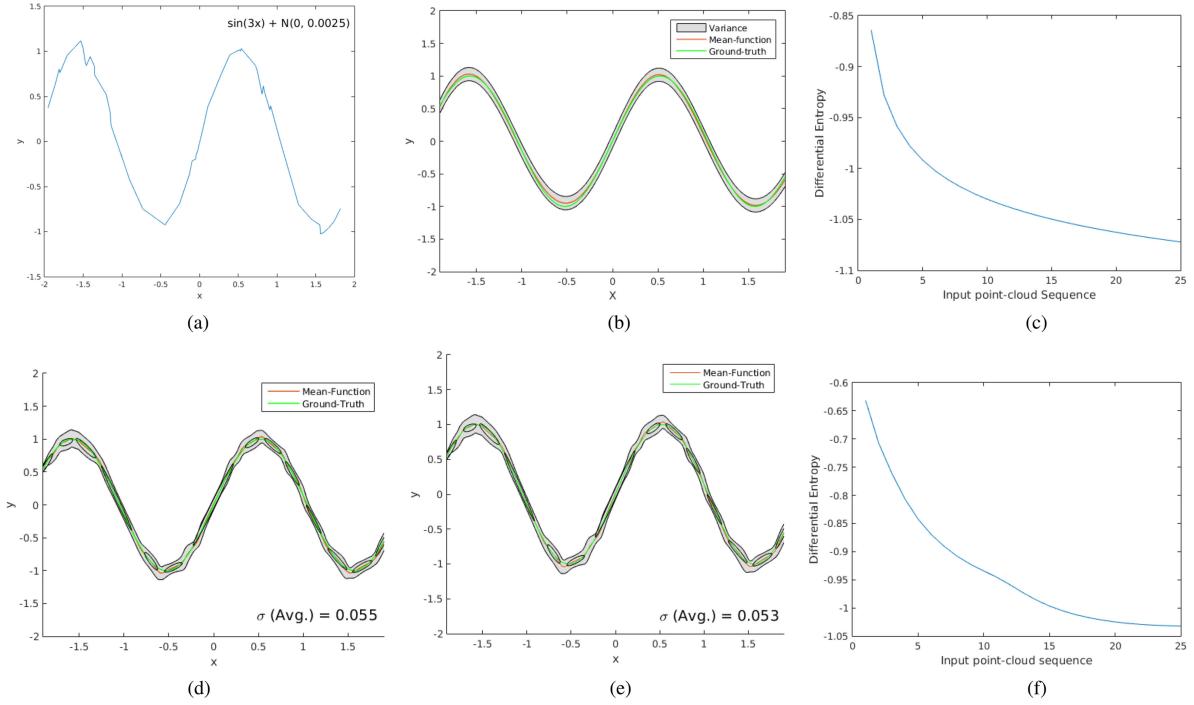


Fig. 12. Characterization of the variance estimate from the proposed framework and Gaussian Process Regression. (a) Sequence of 25 sample-sets each consisting of 500 samples from the simulated noisy function (35) ($\sigma = 0.05$) is provided as input to a GP and the proposed framework. The converged GP mean and variance for a test-set (b) and the initial and final state of the GMM with $\lambda_f = 14$ (d), (e) are shown. The rate of convergence is demonstrated via differential entropy curves (in bits) (c) and (f). Both approaches converge to the correct variance estimate and similar entropy values arising due to appropriate selection of the size of the mixture given the application and environment model. (a) Noisy Input. (b) Converged GP. (c) GP: Entropy Curve. (d) GMM: Point Cloud 1. (e) GMM: Point Cloud 25. (f) GMM: Entropy Curve.

