# Online Learning of a Probabilistic and Adaptive Scene Representation

Zike Yan Xin Wang Hongbin Zha
Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
PKU-SenseTime Machine Vision Joint Lab
zike.yan@pku.edu.cn xinwang_cis@pku.edu.cn zha@cis.pku.edu.cn

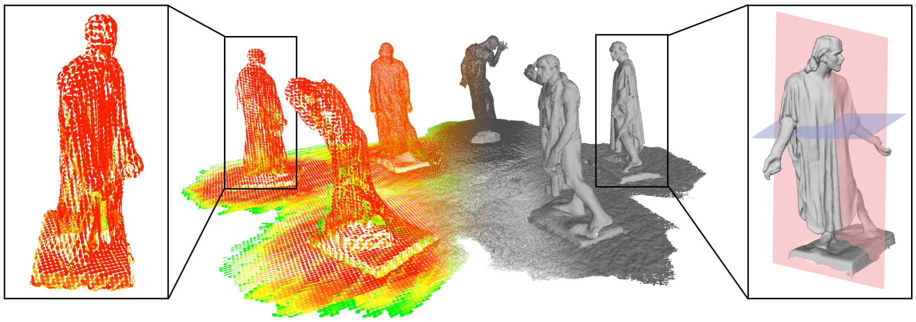


Figure 1: We propose a continuous probability field that can be learned incrementally from streaming data. The probabilistic formulation naturally incorporates both geometry and uncertainty information into a compact parameter space. The generative characteristic allows convenient conversion to different kinds of scene representations.

## Abstract

*Constructing and maintaining a consistent scene model on-the-fly is the core task for online spatial perception, interpretation, and action. In this paper, we represent the scene with a Bayesian nonparametric mixture model, seamlessly describing per-point occupancy status with a continuous probability density function. Instead of following the conventional data fusion paradigm, we address the problem of online learning the process how sequential point cloud data are generated from the scene geometry. An incremental and parallel inference is performed to update the parameter space in real-time. We experimentally show that the proposed representation achieves state-of-the-art accuracy with promising efficiency. The consistent probabilistic formulation assures a generative model that is adaptive to different sensor characteristics, and the model complexity can be dynamically adjusted on-the-fly according to different data scales.*

## 1. Introduction

Simultaneous Localization and Mapping (SLAM) has been recently viewed as a potential perceptual tool towards *Spatial AI* [9] as it allows a mobile device to perceive the world and estimate the sensor state. Along with the evolution of SLAM systems towards spatial perception arises an increasing demand for a more expressive map that can incrementally distill knowledge from different kinds of data into a compact parameter space. Finding an appropriate representation has been a central task of establishing such a comprehensive map.

In this paper, we aim to maintain a continuous probability field that allows for storing data into a unified probabilistic form. The probability field offers a generative extension to different spatial representations, *e.g.*, point cloud, occupancy grid, mesh at arbitrary resolution. Practically, we propose a continuous probability density function as the map representation using a Bayesian nonparametric mixture model. When obtaining 3D point cloud data, the scene geometry is depicted as a continuous probability field of spatial occupancy status. This representation owns the following properties: *1) Probabilistic.* The Bayesian fashion not only quantifies uncertainties explicitly, but also allows to incorporate all sorts of information from different sensor inputs in a unified probabilistic manner; *2) Adaptive and dynamic.* The nonparametric property offers an inherently infinite capacity [68] that guarantees an adaptive model complexity with respect to acquired data scale; *3) Compact and expressive.* The mixture model maintains a continuous and dense probability field in a discrete and sparse parameter space, where both geometry and uncertainty are kept within Gaussian components.

Specifically, we formulate the mapping as an online Bayesian learning problem: the map provides a generative process of the observations, and we use acquired streaming data to learn it incrementally. The incremental inference can be viewed as a transition from geometry prior to posterior given streaming data. As the posterior is intractable to compute and represent, we resort to a parallel and incremental approach. The global distribution is parallelly distributed to local processing in an incremental fashion, guaranteeing efficient inference for accurate scene geometry.

In summary, our major contributions include a novel scene representation using the Bayesian nonparametric mixture model and a principled way of online Bayesian learning for efficient map updating. Our method obtains a continuous high-quality scene representation incrementally, and achieves state-of-the-art results as demonstrated in the qualitative and quantitative experiments.

## 2. Related Work

In this paper, we introduce a novel scene representation that is probabilistic, adaptive, and can be learned incrementally from sequential data. Here we review the most closely related scene representations and indicate the major differences between ours.

### 2.1. 3D Scene Representations

Commonly-used 3D scene representations can be broadly categorized into three kinds: global function based, local primitive based, and neural representations.

**Global function based.** Global function based approaches represent the scene geometry as a continuous scalar field and maintain a global function to map *xyz* coordinate to the field. Signed distance function (SDF) is one commonly used implicit function to represent zero level-set surfaces. Prevalent approaches [7, 38] tend to discretize the space into regularly-partitioned voxel grids and directly maintain a discrete signed distance field. Though this volumetric representation is easy to manage and allows convenient rendering and data fusion, the signed distance function based approaches rely highly on the voxelization as it contains barely geometric property, hence struggling against scalability and flexibility [6, 41, 64, 8]. Besides, the continuity of the distance function is broken due to voxelization.

Another kind of approach represents the scene with a continuous probability density function (PDF) to maintain per-point occupancy probability, which is similar to our approach. Commonly used representations include Gaussian mixture model (GMM) [59, 16, 29, 44, 61] and Gaussian process (GP) [42, 43, 35, 62, 26]. GMM is commonly used as a compact generative model for scene geometry. The uncertainty-aware nature makes it appropriate for robust point cloud registration [15, 18, 17]. However, GMM requires a pre-defined number of mixtures, which is nontrivial to be applied for sequential data. On the other hand, Gaussian process is a Bayesian nonparametric model that is closely related to ours. Mapping with a Gaussian process is cast as a surface function regression problem. A similar idea is applied to Hilbert map [49, 20, 56] that projects observations into a reproducing kernel Hilbert space. However, online operation with the Gaussian process representation is restrained by the requirement to cache training data during inference and the computationally-burdensome inversion of covariance matrix [59]. In contrast, we achieve real-time performance through incremental and parallel inference.

