Simultaneous Localization and Mapping of Subterranean Voids with Gaussian Mixture Models

Wennie Tabib and Nathan Michael

Abstract This paper presents a real-time viable method for Simultaneous Localization and Mapping (SLAM) using Gaussian mixture models (GMMs) for compute-constrained systems that operate in subterranean environments. The two contributions of this work are (1) a SLAM formulation that uses a GMM-based map representation for pose estimation, mapping and loop closure and (2) an Expectation Maximization (EM) formulation that significantly reduces the time to learn a GMM from a sensor observation by exploiting the insight that although Gaussian distributions have infinite support, a substantial amount of the support is contained within a finite region. An on-manifold distribution-to-distribution registration approach is used to estimate pose between consecutive GMMs and the Cauchy-Schwarz divergence is employed to calculate the difference between the distributions to identify loop closures. The method is evaluated in mine and unstructured cave environments. The results demonstrate superior performance in leveraging the compact representation of the GMM as compared to traditional pose graph SLAM techniques that rely on pointcloud-based methods. Further, exploiting the sparsity of the compact support significantly reduces training time towards enabling real-time viability.

1 Introduction

Search and rescue operators benefit from enhanced situational capabilities provided by robots during time-critical and life-threatening operations [19]. Robot perception in disaster environments is challenging as these environments are highly cluttered, unstructured and unpredictable. In addition, communications infrastructure may deteriorate or become completely disabled over extensive areas for hours or days during natural disasters [20] hindering search and rescue operations. These

Wennie Tabib
Carnegie Mellon University, Pittsburgh, PA 15213, USA e-mail: wtabib@cmu.edu

Nathan Michael
Carnegie Mellon University, Pittsburgh, PA 15123, USA e-mail: nmichael@cmu.edu

environmental constraints create the need for compact, efficient environment representations that are transmissible over low bandwidth communications networks.

GMMs compactly represent high-resolution sensor observations and have been demonstrated to enable occupancy modeling [21] and estimate pose [26] with orders of magnitude reduction in memory required to store and transmit the data as compared to raw sensor observations [27]. While Eckart [10] provides a qualitative evaluation of a laser-based SLAM formulation leveraging hierarchical GMMs, to the best of the authors' knowledge, a quantitative analysis of GMM-based SLAM that incorporates loop closures has not been conducted. The proposed work bridges this gap in the state of the art by developing GMM mapping that incorporates global consistency and exploits the sparse compact support to reduce the dimensionality of calculations.

The paper is organized as follows: Section 2 details related work followed by a description of the methodology in Section 3. Section 4 provides an analysis of the proposed approach as compared to the state of the art and Sections 5 and 6 conclude the paper with a discussion of the limitations of the method and future work.

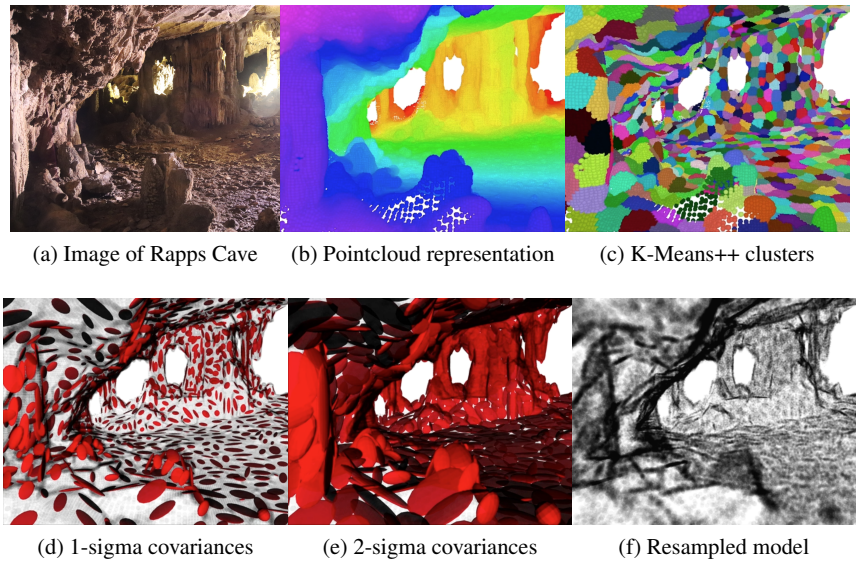
2 Related Work

Generative modeling has experienced a resurgence in recent years due to the vision that disparate pointcloud-based perception algorithms may be unified into a common pipeline [10]. A top-down hierarchical GMM approach is developed by Eckart et al. [8] to accelerate learning by employing a sparsification technique that adjusts the posterior between levels so that child mixture components share geometric context information in a soft partitioning scheme. The proposed approach does not employ a hierarchy but leverages the Mahalanobis distance as a sparsification technique in Expectation Maximization. Srivastava and Michael [23] develop a bottom-up hierarchical approach that assumes known pose estimates and merges partial sensor observations into one monolithic GMM. This approach is susceptible to accumulation of noise and pose drift, which makes corrections to enable global consistency difficult. In contrast, the proposed approach represents each sensor observation as a GMM to enable corrections when closing the loop.

The Normal Distribution Transform map is learned by voxelizing a sensor observation and calculating a Gaussian density within each cell [24]. Pose is estimated by minimizing the L2 distance between the two distributions. While fast, this approach has been determined to be less accurate than point-based approaches like Generalized-Iterative Closest Point (GICP) [26].

