

Variable Resolution Occupancy Mapping using Gaussian Mixture Models

Cormac O'Meadhra, Wennie Tabib, and Nathan Michael

Abstract—Occupancy mapping is fundamental for active perception systems to enable reasoning about known and unknown regions of the environment. The majority of occupancy mapping approaches enforce an *a priori* discretization on the environment, resulting in a fixed resolution map that limits the expressiveness of the representation. The proposed approach removes this *a priori* discretization, learns continuous representations for the evidence of occupied and free space to derive the probability of occupancy, and enables occupancy grid maps to be generated at arbitrary resolution. Efficient methods are also presented that accurately evaluate the probability of occupancy in individual cells and enable multi-resolution mapping and local occupancy evaluation. The efficacy of the approach is demonstrated by comparison to state-of-the-art discrete and continuous mapping techniques in both 2D and 3D. The core contribution of this work is a memory-efficient method for deriving occupancy that is amenable to small or large corrections in pose without the need to regenerate the entire map. The applications under consideration are low-bandwidth scenarios (e.g. multi-robot exploration) and operations in expansive environments where storing an occupancy grid map of the entire environment would be prohibitive.

Index Terms—Mapping, RGB-D Perception.

I. INTRODUCTION

OCCUPANCY mapping is often applied as a framework for core capabilities of autonomous systems such as localization, collision avoidance, exploration, environment reasoning, etc. Representing occupancy in a probabilistic manner enables reasoning about the uncertainty in the occupancy model that arises due to imperfect state estimation and noise in sensor measurements. Appropriate accounting for this uncertainty increases robustness and is critical where error in the environmental model can result in catastrophic failure.

Although the world we seek to model is continuous, reasoning over continuous models is difficult due to the requisite integration over non-trivial spaces, which often necessitates numerical evaluation. A commonly employed mitigation strategy is the pre-emptive discretization of the environment, typically through the imposition of a grid structure [1–6], which renders evaluation of the model significantly less complex. However, *a priori* discretization results in loss of information due to the

Manuscript received: August, 3, 2018; Revised October, 27, 2018; Accepted December, 4, 2018.

This paper was recommended for publication by Editor Cyrill Stachniss upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the NASA Early Stage Innovations grant NNX16AD98G, NASA Space Technology Research Fellowship NNX14AL66H and the NASA Space Technology Research Program, NASA STTR grant NNX16CK16C, and DTRA grant HDTRA1-13-0026.)

The authors are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, USA {comeadhra, wtabib, nmichael}@cmu.edu

Digital Object Identifier (DOI): see top of this page.

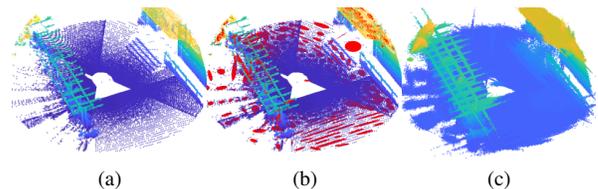


Fig. 1: Gaussian mixture model occupancy mapping pipeline. (a) input pointcloud. (b) 850 component GMM with laser pointcloud overlaid. (c) The proposed approach performs Monte-Carlo sampling of the GMM and raytraces through an occupancy grid map to the sensor origin while updating the probability of occupancy of cells along the beam.

non-invertible mapping from measurements to discrete cells, i.e. it is known that a cell contains a measurement, but it is not known where in the cell it lies. Approaches exist that mitigate against this ambiguity through increased resolution made feasible by multiple levels of discretization that can be matched to the observed data [2], but still requires imposition of a minimum discretization level.

There has been a growing interest in direct modeling of the continuous occupancy distribution, from which standard occupancy grid maps can be sampled, or in some cases, the models can be leveraged for direct evaluation [7, 8]. However, direct modeling of occupancy scales with the volume of the region being modeled, requiring models of increasing size or capacity in order to maintain high fidelity. The scaling issues of direct modeling approaches can be overcome through the use of surface modeling which implicitly captures occupancy. In this work, we propose to follow recent applications of Gaussian Mixture Models (GMMs) as a compact representation of sensor observations [9–12]. Employing GMMs in this manner implicitly captures the occupied surface and free space information required to reconstruct a probabilistic representation of occupancy, while avoiding the cost of direct volumetric modeling.

While the continuous nature and shape of the Gaussian distribution may appear to be ill-suited to modeling discontinuous surfaces of uniform occupancy, we note that in practice, this does not prove to be a substantial issue. Firstly, the support of the Gaussian distribution is effectively compact, with $< 0.3\%$ of the probability mass lying outside the three standard deviation ellipsoid, which can be leveraged to approximate discontinuous boundaries to a high degree of accuracy. Secondly, although the sensor observations are generated by sampling from uniformly dense surfaces according to the sensor sampling distribution, empirical results show a GMM with a sufficient number of components can well model the surface distribution.

By leveraging GMMs (Sections III-A and III-B) as a model of observed surfaces, we build what we term as a mixture

of Gaussian cones (Section III-C), which fully describes the continuous distribution of free space evidence contained within the measurement. In this manner, we determine the evidence of free space approximately for arbitrary shapes in 2D (Section III-C1) and polyhedra in 3D (Section III-C2). We utilize a Bernoulli Distribution to model the opposing evidence for occupied space encoded by the GMM and the evidence for free space derived from the corresponding cones (Section III-C3). We demonstrate the accuracy and efficiency of this approach compared to state-of-the-art discrete and continuous methods (Section IV) and present a study to determine the optimal number of components in practice (Section IV-C).

II. RELATED WORKS

Occupancy grid mapping is most commonly used for modeling occupancy, but suffers from fixed grid resolution, which leads to high memory complexity - quadratic in 2D and cubic in 3D, and an independence assumption between cells in the grid. Variants have been proposed that seek to overcome these limitations. OctoMap [2] reduces the impact of fixed resolution through a memory efficient hierarchical structure, enabling the use of significantly higher resolution. However, this technique incurs discretization errors due to fixed resolution at the lowest level. Recent work by Sun *et al.* proposes to address the independence assumption in grid maps through a Gaussian Markov Random Field by decomposing a grid model into sub-grids with known inter- and intra sub-grid dependencies [13]. Similarly, Osman *et al.* propose to use ray potentials and surface priors to update conditionally dependent occupancy cells in a consistent manner [14].