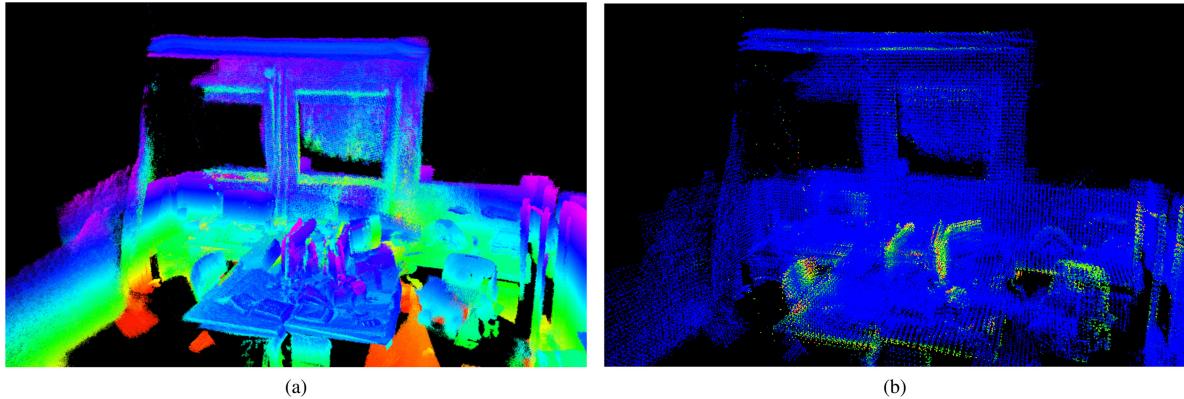


Fig. 13. Qualitative evaluation of the variance estimate obtained from the proposed framework. For a snapshot from **FR_ROOM** (a), the variance estimate is calculated for a set of uniformly sampled locations on the surface and visualized via a heat-map (b) (variance growing from blue to yellow). The variance estimate is higher at locations where the sensor measurements are expected to be noisy (table-edges) and the monitor surface which is visibly noisy in the input point cloud.

The proposed approach, thus, yields similar performance to a GP regression framework while eliminating the need to store input data in memory. However, it should be noted that GP-based modeling techniques enable extrapolation of the variance estimate that is not viable with a GMM-based framework.

A qualitative visualization of the measure of uncertainty is provided in Fig. 13 via a heat map of the variance estimate associated with the model. The model is trained on the data set, **FR_ROOM**, and variance is queried for a set of uniformly sampled points on the surface. It is observed that the regions with

a higher variance estimate correspond to regions with noisier sensor observations such as table edges and edges of the monitor (observed to be state-estimation noise), which aligns with reasonable expectation.

F. Viability of Real-Time Operation

The high degree of parallelizability of expectation maximization, likelihood estimation, and posterior probability calculation is exploited via a GPU-based implementation of the proposed

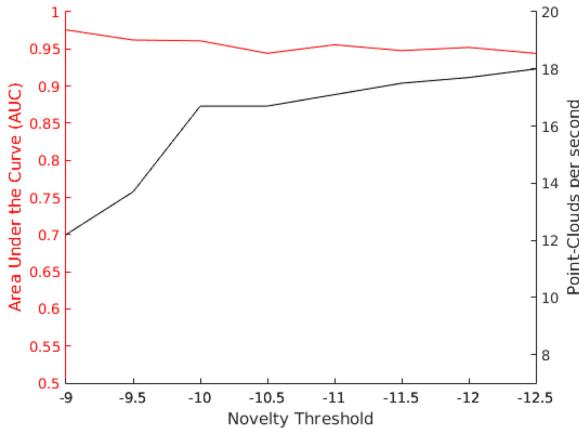


Fig. 14. Variation of the AUC and the rate of operation of the proposed approach with variation in novelty threshold. The variation in AUC is observed to be minimal and the rate of operation drops significantly for higher values of the novelty threshold.

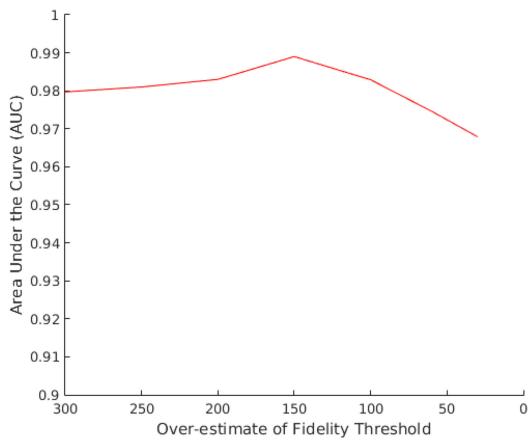


Fig. 15. Variation of the accuracy of the proposed approach for different values of λ_{fo} , with $\lambda_f = 138$. The variation is minimal for values greater than λ_f and registers a drop for significantly low values of λ_{fo} .

framework. Altinigneli *et al.* [44] provided a computational complexity analysis for a parallelized EM implementation. Also, the sequential nature of the motion of mobile robots (aerial or ground) helps to bound the computation required per point cloud, as the percentage of novel information is limited. The run time complexity of the framework thus scales with novelty of subsequent sensor observations. The complexity is further reduced by leveraging robustness of the HGMM formulation to measurement sparsity via subsampling. The cost of divergence calculation scales quadratically with the size of the GMM. This operation is performed on a CPU in favor of a GPU considering the tradeoff between data transfer overhead and parallelization gains.

The proposed approach is implemented on NVIDIA Jetson TX2³, an embedded level System-on-Chip with a discrete CUDA-enabled GPU designed for constrained autonomous

systems. The framework is observed to operate at a rate of **24** point clouds per second on the TX2 for **FR_ROOM** data set and **16** point clouds per second for **MINE** and **PIT** data sets.