**Local primitive based.** Local primitive based approaches represent a scene with a set of discrete geometric primitives. Inference on this kind of representation is performed by fitting local geometry, usually planar surfaces, with the primitives. Commonly used primitives include surfel [65, 53, 28], mesh triangle [10, 11], voxel grid [24], and 3D Gaussian (ellipsoid) [12].

Surfel [47] represents local geometry as an oriented disk. The unstructured nature makes it flexible for deformation and adaptive to different geometric frequencies. However, surfel is inherently sparse, thus leading to a discrete and incomplete scene model. Mesh provides a watertight surface model that is applicable for action and rendering. However, the topology changes for mesh representation are computationally expensive. Incremental mesh extraction is usually derived from other representations such as volumetric SDF [13] and surfel [54].

Another line of research maintains local occupancy status within a sparsely partitioned area. Octomap [24] represents local geometry with uncertainty-aware voxel grids. The uncertainty of occupied, free, or unknown status is assumed to be consistent within a voxel. [12] further maintains a set of unstructured ellipsoids parametrized by 3D Gaussians. Local geometry is assumed to share the same spatial distribution, where each ellipsoid is a 3D probabilistic extension of the surfel primitive. Normal distributions transform (NDT) [2, 51, 52, 55] can be viewed as a combination of voxel grid and 3D Gaussian. The occupancy status within a voxel is no longer a single scalar value but a more expressive Gaussian distribution. Though NDT is usually defined as a continuous representation from a voxelized GMM perspective, the voxel-wise local processing lacks a global constraint.

**Neural representations.** Neural representations learn to parameterize the shape manifold with neural networks. The insights behind neural representations usually derive from a view synthesis perspective [37, 40, 58] or from conventional representations mentioned above, *e.g.*, DeepSDF [45], PointGMM [23], ONet [36], LDIF [19]. The network is expected to learn class-specific shape priors that allow shape completion, interpolation, and generation. Though most works in the area are restricted to an object-level reconstruction, progress has been made recently that achieves detailed scene-level reconstruction [27, 46, 5]. However, these approaches are prevented from online operation with sequential data due to the batch-training fashion. Our method, on the other hand, adopts an efficient and incremental inference that resorts to an uncertainty-aware and interpretable Bayesian learning fashion [68].

## 2.2. Probabilistic 3D Data Fusion in Real-time

3D sequential data are usually redundant and noisy. To ensure scalable exploration and real-time action capability for geometry-dependent mobile devices, probabilistic fusion is performed to compress observed noisy data into a clean and compact form. Acquired data are usually assumed as Gaussian-distributed noisy observations. Hence, weighted averaging is required to incrementally update the representation parameters according to the data. Voxel-based representations, *e.g.*, volumetric TSDF [7], NDT [2], occupancy grid [24], assign each point to a voxel to update the corresponding geometric property, while unstructured representations such as surfel [65] and 3D Gaussian [12] assign each point to a geometric primitive through projective association.

Follow-ups further improve the robustness against noise and outliers by designing more reasonable weight calculations or introducing more complex distributions over the parameter space. Yan *et al.* [69] encode uncertainties into the surfel map by maintaining a 3D positional covariance and a 1D illuminational covariance. Lee *et al.* [31] utilize a more expressive Gaussian process over the SDF value to maintain a continuous implicit surface function. Dong *et al.* [14] add additional uniform distribution to handle outliers and explicitly model directional sensor noise. The literature in sensor measurement model [39, 25, 30, 34] is also vast, but the field is beyond our scope. Recently, RoutedFusion [63] proposes a 2D depth routing network and a 3D depth fusion network to learn non-linear TSDF updates in real-time and achieves state-of-the-art performance.

We, on the other hand, share a similar idea with [67] to learn a generative model of the observation process. By directly modeling a continuous spatial distribution, the uncertainty-aware characteristic is naturally incorporated in a theoretically-principled way, and a probabilistic framework is established systematically.

# 3. Overview

In this section, we introduce the general idea of how the proposed representation is learned incrementally in real-time. The mathematical formulation from an online learning perspective is first presented, followed by a scene representation definition. A parallel and incremental scheme for efficient inference is then introduced.

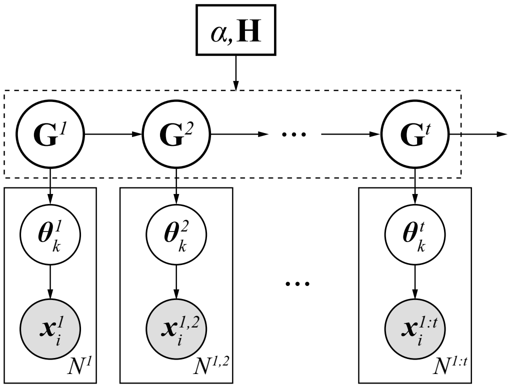


Figure 2: The construction process of the proposed mixture model. Arrows indicate the conditional dependence. Hyper-parameters $\{\alpha, \mathbf{H}\}$ enforce a globally-consistent constraint. The measured probability field $\mathbf{G}^t$ progressively evolves as new data streamed in.

## 3.1. Problem Formulation

We aim to maintain a spatial distribution $\mathbf{G}$ to represent the scene geometry. Let $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^t, \cdots\}$ be the streaming observations, where each set $\mathbf{X}^t$ consists of $N^t$ data points $\mathbf{x}_i^t \in \mathbf{X}^t$. We assume that observed data are i.i.d. samples drawn from the global distribution. The objectiveness is then to maintain and update a parameter space $\boldsymbol{\theta}_k^t \in \mathbf{\Theta}^t$ incrementally as the measurement of the spatial distribution. $\mathbf{\Theta}^t$ can be estimated by computing the posterior through Bayesian theorem recursively as:

$$p(\mathbf{\Theta}^t|\mathbf{X}^{1:t}) = \frac{p(\mathbf{X}^t|\mathbf{\Theta}^t)p(\mathbf{\Theta}^t|\mathbf{X}^{1:t-1})}{p(\mathbf{X}^t|\mathbf{X}^{1:t-1})}. \tag{1}$$

Under a Markov assumption, $\mathbf{X}^t$ is independent of $\mathbf{X}^{1:t-1}$. The posterior can then be transformed as:

$$p(\mathbf{\Theta}^t|\mathbf{X}^{1:t}) \propto \prod_{i=1}^{N^t} p(\mathbf{x}_i^t|\mathbf{\Theta}^t)p(\mathbf{\Theta}^t). \tag{2}$$

Through Eq. 2, online Bayesian learning can be understood as a gradual transition from the geometric prior $p(\mathbf{\Theta}^t)$ to the posterior $p(\mathbf{\Theta}^t|\mathbf{X}^{1:t})$. Knowledge is incrementally learned from data, describing the generative process of streaming observations $\mathbf{X}^{1:t}$ under the routine of Bayesian theorem.