Evangelidis and Horaud [12] develop a batch registration algorithm for multiple point sets that estimates GMM parameters, rotations, and translations via an Expectation Maximization algorithm. While accurate, it is not real-time viable. Eckart et al. [9] develop an approach that simultaneously trains and registers GMMs. The Mahalanobis distance is used to compute the distance between points and estimate optimal rotation and translation for registration. While the accuracy is on par with several ICP variants, the approach is evaluated on a NVIDIA Titan X GPU, which is prohibitive for use on size, weight, and power constrained systems.

Behley and Stachniss [2] develop a SLAM formulation for 3D laser range data, but require a GPU to achieve real-time results. The authors opt for a frame-to-model ICP that registers a pointcloud to surfel model. Loop closure detection proceeds by checking nearby poses against the current laser scan with ICP and rendering a view of the current map to determine if the loop closure leads to a consistent map given the current scan. Multiple initializations of ICP with different translations and rotations are required to determine an adequate pose estimate. In contrast to ICP, GICP has been demonstrated to be more robust to large changes in rotation and translation [22]. The proposed approach is compared to [2].



(a) Image of Rapps Cave (b) Pointcloud representation (c) K-Means++ clusters

(d) 1-sigma covariances (e) 2-sigma covariances (f) Resampled model

Fig. 1: Overview of the process to learn a GMM from a sensor observation. (a) An image taken from Rapps Cave in WV, USA. (b) A corresponding pointcloud colored according to viewing distance (red is further away). (c) Each color corresponds to a cluster learned with the K-Means++ algorithm. The red ellipsoids in (d) and (e) are visualizations of the 1-sigma and 2-sigma contours of constant probability of the Gaussian mixture components after running EM, respectively. The 1-sigma visualization represents approximately 20% of the probabilistic coverage of the underlying point density and 2-sigma approximately 74%. Because the GMM is a generative model, samples may be drawn from the distribution to obtain the reconstruction in (f).

3 Methodology

In this work, each sensor observation is represented as a GMM. Figure 1 illustrates the method by which a GMM is learned from a sensor observation to generate an environment map. A mathematical description of the GMM and the approach to enable real-time viable parameter estimation is detailed in Section 3.1. Section 3.2 describes the distribution-to-distribution registration, SLAM, and loop closure approaches.

3.1 Gaussian Mixture Model

A GMM is a probability distribution that represents multivariate data as a weighted combination of M Gaussian distributions. The probability density of the GMM is represented as

$$p(\boldsymbol{x}|\boldsymbol{\xi}) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$$

where $\boldsymbol{x} \in \mathbb{R}^D$, π_m is a weight such that $0 \leq \pi_m \leq 1$, $\sum_m^M \pi_m = 1$, and $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda})$ is a D-dimensional Gaussian density function with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Lambda}$.

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \frac{|\boldsymbol{\Lambda}|^{-1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Lambda}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \right)$$

The parameters of the distribution are compactly represented as $\boldsymbol{\xi} = \{\pi_m, \boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m\}_{m=1}^M$. Estimating the parameters of a GMM remains an open area of research [14]. Given the density function $p(\boldsymbol{x}|\boldsymbol{\xi})$ and observations $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{x} \in \mathbb{R}^D$ assumed to be independent and identically distributed with distribution p, the density for the samples is

$$p(\boldsymbol{X}|\boldsymbol{\xi}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\xi}) = \mathscr{L}(\boldsymbol{\xi}|\boldsymbol{X})$$

where $\mathscr{L}(\boldsymbol{\xi}|\boldsymbol{X})$ is called the likelihood function and the goal is to find the $\boldsymbol{\xi}^*$ that maximizes $\mathscr{L}$ [3]

$$\boldsymbol{\xi}^* = \arg\max_{\boldsymbol{\xi}} \ \mathscr{L}(\boldsymbol{\xi}|\boldsymbol{X})$$

It is analytically easier to maximize $\ln(\mathscr{L}(\boldsymbol{\xi}|\boldsymbol{X}))$, but the presence of the summation over m inside the logarithm make $\boldsymbol{\xi}$ difficult to compute and taking the derivative of this log likelihood function and setting to zero is made intractable because the resulting equations are no longer in closed form [4]. Instead, latent variables $b_{nm} \in \boldsymbol{B}$ are introduced that take a value of 1 if the sample $\boldsymbol{x}_n$ is in cluster m and 0, otherwise (called a 1-of-M coding scheme). A new likelihood function is defined $p(\boldsymbol{X}, \boldsymbol{B}|\boldsymbol{\xi}) = \mathscr{L}(\boldsymbol{\xi}|\boldsymbol{X}, \boldsymbol{B})$, called the complete data likelihood.

The Expectation step in Expectation Maximization finds the expected value of $\ln p(\boldsymbol{X}, \boldsymbol{B}|\boldsymbol{\xi})$ by the following function [3]

$$Q(\boldsymbol{\xi}, \boldsymbol{\xi}^i) = E\left[\ln p(\boldsymbol{X}, \boldsymbol{B}|\boldsymbol{\xi}) | \boldsymbol{X}, \boldsymbol{\xi}^i \right]$$

The Maximization step maximizes the expectation of the previous equation:

$$\boldsymbol{\xi}^{i+1} = \arg\max_{\boldsymbol{\xi}} Q(\boldsymbol{\xi}, \boldsymbol{\xi}^i)$$

Each iteration of these steps is guaranteed to increase the log likelihood and ensure that the algorithm converges to a local maximum of the likelihood function [3].