G. Sensitivity to Parameters

The sensitivity of the proposed approach to the overestimate of fidelity threshold, λ_{fo} , and the novelty threshold, λ_n , is analyzed in this section. For λ_n , multiple instances of the proposed model are trained on a sequence of point clouds from **MINE** with the novelty threshold varied from -13.5 to -9 . It is observed (see Fig. 14) that the variation in accuracy is minimal and the operating rate drops for significantly high values of λ_n . A higher novelty threshold results in a greater fraction of the point cloud to be designated as novel requiring more EM instances and leading to higher computational complexity. For λ_{fo} , the proposed approach is trained on a point cloud from **MINE** for which λ_f is determined to be 138. The accuracy is observed to be stable for high values but drops for values significantly smaller than λ_f (see Fig. 15).

V. CONCLUSION AND FUTURE WORK

An approach to enable multifidelity probabilistic representations of diverse environments is presented in this work. Gaussian mixture models are employed to provide a continuous approximate spatial distribution. The proposed approach is shown to have a significantly reduced memory footprint as compared to state of the art, enabling operation in potentially large environments. The approach is also demonstrated to be robust to environmental peculiarities including scale, clutter, and type of structure, as well as to different sensor types. A measure of uncertainty is associated with model predictions and shown to be correct.

The model, as presented in this work, supports range measurements and provides a high fidelity representation of occupancy in the environment. An extension of the representation to incorporate information from multiple sensing modalities would provide increased robustness and enable multimodal reasoning and inference. Also, principled handling of state estimation uncertainty and incorporation of global consistency in the formulation via detection of place revisits and bundle adjustment would be the logical next step to enable application to large-scale environments. A principled online estimation of the number of levels in the hierarchy depending on application requirements is also being considered.

REFERENCES

- [1] V. Indelman, L. Carlone, and F. Dellaert, “Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments,” *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 849–882, 2015.
- [2] S. Pathak, A. Thomas, and V. Indelman, “A unified framework for data association aware robust belief space planning and perception,” *Int. J. Robot. Res.*, vol. 37, no. 2–3, pp. 287–315, 2018.
- [3] A. Elfes, “Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation.” Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, 1989.
- [4] H. P. Moravec, “Sensor fusion in certainty grids for mobile robots,” *AI Mag.*, vol. 9, no. 2, pp. 61–74, 1988.

³NVIDIA Jetson TX2. <https://developer.nvidia.com/embedded/buy/jetson-tx2> [Accessed on: 19th July, 2018]