### 3.2. Scene Representation

In this paper, we introduce a Dirichlet process (DP) mixture model as the scene representation. As illustrated in Fig. 2, the generative procedure of observations is well-explained by the model construction as:

$$\mathbf{x}_i^t \sim \boldsymbol{\theta}_k^t, \boldsymbol{\theta}_k^t \sim \mathbf{G}^t, \mathbf{G}^t \sim \mathrm{DP}(\alpha, \mathbf{H}), \tag{3}$$

where $\boldsymbol{\theta}_k^t \in \boldsymbol{\Theta}^t$ is the $k$th mixture component parameterized by $[\omega_k^t, \boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t]$. $\mathbf{G}^t$ is the global distribution measurement at time $t$. The concentration parameter $\alpha$ determines the sensitivity of component instantiation: the larger $\alpha$ leads to an easier instantiation strategy. On the other hand, the base distribution $\mathbf{H}$ determines the initialization of the newly-instantiated component.

In the view of mixture construction, $\mathbf{G}^t$ is a distribution over sparse partitions [48] and can be discretized into countably-infinite components. Hence, it is usually viewed as an infinite-dimensional extension of the Gaussian mixture model [50]. The model can be intuitively understood using a Chinese restaurant process (CRP) metaphor: the $i$th customer $\mathbf{x}_i^t$ walks into a Chinese restaurant with an infinite number of tables and choose to sit at an already occupied table $\boldsymbol{\theta}_k^t$ or a new table $\boldsymbol{\theta}_{K^t+1}^t$. For our case, the sequential construction of the mixture model is handled with dynamic components creation and deletion [33], thus leading to an adaptive model complexity according to the data scale.

### 3.3. Online Bayesian Learning

One bottleneck for Bayesian nonparametric learning lies in the fact that our objective posterior is intractable to compute and represent. We here resort to a parallel and incremental inference: the streaming data can be viewed as a sequence of mini-batches that arrive at consecutive epochs [3, 1]. Each subset of data is then assigned to a thread-safe processing unit for local inference. Hence, the Dirichlet process mixture model is re-parameterized as a mixture of DPs, where inference on each DP is performed in parallel with the associated mini-batch data stream.

Let $\pi_i^t = j$ be the processor indicator for each observation $\mathbf{x}_i^t$ and $J^t$ be the number of processors to be allocated at time $t$. Assuming that data inside each epoch are exchangeable [1] and thus conditionally independent, our objective posterior in Eq. 1 can be decomposed into multiple local DPs. Following AVparallel [66], the generative procedure of the mixture model in Eq. 3 can be re-written as a mixture of DPs:

$$\mathbf{G}^t = \sum \phi_j \mathbf{G}_j^t \sim \mathrm{DP}(\sum \alpha_j, \frac{\sum \alpha_j \mathbf{H}_j)}{\sum \alpha_j}, \tag{4}$$

where the construction of each DP is formulated as:

$$\mathbf{x}_i^t \sim \boldsymbol{\theta}_{jk}^t, \boldsymbol{\theta}_{jk}^t \sim \mathbf{G}_j^t, \mathbf{G}_j^t \sim \mathrm{DP}_{[j]}(\frac{\alpha}{J^t}, \mathbf{H}). \tag{5}$$

In Sec. 4.2, we will explain the sequential inference conducted within each processor that turns the problem into an adaptive component assignment progress. New Gaussian components will be instantiated on-the-fly with knowledge learned from previous observations, guaranteeing an adaptive number of components locally under a globally consistent constraint.

## 4. Implementation

Our pipeline is illustrated in Fig. 3. The obtained data are first assigned to different processing units (Sec. 4.1). Afterwards, local DP is inferred in parallel constrained by hyper-priors. Learned parameters are then streamed to host, reweighted and refined as the map measurement $\boldsymbol{\Theta}^t$.

### 4.1. Initialization

We specify the processor indicator $\pi_i^t$ that distributes data mini-batches at each time to $J^t$ processors using spatial hashing algorithm [41]. Spatial hashing guarantees an $O(1)$ indexing from the coordinate $\mathbf{x}_i^t = (x, y, z)$ to the corresponding processor as:

$$H(x, y, z) = (x \cdot p_1 \oplus y \cdot p_2 \oplus z \cdot p_3) \ mod \ n. \tag{6}$$

We follow VoxelHashing [41] to subdivide the space into voxel blocks, where each block contains $8^3$ voxels. New blocks will be allocated once it falls into the footprint of a new observation. We adopt a lock-based block allocation [13] to avoid thread conflicts. 3D data that are associated with the same mixture component share the same processor indicator, where each processor maintains multiple components $\boldsymbol{\theta}_k^t$ that are corresponding to the same local DP.

### 4.2. Local Inference

The local inference is conducted in parallel between processors. We here resort to a Chinese restaurant process (CRP) implementation to incrementally update the local DP. By marginalizing over the infinite length partitions for $\boldsymbol{\theta}_k^t$, the parameter updating can be viewed as a procedure of adding and refining mixture components on-the-fly when needed, which resembles the fusion-based map updating in a globally consistent manner.