3.1.1 Initialization

Kolouri et al. [18] find the EM algorithm to be sensitive to the choice of initial parameters and Jian and Vemuri [15] prove that random initialization causes the EM algorithm to converge to a bad critical point with high probability. Because the EM algorithm does not guarantee convergence to a global optimum, the initialization is critical for convergence to a good stationary point. The proposed approach implements the K-Means++ algorithm [1], which is an unsupervised learning algorithm that provides an initial clustering of the sensor data (see Fig. 1c) and has advantages over the standard K-Means algorithm in that it provides approximation guarantees for the optimality of the algorithm that improve the speed and accuracy. Several variants of K-Means are proposed in the literature [6, 11, 13], but this work leverages the method proposed in Elkan [11] as it was found to achieve the best performance. Elkan [11] increases efficiency by employing the triangle inequality and maintaining lower and upper bounds on distances between points and centers. Given the range sensor data in Fig. 1b, the K-Means++ algorithm outputs the clustering shown in Fig. 1c. These clusters are used to seed the EM algorithm detailed in the next section.

3.1.2 Expectation Maximization

The EM algorithm proceeds with the following steps:

1. Initialize $\boldsymbol{\mu}_m$, $\boldsymbol{\Lambda}_m$ and π_m with the method detailed in Section 3.1.1.
2. **E step**. Evaluate the responsibilities γ_{nm} using the current parameters $\boldsymbol{\mu}_m$, $\boldsymbol{\Lambda}_m$ and π_m:

$$\gamma_{nm} = \frac{\pi_m \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)}{\sum_{j=1}^{M} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Lambda}_j)} \tag{1}$$

3. **M step**. Estimate the new parameters $\boldsymbol{\mu}_m^{i+1}$, $\boldsymbol{\Lambda}_m^{i+1}$ and π_m^{i+1} using the current responsibilities, γ_{nm}.

$$\boldsymbol{\mu}_m^{i+1} = \frac{\sum_{n=1}^{N} \gamma_{nm} \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_{nm}} \tag{2}$$

$$\boldsymbol{\Lambda}_m^{i+1} = \frac{\sum_{n=1}^{N} \gamma_{nm} (\boldsymbol{x}_n - \boldsymbol{\mu}_m^i)(\boldsymbol{x}_n - \boldsymbol{\mu}_m^i)^T}{\sum_{n=1}^{N} \gamma_{nm}} \tag{3}$$

$$\boldsymbol{\pi}_m^{i+1} = \sum_{n=1}^{N} \frac{\gamma_{nm}}{N} \tag{4}$$

4. Evaluate the log likelihood

$$\ln p(\boldsymbol{X}|\boldsymbol{\xi}) = \sum_{n=1}^{N} \ln \left(\sum_{m=1}^{M} \pi_m \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) \right) \tag{5}$$

   and check for convergence of either the parameters or the log likelihood. If convergence is not achieved, iterate again from step 2.

Figures 1d and 1e provide a visualization of the GMM with 1- and 2-sigma covariances after running EM. While the K-means algorithm performs a hard assignment of data points to clusters, the EM algorithm makes a soft assignment based on posterior probabilities. The intuition behind the soft assignment yields one of the contributions of this paper which is that because Gaussians fall off quickly, points far away from an initialized density will have a small effect on the updated parameters for that density. The responsibility matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{N \times M}$ scales with the number of samples N and the number of components M, so to reduce the computation time, an approximation is made to ignore points that lie outside a Mahalanobis distance greater than λ for the initialized density. EM is modified in the following way:

1. Initialize $\boldsymbol{\mu}_m^1$, $\boldsymbol{\Lambda}_m^1$ and π_m^1 with the method detailed in Section 3.1.1.
2. For a given component m, evaluate only the γ_{nm} that satisfy Mahalanobis-bound:

$$\lambda < \sqrt{(\boldsymbol{x}_n - \boldsymbol{\mu}_m^1)^T (\boldsymbol{\Lambda}_m^1)^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_m^1)} \tag{6}$$

3. Estimate the updated parameters $\boldsymbol{\mu}_m^{i+1}$, $\boldsymbol{\Lambda}_m^{i+1}$, and π_m^{i+1} with the current responsibilities γ_{nm} and Eqs. (2) to (4).
4. Evaluate the log likelihood (Eq. (5)) and iterate again from step 2 if convergence is not achieved.

3.2 Pose Graph SLAM via GMM Registration

Each sensor observation is represented as a GMM with M components and successive GMMs are registered together using the approach detailed in [26], which is summarized in Section 3.2.1. Loop closures are detected by aligning observations within a given radius r of the current pose until a match is found that achieves the fitness threshold α.

The pose graph is formulated as a factor graph [16] where factors represent constraints between poses, or nodes. The factor graph $G = (\mathscr{F}, \mathscr{V}, \mathscr{E})$ is composed of factor nodes $f_i \in \mathscr{F}$ and variable nodes $v_j \in \mathscr{V}$ with edges $e_{ij} \in \mathscr{E}$ connecting the factor nodes and variable nodes. The factor graph finds the variable assignment $\mathscr{V}^*$ that maximizes

$$\mathscr{V}^* = \arg\max_{V} \prod_i f_i(V_i) \tag{7}$$

where V_i is the set of variables adjacent to the factor f_i and independence relationships are encoded by the edges e_{ij} such that each factor f_i is a function of the variables in V_i. The pose graph uses relative constraints from GMM registration with a covariance based on the depth sensor model.