Continuous and semi-continuous approaches have been presented to mitigate against discretization errors. The Normal Distribution Transform Occupancy Map (NDT-OM) [4] builds a 3D occupancy grid map where each occupied cell stores a uniformly weighed Gaussian density. However, NDT mapping suffers from large memory requirements due to the need to retain the voxelized representation of the environment and Srivastava [15] demonstrates that voxelization introduces discretization at the edges of the cell. Gaussian Process (GP) occupancy mapping techniques first introduced by O’Callaghan and Ramos [1] learn a continuous representation of the environment using GP regression. GP occupancy maps (GPOMs), as first introduced, suffer from poor training and test complexity, $O(N^3)$ and $O(N^2M)$, respectively where N and M are the number of training points and testing points for model learning and evaluation, respectively. Kim and Kim [16] develop a more computationally tractable method by dividing the training data into small subsets and applying a mixture of GPs and subsequently further improve the efficiency through the application of sparse Gaussian processes [17]; however, the training time remains too large for real-time applications. Jadidi *et al.* [18] develop incremental GPOM methods that efficiently learn separate models for free and occupied space, which are combined to generate a probability of occupancy. An efficient variant of the GPOMs was introduced by Doherty *et al.* [8], in which Bayesian kernel inference and sparse kernels in an octree data structure are

employed to reduce training complexity via recursive updates and testing complexity through test data partitioning. The proposed approach compares favorably to BGK-OM both in terms of memory and accuracy as show in Section IV-B. Hilbert Maps [7] have also been proposed as an efficient method of learning continuous occupancy models by training a classifier in a reproducing kernel Hilbert space. Evaluation in 2D environments demonstrate the increased representational capacity of the proposed approach when compared to Hilbert maps (see Section IV-A).

Gaussian Mixture Models have been shown to provide a high fidelity generative of model of point cloud data [10]. Furthermore, while GMMs have historically been slow to train, Eckart *et al.* [10] demonstrated an efficient coarse-to-fine expectation-maximization implementation that enables conversion from dense depth data to a GMM at sensor rate on a mobile platform. Recently, Marriott *et al.* [12] presented an application of GMMs to surface modeling via a Gaussian mixture regression formulation for plane extraction, which can be considered as a constrained variant of the fitting procedure employed in this work. Furthermore, GMMs have been shown to be useful in other stages of the robotics pipeline. For example, in prior work, the authors demonstrated the application of GMMs for robust registration of depth observations [11].

III. METHODOLOGY

A. Gaussian Mixture Model as a Generative Observation Model

We elect to model an observed point cloud, \mathcal{Z} , as a sample drawn from an underlying probabilistic mixture model, $p(x)$. We assume $p(x)$ to be well-modeled by a J -component Gaussian Mixture Model, \mathcal{G} , which is defined as

$$\mathcal{G} = \sum_{j=1}^J \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \quad (1)$$

where the hyper-parameters $\theta_j = \{\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$ are the component-wise prior, mean, and covariance, respectively. The point cloud \mathcal{Z} is comprised of N points $z_i \in \mathcal{Z}$, where $z_i \in \mathbb{R}^n$, for $n \in \{2, 3\}$ in this work. Each z_i is assumed to be *i.i.d* according to \mathcal{G} . The hyper-parameters are then learned such that the GMM is the maximum-likelihood estimator of \mathcal{Z} via the Expectation-Maximization (EM) algorithm. EM is an iterative procedure that optimizes a lower bound on the MLE. For more information on the EM algorithm applied to GMMs see [19, Chapter 9] and [10].

The GMM \mathcal{G} encodes the spatial density of observed points. Given the support size N and the GMM describing a series of measurements, $\mathcal{G} = p(\mathbf{x}) = \sum_{j=1}^J \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, we can compute the number of points observed in any volume V , through integrating $p(\mathbf{x})$ over this volume. We leverage this fact to interpret \mathcal{G} as model of the evidence for occupancy in any volume in space. The GMM encodes sensor noise implicitly within the surface model. As depth sensors observe surfaces, the resulting GMM will typically have near degenerate dimension directed along the normal vector to the surface. The variance of the data along this dimension, which is captured in the component covariance, describes the sensor noise. As each

component corresponds to an individual measurement and each measurement is invariant to pose uncertainty, corrections can be applied on a component-wise basis via the following equations that transforms the density through a rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ and translation vector $\mathbf{t} \in \mathbb{R}^3$.

$$\boldsymbol{\mu}' = \mathbf{R}\boldsymbol{\mu} + \mathbf{t} \quad \boldsymbol{\Sigma}' = \mathbf{R}\boldsymbol{\Sigma}\mathbf{R}^T \quad (2)$$

B. Free Space Modeling

While a GMM \mathcal{G} models the observed space (both occupied and max-range), it ignores the information contained in the beams between the sensor and the end points. Free space can be modeled explicitly by additionally learning a free space GMM \mathcal{F} , whose components lie along the beams associated with a component of \mathcal{G} . However, this approach has two principle limitations: Firstly, each occupied point corresponds to an infinite number of free space points lying on the beam from the occupied point to the sensor location; Secondly, the free space evidence resulting from an individual point is constant along the entire beam between the sensor location and the occupied point, making Gaussian decay an inappropriate modeling choice. We can overcome both of these limitations by acknowledging the implicit modeling of free space embedded in the GMM \mathcal{G} .

C. Mixtures of Gaussian Cones

Making use of the fact that the free space evidence is entirely defined by the beam connecting the sensor location and the observed points, we propose to derive the free space model from the tuple $\{\mathcal{G}_t, \mathbf{s}_t\}_{t=1, \dots, T}$, where \mathbf{s}_t is the sensor pose corresponding to the observation \mathcal{Z}_t on which the mixture model \mathcal{G}_t was trained. The free-space beams that connect the sensor location to a given component, when clustered, form an oblique cone, as shown in Fig. 2. The evidence of free space at a given point \mathbf{x} is the evidence of an occupied or max range point lying beyond \mathbf{x} along the ray originating at the sensor location and passing through \mathbf{x} . The evidence for free-space in a region V , is given by the evidence of occupancy contained within the truncated polygonal pyramid, with apex and shape defined by the spherical polygon derived from the projection of the volume V onto a sphere centered at the sensor location. We utilize this fact to derive an analytic expression for the free-space evidence in 2D. However, due to a non-trivial region of integration, to the best of the authors' knowledge, direct extension of this approach to 3D is not possible (see Section VI for more information). Instead, we utilize a fast Monte Carlo ray tracing technique.

