Incremental Multimodal Surface Mapping via Self-Organizing Gaussian Mixture Models

Kshitij Goel and Wennie Tabib

***Abstract*— This letter describes an incremental multimodal surface mapping methodology, which represents the environment as a continuous probabilistic model. This model enables high-resolution reconstruction while simultaneously compressing spatial and intensity point cloud data. The strategy employed in this work utilizes Gaussian mixture models (GMMs) to represent the environment. While prior GMM-based mapping works have developed methodologies to determine the number of mixture components using information-theoretic techniques, these approaches either operate on individual sensor observations, making them unsuitable for incremental mapping, or are not real-time viable, especially for applications where high-fidelity modeling is required. To bridge this gap, this letter introduces a spatial hash map for rapid GMM submap extraction combined with an approach to determine relevant and redundant data in a point cloud. These contributions increase computational speed by an order of magnitude compared to state-of-the-art incremental GMM-based mapping. In addition, the proposed approach yields a superior tradeoff in map accuracy and size when compared to state-of-the-art mapping methodologies (both GMM- and not GMM-based). Evaluations are conducted using both simulated and real-world data. The software is released open-source to benefit the robotics community.**

I. Introduction

Robotic exploration systems are being deployed to automate multimodal data collection for applications like artifact detection [1], active thermal mapping [2], and planetary exploration [3]. For example, multi-instrument payloads (e.g. range and thermal sensors) are critical for mapping planetary caves [4]. These missions require an autonomous agent to incrementally create a mathematical model of the environment using onboard sensors and computers. The model must accurately represent fine details to enable operation in close proximity to complex, unknown structures (e.g. stalactites, thin wires, etc) and be compact to transfer via low-bandwidth communication channels [4]. Prior work has demonstrated that the exploration performance of a multi-agent team is impacted by the compactness of the environment model used in communications restricted environments due to the ability (or lack thereof) to share information [5, 6].

Few environment models are compact while enabling high-resolution reconstruction and safe navigation. Octomap [7], Voxblox [8], and GMM maps [9] are the key methodologies that have been used recently for communication-constrained exploration by Agha et al. [10], Tranzatto et al. [1], and Goel et al. [5], respectively. These methods enable safe navigation but suffer from the same limitation in terms of fixing the highest achievable map fidelity throughout exploration (e.g., minimum leaf size for Octomap, voxel size for Voxblox, and number of mixture components for GMM maps). This is inefficient because all parts of the environment are assigned the same level of highest fidelity, leading to larger map sizes and communication inefficiency [6].

The authors are with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA {kshitij,wtabib}@cmu.edu

Fig. 1: A reconstructed point cloud using spatial and intensity information inferred from the compact multimodal point cloud model created using the proposed approach. The representation leverages a formulation that has been demonstrated to be amenable for higher level robot autonomy objectives like exploration in complex, unstructured 3D environments. A video is available at: https://youtu.be/VgPEEcbUAnY.

To enable large-scale exploration with many robots simultaneously sharing information, the mapping algorithm must adapt to the scene complexity. Our recent work [11], the Self-Organizing Gaussian Mixture Models (SOGMMs), proposes an information-theoretic approach for automatically determining the number of mixture components from the underlying sensor data; however, the approach operates on single observations. Because consecutive sensor observations have significant overlap, the formulation in [11] is not well-suited for incremental mapping.

Prior GMM mapping approaches not demonstrated onboard a robot either require iterating over the mixture components [12] or use approximate geometric projection methods [13, 14]. The computational requirements for these methods increase prohibitively with the number of components in the GMM map. To bridge these gaps, we extend the

approach of [11] and provide the following novel contributions (Fig. 1):

1) a methodology which alleviates the computational burden of submap extraction by innovating a spatial hash table of mixture components;
2) an efficient method to incrementally update the global environment model leveraging the log-likelihood of the point cloud data with respect to the extracted submap;
3) extensive evaluations on simulated and real-world datasets with comparisons against state-of-the-art approaches (both GMM-based and not GMM-based); and
4) an open-source release of the approach for the benefit of robotics research and industry.

The letter proceeds with an overview of related work (Section II). Section III presents the proposed approach and Section IV details the results. Limitations of the approach are detailed in Section V and the letter is concluded with Section VI.

II. Related Work

This section reviews mapping methods that enable high-resolution multimodal reconstruction and occupancy modeling for 3D robotic exploration.

Early exploration works leveraged discrete occupancy grid maps [15] for rapid volumetric querying [16, 17]. However, voxel grids are subject to aliasing due to the independence assumption between cells (voxels) leading to poor multimodal reconstruction. To alleviate the effects of aliasing, smaller voxel sizes can be used; however, the memory footprint substantially increases as the number of voxels scales cubically according to the environment extents.

To mitigate the memory challenges, Hornung et al. [7] leverages octrees to enable a multi-resolution and hierarchical volumetric representation. This method has been used by recent exploration frameworks due to its scalability improvements over the occupancy grids at small voxel sizes [1, 18]. However, the insertion cost for new point clouds into Octomap and the cost of voxel state access (for the purposes of informative planning) is higher than occupancy grids. To decrease these costs while leveraging a hierarchical approach, the OpenVDB method by Museth [19] utilizes B+ trees, which yields increased efficiency in frontier extraction at large spatial scales [20]. An alternate approach by Oleynikova et al. [8] called Voxblox enables superior, constant-time, insertion and queries to the voxel grid while enabling informative planning and multimodal reconstruction capabilities [1]. Voxblox uses a regular occupancy grid and Truncated Signed Distance Fields (TSDFs) to approximately alleviate the aliasing due to the discrete representation. However, in terms of communication efficiency the representation suffers from the same drawbacks as occupancy grids. In this work, we present a mapping method that achieves higher multimodal reconstruction accuracy than Voxblox and Octomap through compact and continuous point cloud models.

Saarinen et al. [21] motivate the development of NDTMap, which uses a Gaussian density in each cell, by arguing that larger voxels may be used since the Gaussian density better approximates the surface geometry. However, this representation also suffers from the aliasing challenges of a regular occupancy grid as each Gaussian density is considered an independent component of a uniformly-weighted Gaussian mixture model (GMM). [9, 22] relax the assumption of uniform weights, by using a maximum-likelihood fit over the point cloud data to create a global map that is represented as a GMM without the use of a discrete grid. However, these works require specifying the number of mixture components before operation which limits the maximum achievable fidelity of the map. We bridge this gap by proposing a GMM-based approach that enables creating a map representation that increases the model fidelity incrementally via an information-theoretic self-organizing approach [11] and enables scalable and efficient inference via spatial hashing.