3.2.1 Registration

Following the approach of [26], let $\mathscr{G}_i(\boldsymbol{x})$ and $\mathscr{G}_j(\boldsymbol{x})$ denote GMMs learned from sensor observations $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_j$, respectively,

$$\mathscr{G}_i(\boldsymbol{x}) = \sum_{m}^{M} \pi_m \mathscr{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$$
$$\mathscr{G}_j(\boldsymbol{x}) = \sum_{k}^{K} \tau_k \mathscr{N}(\boldsymbol{x}|\boldsymbol{\nu}_k, \boldsymbol{\Omega}_k)$$

and let $T(\cdot, \boldsymbol{\theta})$ denote the rigid transformation consisting of a rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$. To register $\mathscr{G}_j(\boldsymbol{x})$ into the frame of $\mathscr{G}_i(\boldsymbol{x})$, optimal rotation and translation parameters must be found such that the squared L2 norm between the distributions $\mathscr{G}_i(\boldsymbol{x})$ and $T(\mathscr{G}_j(\boldsymbol{x}), \boldsymbol{\theta})$ is minimized. The transformation parameters $\boldsymbol{\theta}$ consisting of $\boldsymbol{R}$ and $\boldsymbol{t}$ may be applied to a GMM $\mathscr{G}_j(\boldsymbol{x})$ in the following way

$$T(\mathscr{G}_j(\boldsymbol{x}), \boldsymbol{\theta}) = \sum_{k=1}^{K} \tau_k \mathscr{N}(\boldsymbol{x}|\boldsymbol{R}\boldsymbol{\nu} + \boldsymbol{t}, \boldsymbol{R}\boldsymbol{\Omega}_k\boldsymbol{R}^T)$$

The cost function is

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \int ||\mathscr{G}_i(\boldsymbol{x}) - T(\mathscr{G}_j(\boldsymbol{x}), \boldsymbol{\theta})||_2^2 d\boldsymbol{x} \tag{8}$$
$$= \arg\min_{\boldsymbol{\theta}} \int ||\mathscr{G}_i(\boldsymbol{x})||_2^2 + ||T(\mathscr{G}_j(\boldsymbol{x}), \boldsymbol{\theta})||_2^2 - 2\mathscr{G}_i(\boldsymbol{x})T(\mathscr{G}_j(\boldsymbol{x}), \boldsymbol{\theta}) d\boldsymbol{x} \tag{9}$$

The first term in Eq. (9) does not depend on the transformation parameters $\boldsymbol{\theta}$ and remains constant; therefore, it may be eliminated. The second term is invariant under rigid transformation and also may be eliminated. Thus, Eq. (8) may be rewritten as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} - \int 2\mathscr{G}_i(\boldsymbol{x})T(\mathscr{G}_j(\boldsymbol{x}), \boldsymbol{\theta}) d\boldsymbol{x}$$

which has a closed form solution as shown in [26]

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} - \sum_{m=1}^{M}\sum_{k=1}^{K} \pi_m \tau_k \mathscr{N}(\boldsymbol{\mu}_m|\boldsymbol{R}\boldsymbol{\nu}_k + \boldsymbol{t}, \boldsymbol{\Lambda}_m + \boldsymbol{R}\boldsymbol{\Omega}_k\boldsymbol{R}^T)$$

The optimal rigid transformation parameters may be solved for by employing a Riemannian trust-region method with conjugate gradients [5]. The registration method used in this work is the Isoplanar variant from [26] that modifies the covariances prior to registration to smooth the cost function. The modified covariance is computed by calculating the eigen decomposition $\boldsymbol{\Lambda}_m = \boldsymbol{U}_m\boldsymbol{D}_m\boldsymbol{U}_m^T$ and replacing the

matrix of eigenvalues, $\boldsymbol{D}_m$, with diag$(\begin{bmatrix}1 & 1 & \varepsilon\end{bmatrix}^T)$ where ε is a small constant (e.g., 0.001) that represents the smallest eigenvalue.

The GMM provides a compressed representation of tens or hundreds of thousands of points in a sensor observation with a small number (several tens or hundreds) of mixture components. Representing the sensor observation in this way yields fast and robust registration because the compactness of this representation enables each pair of mixture components from the source and target distributions to be compared in the registration cost function much more quickly than would be possible when operating directly on the uncompressed pointcloud. In contrast, pointcloud-based techniques like ICP and GICP must bound the search for matching points in order to remain real-time viable.

3.2.2 Loop Closure

When the current estimated pose j is within a fixed radius r away from a previously visited pose i represented in the factor graph, the poses are considered candidates for loop closure. The estimated pose difference is used to seed the registration between GMMs $\mathcal{G}_i(\boldsymbol{x})$ and $\mathcal{G}_j(\boldsymbol{x})$ associated with poses i and j in the factor graph. After registration, the updated pose $\boldsymbol{\theta}$ is used to transform $\mathcal{G}_j(\boldsymbol{x})$ into the frame of $\mathcal{G}_i(\boldsymbol{x})$ by applying the transform $T(\mathcal{G}_j(\boldsymbol{x}),\boldsymbol{\theta})$. To determine if the GMMs partially overlap the same scene, the Cauchy-Schwarz divergence [17] is used to measure the difference between $T(\mathcal{G}_j(\boldsymbol{x}),\boldsymbol{\theta})$ and $\mathcal{G}_i(\boldsymbol{x})$. This measure is employed by Kampa et al. [17] as an entropic measure for classification for distributions and has the same extrema as that of the cost function used in registration. The loop closure problem may be viewed as a classification problem where the goal is to determine whether the scene under consideration by the current view has been previously observed.