The free space evidence integral can be expressed as

$$Evidence(\mathbf{x}) = \int_V p(\mathbf{x}) d\mathbf{x} \quad (3)$$

The nature of the volume V makes parameterization in spherical coordinates a natural choice. Before performing the coordinate transformation, it is convenient to remove the effect of the anisotropic data covariance, which is achieved by applying the following isotropic mapping: $\mathbf{x}_I = \mathbf{S}_k^{-1/2} \mathbf{V}_k^T (\mathbf{x} - \boldsymbol{\mu}_k)$, where $\mathbf{V}_k \mathbf{S}_k \mathbf{V}_k^T = \boldsymbol{\Sigma}_k$ is the singular-value decomposition of the anisotropic covariance matrix.

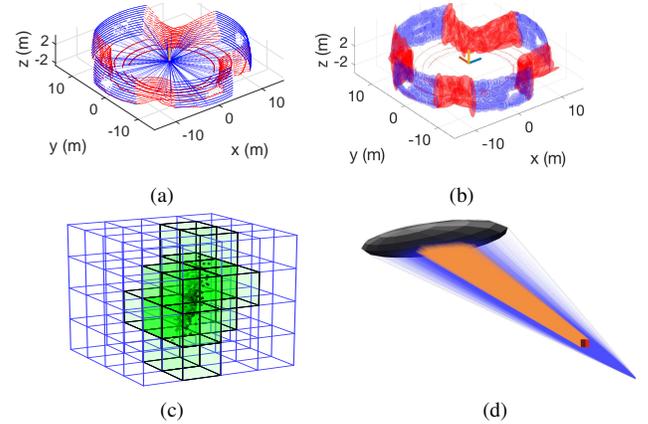
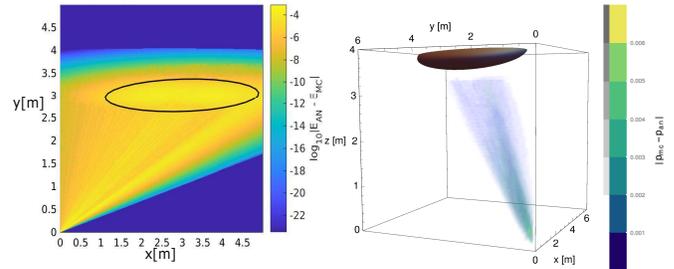


Fig. 2: Occupancy reconstruction pipeline. (a) The system takes as input a sensor observation with occupied points (red), max-range points (blue) and a sensor location, from which the free space beams (blue lines) originate. (b) A GMM is fitted to the occupied points (red) and maximum range points (blue). (c) The occupied space evidence (N_V^H) is then computed using Monte Carlo integration over the occupied space components (red components) only, which amounts to resampling and binning in a voxel grid. (d) The free space evidence (N_V^M) is computed by evaluating the subset (orange density) of Gaussian cone (blue density) passing through a given voxel V . This evaluation is achieved using Monte Carlo raytracing or the presented analytic approximations. The probability of occupancy is the reconstructed for a voxel V using Eq. (11).



(a) Error in Eq. (5) for 1cm cell. (b) Error in Eq. (9) for 10cm voxel.

Fig. 3: Absolute error in the free space approximation for 2D (a) and 3D (b). The error was calculated as the difference between the Monte Carlo estimate and the analytic value for the probability of occupancy due to a single Gaussian component observed from the origin. In both cases, the peak absolute error is empirically found to be low.

A single Gaussian component in this new space is given by $p(\mathbf{x}_I) = \frac{1}{(2\pi)^{3/2}} \exp(-\frac{1}{2}(\mathbf{x}_I - \mathbf{p}_k)^T (\mathbf{x}_I - \mathbf{p}_k))$, where \mathbf{p}_k is the sensor location corresponding to component k . Finally, we apply a rotation \mathbf{R}_k , such that the component mean lies along a co-ordinate axis, i.e. in 2D, we rotate the mean onto the x-axis, such that $\mathbf{R}_k \mathbf{p}_k = [p_{k,x} \ 0]^T$.

1) *2D Occupancy Grids*: In the 2D case, we reparameterize Eq. (3) to polar coordinates and rotate the mean onto the x-axis

$$\begin{aligned} E(x) &= \frac{1}{2\pi} \int_V r e^{-\frac{1}{2} \left\| \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} - \begin{bmatrix} p_x \\ 0 \end{bmatrix} \right\|_2^2} dr d\theta \\ &= \frac{1}{2\pi} \int_V e^{-\frac{1}{2} p_x^2 \sin^2 \theta} r e^{-\frac{1}{2} (r - p_x \cos \theta)^2} dr d\theta \\ &= \frac{1}{2\pi} \int_{\theta} e^{-\frac{1}{2} p_x^2 \sin^2 \theta} \int_{r'}^{\infty} r e^{-\frac{1}{2} (r - p_x \cos \theta)^2} dr d\theta \quad (4) \\ &= \frac{1}{2\pi} \int_{\theta} e^{-\frac{1}{2} p_x^2 \sin^2 \theta} \left[e^{-\frac{1}{2} (r' - p_x \cos \theta)^2} + \right. \\ &\quad \left. \left(1 - \operatorname{erf} \left(\frac{r' - p_x \cos \theta}{\sqrt{2}} \right) \right) \sqrt{\frac{\pi}{2}} p_x \cos \theta \right] d\theta \end{aligned}$$

The final integral has no closed form solution in general. However, we note that due to the phase offset between sine and

cosine, the first term has negligible contribution except around $\theta \approx 0$. In this case the term reduces to $\exp(-(1/2)(r' - p_x)^2)$, which has value of 1 for $r' \approx p_x$. However, since the mean p_x is typically very large (due to the isotropic transform significantly stretching the along beam direction, see Fig. 2), the contribution of this term to the integral is negligible. We can apply the same logic to approximate the term inside the error function as constant evaluated at $\theta = 0$.