Neural Radiance Fields (NeRFs) [23] enable photorealistic environment rendering at lower memory costs; however, incremental mapping with implicit representations are known to suffer from catastrophic forgetting [24]. Catastrophic forgetting is the problem of forgetting old knowledge after training with new data [25]. To mitigate this issue, some incremental NeRF mapping approaches [26, 27] retain keyframes from historical data and replay them with current data to train the network; however, this approach requires more memory to store the keyframes [24]. Zhong et al. [24] develop a technique for large scale Signed Distance Field (SDF) mapping using LiDAR, but it is not clear how robust it will be to catastrophic forgetting when intensity data is incorporated. In contrast, the proposed approach adaptively increases the fidelity of the parametric GMM-based environment model depending on the scene complexity. This way, while offering an implicit representation of the surface, no special consideration for catastrophic forgetting is required.

Finally, note that the current state-of-the-art in radiance field rendering in the computer graphics literature leverages 3D Gaussian densities in place of neural networks [28, 29], demonstrating superior performance in both training and inference compared to NeRFs. Our method uses a mixture of 4D Gaussian densities to jointly model intensity and spatial data. Therefore, incorporating radiance field rendering within the proposed incremental mapping method is an exciting direction for future work.

III. Approach

This section provides details of the mapping methodology. The incremental mapping methodology is presented in Section III-A and Section III-B describes the inference method used for multimodal surface reconstruction.

Notation. We follow the notation from [11], which uses small letters for scalars (e.g., x, y), bolded small letters for vectors (e.g., $\mathbf{x}$, $\mathbf{y}$), bolded capital letters for matrices (e.g., $\mathbf{X}$, $\mathbf{Y}$), and calligraphic letters for sets (e.g., $\mathcal{I}$, $\mathcal{X}$, $\mathcal{Y}$).

A. Incremental Multimodal Surface Mapping

Before describing the algorithm, three key data structures are described: the Local SOGMM, Global SOGMM, and Spatial Hash table.

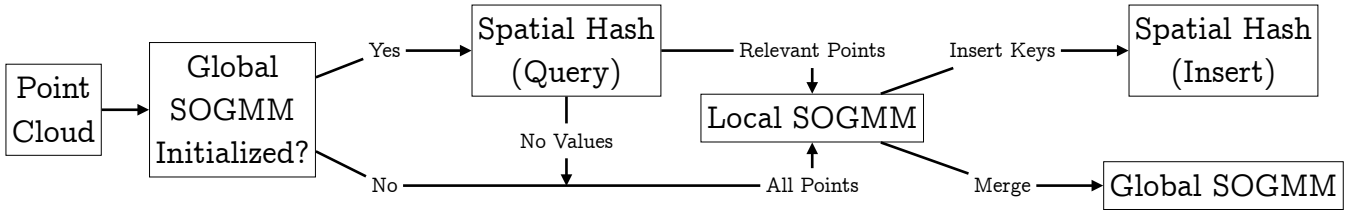


Fig. 2: Information flow during surface point cloud modeling via the proposed incremental mapping approach (Section III-A).

Local SOGMM. A GMM model created via the SOGMM method using the relevant points corresponding to the latest multimodal point cloud. Each point $\mathbf{z}$ in this point cloud is assumed to be of the form, $\mathbf{z} = \{(\mathbf{x}, i) \mid \mathbf{x} \in \mathbb{R}^3, i \in [0.0, 1.0]\}$. Formally, the GMM model is represented as the function $p_L \equiv p_L(\mathbf{z}) = \sum_{j \in \mathcal{J}} \pi_j \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, where π_j, $\boldsymbol{\mu}_j$, and $\boldsymbol{\Sigma}_j$ are the weight, mean, and covariance for the mixture component associated with index j in p_L. Each mixture component is a Gaussian probability density $\mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$. The set of indices is denoted by $\mathcal{J}$. The sum of weights must be 1, $\sum_{j \in \mathcal{J}} \pi_j = 1$, for a valid p_L.

Global SOGMM. A GMM model created after merging all prior p_L models. The global model contains $|\mathcal{K}| \geq |\mathcal{J}|$ mixture components. Formally, $p_G \equiv p_G(\mathbf{z}) = \sum_{k \in \mathcal{K}} \tau_k \mathcal{N}(\mathbf{z} \mid \boldsymbol{\nu}_k, \boldsymbol{\Lambda}_k)$, where different index and parameter symbols are used to notationally differentiate p_G from p_L. Similar to p_L, the set of indices is denoted by $\mathcal{K}$ and $\sum_{k \in \mathcal{K}} \pi_k = 1$ must hold for a valid p_G.

Spatial Hash. A hash table $H : \mathcal{M} \to \mathcal{Q}, m \mapsto \mathbf{q}^{\mathcal{A}}[h(m)]$ is created to map any point in 3D space (key $m \in \mathcal{M}$) to a vector of mixture component indices in p_G (value $\mathbf{q}^{\mathcal{A}} \in \mathcal{Q}$) that are within a fixed volume around the point. This fixed volume is a cube with side length α. The hash function $h : \mathcal{M} \to \mathcal{A}$ maps the keys to an index set $\mathcal{A} = \{0, 1, \ldots, |\mathcal{A}| - 1\}$ used to insert into and query from H.

Figure 2 provides an overview of the multimodal surface mapping method. There are three steps: (1) creating p_L, (2) merging p_L into p_G, and (3) spatially hashing p_L into H. Details of each step are provided in the following sections.

1) Creating p_L: For each point cloud $\mathcal{Z}$, the relevant subset of the point cloud $\mathcal{Z}^r$ is determined. If the number of points in this set is greater than a pre-specified threshold, then p_L is created; otherwise, $\mathcal{Z}^r$ is cached and used along with the subsequent frames. $\mathcal{Z}^r$ represents points not already modeled by p_G. Therefore, if p_G is not initialized, all points are treated as relevant ($\mathcal{Z}^r = \mathcal{Z}$). Otherwise, $\mathcal{Z}^r$ is determined using a threshold (ϕ) on the log-likelihood [12] that points $\mathcal{Z}$ originated from the model p_G,

$$\mathcal{Z}^r = \{\mathbf{z} \in \mathcal{Z} \mid \mathcal{L}(\mathbf{z}) = \ln\left(p_G(\mathbf{z})\right), \mathcal{L}(\mathbf{z}) < \phi\}. \quad (1)$$

However, this approach has two drawbacks when used with multimodal point clouds. First, thresholding the log-likelihood scores via Eq. (1) over the multimodal point cloud directly does not yield the intended $\mathcal{Z}^r$ as the 4^{th} dimension contains intensity data, which is not in the metric space of the other three dimensions. Second, as the size of the model p_G (i.e., the value of K) increases over time, the time complexity of calculating Eq. (1) increases linearly. Performing this computation for all points in $\mathcal{Z}$ is prohibitive for real-time operation on computationally-constrained robotic systems.