Equation (10) is the Cauchy-Schwarz divergence that measures the difference between probability density functions. A closed-form expression for this equation for mixtures of Gaussians is derived in [17].

$$D_{CS}\Big(\mathcal{G}_i(\boldsymbol{x}), T(\mathcal{G}_j(\boldsymbol{x}),\boldsymbol{\theta})\Big) = -\log\left(\frac{\int \mathcal{G}_i(\boldsymbol{x}) T(\mathcal{G}_j(\boldsymbol{x}),\boldsymbol{\theta}) d\boldsymbol{x}}{\sqrt{\int \mathcal{G}_i(\boldsymbol{x})^2 d\boldsymbol{x} \int T(\mathcal{G}_j(\boldsymbol{x}),\boldsymbol{\theta})^2 d\boldsymbol{x}}}\right) \tag{10}$$

This measure is not a metric because it does not satisfy the triangle inequality, but $0 \leq D_{CS}(\mathcal{G}_i(\boldsymbol{x}), T(\mathcal{G}_j(\boldsymbol{x}),\boldsymbol{\theta})) \leq \infty$ and it satisfies the symmetry property, meaning that $D_{CS}(\mathcal{G}_i(\boldsymbol{x}),\mathcal{G}_j(\boldsymbol{x})) = D_{CS}(\mathcal{G}_j(\boldsymbol{x}),\mathcal{G}_i(\boldsymbol{x}))$. Furthermore, the distributions $\mathcal{G}_i(\boldsymbol{x})$ and $\mathcal{G}_j(\boldsymbol{x})$ are the same only when $D_{CS}(\mathcal{G}_i(\boldsymbol{x}),\mathcal{G}_j(\boldsymbol{x})) = 0$.

If $D_{CS}(\mathcal{G}_i(\boldsymbol{x}), T(\mathcal{G}_j(\boldsymbol{x}),\boldsymbol{\theta}))$ is less than a pre-defined threshold, an edge is added between the poses i and j in the factor graph. Quantifying how different one distribution is from another provides a robust threshold for determining the existence of loop closures because the distribution represents the spread of the samples in the environment. Points sample the 3D world but there is no guarantee that exactly the same point is observed from consecutive observations. Because observations are composed of thousands of points, only distances between nearest points are considered to remain tractable.

(a) Aerial system flies in Rapps Cave (b) Aerial system

Fig. 2: (a) A custom-built aerial system collects data in an expansive cavern of Rapps Cave in WV, USA. (b) The aerial system is equipped with a VLP-16 laser scanner.

4 Results

The proposed approach is compared to the Surfel Mapping (SuMa) approach developed in [2], which is a highly optimized, parallelized implementation for GPU. The GMM formulation is run single-threaded so the timing performance is not one-for-one comparable. 420 LiDAR[1] observations are collected from a ground vehicle in a mine and 450 observations are collected from an aerial system (shown in Fig. 2b) in a cave.

All of the experiments are run on a low-power, embedded Gigabyte Brix with an Intel i7 8550U CPU, four cores (8 hyperthreads), and 32GB of RAM suitable for use on a SWaP-constrained robotic system. The Cauchy-Schwarz divergence loop closure threshold is set to $-\log(1\times 10^{-6})$. Both SLAM implementations employ the GTSAM framework [7]. 100-component GMMs are used in both experiments and the parameters are kept constant for both experiments.

SuMa is originally developed for the Velodyne HDL-64E[2] so the following parameters are updated to work with the Velodyne VLP-16. The data width and height are changed to 500 and 16, respectively. The model width and model height are changed to 500 and 16, respectively. The fields of view up and down are changed to 15 and -15. The map max. angle is changed to 30, sigma angle to 2, and sigma distance to 2. The loop closure distance is also changed to match the setting of the GMM approach.

Two measures are used to evaluate the SLAM results. The Root Mean Square Error (RMSE) as detailed in [25] and the odometric error. The RMSE is defined as the relative pose error at time step i:

$$\boldsymbol{E}_i := \left(\boldsymbol{Q}_i^{-1}\boldsymbol{Q}_{i+1}\right)^{-1}\left(\boldsymbol{S}_i^{-1}\boldsymbol{S}_{i+1}\right) \tag{11}$$

$$RMSE(\boldsymbol{E}_{1:n}) := \left(\frac{1}{n-1}\sum_{i=1}^{n-1}\|trans(\boldsymbol{E}_i)\|^2\right)^{1/2} \tag{12}$$

[1] https://velodynelidar.com/vlp-16.html

[2] https://velodynelidar.com/hdl-64e.html

where $trans(\boldsymbol{E}_i)$ refers to the translational components of the relative pose error $\boldsymbol{E}_i$, the estimated trajectory $\boldsymbol{S}_1,\ldots\boldsymbol{S}_n \in SE(3)$ and the ground truth trajectory $\boldsymbol{Q}_1,\ldots,\boldsymbol{Q}_n \in SE(3)$. The odometric error is computed as the translation and rotation error between frames 1 and j where $j \in [1,n]$:

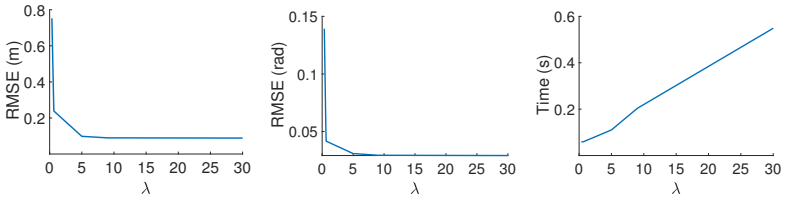
$$\boldsymbol{E}_j := \left(\boldsymbol{Q}_1^{-1}\boldsymbol{Q}_j\right)^{-1}\left(\boldsymbol{S}_1^{-1}\boldsymbol{S}_j\right) \tag{13}$$

$$OE(\boldsymbol{E}_{1:n}) := \|trans(\boldsymbol{E}_j)\| \tag{14}$$

For both relative pose and odometric errors, the rotation errors are similarly computed. Ground truth for the mine dataset is provided by Near Earth Autonomy[3]. GPS is unavailable in the cave so ground truth estimates are obtained using a map generated from a survey-grade, high accuracy FARO scanner[4].

EM Variant	**Mine Avg. Train. Time (s)**	**Cave Avg. Train. Time (s)**
Standard EM	1.888	2.365
Mahalanobis EM	0.848	1.054
Mahalanobis EM & KMeans++ Redux	0.2406	0.322

Table 1: Timing analysis for the Mine (also shown in the bar graph in Fig. 4f) and Cave (also shown in the bar graph in Fig. 5h) datasets. The Mahalanobis EM variant is approximately 2.25 times faster than the standard EM approach. With the K-Means++ reduction and point filtering, the runtime reduces approximately by a factor of 7.



(a) (b) (c)

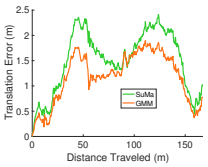
Fig. 3: The λ parameter from Eq. (6) is varied to determine a suitable value that balances accuracy of registration with time to compute the GMM. (a) and (b) demonstrate that for smaller values of λ the accuracy of the pose estimate decreases and (c) demonstrates the time to run EM without initialization increases as the value of λ increases. $\lambda = 5$ is chosen to achieve accurate pose estimation while remaining real-time viable.

A timing analysis for training the GMM is provided in Table 1. The table illustrates that the Mahalanobis variant of EM from Eq. (6) reduces the timing by a factor of approximately 2.25. The remaining time for the Mahalanobis EM is due to initialization. To decrease the time even further, points outside of a 15 m range are removed for both GMM initialization and training. This is not done for the SuMA approach because it negatively impacts the quality of the pose estimates. The time for GMM initialization is further reduced by using a downsampled set of points

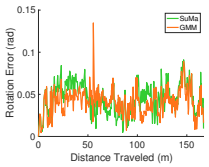
[3] https://www.nearearth.aero/

[4] https://www.faro.com/

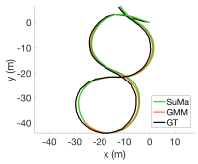
(every fifth point) for initialization and then assigning all remaining points to the closest cluster. Processing the data in this way leads to the training times labeled *Mahalanobis EM & KMeans++ Redux* in Table 1, which is approximately 7 times faster than the standard EM. This initialization and EM approach are used for training GMMs in the mine and cave experiments. Registration times are also provided in Figs. 4e and 5g. The frame-to-frame registration times for SuMa are reported for timing to compare fairly; however, the more accurate frame-to-model SLAM approach is used in all other plots and tables. For both evaluations, the GMM approach requires less data to transmit than the SuMa approach (Figs. 4g and 5i) due to the compactness of the GMM representation.



(a) Trans. Error



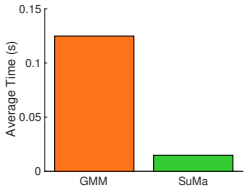
(b) Rot. Error



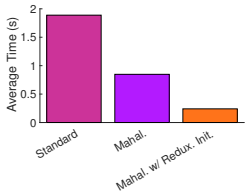
(c) Traj. (XY View)

	RMSE (m)
SuMa	9.88×10^{-2}
GMM	9.48×10^{-2}
	RMSE (rad)
SuMa	2.07×10^{-2}
GMM	2.11×10^{-2}

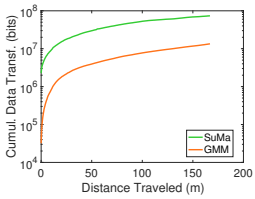
(d) RMSE



(e) Registration Times



(f) EM Variant Times



(g) Data Transferred

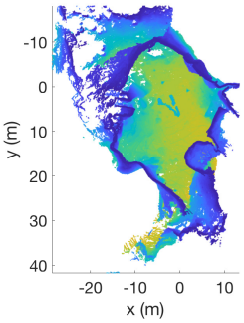
Fig. 4: Results for the Mine dataset. (a) and (b) illustrate the odometric errors as a function of distance traveled. (c) presents a view of the ground truth trajectory in black with each approach overlaid in different colors. (d) presents the RMSE values to evaluate the relative pose error between consecutive sensor observations. (e) presents the registration times and (f) illustrates the timing comparison from the second column of Table 1 as a bar chart. (g) illustrates the cumulative data transferred by each approach as a function of distance traveled.