$$\begin{aligned} E(x) &\approx \frac{1}{\sqrt{8\pi}} \int_{\theta} e^{-\frac{1}{2}p_x^2 \sin^2 \theta} \left(1 - \operatorname{erf}\left(\frac{r' - p_x \cos \theta}{\sqrt{2}}\right)\right) p_x \cos \theta d\theta \\ &\approx \frac{1}{\sqrt{8\pi}} \left(1 - \operatorname{erf}\left(\frac{r' - p_x}{\sqrt{2}}\right)\right) \int_{\theta} e^{-\frac{1}{2}p_x^2 \sin^2 \theta} p_x \cos \theta d\theta \\ &\approx \frac{1}{4} \left(1 - \operatorname{erf}\left(\frac{r' - p_x}{\sqrt{2}}\right)\right) \left(\operatorname{erf}\left(\frac{p_x \sin \theta_2}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{p_x \sin \theta_1}{\sqrt{2}}\right)\right) \end{aligned} \quad (5)$$

Here, we determine θ_1 and θ_2 from angle centered on the sensing location that bounds the transformed cell. Additionally, the radial term r' is given by the closest radial point, which is a faithful approximation for cells outside of the 3-sigma occupied ellipse. Fig. 3a shows the approximation error for a range of values for r' and p_x .

2) *3D Voxel Grid* : Direct extension of the 2D approach to 3D is not possible, see Section VI. However, noticing that components are approximately planar (see Fig. 2b), the problem can be converted to the integration of a bivariate Gaussian over a convex polygonal domain. The domain of integration is determined by the projection of the voxel of interest to the plane defined by the eigenvector of the Gaussian covariance corresponding to the smallest eigenvalue. The bivariate Gaussian is defined by the remaining two dimensions of the 3D Gaussian and the isotropic transform outlined in Section III-C is applied. As this is an affine transform, the convexity of the projected polygon is preserved. This 2D integral has received some attention in the numerical integration literature [20, 21], however, these approaches are designed to provide significantly higher accuracy than is necessary for our application. A less complex, albeit less accurate, strategy can then be achieved through application of Green's Theorem

$$\frac{1}{2\pi} \int_P e^{-\frac{1}{2}(x^2+y^2)} dx dy = \frac{1}{\sqrt{8\pi}} \oint_{\partial P} e^{-\frac{1}{2}x^2} \operatorname{erf}\left(\frac{y(x)}{\sqrt{2}}\right) dx \quad (6)$$

The path of integration ∂P is defined piece-wise as the line segments defining the edges of the polygon traversed in counter clock-wise order. Unfortunately, the path integral Eq. (6) has no-analytic solution. This is remedied by introducing the function $e(x) = 1 - \exp(-\frac{2}{\pi}x(x + \sqrt{\pi}))$, from which a piece-wise approximation of the error function is constructed.

$$\operatorname{erf}(x) \approx \hat{e}(x) = \begin{cases} e(x) & x \geq 0 \\ -e(-x) & x < 0 \end{cases} \quad (7)$$

This approximation has a maximum absolute error of 0.0140, a maximum absolute relative error of 0.0166 and an average absolute error of 0.0048, which is sufficiently accurate for our application.

Each edge k of the N_P edges in ∂P is defined by the line $y = m_k x + c_k$. The path integral Eq. (6) can be decomposed into a sum of integrals along the edges

$$\frac{1}{\sqrt{8\pi}} \sum_{k=1}^{N_P} \int_{x_k}^{x_{k+1}} e^{-\frac{1}{2}x^2} \hat{e}(m_k x + c_k) dx \quad (8)$$

The solution for each edge integral, I_k , is given by

$$\frac{\pi}{\sqrt{2}} \left(\frac{e^{\frac{m_k^2 - c_k(2s_y c_k + \sqrt{2\pi})}{d_k}}}{\sqrt{d_k}} \operatorname{erf}\left(\frac{m_k(2s_y c_k + \sqrt{2\pi}) + s_y d_k x}{\sqrt{2\pi d_k}}\right) - s_y \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) \right)$$

where $d_k = 2m_k^2 + \pi$ and $s_y = \operatorname{sign}(y(x))$. Due to the piece-wise definition of $\hat{e}(x)$, it is necessary to evaluate I_k in a piece-wise manner. If the sign of y_k and y_{k+1} differ, then the I_k must be evaluated at the x intercept ($x_z = -\frac{c}{m}$). In general, the solution to Eq. (6) is given by

$$\frac{1}{\sqrt{8\pi}} \sum_{k=1}^{N_P} (I_k|_{x_{k+1}} - I_k|_{x_k}) \quad (9)$$

3) *Monte Carlo Estimation*: While the previous expression is useful for computing the free-space evidence at a point, a more efficient method for estimating occupancy over a larger volume is to sample points from the occupancy model, ray-trace through a voxel grid representation [22] to the sensor origin to determine intersecting cells, and update the probability of occupancy inside each cell.

D. Occupancy Modeling

As mentioned in Section III-A, it is possible to determine the number of observed points in any volume by solving the integral

$$N_V^H = N \sum_{k=1}^K \pi_k \int_V \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k) \quad (10)$$

Since this integral cannot be computed in closed form, we evaluate Eq. (10) for \mathcal{G} using Monte Carlo sampling on the non-max-range components. The free space evidence N_V^M is then determined by performing ray casting using points resampled from all components in \mathcal{G} as described in Section III-B. Alternatively, we can compute the free space evidence using the result Eq. (5) in 2D or Eq. (9) in 3D. We model the probability of occupancy as a Bernoulli distribution over every volume V in space. Specifically, we have $p(o|V) \sim \theta^{(o)}(1-\theta)^{(1-o)}$, where $o = \{0, 1\}$ is the binary occupancy variable. The maximum *a posteriori* estimate of the distribution parameters is given by

$$\theta^{MAP} = \frac{N_V^H + N_p}{N_V^H + N_V^M + 2N_p} \quad (11)$$

where N_p is the contribution due to the prior and is user-defined to control the sensitivity of the distribution to new observations. The prior model forces all unobserved points to have a probability of occupancy of 0.5.

The algorithm **GenerateOccupancy3D** provides pseudocode for deriving occupancy from resampled points in an occupied GMM. Line 1 samples N points from the GMM \mathcal{G} . The list

of cells between the sensor origin s and the endpoint p_n are obtained as C_M , where $m \in [1, \dots, M]$ in line 3. If $m < M$, the number of miss points, N_V^M , is incremented as shown on line 6. c_M is the cell corresponding to the occupied point p_N , so the contents of the cell are updated by incrementing N_V^H shown on line 8. Removing lines 5 and 7-9 yields the algorithm for updating the cells via a max-range GMM. In this work, the `getRayCells` function is implemented via the approach proposed by Amanatides *et al.* [22].