The first challenge is addressed by utilizing the marginal probability density $p(\mathbf{x})$ instead of $p(\mathbf{z})$ for the log-likelihood calculation, i.e.,

$$\mathcal{Z}^r = \{\mathbf{z} \in \mathcal{Z} \mid \mathbf{z} = (\mathbf{x}, i), \mathcal{L}(\mathbf{x}) < \phi\} \text{ where,} \quad (2)$$

$$\mathcal{L}(\mathbf{x}) = \ln\left(\sum_{k \in \mathcal{K}} \tau_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\nu}_k^{\mathbf{x}}, \boldsymbol{\Lambda}_k^{\mathbf{xx}})\right), \quad (3)$$

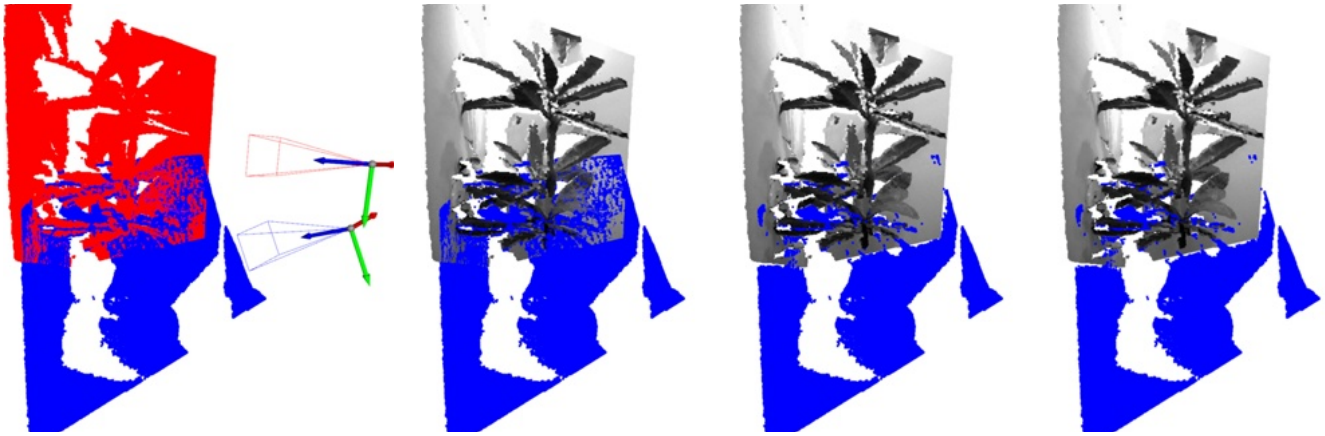
$$\boldsymbol{\nu}_k = [\boldsymbol{\nu}_k^{\mathbf{x}}, \boldsymbol{\nu}_k^{i}]^\top, \text{ and } \boldsymbol{\Lambda}_k = \begin{bmatrix} \boldsymbol{\Lambda}_k^{\mathbf{xx}} & \boldsymbol{\Lambda}_k^{\mathbf{x}i} \\ \boldsymbol{\Lambda}_k^{i\mathbf{x}} & \boldsymbol{\Lambda}_k^{ii} \end{bmatrix}. \quad (4)$$

Figure 3 details the effect of using Eq. (2) instead of Eq. (1). The value of ϕ is determined empirically, as done in [31], and it is fixed for the synthetic and real-world scenarios.

The second challenge is addressed by selecting only the subset of mixture components from p_G (i.e., selecting $\mathcal{B} \subseteq \mathcal{K}$) that are overlapped by or close to the points in $\mathcal{Z}$. This way we can reduce the number of summands in Eq. (3). The hash table H is leveraged for this purpose. Each 3D point $\mathbf{x}$ from $\mathbf{z} \in \mathcal{Z}$ is a key m for the hash function h that is used to search H for the closest vector of mixture component indices, $\mathbf{q}^{\mathcal{A}}[h(m)]$. After attaining these vectors for all points in $\mathcal{Z}$, the unique set of mixture component indices form the index set $\mathcal{B}$. For the example scenario in Fig. 3, Fig. 3d shows the output after this approximation. The output is similar to the case when the original set of components $\mathcal{K}$ (Fig. 3c) is used but $\mathcal{B}$ is smaller (480 elements instead of 1165). Consequently, for this example the time taken to compute Eq. (3) with $\mathcal{K}$ is $0.35\,\text{s}$ whereas with $\mathcal{B}$ it is $0.25\,\text{s}$ (28% faster). Note that the ratio $|\mathcal{K}|/|\mathcal{B}|$ grows over time as the size of p_G increases when point clouds from new regions are observed. An analysis of the computational savings using this approach is provided in Section IV-A.

2) Merging p_L into p_G: After the p_L model is created using $\mathcal{Z}^r$, it is merged with the global point cloud model p_G by appending the parameters and re-normalizing the weights. Let the global model before merging be p_G^t and after merging be p_G^{t+1}. The parameters for p_G^{t+1} are given by $\boldsymbol{\tau}^{t+1} = [\boldsymbol{\tau}^t, \boldsymbol{\pi}]^\top$ such that $\sum_{b \in \mathcal{B}} \tau_b^{t+1} = 1$, $\boldsymbol{\nu}^{t+1} = [\boldsymbol{\nu}^t, \boldsymbol{\mu}]^\top$, and $\boldsymbol{\Lambda}^{t+1} = [\boldsymbol{\Lambda}^t, \boldsymbol{\Sigma}]^\top$. The index set for the global model is also updated and the number of components increase accordingly, $|\mathcal{K}|^{t+1} = |\mathcal{K}|^t + |\mathcal{J}|$.

3) Spatially Hashing p_L into H: In addition to the global model, the hash table H is updated using the mixture components from the latest local model p_L. A total of $|\mathcal{J}|$ hash keys are inserted into the table where each hash key m_j is the spatial part of the mean position $\boldsymbol{\mu}_j^{\mathbf{x}}$ along with points



(a) (b) (c) (d)

Fig. 3: Illustration of the relevant point cloud calculation using two multimodal point clouds, $\mathcal{Z}_1$ and $\mathcal{Z}_2$ (Section III-A.1). The objective is to find the relevant point cloud, $\mathcal{Z}_2^r$, from $\mathcal{Z}_2$ using p_G, which is created from $\mathcal{Z}_1$. (a) shows the 3D parts of these point clouds in different colors and the associated 3D poses. (b) shows $\mathcal{Z}_2^r$ and $\mathcal{Z}_1$ with intensity values, when Eq. (1) is used. (c) shows the same but when Eq. (2) is used. Notice that in the former case $\mathcal{Z}_2^r$ contains more misclassified points that overlap with p_G than in the latter case. (d) shows the output $\mathcal{Z}_2^r$ when only a subset ($|\mathcal{B}| = 480$) of components in p_G ($|\mathcal{K}| = 1165$) derived using the hash table H are used. This output is similar to (c). The point clouds are sourced from from the real-world Lounge dataset [30]. *This figure is best viewed in color.*

generated at constant probability ellipsoids corresponding to 68% (1-sigma), 95% (2-sigma) and 99.7% (3-sigma) of the data points, for all $j \in \mathcal{J}$.