- [5] P. Payeur, P. Hebert, D. Laurendeau, and C. M. Gosselin, "Probabilistic octree modeling of a 3d dynamic environment," in *Proc. Int. Conf. Robot. Autom.*, Apr. 1997, vol. 2, pp. 1289–1296.
- [6] D. Meagher, "Geometric modeling using octree encoding," *Comput. Graph. Image Process.*, vol. 19, no. 2, pp. 129–147, 1982.
- [7] J. Fournier, B. Ricard, and D. Laurendeau, "Mapping and exploration of complex environments using persistent 3d model," in *Proc. 4th Can. Conf. Comput. Robot Vis.*, May 2007, pp. 403–410.
- [8] N. Fairfield, G. Kantor, and D. Wettergreen, "Real-time slam with octree evidence grids for exploration in underwater tunnels," *J. Field Robot.*, vol. 24, nos. 1/2, pp. 3–21, 2007.
- [9] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [10] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 2276–2282.
- [11] S. Thrun, "Learning occupancy grids with forward models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, vol. 3, pp. 1676–1681.
- [12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Ser. B (Methodological)*, vol. 39, pp. 1–38, 1977.
- [13] S. Thrun *et al.*, "A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 433–443, Jun. 2004.
- [14] M. Veeck and W. Veeck, "Learning polyline maps from range scan data acquired with mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 2, pp. 1065–1070.
- [15] L. Pedraza, D. Rodriguez-Losada, F. Matia, G. Dissanayake, and J. V. Miró, "Extending the limits of feature-based slam with b-splines," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 353–366, Apr. 2009.
- [16] M. Paskin and S. Thrun, "Robotic Mapping with Polygonal Random Fields," in *Proc. Twenty-First Conf. Uncertainty Artif. Intell.*, Jul. 26–29, 2005, pp. 450–458.
- [17] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 3, pp. 2743–2748.
- [18] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.
- [19] J. P. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, "3D normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1627–1644, 2013.
- [20] S. Srivastava and N. Michael, "Approximate continuous belief distributions for precise autonomous inspection," in *Proc. IEEE Int. Symp. Saf. Secur. Rescue Robot.*, Oct. 2016, pp. 74–80.
- [21] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *Int. J. Robot. Res.*, vol. 31, no. 1, pp. 42–62, 2012.
- [22] S. Kim and J. Kim, "GPmap: A unified framework for robotic mapping based on sparse gaussian processes," in *Field and Service Robotics*. New York, NY, USA: Springer, 2015, pp. 319–332.
- [23] V. Tresp, "A bayesian committee machine," *Neural Comput.*, vol. 12, no. 11, pp. 2719–2741, 2000.
- [24] S. Kim *et al.*, "Recursive Bayesian updates for occupancy mapping and surface reconstruction," in *Proc. Australas. Conf. Robot. Automat.*, 2014, pp. 354–361.
- [25] J. Wang and B. Englot, "Fast, accurate gaussian process occupancy maps via test-data octrees and nested Bayesian fusion," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2016, pp. 1003–1010.
- [26] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1717–1730, 2016.
- [27] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1177–1184.
- [28] C. K. I. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst.*, 2000, pp. 661–667. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3008751.3008847>.
- [29] V. Guizilini and F. Ramos, "Large-scale 3d scene reconstruction with hilbert maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2016, pp. 3247–3254.
- [30] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical dirichlet processes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1385–1392.
- [31] B. Eckart, K. Kim, A. Troccoli, A. Kelly, and J. Kautz, "Accelerated generative models for 3d point cloud data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5497–5505.
- [32] B. Eckart, K. Kim, A. Troccoli, and A. Kelly, "MLMD: Maximum likelihood mixture decoupling for fast and accurate point cloud registration," in *Proc. IEEE Int. Conf. 3D Vis.*, 2015, pp. 241–249.
- [33] S. Srivastava and N. Michael, "Approximate continuous belief distributions for precise autonomous inspection," in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, 2016, pp. 74–80.
- [34] B. Eckart and A. Kelly, "Rem-seg: A robust em algorithm for parallel segmentation and registration of point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4355–4362.
- [35] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
- [36] J. Goldberger, S. Gordon, and H. Greenspan, "An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 487–493.
- [37] V. Satopää, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a needle in a haystack: detecting knee points in system behavior," in *Proc. 31st IEEE Int. Conf. Distrib. Comput. Syst. Workshops*, 2011, pp. 166–171.
- [38] P. M. Engel and M. R. Heinen, "Incremental learning of multivariate gaussian mixture models," in *Proc. Braz. Symp. Artif. Intell.*, 2010, pp. 82–91.
- [39] H. G. Sung, "Gaussian Mixture Regression and Classification," Ph.D. dissertation, Rice Univ., Houston, TX, USA, 2004.
- [40] K. Mardia, J. Kent, and J. Bibby, "Multivariate analysis," *Ser. Probability Math. Stat.*, Academic Press, 1979. [Online]. Available: <https://books.google.com/books?id=bxjvAAAAMAAJ>
- [41] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. Int. Conf. Intell. Robot Syst.*, Oct. 2012, pp. 573–580.
- [42] M. Quigley, J. Faust, T. Foote, and J. Leibs, "ROS: An open-source robot operating system," in *Proc. Int. Conf. Robot. Autom. Workshop Open Source Softw.*, vol. 3, no. 3.2, May 2009, p. 5.
- [43] P. Yalamanchili *et al.*, "ArrayFire - A high performance software library for parallel computing with an easy-to-use API," Atlanta, 2015. [Online]. Available: <https://github.com/arrayfire/arrayfire>.
- [44] M. C. Altinigneli, C. Plant, and C. Böhm, "Massively parallel expectation maximization using graphics processing units," in *Proc. 19th ACM Int. Conf. Knowl. Discovery Data Min.*, 2013, pp. 838–846.



Shobhit Srivastava received the Bachelor's degree in computer science from National Institute of Technology, Allahabad, India, and the Master's degree in robotics from the Robotics Institute, Carnegie Mellon University (CMU), Pittsburgh, PA, USA, in 2017.

At CMU, he worked with the Resilient Intelligent Systems Lab on machine vision and perception. Since graduating in 2017, he has been a member of the Autonomy team at Shield AI. His primary research interest include robust perception to enable large scale intelligent autonomy. Prior to joining CMU, he was a part of the Corporate R&D Department, Qualcomm, where he worked on mobile platform virtualization, optical character recognition, and image stabilization.



Nathan Michael received the Ph.D. degree in mechanical engineering from University of Pennsylvania, Pennsylvania, in 2008. He is an Associate Research Professor with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA; the Director of the Resilient Intelligent Systems Lab; author of more than 160 publications on control, perception, and cognition for resilient intelligent single and multirobot systems. He develops resilient intelligent autonomous systems capable of individual and collective intelligence through introspection, adaptation, and evolution in challenging domains.

Mr. Michael is a nominee or recipient of nine best paper awards; recipient of the Popular Mechanics Breakthrough Award and Robotics Society of Japan Best Paper Award (of 2014); PI of past and ongoing research programs supported by ARL, AFRL, DARPA, DOE, DTRA, NASA, NSF, ONR, and industry; and Chief Science Officer of Shield AI.