It is important to note that the voxelization presented here is not the same as standard voxel grid mapping, since the voxel grid is used as a convenient interpretation of the underlying, continuous GMM. We could equivalently compute the occupancy in an arbitrary discretization of the space. However, this would likely preclude the use of the proposed efficient Monte Carlo ray tracing strategy and would instead require computation of the intersection between the discretized shapes and the sampled beams.

GenerateOccupancy3D(s, \mathcal{G}, N) Pseudocode to update the probability of occupancy of occupied GMM \mathcal{G} given sensor location $s \in SE(3)$ and number of points to sample N

```

1:  $\mathcal{P}_N = \text{samplePoints}(\mathcal{G}, N)$ 
2: for  $p_n \in \mathcal{P}_N$  do
3:    $C_M = \text{getRayCells}(s, p_n)$ 
4:   for  $c_m \in C_M$  do
5:     if  $m \neq M$  then
6:        $\text{updateMiss}(c_m)$  // increment  $N_V^M$  for cell  $c_m$ 
7:     else
8:        $\text{updateHit}(c_m)$  // increment  $N_V^H$  for cell  $c_m$ 
9:     end if
10:  end for
11: end for

```

IV. RESULTS

The GMM implementation used throughout this work is derived from the `scikit-learn`¹ toolkit and ported to C++ to decrease the runtime. The runtimes are collected on a late 2013 15" Macbook Pro with 2.60GHz Intel Core i7-4960HQ and 16GB RAM. All approaches are tested single-threaded.

A. 2D GMM Occupancy Modeling

The 2D occupancy model is tested on the Intel Research Lab environment from the Robotics Data Set Repository [23]. The Monte Carlo and Analytic Approximation GMM occupancy reconstruction approaches are compared against standard occupancy grid mapping and the recently proposed Hilbert maps [7]. Code provided by the authors is used to compare against Hilbert Maps² with the recommended parameter settings as in [7].

The accuracy and robustness of the proposed approach to sparsity in the observed data is shown in Fig. 4. For 10cm resolution, the grid size is well matched to the density of the sensor and occupancy grid maps yield very strong

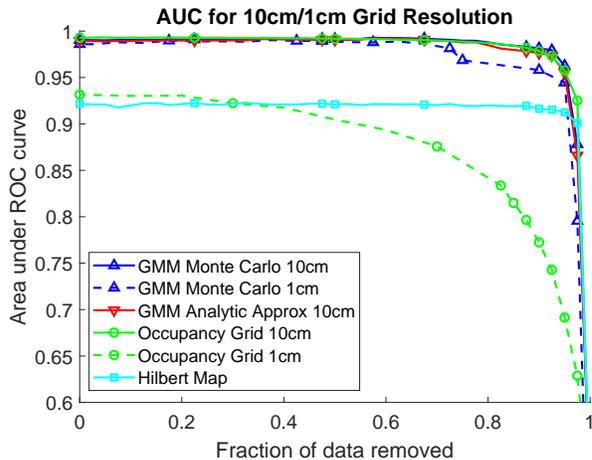


Fig. 4: Area under the ROC curve for 10cm and 1cm resolution occupancy grids. Raw occupancy maps and both GMM variants show equivalent performance superior to that of Hilbert maps. The strong performance of standard occupancy grids can be attributed to the density of the data relative to the resolution. In the case of 1cm grids, this measurement density is no longer sufficient demonstrating that the proposed approach is capable of reconstructing resolutions beyond that of the sensor.

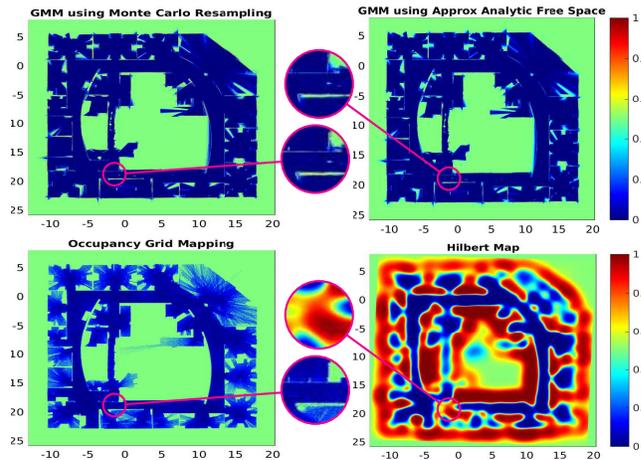


Fig. 5: Occupancy Grid maps generated at 1cm resolution. Both GMM methods demonstrate the ability to generalize beyond the raw sensor measurement. The basic occupancy grid map (bottom right) shows inadequacy of the sensor density for this mapping resolution. The Hilbert Map generates an overly conservative estimate, resulting in incorrectly modeling free space as occupied.

performance, which is matched by that of the two variants of the proposed approach. As the resolution is increased to 1cm, the sparsity in the data and the independence assumption inherent in occupancy grids results in reduced performance. Since the proposed analytic approach is less efficient for large grid sizes, we evaluate only the Monte Carlo strategy. We see comparable performance to the 10cm resolution grid for up to 70% data sparsity, after which point the sparse data combined with the high resolution results in regions with little evidence. However, we note that at both resolutions, the proposed approach outperforms Hilbert Maps, which has a fixed intrinsic resolution determined by the kernel parameters used to learn the classifier.

The performance of the proposed approaches is compared qualitatively in Fig. 5. In the top row, the dense occupancy grids demonstrate the ability for the proposed approaches to reconstruct at high resolution. In the bottom left, the sparsity in standard occupancy grid mapping can be clearly observed. In

¹<http://scikit-learn.org/stable/modules/mixture.html>
²https://bitbucket.org/LionelOtt/hilbert_maps_rss2015

the bottom right, we see the continuous occupancy distribution generated using Hilbert Maps. The magnified regions illustrate the sparsity in occupancy mapping and the conservative nature of Hilbert Maps, resulting in designation of the corridor as occupied. For the two proposed approaches, we see some free space evidence outside of the corner, which occurs due to the infinite extent of the GMM. However, we note that the occupied space is appropriately designated.