An analysis is conducted to determine an adequate value for λ using the mine dataset (see Fig. 3). For the following experiments, $\lambda = 5$ as it yields accurate pose estimates while remaining real-time viable.

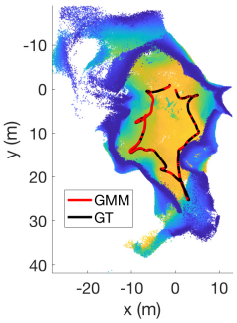
4.1 Mine

The experiment consists of 420 LiDAR observations in a mine environment taken from a ground vehicle. Figures 4a and 4b illustrate the error between the estimated pose as a function of distance traveled for the path taken by the vehicle shown in Fig. 4c. Figure 4d presents the RMS errors for consecutive pose estimates as a table. The translation RMSE values are slightly lower for the GMM approach and the rotation RMSE values are slightly lower for the SuMa approach. The odometric errors approximately follow the trends of the RMSE values.

The timing results in Figs. 4e and 4f demonstrate that the Mahalanobis EM with the KMeans++ reduction approach is able to significantly reduce the time to create a GMM. SuMa takes the least time but this approach is parallelized on the integrated GPU in the 8550U.



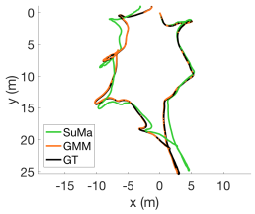
(a) Ground Truth Model



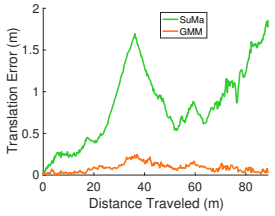
(b) GMM Reconstruction

	RMSE (m)
SuMa	3.78×10^{-2}
GMM	2.67×10^{-2}
	RMSE (rad)
SuMa	4.57×10^{-3}
GMM	5.05×10^{-3}

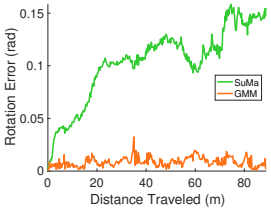
(c) RMSE



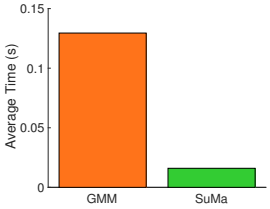
(d) Trajectory (XY View)



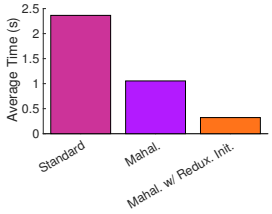
(e) Trans. error



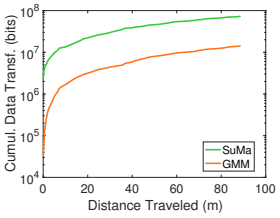
(f) Rot. error



(g) Registration Times



(h) EM Variant Times



(i) Data Transferred

Fig. 5: Results for the Cave dataset. (a) and (b) illustrate the ground truth and GMM reconstruction of the environment model. (c) presents the RMSE values for consecutive sensor observations for each approach after running SLAM. (d) presents a view of the ground truth trajectory in black with each approach overlaid in different colors. (e) and (f) illustrate the odometric error as a function of distance traveled. (g) and (h) present a timing comparison for the registration times and EM variants, respectively. (i) illustrates the cumulative data transferred by each approach as a function of distance traveled.

4.2 Cave

Figure 5a illustrates a cross section of the ground truth environment model created from LiDAR observations taken from the aerial system shown in Fig. 2. The path taken by the vehicle is shown in Fig. 5d. In this trial, the GMM approach is able to close the loop between the start and end points of the trajectory which leads to significantly lower overall error. One of the limitations of ICP is the cost function exhibits many local minima that are difficult to overcome when registering point-based sensor observations. Behley and Stachniss [2] attempt to overcome this limitation by trying multiple different initializations for the frame-to-model ICP, but this requires careful tuning of the parameters to ensure that enough variations in the translation and rotation are tested to successfully register the observations. The RMS errors in Fig. 5c demonstrate an overall translation error that is slightly lower for the GMM approach than SuMa. The odometric errors for the trajectory are shown in Figs. 5e and 5f.

Figures 5g and 5h illustrates the almost $7\times$ reduction in training time for the GMM as opposed to the standard EM method. SuMa runs on the integrated GPU in a highly optimized and parallelized way to achieve the reported runtimes. Figure 5b illustrates the GMM reconstruction of the environment by sampling points from the distribution.

5 Discussion and Future Work

While the results for the GMM approach presented in Section 4 cannot be run in real-time on a single thread, there is promise for a multi-threaded implementation. The distance calculations in K-Means++, responsibilities in EM, and the correspondence between pairs of mixture components in the GMM registration are all readily parallelizable. Offloading these calculations to a GPU is left as future work to enable real-time performance.

6 Conclusion

This paper demonstrated a real-time viable method for GMM SLAM for compute-constrained systems by formulating an EM algorithm that exploits sparsity to significantly reduce the time to learn a GMM. GMMs are trained from pointcloud data, used to estimate pose, and build a map of the environment. A method to close the loop is presented and the approach is evaluated with real-world data of challenging mine and unstructured cave environments.