Inference with CRP is trivial by first calculating the component assignment $z_i^t \in \mathbf{Z}^t$ and then updating the parameters $\boldsymbol{\Theta}^t$. Component assignment is done parallelly within the associated processor by assigning the point to an existing component k with the probability of:

$$p(z_i = k | \mathbf{Z}_{-i}, \alpha) = \frac{n_{-i,k}}{n - 1 + \frac{\alpha}{J^t}}, \tag{7}$$

or instantiating a new component $K^t + 1$ with:

$$p(z_i = K^t + 1 | \mathbf{Z}_{-i}, \alpha) = \frac{\frac{\alpha}{J^t}}{n - 1 + \frac{\alpha}{J^t}}, \tag{8}$$

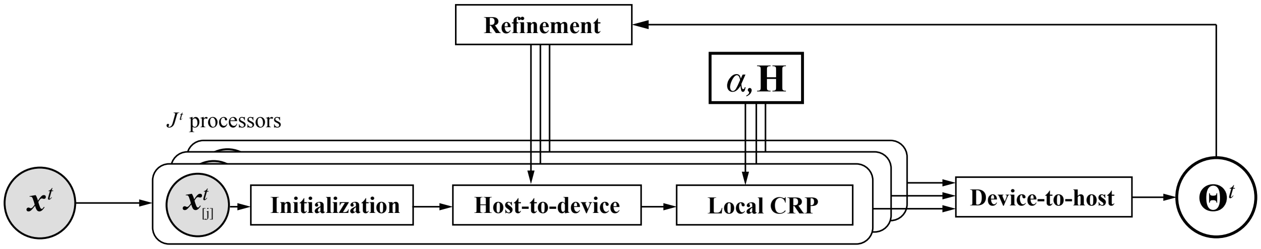


Figure 3: The proposed online Bayesian learning pipeline. The data are assigned to local processors for parallel updating, where local mixtures are constrained by the global hyper-parameters $\{\alpha, \mathbf{H}\}$. The local mixtures are then reweighted and refined as the updated global parameters.

where the subscript $-i$ denotes all indices before $i$th point arrives. $n_{-i,k}$ denotes the number of data that are marginalized out by all mixture components within the processor before $i$th point arrives.

After assigning the data to a specific component, the component parameter can be incrementally updated following [21] as:

$$\omega_{i,n_k+1} = \omega_{i,n_k} + 1, \tag{9}$$

$$\boldsymbol{\mu}_{i,n_k+1} = \frac{\omega_{i,n_k}\boldsymbol{\mu}_{i,n_k} + \mathbf{x}_i}{\omega_{i,n_k} + 1}, \tag{10}$$

$$\mathbf{\Sigma}_{i,n_k+1} = \mathbf{\Sigma}_{i,n_k} + \frac{\omega_{i,n_k}}{\omega_{i,n_k} + 1}(\mathbf{x}_i - \boldsymbol{\mu}_{i,n_k})^2. \tag{11}$$

### 4.3. Map Refinement

Though the hyper-priors for CRP enforce component instantiation when needed, the inference within a processor is conducted sequentially. Hence, a large amount of newly instantiated components may make the system computationally intractable even with a GPU acceleration. Practically, we perform truncation and pruning to maintain a clean and compact parameter space. Truncation is implemented by setting an upper bound of the mixture number for each processor as $T$. By enforcing $p(z_i = k) = 0$ for $k \geq T$, memory pre-allocation and fast indexing is guaranteed within each processor.

On the other hand, it is still possible that some of the components are redundant. We follow the Sequential Variational Approximation (SVA) [33] to explicitly maintain an accumulated weight for each component and adopt a thresholding pruning when necessary. The weight takes both point-Gaussian distance and data fidelity into consideration. An example of map updating from noisy RGB-D data is illustrated in Fig. 4. Our strict component instantiation strategy guarantees a clean and compact mixture of Gaussians.

To measure the data fidelity, we here view acquired 3D data as Gaussian-distributed noisy observations from samples of the global distribution as $\hat{\mathbf{x}}_i^t \sim \mathcal{N}(\mathbf{x}_i^t, \mathbf{\Sigma}_{\mathbf{x}_i^t})$, which can be optionally replaced by a specific model for a particular sensor input such as [39] and [25]. Following [4, 12], the covariance is represented as:

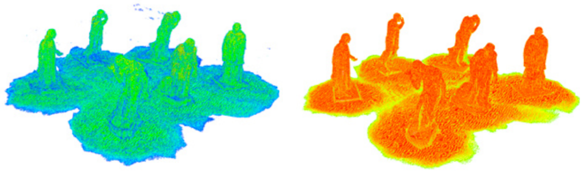
$$\mathbf{\Sigma}_{\mathbf{x}} = \mathbf{J}_{\mathbf{x}} \cdot \mathrm{diag}(\sigma_u^2, \sigma_v^2, \sigma_z^2) \cdot \mathbf{J}_{\mathbf{x}}^T, \tag{12}$$

where $\sigma_u^2$and $\sigma_v^2$ are pixel-positional variance set to be half-pixel size $0.5^2$. $\sigma_z^2$ is the depth variance obtained by [39]. $\mathbf{J}_{\mathbf{x}}$ is the Jacobian matrix as:

$$\mathbf{J}_{\mathbf{x}} = \begin{bmatrix} f_x^{-1} & 0 & (u - c_x)f_x^{-1} \\ 0 & f_y^{-1} & (v - c_y)f_y^{-1} \\ 0 & 0 & 1 \end{bmatrix}. \tag{13}$$

## 5. Experiments

In this section, we first present our experimental setup and evaluation protocols. Afterwards, we compare our representation against other related representations with state-of-the-art performances in terms of accuracy and efficiency. Qualitative results of the proposed representation on different datasets are also presented.



(a) Samples from original map (b) Samples from pruned map

Figure 4: Map refinement from noisy RGB-D data stream.

### 5.1. Experimental Setup

The experiments were conducted on a desktop PC with an Intel Core i7-6700 (8 cores @ 4 GHz), 16GB of RAM, and a single NVIDIA GeForce RTX 2080Ti with 11GB of memory.

**State-of-the-art methods.** Due to our spatial hashing implementation, we compare our method against state-of-the-art voxel-based probabilistic representations that can be updated in real-time. Default parameters are taken for each to perform confidence reasoning and outlier rejection.