B. 3D GMM Occupancy Modeling

The GMM occupancy model is evaluated with two datasets: a mine dataset (reconstruction shown in Fig. 6c) and the Freiburg campus dataset (raw pointcloud shown in Fig. 7a). For each pose and pointcloud, a free and occupied GMM are computed to represent the environment. 10^6 points are sampled from the distribution and raytraced to the sensor origin and the probability of occupancy is updated for each 25 cm voxel along the ray. To determine the accuracy of reconstruction, the Area under ROC (AUROC) is computed while varying the percentage of observations removed. The proposed approach is compared against the standard occupancy grid map and Bayesian Generalized Kernel Inference for occupancy map prediction [8] (BGK-OM) approaches in Fig. 6a, and demonstrates superior results in terms of inferring probability as the number of training points decreases. Timing results shown in Fig. 6b demonstrate the superior performance of the proposed approach. While the times reported in [8] are significantly lower, than what is reported here, and the discrepancy can be accounted for by the following: 1. BGK-OM is highly parallelized with 8 threads so for a fair comparison the multi-threading is disabled; 2. the max range of the sensor in these tests is higher (15 m) than what is reported in Doherty et al. [8] (4 and 8 m); 3. the environment is larger ($100\text{m} \times 100\text{m} \times 8\text{m}$) as compared to the largest environment reported in Doherty et al. [8] ($43.8\text{m} \times 18.2\text{m} \times 3.3\text{m}$); 4. the maximum number of points used per scan reported in is 7601 but the tests in Fig. 6b contain between 15,000-20,000 points when 0% of the observations are removed and after downsampling with the voxel grid filter. The dense voxel filter resolution is left unchanged from the default parameters (0.1 m) but when the sensor observations are sparse and the environment is large, few points can be removed. The priors are set to 0.5, block size is set to 1, and the resolution is 0.25 in keeping with the other approaches. The other parameters are unchanged.

The key advantages of the GMM occupancy model over state-of-art occupancy mapping approaches like the occupancy

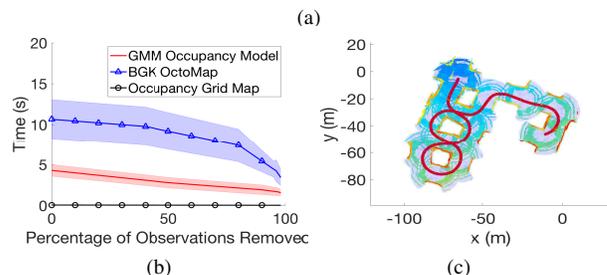
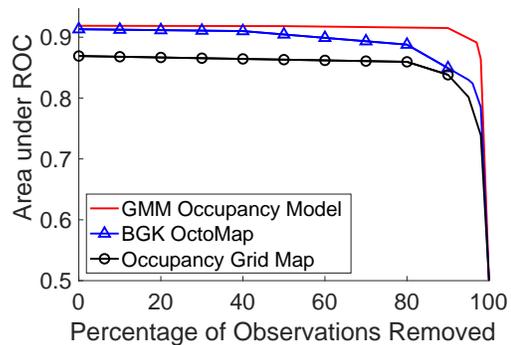


Fig. 6: (a) The effect of data sparsity on the fidelity of the map is measured by computing the area under curve for several models constructed by increasingly reducing the percentage of observations removed from the original data. (b) provides timing analysis for each method along with variance in the lighter shaded region. (c) illustrates the reconstruction of the map after sampling from the GMM. 100-component GMMs were used to generate these results.

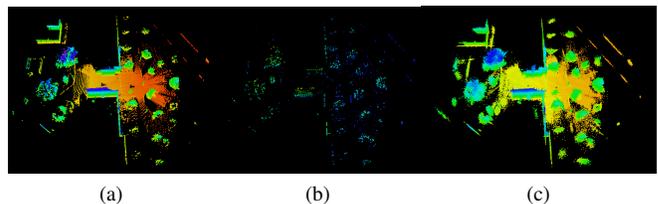


Fig. 7: (a) Raw pointcloud from Freiburg campus dataset with coloring according to localization along the z-axis (total is 447,528 points and approximately 5.12MB of data assuming 32-bit floats). (b) Reconstruction of occupied regions with BGK inference for occupancy maps using 9472 points (111 KB data storage requirement). The voxels displayed are those with a probability of occupancy greater than or equal to 0.65. The AUC is 0.72. (c) Reconstruction of GMM map using 1000 components (40 KB data storage requirement). The AUC is 0.82.

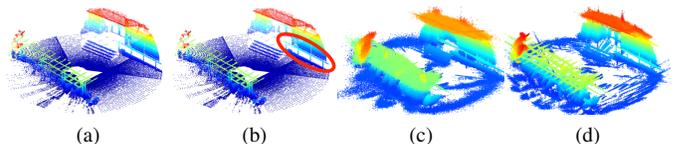


Fig. 8: (a) A pointcloud of a courtyard from the Freiburg campus dataset with (b) a railing outlined in red. (c) The occupied points from the grid map derived by Monte Carlo sampling a 200-component GMM and (d) 850-component GMM are shown after applying a threshold using variance = 0.06. The AUC for (c) is 0.79 and the AUC for (d) is 0.82. In both cases the railing is properly reconstructed.

grid map, Octomap, or BGK-OM is the ability to correct for small or large changes in pose without the need to regenerate the entire map via (2). BGK-OM requires significantly more data storage to obtain the same level of accuracy as can be seen in the results from the Freiburg Campus Dataset in Table I. Because the standard occupancy grid map under performs as compared to both the proposed and BGK-OM approaches in Fig. 6, the memory requirements would be larger to achieve the same AUC. Furthermore, the timing needs for regenerating the entire map are prohibitive for real-time applications. Individual voxels cannot be meaningfully rotated and translated; however, individual components of a GMM are

	AUC	Memory Usage (Bytes)	
GMM			Number of Components
	0.8179	40,000	1000
	0.8072	20,000	500
	0.8055	10,000	250
	0.7849	4,000	100
BGK			Number of Points
	0.7603	204,132	17,011
	0.7182	113,664	9,472
	0.6338	60,120	5,010
	0.5845	37,140	3,095
	0.5428	18,660	1,555

TABLE I: AUC for given memory usage of GMM occupancy model compared to the performance of BGK inference occupancy maps. The GMM is better able to reconstruct the occupancy information in the environment with $50\times$ less data than the Bayesian Generalized Kernel inference approach for occupancy mapping.

easily translated and rotated to correct for errors in pose [24].