The hash function h that maps m_j to an index in $\mathcal{A}$ is given by $h(m_j) \equiv h(\boldsymbol{\mu}_j^{\mathbf{x}}) = N_z(r(\boldsymbol{\mu}_j^{\mathbf{x}})N_x + c(\boldsymbol{\mu}_j^{\mathbf{x}})) + s(\boldsymbol{\mu}_j^{\mathbf{x}})$ such that, $r(\boldsymbol{\mu}_j^{\mathbf{x}}) = \lfloor(\boldsymbol{\mu}_j^y - \mathbf{o}^y)/\alpha\rfloor$, $c(\boldsymbol{\mu}_j^{\mathbf{x}}) = \lfloor(\boldsymbol{\mu}_j^x - \mathbf{o}^x)/\alpha\rfloor$, and $s(\boldsymbol{\mu}_j^{\mathbf{x}}) = \lfloor(\boldsymbol{\mu}_j^z - \mathbf{o}^z)/\alpha\rfloor$. Here, $[N_x, N_y, N_z]$ are the number of cells along each axis of a 3D regular grid of spatial resolution α, $\mathbf{o} = -\frac{1}{2}\alpha[N_x, N_y, N_z]$ is the origin position of this grid, and $\lfloor.\rfloor$ is the floor operator. Intuitively, the hash function h assigns the mean positions $\boldsymbol{\mu}^{\mathbf{x}}$ into a sparse grid of a pre-specified extent $[N_x, N_y, N_z]$ and resolution α.

The value corresponding to each key is the index of the component j in the global model after the merging step (Section III-A.2) is complete. Thus, the value corresponding to the hash key m_j is $|\mathcal{K}|^t + j$. It is possible that multiple hash keys are mapped to the same cell in the grid (i.e., hash collisions are possible). An example scenario is when α is large and a subset of means $\boldsymbol{\mu}^{\mathbf{x}}$ are spatially within α distance. In this case, we want to store all the values in a vector. This is why the value set $\mathcal{Q}$ is defined as a set of vectors as opposed to a set of integers. If a hash collision occurs for any two keys m_f and m_g (i.e., $h(m_f) = h(m_g)$), the values are appended into the vector $\mathbf{q}^{\mathcal{A}}[h(m_f)]$.

B. Global Spatial and Intensity Inference

Given the global model p_G, we want to reconstruct the environment spatially along with the intensity values. The marginal global model given by $\boldsymbol{\tau}$, $\boldsymbol{\nu}^{\mathbf{x}}$, and $\boldsymbol{\Lambda}^{\mathbf{xx}}$ (as defined by Eq. (4)) is used for spatial inference and densely sampled using the Box-Muller transform [32]. The conditional probability density $p(i \mid \mathbf{x})$ (as noted in [11]), is used to infer intensity at the sampled spatial points. This inference is performed in batches of components from p_G. The batch size is determined based on the available memory on the CPU used to perform inference. An example of the reconstruction obtained for the Living Room dataset [33] is shown in Fig. 1.

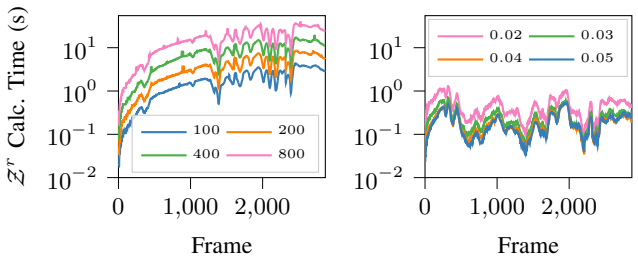
IV. Results

The experimental results are divided into two parts. In Section IV-A, the computational performance gain offered by the proposed incremental mapping approach due to the spatial hash formulation is compared with the prior work on GMM-based multimodal mapping [12]. In Section IV-B the reconstruction accuracy from the global map created through the proposed approach is compared with Octomap [7], Voxblox (Nvblox[1]) [8], and GMM-based maps that use a fixed number of components (FCGMM) [14]. These methods are chosen as baselines because they have been used for our target application, multi-robot 3D exploration. We will use the "Method-Parameter" notation to denote the parameter being used. For example, Proposed-0.02 denotes the proposed approach with the bandwidth parameter $\sigma = 0.02$.

One synthetic (D1: Living Room from the Augmented ICL-NUIM datasets [33]) and three real-world datasets (D2: Lounge, D3: Copyroom, and D4: Stonewall from the Redwood datasets [30]) are used for qualitative and quantitative evaluation of the proposed approach and its comparison with the baseline methods. All datasets contain 640×480 RGB and depth images along with the corresponding camera poses. All methods are provided reduced resolution 128×96 images for incremental mapping. The computer used for all experiments contains an Intel Core i9-10900K CPU (20 threads, maximum clock speed 5.3 GHz, 32 GB RAM) and a NVIDIA GeForce RTX 3060 GPU (12 GB RAM). The CPU implementation of the Octomap approach for color data is used. Nvblox and the proposed method[2] use the CPU and GPU for incremental mapping. Nvblox uses both

[1]Nvblox is the GPU-accelerated extension of Voxblox: `https://github.com/nvidia-isaac/nvblox`

[2]Release `0.1.0` of `https://github.com/gira3d/gira3d-reconstruction`.