- MRFMap[1] [57] maintains a forward ray sensor model via a Markov random field. The experiments are conducted on selected keyframes due to the increasing computational cost along with the graph size.
- KD-NDT[2] [55] maintains local Gaussian distribution within overlapped grid cell indexed by multiple kd-trees to mitigate the discretization error induced by voxelization. The CPU implementation without parallelization makes the method computationally intractable. The experiments are conducted on selected keyframes with downsampled depth data.
- PSDF[3] [14] maintains a joint distribution of SDF value and its inlier probability and outperforms traditional TSDF-fusion methods with noise and outlier handling.
- RoutedFusion[4] [63] trains a 2D depth routing network and a 3D depth fusion network to handle anisotropically distributed data fusion. We use the pre-trained model for evaluation.

**Metrics.** We evaluate our representation quality by computing the mean and the standard deviation (std.) of the cloud-mesh distance using *CloudCompare*[5] software. Since output formats vary between different representations, we randomly sample 150,000 points from each representation for quantitative evaluation. As KD-NDT only outputs means of Gaussians, we directly take downsampled mean values as sampled points. For our representation, samples are generated using importance sampling [16], which approximates the global distribution and is noisier compared to the mean value.

**Datasets.** We mainly evaluate quantitatively on the synthetic ICL-NUIM livingroom dataset [22]. Additional evaluations are performed on TUM RGB-D Dataset (TUM) [60] and 3D Scene Data (Zhou) [70] with real scans.

### 5.2. Representation Quality

We conduct qualitative and quantitative evaluation to measure how well the proposed representation can describe the generative process of a scene. The visualization of the proposed representation is illustrated in Fig. 5. We compare against other representations on both clean and noisy sequences of the ICL-NUIM dataset to further demonstrate the noise-handling ability of each representation. As shown in Tab. 1 and Fig. 6, our representation achieves a much lower error compared to other baselines. We provide various voxel resolution configurations and obtain consistent findings:

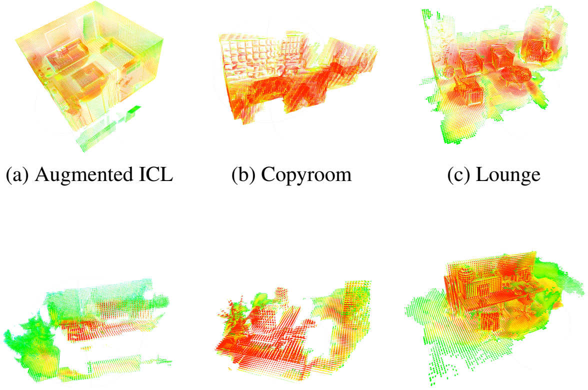

(a) Augmented ICL (b) Copyroom (c) Lounge
(d) fr1_xyz (e) fr2_xyz (f) fr3_long_office

Figure 5: Visualization of the proposed scene representation on various datasets. Color denotes the local accumulated confidence (red→blue: high→low confidence).

1) Descriptiveness: It is noteworthy that at a low voxel resolution, Gaussian-based representations such as KD-NDT and ours are more capable of modeling clean and thin surfaces. As we maintain an adaptive number of Gaussians within each voxel grid, the representation is more descriptive compared to the NDT-based representations. Even though our parameter space is sparse and discrete, the representation itself is a continuous probability field. Hence, the map serves as a generative model where we can sample arbitrary number of points. Besides, the sampled density reflects the local geometric confidence (Fig. 6e).

2) Noise-handling ability: It can also be noted that our representation achieves the lowest error and deviation on noisy sequences compared to other competitive representations. The systematically established probabilistic formulation along with the truncation and pruning strategies guarantee a promising accuracy. We clarify that we do not train networks provided by RoutedFusion as we target an online learning fashion. Quantitative evaluation with re-trained networks for RoutedFusion is demonstrated in the supplementary materials.

### 5.3. Representation Efficiency

We measure the representation efficiency in terms of accuracy *vs*. runtime/parameter number at different voxel

[1] https://github.com/mrfmap/mrfmap
[2] https://github.com/cogsys-tuebingen/cslibs_ndt
[3] https://github.com/theNded/MeshHashing
[4] https://github.com/weders/RoutedFusion
[5] http://www.danielgm.net/cc/