The GMM infers the presence of occupied space in the neighborhood of observed points. However, the inferred occupied space may be unobservable from the sensor location due to occlusions from other surfaces. As a result, the inferred density may contribute erroneous free space evidence. For example, Fig. 8a depicts a laser scan from the Freiburg campus dataset that exhibits clutter and challenging surfaces to model in the form of a railing outlined in Fig. 8b. The inferred, but unobserved, portion of the wall contributes free space evidence to the voxels containing the railing, resulting in the probability of occupancy dropping for the railing voxels. To account for such situations, we leverage the variance of the voxels to determine the occupancy classification levels. The variance of a voxel is simply the variance of a Bernoulli random variable

$$\sigma^2 = p(x_i)(1 - p(x_i)) \quad (12)$$

The variance quantifies the uncertainty in the occupancy value of the cell and introduces a dead-band region in which voxels remain unclassified. The occupied/free thresholds are given by $\frac{1}{2}(1 \pm \sqrt{1 - 4\sigma^2})$. Appropriate selection of $\sigma^2 \in [0, 0.25]$ enables a threshold to be applied depending on how conservative we wish to be for a given application. Fig. 8c illustrates the occupied cells when a variance thresholding is applied when reconstructing a GMM with 200 components. The railing is well represented in the reconstruction and the AUC is 0.79. Fig. 8d illustrates the same result with 850 GMM components and an AUC of 0.82.

C. Component Selection

Selecting the right number of components to model a GMM is an open area of research. A hierarchical, top-down strategy is developed by Eckart *et al.* [10] that produces a GMM by successively partitioning the data into J leaf nodes such that each leaf is a Gaussian density and $J \ll N$, where N is the number of points in the pointcloud. The downside of this approach is that user-defined convergence criterion must be set to determine when the point cloud has been sufficiently segmented. Srivastava and Michael [9] propose a bottom-up strategy that successively merges components until a knee-point is achieved. In addition, information-theoretic criterion such as the Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) have also been used to determine the elbow-point where adding an additional component does not significantly add information; however, the elbow-point cannot always be unambiguously identified. The plots of BIC and AIC for a pointcloud shown in Fig. 2a are shown in Fig. 9a and Fig. 9b, respectively, for numbers of components between 1 and 200.

We propose another method to determine the number of components to ensure a maximum resolution inspired by the work of Isler *et al.* [25] that measures surface coverage by discretizing the space into a voxel grid with very small cell size and counting the number of rays that collide with a surface in the voxel. For a given desired maximum resolution, the number of components needed to achieve that resolution may be determined by computing a GMM, resampling from the

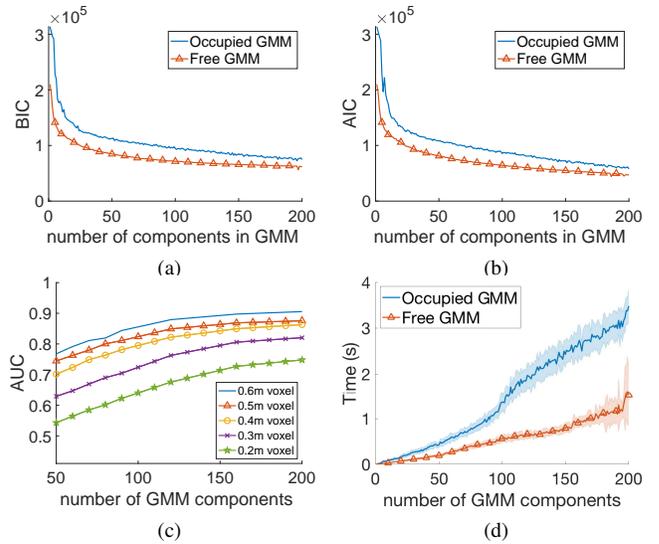


Fig. 9: (a) The BIC and (b) AIC are often used to determine the number of components to represent a distribution. The proposed approach (c) ensures max resolution through comparison to an occupancy grid map. (d) Timing results for the occupied and free space GMMs as the number of components increases. A visualization of the 200-component GMMs may be seen in Fig. 2b.

distribution a number of points equal to the support size, and raytracing through an occupancy grid map (note: not a Bernoulli grid). The AUC is computed by comparing the resulting occupancy grid map with a ground truth occupancy grid map whose values are determined by raytracing the original point cloud. Figure 9c illustrates the AUC scores when varying the number of components in the GMMs for free and occupied points.

The intuition behind this approach is that as the number of components in the model approaches the number of points in the sensor observation, the resulting AUC score will approach 1. It will be 1 only when the number of components exactly matches the number of points in the sensor observation. This representation also gives a more intuitive understanding of how selecting the number of components will affect the reconstruction. For example, when the desired cell size is 0.6m, 50 components may be sufficient, but when increasing the resolution to 0.3m cell size, 200 components would be needed to obtain similar AUC accuracy. To obtain good performance a trade-off must occur between the time it takes to calculate the GMM and the desired maximum resolution. Figure 9d illustrates the time to compute a GMM as the number of components increases. The times may be significantly reduced by employing the hierarchical strategy of [10]. This is left as future work discussed in the next section.

V. CONCLUSIONS AND FUTURE WORK

This paper proposed a continuous occupancy mapping methodology that enables efficient storage of a high-fidelity model of both observed occupied space and free space while remaining amenable to local or global updates in pose. Through Monte-Carlo sampling and ray-tracing, resolution beyond that of the sensor can be achieved. Furthermore, methods for accurately determining the free space evidence for arbitrary (in 2D) and polyhedral (in 3D) regions were presented. The fidelity of the approach was demonstrated both

quantitatively and qualitatively against Hilbert Maps [7] in 2D and Bayesian Generalized Kernel Inference based occupancy maps [8] in 3D. The results show a substantial reduction in complexity of the representation, while achieving superior modeling accuracy. Future work will focus on efficient evaluation of the analytic approximate solutions, consideration of non-voxel based representations and the adaptive variance determination to enable better matching of the model to the environmental resolution, and develop strategies to enable faster calculation of GMMs.