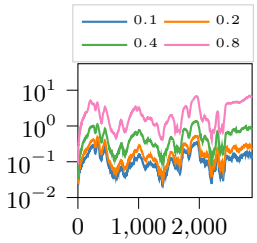


(a) Baseline [12] with varying $|\mathcal{J}|$

(b) Proposed with varying σ

Method	Time (s)	Δ
FCGMM-800	44751.6	–
FCGMM-400	20702.3	–
FCGMM-200	9957.6	–
FCGMM-100	4774.7	–
Proposed-0.02	1239.6	**36.1**×
Proposed-0.03	666.1	**31.0**×
Proposed-0.04	488.0	**20.4**×
Proposed-0.05	436.9	**10.9**×

(c) Cumulative time taken



(d) Ablation for α

Fig. 4: Comparison of the relevant subset $\mathcal{Z}^r$ calculation time between the prior work on multimodal GMM mapping [12] and the proposed approach. The per-frame calculation time in seconds is plotted for (a) different values of fixed numbers of components $|\mathcal{J}|$ and (b) different values of the bandwidth parameter σ for the proposed method. (c) Notice that the spatial hash (Section III-A.3) enables an order of magnitude improvement and that the performance gains increase monotonically with model size. (d) shows an ablation of calculation times for different values of the spatial hash resolution parameter α.

depth and color data by default. It is modified to use depth and grayscale images for the comparison presented in this section. Since the software for prior GMM map works [12, 14, 31] is not openly available, the codebase for the proposed approach is modified to use a fixed number of components for the FCGMM comparison. The FCGMM approach uses the GPU for EM execution but CPU for the $\mathcal{Z}^r$ calculation because it requires access to more RAM than is available to the GPU.

A. Relevant Point Cloud Calculation

As mentioned in Section III-A.1, a drawback of prior work [12] is that the relevant point cloud subset $\mathcal{Z}^r$ calculation in Eq. (1) is not real-time viable, especially as the number of components $|\mathcal{K}|$ increase in the global point cloud model p_G. The proposed spatial hashing approach reduces this computation cost by selecting a subset of components $\mathcal{B}$ that geometrically overlaps the point cloud. Figure 4 demonstrates the performance gains due to the proposed approach. While the methodology proposed in [12] is hierarchical, we choose to compare against their highest fidelity model (i.e., lowest layer) in the hierarchy for a fair comparison.

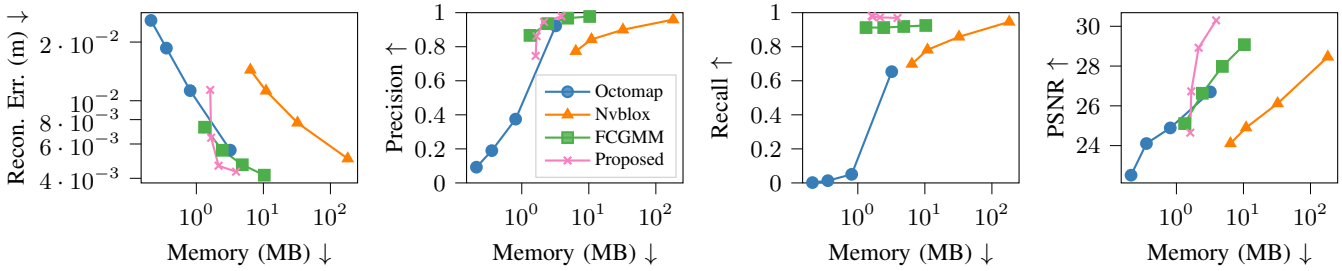
For the baseline approach (Fig. 4a), the calculation times per frame are shown for increasing values of number of mixture components $|\mathcal{J}|$, $\boldsymbol{J} = \{100, 200, 400, 800\}$. The proposed methodology enables an order of magnitude faster $\mathcal{Z}^r$ calculation as compared to the baseline approach, because the log-likelihood operates over all mixture components and points. The spatial hash resolution α is fixed at $0.2\,\mathrm{m}$ for all values of σ. Because lower σ yields higher resolution reconstruction, the computation time increases as σ decreases. Figure 4c presents the cumulative $\mathcal{Z}^r$ calculation times for the D1 dataset. The Δ columns show the order of magnitude improvement via the proposed approach. Notice that while the performance gain of Proposed-0.05 compared to FCGMM-100 is nearly $\mathbf{10}\times$, it increases with the model fidelity; the Proposed-0.02 is about $\mathbf{30}\times$ faster than FCGMM-800. Finally, Fig. 4d compares the $\mathcal{Z}^r$ calculation times for $\alpha = \{0.1, 0.2, 0.4, 0.8\}$. Increasing α results in overall higher calculation times because the size of $\mathcal{B}$ gets larger. Before using the proposed method, the value of α should be set according to the available CPU computation resources.

B. Global Map Accuracy and Compression

For Octomap and Nvblox, the predicted point cloud $\mathcal{Z}^{\text{pr}}$ and mesh are constructed, respectively, after processing all the frames in a given dataset. For FCGMM and the proposed approach, the predicted point cloud is inferred from the global model p_G using the method in Section III-B. The Octomap method requires specifying a minimum leaf size for the underlying octree used for modeling and inference. We use a range of leaf sizes for the experiments, $\boldsymbol{\alpha}_{\text{om}} = \{0.02, 0.04, 0.06, 0.08\}\mathrm{m}$. The same set of values are used for the voxel sizes required in the Nvblox method, $\boldsymbol{\alpha}_{\text{nv}} = \boldsymbol{\alpha}_{\text{om}}$. For FCGMM and the proposed approach, the same set of parameters are used as in Section IV-A. The ground truth point cloud, $\mathcal{Z}^{\text{gt}}$, is constructed by appending all the point clouds corresponding to the images and poses in the dataset followed by downsampling using a voxel grid filter with a small voxel size ($0.01\,\mathrm{m}$ for all experiments in this section).

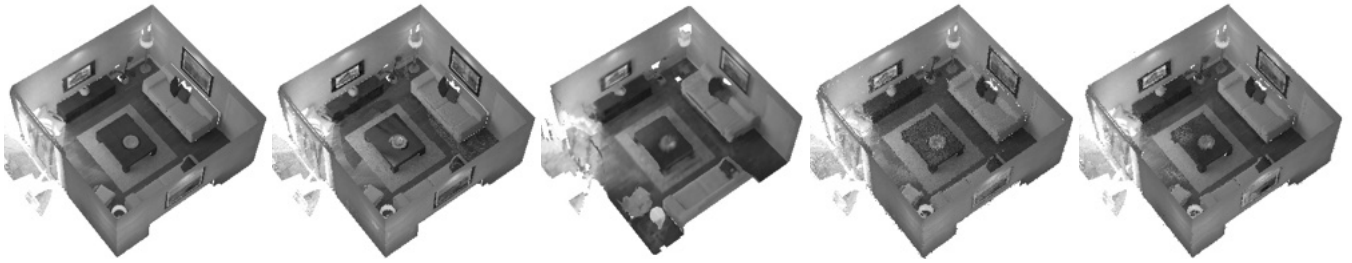
The performance measures for 3D reconstruction are (1) Mean Reconstruction Error (MRE), which is the average distance of the closest points between $\mathcal{Z}^{\text{pr}}$ and $\mathcal{Z}^{\text{gt}}$ (lower is better), (2) Precision of 3D reconstruction, which measures the fraction of points in $\mathcal{Z}^{\text{pr}}$ that lie within $0.01\,\mathrm{m}$ of a point in $\mathcal{Z}^{\text{gt}}$ (higher is better), and (3) Recall of 3D reconstruction, which measures the fraction of points in $\mathcal{Z}^{\text{gt}}$ that lie within $0.01\,\mathrm{m}$ of a point in $\mathcal{Z}^{\text{pr}}$ (higher is better). Intuitively, this measure computes the degree of "completeness" in the reconstruction. The performance measure for intensity reconstruction is the peak-signal-to-noise ratio (PSNR) calculated using the mean squared error (MSE) between the intensity values of the closest points in $\mathcal{Z}^{\text{pr}}$ and $\mathcal{Z}^{\text{gt}}$ (higher PSNR is better). These measures are computed using the closest point distance computation functions for point clouds in Open3D [34]. For the Nvblox case in particular, the mesh output is uniformly and densely sampled to create a point cloud with number of points equal to the ground truth point cloud.