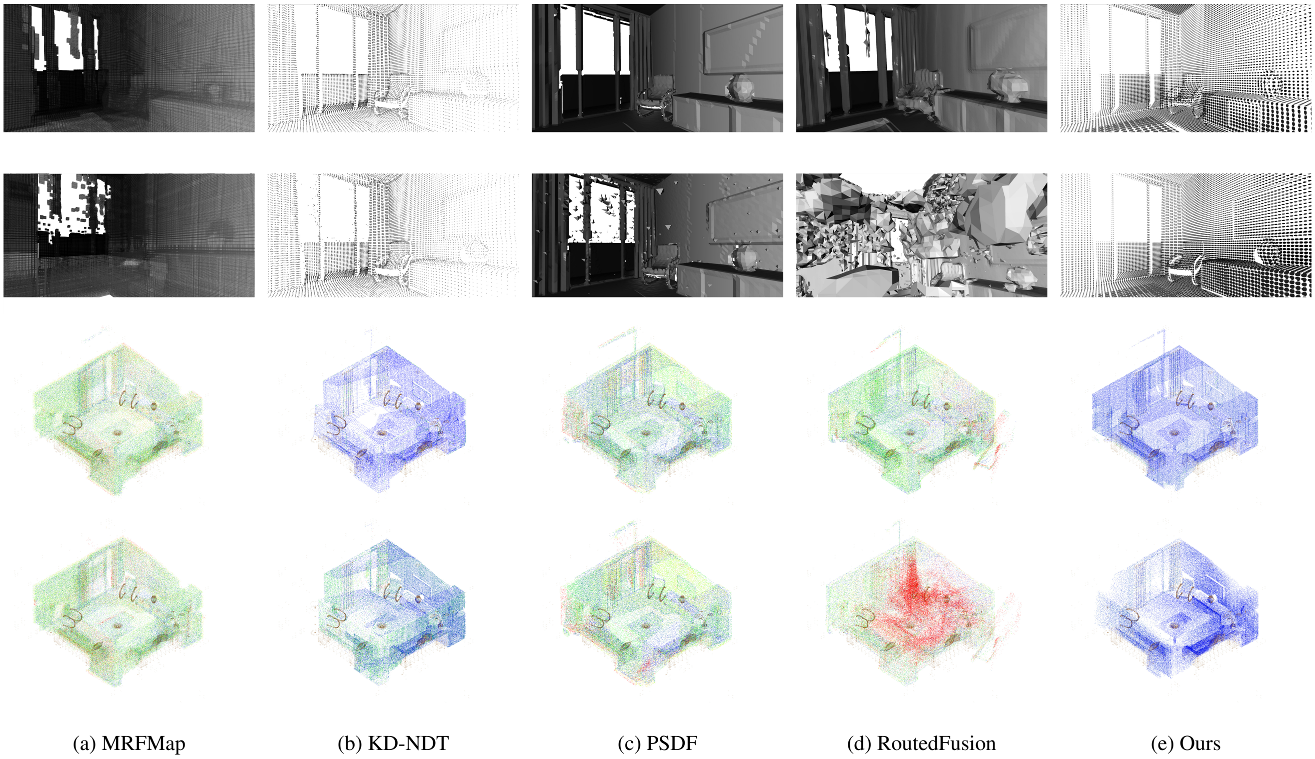


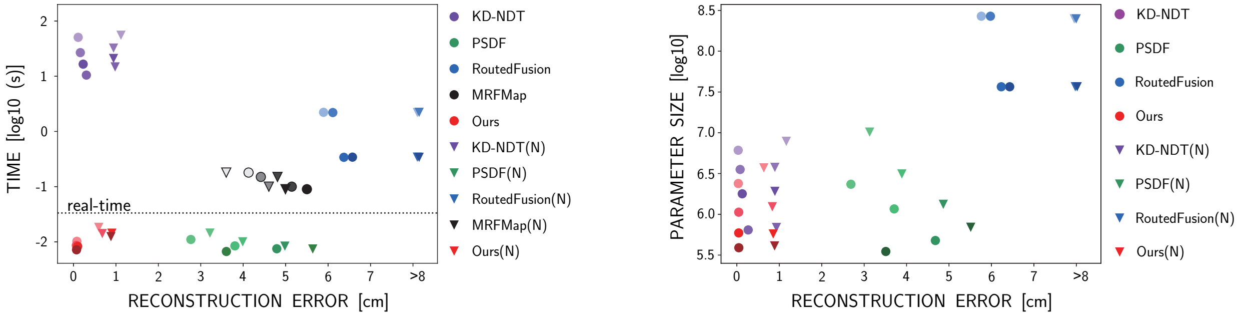
Figure 6: Qualitative comparisons of the representation quality. First two rows: visualization of each scene representation on kt1 clean/noisy sequences. The voxel resolution is set as 4 cm. Last two rows: Error heatmap on kt2 clean/noisy sequences. Color denotes the point-to-mesh distance (blue-red: 0-10cm). The voxel resolution is set as 2cm.

Table 1: Cloud-to-mesh statistics (cm) on the ICL-NUIM dataset. The voxel resolution is set as 5cm.

| Method | kt1 (*clean*) | | kt1 (*noisy*) | | kt2 (*clean*) | | kt2 (*noisy*) | |
|---|---|---|---|---|---|---|---|---|
| | mean | std. | mean | std. | mean | std. | mean | std. |
| MRFMap [24] | 5.3759 | 3.3863 | 5.3028 | 4.9927 | 5.3979 | 3.3210 | 5.2735 | 4.4339 |
| KD-NDT [55] | 0.2675 | 0.4948 | 1.0688 | 1.2475 | 0.2492 | 0.4603 | 1.4877 | 1.8574 |
| PSDF [14] | 3.5104 | 3.4741 | 5.5026 | 9.9667 | 4.2282 | 3.6356 | 5.2325 | 6.1295 |
| RoutedFusion [63] | 6.5169 | 3.3753 | 20.3565 | 19.1897 | 5.0746 | 3.1432 | 23.7922 | 27.0850 |
| Ours | **0.0752** | **0.1321** | **0.8709** | **1.0549** | **0.0659** | **0.1195** | **1.0078** | **0.8658** |

resolution (2cm-5cm) configurations. All baselines except KD-NDT are implemented in parallel on a single GPU. As illustrated in Fig. 7a, the proposed method yields a good trade-off between accuracy and computational efficiency. Our spatial hashing scheme is similar to PSDF [14]. Though our local sequential inference leads to additional computational cost compared to SDF updating, we achieve efficient inference at a high voxel resolution. It can be explained twofold: Firstly, new components can hardly be instantiated as high voxel resolution leads to a large $J^t$ and in turn a small $\frac{\alpha}{J^t}$. Secondly, the size of minibatch data within each processor is small, thus leading to lower complexity of local sequential inference. It should also be noted that the computational and memory efficiency of RoutedFusion is up to the defined volume size. For livingroom dataset at the size of about 6m*3m*9m, the voxel grid is allocated to be $256^3$ for 4cm and 5cm resolution and $512^3$ for 2cm and 3cm.

Furthermore, Fig. 7b depicts the trade-off between memory consumption and accuracy. It can be noticed that we achieve high accuracy at a relatively low memory consumption. The parameter size is calculated by multiplying the allocated voxel number by the parameter number within each voxel. It should be noted that we do not provide the param-



(a) Accuracy vs. runtime

(b) Accuracy vs. memory

Figure 7: Representation efficiency on ICL-NUIM kt1 sequence. For each method, we evaluate the trade-offs between accuracy-time and accuracy-parameter number on clean and noisy [N] datasets. The voxel resolution is set to be 2cm-5cm, where the lighter color denotes a higher resolution.