Acknowledgments The authors would like to thank Aditya Dhawale, Alexander Spitzer, and Kumar Shaurya Shankar for providing feedback on drafts of this manuscript. The authors would also like to thank Curtis Boirum and Brian Osbun for assisting in data collection at Rapps Cave as well as Carroll Bassett of the West Virginia Cave Conservancy for granting access to Rapps Cave.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proc. Eighteenth Annu. ACM-SIAM Symp. Discrete Algorithms*, pages 1027–1035, Jan. 2007. doi: 10.1145/1283383.1283494.

[2] J. Behley and C. Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. In *Proc. Robot.: Sci. and Syst.*, June 2018. doi: 10.15607/RSS.2018.XIV.016.

[3] J. A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *Intl. Comput. Sci. Institute*, 4(510):126, 1998.

[4] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, New York, 2 edition, 2007.

[5] N. Boumal, B. Mishra, P. A. Absil, and R. Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *J. Mach. Learn. Research*, 15(1):1455–1459, 2014.

[6] R. R. Curtin. A dual-tree algorithm for fast k-means clustering with large k. In *Proc. SIAM Intl. Conf. on Data Mining*, pages 300–308, 2017. doi: 10.1137/1.9781611974973.34.

[7] F. Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

[8] B. Eckart, K. Kim, A. Troccoli, A. Kelly, and J. Kautz. Accelerated generative models for 3d point cloud data. In *IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pages 5497–5505, 2016. doi: 10.1109/CVPR.2016.593.

[9] B Eckart, K. Kim, and J. Kautz. Hgmr: Hierarchical gaussian mixtures for adaptive 3d registration. In *Proc. European Conf. on Comput. Vision (ECCV)*, pages 705–721, 2018. doi: 10.1007/978-3-030-01267-0_43.

[10] Benjamin Eckart. *Compact Generative Models of Point Cloud Data for 3D Perception*. PhD thesis, Pittsburgh, PA, Oct. 2017.

[11] C. Elkan. Using the triangle inequality to accelerate k-means. In *Proc. 20th Intl. Conf. on Mach. Learn. (ICML-03)*, pages 147–153, Aug. 2003.

[12] G. Dimitrios Evangelidis and R. Horaud. Joint alignment of multiple point sets with batch and incremental expectation-maximization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1397–1410, 2018. doi: 10.1109/TPAMI.2017.2717829.

[13] Greg Hamerly. Making k-means even faster. In *Proc. SIAM Intl. Conf. on Data Mining*, pages 130–140, 2010. doi: 10.1137/1.97816119728012.

[14] Reshad Hosseini and Suvrit Sra. An alternative to em for gaussian mixture models: Batch and stochastic riemannian optimization. *arXiv preprint arXiv:1706.03267*, 2017.

[15] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1633–1645, 2011. doi: 10.1109/TPAMI.2010.223.

[16] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *J. Intl. & Robot. Research*, 31(2):216–235, 2012. doi: 10.1177/0278364911430419.

[17] K. Kampa, E. Hasanbelliu, and J. C. Principe. Closed-form cauchy-schwarz pdf divergence for mixture of gaussians. In *Proc. IEEE Intl. Joint Conf. Neural Networks*, pages 2578–2585, 2011. doi: 0.1109/IJCNN.2011.6033555.

[18] S. Kolouri, G. K. Rohde, and H. Hoffmann. Sliced wasserstein distance for learning gaussian mixture models. In *IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pages 3427–3436, June 2018. doi: 10.1109/CVPR.2018.00361.

[19] Y. Liu and G. Nejat. Robotic urban search and rescue: A survey from the control perspective. *J. of Intell. & Robot. Syst.*, 72(2):147–165, 2013. doi: 10.1007/s10846-013-9822-x.

[20] G. O'Reilly, A. Jrad, R. Nagarajan, T. Brown, and S. Conrad. Critical infrastructure analysis of telecom for natural disasters. In *Netw. 2006. 12th Intl. Telecommunications Netw. Strategy and Planning Symp.*, pages 1–6, 2006. doi: 10.1109/NETWKS.2006.300396.

[21] C. OMeadhra, W. Tabib, and N. Michael. Variable resolution occupancy mapping using gaussian mixture models. *IEEE Robot. and Autom. Lett.*, 4(2):2015–2022, Apr. 2019. doi: 10.1109/LRA.2018.2889348.

[22] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Proc. Robot.: Sci. and Syst.*, June 2009. doi: 10.15607/RSS.2009.V.021.

[23] S. Srivastava and N. Michael. Efficient, multifidelity perceptual representations via hierarchical gaussian mixture models. *IEEE Trans. Robot.*, 35(1):248–260, 2019. doi: 10.1109/TRO.2018.2878363.

[24] T. Stoyanov, M. Magnusson, and A. J. Lilienthal. Point set registration through minimization of the l 2 distance between 3d-ndt models. In *Proc. IEEE Intl. Conf. on Robot. and Autom.*, pages 5196–5201, 2012. doi: 10.1109/ICRA.2012.6224717.

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots and Syst.*, Oct. 2012. doi: 10.1109/IROS.2012.6385773.

[26] W. Tabib, C. OMeadhra, and N. Michael. On-manifold gmm registration. *IEEE Robot. and Autom. Lett.*, 3(4): 3805–3812, 2018. doi: 10.1109/LRA.2018.2856279.

[27] W. Tabib, K. Goel, J. Yao, M. Dabhi, C. Boirum, and N. Michael. Real-time information-theoretic exploration with gaussian mixture model maps. In *Proc. Robot.: Sci. and Syst.*, FreiburgimBreisgau, Germany, June 2019. doi: 10.15607/RSS.2019.XV.061.