The memory storage efficiency of the multimodal environment representations is measured by calculating the size (measured in bytes) of the models that can be loaded from disk to create $\mathcal{Z}^{\text{pr}}$. The models are chosen so that they can enable reconstruction of the surface and occupancy modeling for other robots in a multi-robot exploration scenario. For Octomap, the model size corresponds to the output `.ot`



(a) MRE (lower is better) (b) Precision (higher is better) (c) Recall (higher is better) (d) PSNR (higher is better)

Fig. 5: Quantitative comparison of (a) reconstruction error, (b) precision, (c) recall, and (d) PSNR as a function of the map size in megabytes (MB) for each approach. The dataset under consideration is the synthetic D1 dataset shown in Fig. 6a. Note that the proposed approach yields a map that requires less disk space than the competing methods while demonstrating at par or better reconstruction accuracy (i.e., low reconstruction error and high precision).



(a) Synthetic Dataset (D1) (b) Octomap (0.02 m, 3.2 MB) (c) Nvblox (0.08 m, 6.4 MB) (d) FCGMM (400, 4.8 MB) (e) Proposed (0.02, 2.1 MB)

(f) Real-World Dataset (D2) (g) Octomap (0.02 m, 5.9 MB) (h) Nvblox (0.06 m, 8.2 MB) (i) FCGMM (400, 6.2 MB) (j) Proposed (0.02, 5.7 MB)

Fig. 6: Qualitative comparison of the reconstructions obtained by baseline methods and the proposed approach at similar values of map size for (a) D1 and (f) D2 datasets. The highest achievable resolution used during execution and resulting map sizes are reported in the sub-captions. (b) and (g) visualize the lowest level of the Octomap octree. Incorrect intensity values are visible due to the color averaging within the octree. (c) and (h) illustrate the mesh extracted from the stored TSDF for Nvblox. Aliasing is visible in the meshes due to large voxel sizes required for a lower memory footprint. (d), (i) FCGMM and (e), (j) the proposed method enable qualitatively similar high-resolution dense reconstructions; however, the FCGMM output requires a much longer time to process incremental observations (see Fig. 4). A video of the proposed approach reconstructing the D1 dataset is available at `https://youtu.be/VgPEEcbUAnY`.

file [7], as opposed to the binary `.bt` file that does not retain occupancy information. Due to the same reason, for Nvblox the SQLite3 database (`.db` file) output is used instead of the output `.ply` mesh file. For the FCGMM and proposed methodologies, the memory occupied by the means, covariances, and weights in the global model p_G is calculated (four floats for each mean, one float for each weight, and ten floats for each covariance). Note that occupancy modeling from a stored GMM map with these parameters has been demonstrated in prior work [14, 31, 35, 36]

Figure 5 shows the variation in the performance measures for all methods with respect to the map size on disk for the D1 dataset. Each data point in the plots corresponds to a unique parameter setting in $\boldsymbol{\alpha}_{\text{om}}$, $\boldsymbol{\alpha}_{\text{nv}}$, $\boldsymbol{J}$, and $\boldsymbol{\sigma}$ for Octomap, Nvblox, FCGMM, and the proposed approach respectively. To attain similar levels of mean reconstruction error and precision, the proposed approach, FCGMM, and Octomap require an order of magnitude less memory than Nvblox (Figs. 5a and 5b). This is because Nvblox utilizes a regular grid of fixed resolution and multiple data storage layers while Octomap, FCGMM, and the proposed approach leverage octrees, GMMs, and SOGMMs, respectively. Note that while these values are close for the FCGMM and the proposed methods, in the FCGMM case the time taken to create the model is much higher (Section IV-A).

The FCGMM and the proposed approaches achieve a recall score close to 1.0 (Fig. 5c) demonstrating that for nearly each point in the ground truth point cloud, there is a point in the reconstruction within a 0.01 m ball. Both of these GMM-based methods outperform Octomap because an arbitrarily high number of points can be densely sampled from a GMM (Section III-B) whereas the Octomap outputs the point cloud at its minimum pre-specified leaf size. The Nvblox method output mesh is uniformly sampled; however, a low voxel size is required to achieve similar recall scores.

The highest intensity reconstruction accuracy (i.e., PSNR score in Fig. 5d) attained by Octomap (PSNR $= 26.70$) is much lower than the proposed approach (PSNR $= 30.29$)

Method	Param.	MRE (m) ↓	Prec. ↑	Rec. ↑	PSNR (dB) ↑	Mem. (MB) ↓
Octomap	0.02	0.007	0.81	0.46	18.37	5.94
	0.04	0.011	0.55	0.06	16.76	1.13
	0.06	0.016	0.41	0.02	15.84	0.44
	0.08	**0.021**	**0.33**	**0.01**	**15.30**	**0.23**
Nvblox	0.02	0.006	0.92	0.38	20.63	**114.19**
	0.04	0.008	0.86	0.35	19.08	19.98
	0.06	0.011	0.81	0.31	18.26	8.19
	0.08	0.014	0.76	0.28	17.51	4.22
FCGMM	800	**0.005**	**0.98**	**0.72**	20.25	14.11
	400	0.006	0.97	**0.72**	19.35	6.24
	200	**0.005**	0.96	**0.72**	18.64	2.98
	100	0.006	0.92	**0.72**	17.62	1.46
Proposed	0.02	**0.005**	**0.98**	0.65	**20.74**	5.71
	0.03	**0.005**	0.97	0.64	20.18	2.39
	0.04	0.006	0.96	0.65	19.76	1.32
	0.05	0.006	0.93	0.65	19.12	0.83