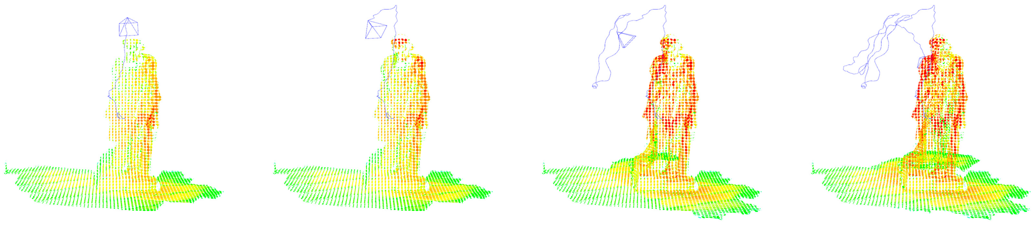


Figure 8: Online learning of the Bayesian nonparametric model. The confidence and the completeness are gradually increasing by gaining knowledge from streaming data.

eter size of MRFMap. Keyframe sensor data are required to be stored to construct the MRF graph. Hence, the memory consumption will increase monotonically when adding more keyframes. Meanwhile, the implemented overlapping grids for KD-NDT lead to an effective resolution of half the resolution of the voxel grid, thus bringing more Gaussian components compared against ours.

## 6. Conclusion and Future Work

In this paper, we introduce a Bayesian nonparametric mixture model as the scene representation, depicting a continuous probability density function. Map updating given streaming data is cast as an online Bayesian learning problem, as illustrated in Fig. 8. A gradual transition from geometry prior to posterior is conducted through parallel and incremental inference in real-time. Experimental results demonstrate that the proposed method achieves state-of-the-art accuracy and efficiency.

We believe that the proposed approach establishes a systematical framework based on probabilistic formulation, revealing potentials for multiple extensions. One interesting direction lies in online learning of scene geometry with neural networks. The proposed approach models the transition from geometry prior to posterior and opens the gate to enforce knowledge transfer [32] from pre-trained features. Recent advances in learning local geometry primitives may obtain a more expressive prior distribution compared to the assumed Gaussian or other distributions. Another direction lies in the graphical applications derived from the proposed representation. As stated in SurfelMeshing [54], online meshing directly from point-wise data is susceptible to noise. The parameter space of the proposed representation can be directly utilized as a probabilistic surface element that is robust to different sensor noise. The generative property also allows the generation of different scene representations from the probability field. We believe that the proposed representation will store and provide more informative cues for diverse kinds of applications.

## 7. Acknowledgement

We would like to thank Wei Dong for feedback on the manuscript, Kangjie Zhou and Jiarui Liang for their help with the cloud-mesh distance calculation, and anonymous reviewers for their helpful feedback. This work is supported by the National Key Research and Development Program of China (2017YFB1002601) and National Natural Science Foundation of China (61632003, 61771026).

## References

[1] Amr Ahmed and Eric Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *SIAM Intl. Conf. on Data Mining (SDM)*, pages 219–230, 2008. 4

[2] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, volume 3, pages 2743–2748, 2003. 2, 3

[3] Trevor Campbell, Julian Straub, John W Fisher III, and Jonathan P How. Streaming, distributed variational inference for bayesian nonparametrics. In *Advances in Neural Information Processing Systems (NIPS)*, pages 280–288, 2015. 4

[4] Yan-Pei Cao, Leif Kobbelt, and Shi-Min Hu. Real-time high-accuracy three-dimensional reconstruction with consumer rgb-d cameras. *ACM Trans. Graphics*, 37(5):1–16, 2018. 5

[5] Rohan Chabra, Jan E. Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conf. on Computer Vision (ECCV)*, pages 608–625, 2020. 3

[6] Jiawen Chen, Dennis Bautembach, and Shahram Izadi. Scalable real-time volumetric surface reconstruction. *ACM Trans. Graphics*, 32(4):1–16, 2013. 2

[7] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312, 1996. 2, 3

[8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graphics*, 36(3):24, 2017. 2

[9] Andrew J Davison. Futuremapping: The computational structure of spatial ai systems. *arXiv preprint arXiv:1803.11288*, 2018. 1

[10] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1486–1493, 2014. 2

[11] Amaël Delaunoy and Emmanuel Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *Intl. J. of Computer Vision*, 95(2):100–123, 2011. 2

[12] Aditya Dhawale and Nathan Michael. Efficient parametric multi-fidelity surface mapping. In *Robotics: Science and Systems (RSS)*, 2020. 2, 3, 5

[13] Wei Dong, Jieqi Shi, Weijie Tang, Xin Wang, and Hongbin Zha. An efficient volumetric mesh representation for real-time scene reconstruction using spatial hashing. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6323–6330, 2018. 2, 4

[14] Wei Dong, Qiuyuan Wang, Xin Wang, and Hongbin Zha. Psdf fusion: Probabilistic signed distance function for on-the-fly 3d data fusion and scene reconstruction. In *European Conf. on Computer Vision (ECCV)*, pages 701–717, 2018. 3, 6, 7

[15] Benjamin Eckart, Kihwan Kim, and Jan Kautz. Hgmr: Hierarchical gaussian mixtures for adaptive 3d registration. In *European Conf. on Computer Vision (ECCV)*, pages 705–721, 2018. 2

[16] Benjamin Eckart, Kihwan Kim, Alejandro Troccoli, Alonzo Kelly, and Jan Kautz. Accelerated generative models for 3d point cloud data. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5497–5505, 2016. 2, 6

[17] Georgios Dimitrios Evangelidis and Radu Horaud. Joint alignment of multiple point sets with batch and incremental expectation-maximization. *IEEE Trans. Pattern Anal. Machine Intell.*, 40(6):1397–1410, 2017. 2

[18] Wei Gao and Russ Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11095–11104, 2019. 2

[19] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4857–4866, 2020. 3

[20] Vitor Guizilini and Fabio Ramos. Towards real-time 3d continuous occupancy mapping using hilbert maps. *Intl. J. of Robotics Research*, 37(6):566–584, 2018. 2

[21] Tom SF Haines and Tao Xiang. Background subtraction with dirichletprocess mixture models. *IEEE Trans. Pattern Anal. Machine Intell.*, 36(4):670–683, 2013. 5

[22] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1524–1531, 2014. 6

[23] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. Pointgmm: a neural gmm network for point clouds. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12054–12063, 2020. 3

[24] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013. 2, 3, 7

[25] Marius Huber, Timo Hinzmann, Roland Siegwart, and Larry H Matthies. Cubic range error model for stereo vision with illuminators. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 842–848, 2018. 3, 5