REFERENCES

- [1] S. T. OCallaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, 2012.
- [2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [3] M. Magnusson, "The Three-Dimensional Normal-Distributions Transform: An Efficient Representation for Registration, Surface Analysis, and Loop Detection," 2013. [Online]. Available: <https://pdfs.semanticscholar.org/120a/f9c331f9159229d0246b1ebb5ae2592cdf6a.pdf>
- [4] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal, "Normal distributions transform occupancy maps: Application to large-scale online 3d mapping," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2233–2238.
- [5] E. Einhorn and H.-M. Gross, "Generic ndt mapping in dynamic environments and its application for lifelong slam," *Robotics and Autonomous Systems*, vol. 69, pp. 28–39, 2015.
- [6] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Warped gaussian processes occupancy mapping with uncertain inputs," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 680–687, 2017.
- [7] F. Ramos and L. Ott, "Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, 2016.
- [8] K. Doherty, J. Wang, and B. Englot, "Bayesian generalized kernel inference for occupancy map prediction," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3118–3124.
- [9] S. Srivastava and N. Michael, "Approximate continuous belief distributions for precise autonomous inspection," in *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 74–80.
- [10] B. Eckart, K. Kim, A. Troccoli, A. Kelly, and J. Kautz, "Accelerated generative models for 3d point cloud data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5497–5505.
- [11] W. Tabib, C. OMeadhra, and N. Michael, "On-manifold gmm registration," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3805–3812, Oct 2018.
- [12] R. T. Marriott, A. Pashevich, and R. Horaud, "Plane-extraction from depth-data using a gaussian mixture regression model," *Pattern Recognition Letters*, vol. 110, pp. 44–50, 2018.
- [13] L. Sun, T. Vidal-Calleja, and J. V. Miro, "Coupling conditionally independent submaps for large-scale 2.5 d mapping with gaussian markov random fields," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3131–3137.
- [14] A. Osman Ulusoy, M. J. Black, and A. Geiger, "Patches, planes and probabilities: A non-local prior for volumetric 3d reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3280–3289.
- [15] S. Srivastava, "Efficient, multi-fidelity perceptual representations via hierarchical gaussian mixture models," Master's thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh PA, August 2017.
- [16] S. Kim and J. Kim, "Building occupancy maps with a mixture of gaussian processes," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4756–4761.
- [17] —, *GPmap: A Unified Framework for Robotic Mapping Based on Sparse Gaussian Processes*, L. Mejias, P. Corke, and J. Roberts, Eds. Cham: Springer International Publishing, 2015. [Online]. Available: https://doi.org/10.1007/978-3-319-07488-7_22
- [18] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Gaussian processes autonomous mapping and exploration for range-sensing mobile robots," *Autonomous Robots*, vol. 42, no. 2, pp. 273–290, Feb 2018. [Online]. Available: <https://doi.org/10.1007/s10514-017-9668-3>
- [19] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, New York, 2007.
- [20] N. Balakrishnan and C.-D. Lai, *Continuous Bivariate Distributions*. Springer, New York, 2009.
- [21] A. Didonato, M. Jarnagin, Jr, and R. Hageman, "Computation of the integral of the bivariate normal distribution over convex polygons," *SIAM Journal on Scientific and Statistical Computing*, vol. 1, no. 2, pp. 179–186, 1980.
- [22] J. Amanatides, A. Woo *et al.*, "A fast voxel traversal algorithm for ray tracing," in *Eurographics*, vol. 87, 1987, pp. 3–10.
- [23] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: <http://radish.sourceforge.net/>
- [24] G. Rigas, C. Nikou, Y. Goletsis, and D. I. Fotiadis, "Hierarchical similarity transformations between gaussian mixtures," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 11, pp. 1824–1835, 2013.
- [25] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3d reconstruction," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3477–3484.

VI. APPENDIX

3D Voxel Grids - Angular Integration

Following a similar approach to the 2D case, we arrive at an integral over the angular coordinates θ and ϕ . Applying the standard transformation to spherical coordinates yields

$$p(r, \theta, \phi) = \frac{1}{(2\pi)^{3/2}} e^{-\frac{1}{2} \left\| \begin{bmatrix} r \cos \phi \sin \theta \\ r \sin \phi \sin \theta \\ r \cos \theta \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ p_z \end{bmatrix} \right\|_2^2} \quad (13)$$

$$= \frac{1}{(2\pi)^{3/2}} e^{-\frac{1}{2} p_z^2 \sin^2 \theta} e^{-\frac{1}{2} (r - p_z \cos \theta)^2} \quad (14)$$

The volume integral can be specified in spherical coordinates as

$$Evidence(\mathbf{x}) = \int_V p(r, \theta, \phi) r^2 \sin \theta dr d\theta d\phi \quad (15)$$

Integrating out the radial dimension (from r' , the distance to the closest point of interest) allows the angular distribution to be specified in terms of the complementary error function $\text{erfc}(x)$. For brevity, we denote $s(\theta) = \sin \theta$ and $c(\theta) = \cos \theta$.

$$\begin{aligned} E(\mathbf{x}) &= \frac{1}{(2\pi)^{\frac{3}{2}}} \int_{\phi, \theta} e^{-\frac{1}{2} p_z^2 s^2(\theta)} s(\theta) \int_{r'}^{\infty} r^2 e^{-\frac{1}{2} (r - p_z c(\theta))^2} dr d\theta d\phi \\ &= \frac{1}{(2\pi)^{\frac{3}{2}}} \int_{\phi, \theta} e^{\frac{1}{2} \mu_z^2 s^2(\theta)} s(\theta) \left[e^{-\frac{1}{2} (r' - p_z c(\theta))^2} (r' + p_z c(\theta)) \right. \\ &\quad \left. + \sqrt{\frac{\pi}{2}} (1 + p_z^2 c^2(\theta)) \text{erfc} \left(\frac{r' - p_z s(\theta)}{\sqrt{2}} \right) \right] d\theta d\phi \end{aligned}$$

When considering the evidence for free space in a voxel, the domain of integration for ϕ, θ is a spherical (or geodesic) polygon defined by the shadow cast by V as viewed from the sensor. However, it is difficult to integrate over this region in closed form, which necessitates resorting to Monte Carlo estimation or the polygon projection technique described in Section III-C2.