(a) Performance measures for the D2: Lounge dataset

Method	Param.	MRE (m) ↓	Prec. ↑	Rec. ↑	PSNR (dB) ↑	Mem. (MB) ↓
Octomap	0.02	0.007	0.83	0.51	18.61	5.93
	0.04	0.011	0.57	0.06	17.35	1.10
	0.06	0.016	0.44	0.02	16.49	0.43
	0.08	**0.021**	**0.36**	**0.01**	**15.95**	**0.23**
Nvblox	0.02	0.006	0.92	0.38	20.94	**96.95**
	0.04	0.009	0.87	0.33	19.71	18.61
	0.06	0.010	0.83	0.29	18.86	8.00
	0.08	0.010	0.81	0.26	18.29	4.40
FCGMM	800	–	–	–	–	–
	400	**0.004**	0.97	**0.83**	20.66	10.68
	200	0.005	0.95	**0.83**	19.70	4.85
	100	0.007	0.91	**0.83**	18.59	2.30
Proposed	0.02	0.005	**0.98**	0.70	**21.42**	5.59
	0.03	0.005	0.97	0.68	20.96	2.48
	0.04	0.005	0.95	0.66	20.56	1.39
	0.05	0.006	0.93	0.66	20.05	0.88

(b) Performance measures for the D3: Copyroom dataset

Method	Param.	MRE (m) ↓	Prec. ↑	Rec. ↑	PSNR (dB) ↑	Mem. (MB) ↓
Octomap	0.02	0.007	0.84	0.62	19.19	5.29
	0.04	0.011	0.59	0.08	18.32	0.95
	0.06	0.016	0.47	0.02	17.61	0.37
	0.08	**0.021**	**0.39**	**0.01**	**17.79**	**0.19**
Nvblox	0.02	0.005	**0.99**	0.42	**24.80**	**155.05**
	0.04	0.005	0.98	0.39	23.41	25.96
	0.06	0.005	0.96	0.37	22.40	9.43
	0.08	0.006	0.94	0.34	21.43	4.89
FCGMM	800	0.005	**0.99**	**0.82**	22.13	13.34
	400	**0.004**	**0.99**	0.80	21.51	5.66
	200	0.005	0.98	0.79	20.99	2.60
	100	0.006	0.96	0.79	20.40	1.24
Proposed	0.02	0.005	**0.99**	0.68	21.83	3.27
	0.03	0.005	0.98	0.66	21.62	1.26
	0.04	0.005	0.97	0.65	21.41	0.63
	0.05	0.006	0.95	0.65	21.16	0.36

(c) Performance measures for the D4: Stonewall dataset

Fig. 7: Quantitative comparison of Octomap, Nvblox, FCGMM, and the proposed approach using the real-world datasets with noisy RGB-D data. The best and worst values in each column are colored green and red respectively. The FCGMM method results in a larger map size compared to the proposed approach and is orders of magnitude slower in execution time (Fig. 4). These results highlight that the proposed approach balances the accuracy and map size better than the state-of-the-art approaches.

at a similar storage cost. This is because the intensity in an Octomap octree node is averaged according to the density of the points around the node. In contrast, the proposed approach treats intensity as a univariate random variable and jointly modeled with the 3D coordinates in the global point cloud model. Inference from this joint probability density leads to a higher accuracy than the averaging in Octomap. Nvblox fuses intensity information into the 3D map using a weighted average update. This process is an improvement over Octomap but still requires a low voxel size to attain a PSNR comparable to GMM-based approaches. Finally, the FCGMM approach demonstrates a lower PSNR than the proposed approach for similar storage costs. This is because the FCGMM uses a fixed number of components for every scene in the dataset while the proposed method uses SOGMM, which adapts the number of components according to the complexity of depth and image data [11]. The impact of these quantitative results is visible in the qualitative comparison shown in Fig. 6. The reconstructions from all methods are shown for comparable map sizes along with the ground truth point cloud for D1 and D2 datasets.

Figure 7 provides performance statistics corresponding to real-world datasets D2, D3 and D4, which exhibit noisy sensor readings. For each performance measure, the best and worst values are highlighted in green and red, respectively. Note that there is no result for the D2 dataset in the FCGMM-800 case because the relevant point cloud calculation required more RAM than the available 32 GB. This is expected since the D2 dataset contains 5490 frames which is nearly twice the other datasets. While using Octomap at a 0.08 m resolution results in the lowest map size (0.23 MB), reconstruction performance is poor compared to all other methods. Nvblox outputs the largest map size on disk at 114.19 MB but does not enable the highest PSNR. The FCGMM and proposed approach enable similar reconstruction accuracy; however, the proposed approach results in smaller map sizes and utilizes less computation as shown earlier. This trend is observed for the D2 and D3 datasets as well. One exception for Nvblox is that it provides a higher PSNR compared to the proposed approach for the D3 dataset but it consumes about 50× more storage.

V. Limitations

The effects of changes in illumination in the scene are not explicitly considered and requires future work. The proposed method assumes drift-free sensor poses are available through a localization system. This assumption is consistent with prior work on multimodal mapping with GMMs [12]. Prior work in sensor localization via point cloud registration [37, 38] and loop closure [39] using GMMs may be leveraged for pose estimation.

VI. Conclusion

This letter detailed an incremental multimodal surface mapping methodology for high-resolution environment reconstruction. State-of-the-art GMM-based perceptual modeling approaches use a pre-specified number of components to enable mapping of environment surfaces, which is memory

inefficient. Inserting a new point cloud observation to an existing GMM map model involves iterating over all the mixture components; which is computationally expensive. To bridge these gaps, this paper formulated methodologies to (1) extract a submap by innovating a spatial hash table of mixture components and (2) incrementally update the global environment model in a computationally efficient manner. The approach was evaluated with synthetic and real-world datasets and the results demonstrated that the proposed approach enables high-fidelity reconstruction at low memory with an order of magnitude increase in speed compared to existing GMM-based mapping methods.

REFERENCES

[1] M. Tranzatto, F. Mascarich, L. Bernreiter *et al.*, "CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge," *FR*, vol. 2, no. 1, pp. 274–324, Mar. 2022.

[2] W. Tabib, R. Whittaker, and N. Michael, "Efficient multi-sensor exploration using dependent observations and conditional mutual information," in *2016 IEEE Int. Symp. Saf. Secur. Rescue Robot. SSRR*, Oct. 2016, pp. 42–47.

[3] A. Arora, P. M. Furlong, R. Fitch *et al.*, "Multi-modal active perception for information gathering in science missions," *Auton Robot*, vol. 43, no. 7, pp. 1827–1853, Oct. 2019.