[26] Maani Ghaffari Jadidi, Jaime Valls Miro, and Gamini Dissanayake. Warped gaussian processes occupancy mapping with uncertain inputs. *IEEE Robotics and Automation Letters*, 2(2):680–687, 2017. 2

[27] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6001–6010, 2020. 3

[28] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *Intl. Conf. on 3D Vision (3DV)*, pages 1–8, 2013. 2

[29] Leonid Keselman and Martial Hebert. Direct fitting of gaussian mixture models. In *Conf. on Computer and Robot Vision (CRV)*, pages 25–32, 2019. 2

[30] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012. 3

[31] Bhoram Lee, Clark Zhang, Zonghao Huang, and Daniel D Lee. Online continuous mapping using gaussian process implicit surfaces. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6884–6890, 2019. 3

[32] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *International Conference on Learning Representations*, 2019. 8

[33] Dahua Lin. Online learning of nonparametric mixture models via sequential variational approximation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 395–403, 2013. 4, 5

[34] Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar. Characterizations of noise in kinect depth images: A review. *IEEE Sensors Journal*, 14(6):1731–1740, 2014. 3

[35] Wolfram Martens, Yannick Poffet, Pablo Ramón Soria, Robert Fitch, and Salah Sukkarieh. Geometric priors for gaussian process implicit surfaces. *IEEE Robotics and Automation Letters*, 2(2):373–380, 2016. 2

[36] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019. 3

[37] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conf. on Computer Vision (ECCV)*, 2020. 2

[38] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011. 2

[39] Chuong V Nguyen, Shahram Izadi, and David Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *Intl. Conf. on 3D imaging, modeling, processing, visualization & transmission*, pages 524–530, 2012. 3, 5

[40] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3504–3515, 2020. 2

[41] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graphics*, 32(6):1–11, 2013. 2, 4

[42] Simon T O'Callaghan, Fabio T Ramos, and Hugh Durrant-Whyte. Contextual occupancy maps incorporating sensor and location uncertainty. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3478–3485, 2010. 2

[43] Simon T O'Callaghan and Fabio T Ramos. Gaussian process occupancy maps. *Intl. J. of Robotics Research*, 31(1):42–62, 2012. 2

[44] Cormac O'Meadhra, Wennie Tabib, and Nathan Michael. Variable resolution occupancy mapping using gaussian mixture models. *IEEE Robotics and Automation Letters*, 4(2):2015–2022, 2018. 2

[45] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 3

[46] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conf. on Computer Vision (ECCV)*, 2020. 3

[47] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *SIGGRAPH*, pages 335–342, 2000. 2

[48] Jim Pitman et al. Combinatorial stochastic processes. Technical report, Technical report, U.C. Berkeley Dept. Statistics, 2002. 4

[49] Fabio Ramos and Lionel Ott. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *Intl. J. of Robotics Research*, 35(14):1717–1730, 2016. 2

[50] Carl Edward Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 554–560, 2000. 4

[51] Jari Saarinen, Henrik Andreasson, Todor Stoyanov, Juha Ala-Luhtala, and Achim J Lilienthal. Normal distributions transform occupancy maps: Application to large-scale online 3d mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2233–2238, 2013. 2

[52] Jari P Saarinen, Henrik Andreasson, Todor Stoyanov, and Achim J Lilienthal. 3d normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments. *Intl. J. of Robotics Research*, 32(14):1627–1644, 2013. 2

[53] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 134–144, 2019. 2

[54] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Surfelmeshing: Online surfel-based mesh reconstruction. *IEEE Trans. Pattern Anal. Machine Intell.*, 2019. 2, 8

[55] Cornelia Schulz, Richard Hanten, and Andreas Zell. Efficient map representations for multi-dimensional normal distributions transforms. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2679–2686, 2018. 2, 6, 7

[56] Ransalu Senanayake and Fabio Ramos. Bayesian hilbert maps for dynamic continuous occupancy mapping. In *Conf. on Robot Learning (CoRL)*, pages 458–471, 2017. 2

[57] Kumar Shaurya Shankar and Nathan Michael. Mrfmap: Online probabilistic 3d mapping using forward ray sensor models. In *Robotics: Science and Systems (RSS)*, 2020. 6

[58] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1121–1132, 2019. 2

[59] Shobhit Srivastava and Nathan Michael. Efficient, multi-fidelity perceptual representations via hierarchical gaussian mixture models. *IEEE Trans. Robotics*, 35(1):248–260, 2018. 2

[60] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012. 6

[61] Wennie Tabib, Kshitij Goel, John Yao, Mosam Dabhi, Curtis Boirum, and Nathan Michael. Real-time information-theoretic exploration with gaussian mixture model maps. In *Robotics: Science and Systems (RSS)*, 2019. 2

[62] Jinkun Wang and Brendan Englot. Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1003–1010, 2016. 2

[63] Silvan Weder, Johannes Schonberger, Marc Pollefeys, and Martin R Oswald. Routedfusion: Learning real-time depth map fusion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4887–4897, 2020. 3, 6, 7

[64] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *Intl. J. of Robotics Research*, 34(4-5):598–626, 2015. 2

[65] Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and Systems (RSS)*, 2015. 2, 3

[66] Sinead Williamson, Avinava Dubey, and Eric Xing. Parallel markov chain monte carlo for nonparametric mixture models. In *Intl. Conf. on Machine Learning (ICML)*, pages 98–106, 2013. 4

[67] Oliver J Woodford and George Vogiatzis. A generative model for online depth fusion. In *European Conf. on Computer Vision (ECCV)*, pages 144–157, 2012. 3

[68] Junyu Xuan, Jie Lu, and Guangquan Zhang. A survey on bayesian nonparametric learning. *ACM Computing Surveys (CSUR)*, 52(1):1–36, 2019. 1, 3

[69] Zhixin Yan, Mao Ye, and Liu Ren. Dense visual slam with probabilistic surfel map. *IEEE Trans. Visualization and Comp. Graphics*, 23(11):2389–2398, 2017. 3

[70] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Trans. Graphics*, 32(4):1–8, 2013. 6