[4] J. J. Wynne, T. N. Titus, A.-a. Agha-Mohammadi *et al.*, "Fundamental science and engineering questions in planetary cave exploration," *Journal of Geophysical Research: Planets*, vol. 127, no. 11, p. e2022JE007194, 2022.

[5] K. Goel, W. Tabib, and N. Michael, "Rapid and High-Fidelity Subsurface Exploration with Multiple Aerial Robots," in *Exp. Robot.*, ser. Springer Proceedings in Advanced Robotics, B. Siciliano, C. Laschi, and O. Khatib, Eds. Cham: Springer International Publishing, 2021, pp. 436–448.

[6] K. Ebadi, Y. Chang, M. Palieri *et al.*, "LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments," in *2020 IEEE Int. Conf. Robot. Autom. ICRA*, May 2020, pp. 80–86.

[7] A. Hornung, K. M. Wurm, M. Bennewitz *et al.*, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

[8] H. Oleynikova, Z. Taylor, M. Fehr *et al.*, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning," in *2017 IEEERSJ Int. Conf. Intell. Robots Syst. IROS*, Sep. 2017, pp. 1366–1373.

[9] W. Tabib, K. Goel, J. Yao *et al.*, "Real-Time Information-Theoretic Exploration with Gaussian Mixture Model Maps," in *Robot. Sci. Syst. XV*. Robotics: Science and Systems Foundation, Jun. 2019.

[10] A. Agha, K. Otsu, B. Morrell *et al.*, "NeBula: TEAM CoSTAR's Robotic Autonomy Solution that Won Phase II of DARPA Subterranean Challenge," *FR*, vol. 2, no. 1, pp. 1432–1506, Mar. 2022.

[11] K. Goel, N. Michael, and W. Tabib, "Probabilistic Point Cloud Modeling via Self-Organizing Gaussian Mixture Models," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2526–2533, May 2023.

[12] S. Srivastava, "Efficient, multi-fidelity perceptual representations via hierarchical gaussian mixture models," Master's thesis, Carnegie Mellon University, Pittsburgh PA, August 2017.

[13] A. Dhawale and N. Michael, "Efficient Parametric Multi-Fidelity Surface Mapping," in *Robot. Sci. Syst. XVI*. Robotics: Science and Systems Foundation, Jul. 2020.

[14] W. Tabib, K. Goel, J. Yao *et al.*, "Autonomous Cave Surveying With an Aerial Robot," *IEEE Trans. Robot.*, pp. 1–17, 2021.

[15] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[16] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proc. Second Int. Conf. Auton. Agents - AGENTS 98*. Minneapolis, Minnesota, United States: ACM Press, 1998, pp. 47–53.

[17] W. Burgard, M. Moors, D. Fox *et al.*, "Collaborative multi-robot exploration," in *Proc. 2000 ICRA Millenn. Conf. IEEE Int. Conf. Robot. Autom. Symp. Proc. Cat No00CH37065*, vol. 1, Apr. 2000, pp. 476–481 vol.1.

[18] Z. Zhang, T. Henderson, S. Karaman *et al.*, "FSMI: Fast computation of Shannon mutual information for information-theoretic mapping," *The International Journal of Robotics Research*, vol. 39, no. 9, pp. 1155–1177, Aug. 2020.

[19] K. Museth, "VDB: High-resolution sparse volumes with dynamic topology," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–22, Jun. 2013.

[20] S. Scherer, "Resilient and Modular Subterranean Exploration with a Team of Roving and Flying Robots," *Field Robot.*, 2022.

[21] J. P. Saarinen, H. Andreasson, T. Stoyanov *et al.*, "3D normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1627–1644, Dec. 2013.

[22] M. Corah, C. O'Meadhra, K. Goel *et al.*, "Communication-Efficient Planning and Mapping for Multi-Robot Exploration in Large Environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1715–1721, Apr. 2019.

[23] B. Mildenhall, P. P. Srinivasan, M. Tancik *et al.*, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *Comput. Vis. – ECCV 2020*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox *et al.*, Eds. Cham: Springer International Publishing, 2020, pp. 405–421.

[24] X. Zhong, Y. Pan, J. Behley *et al.*, "SHINE-Mapping: Large-Scale 3D Mapping Using Sparse Hierarchical Implicit Neural Representations," in *2023 IEEE Int. Conf. Robot. Autom. ICRA*, May 2023, pp. 8371–8377.

[25] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[26] E. Sucar, S. Liu, J. Ortiz *et al.*, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[27] J. Ortiz, A. Clegg, J. Dong *et al.*, "isdf: Real-time neural signed distance fields for robot perception," *arXiv preprint arXiv:2204.02296*, 2022.

[28] L. Keselman and M. Hebert, "Approximate Differentiable Rendering with Algebraic Surfaces," in *Comput. Vis. – ECCV 2022 17th Eur. Conf. Tel Aviv Isr. Oct. 23–27 2022 Proc. Part XXXII*. Berlin, Heidelberg: Springer-Verlag, Oct. 2022, pp. 596–614.

[29] B. Kerbl, G. Kopanas, T. Leimkuehler *et al.*, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139:1–139:14, Jul. 2023.

[30] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–8, Jul. 2013.

[31] S. Srivastava and N. Michael, "Efficient, Multifidelity Perceptual Representations via Hierarchical Gaussian Mixture Models," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 248–260, Feb. 2019.

[32] G. E. Box and M. E. Muller, "A note on the generation of random normal deviates," *The annals of mathematical statistics*, vol. 29, no. 2, pp. 610–611, 1958.

[33] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust Reconstruction of Indoor Scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5556–5565.

[34] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A Modern Library for 3D Data Processing," Jan. 2018.

[35] B. Eckart, K. Kim, A. Troccoli *et al.*, "Accelerated Generative Models for 3D Point Cloud Data," in *2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 5497–5505.

[36] C. O'Meadhra, W. Tabib, and N. Michael, "Variable Resolution Occupancy Mapping Using Gaussian Mixture Models," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2015–2022, Apr. 2019.

[37] B. Eckart, K. Kim, and J. Kautz, "HGMR: Hierarchical Gaussian Mixtures for Adaptive 3D Registration," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 705–721.

[38] W. Tabib, C. O'Meadhra, and N. Michael, "On-Manifold GMM Registration," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3805–3812, Oct. 2018.

[39] W. Tabib and N. Michael, "Simultaneous Localization and Mapping of Subterranean Voids with Gaussian Mixture Models," in *Field Serv. Robot.*, ser. Springer Proceedings in Advanced Robotics, G. Ishigami and K. Yoshida, Eds. Singapore: Springer, 2021, pp. 173–187.