MR-GMMapping: Communication Efficient Multi-Robot Mapping System via Gaussian Mixture Model

Haolin Dong, Jincheng Yu, Yuanfan Xu, Zhilin Xu, Zhaoyang Shen, Jiahao Tang, Yuan Shen, and Yu Wang

***Abstract*—Collaborative perception in unknown environments is a critical task for multi-robot systems. Without external positioning, multi-robot mapping systems have relied on the transfer of place recognition (PR) descriptors or sensor data for the relative pose estimation (RelPose) and share their local maps for relative localization. Thus, in a communication limited environment, data transmission can become a significant communication bottleneck in the multi-robot mapping system. To address this limitation, we propose MR-GMMapping, a Multi-Robot GMM-based mapping system in which robots perform relative localization only with GMM submaps to reduce data transmission and storage. For mapping, we propose GMM submap construction strategy with an adaptive model selection method, which makes robots dynamically select the appropriate number of Gaussian components. For applications, we realize fully GMM-submap-based PR, RelPose, and local planner. Robots are able to perform relative localization without the aid of other forms of maps or information, which makes them favorable for environments with communication constraints. Experiments show that our GMM Submap extraction strategy improves 11% translation precision and 30% rotation precision in RelPose, compared to RelPose on point clouds and GMM frames. Our experiments also show the feasibility of the GMM-based local planner and a 98% data transmission reduction compared to point cloud maps. MR-GMMapping is published as an open-source ROS project at https://github.com/efc-robot/gmm_map_python.git.**

***Index Terms*—Multi-robot SLAM, mapping.**

I. Introduction

RAPID perception in an unknown environment is important for disaster search and rescue applications, where communication is constrained and external positioning methods fail. The multi-robot system can improve the efficiency of perception, but in the absence of external positioning, robots have to transfer sensor data and maps for relative pose estimation (RelPose), which may become the system bottleneck in communication-limited environments. For example, the data rate of the Mars-to-Earth communications is constrained to approximately 10 KB/s [1]. However, the transfer of 3D grid maps and place recognition (PR) descriptors consumes 2.6 MB/s [2], which precludes the transmission of perceptual models and human operations between Mars and Earth.

The Gaussian Mixture Model (GMM) is favourable for mapping in communication-constrained and storage-limited environments due to its small data volume and strong representation ability as a continuous probability model. A GMM map reduces the data volume by approximately 200× that of occupancy grid maps (from 1 MB/s to 5 KB/s) and meets the Earth-to-Mars communication bound [3]. When the robots know each other's relative pose, the GMM map is also proved to be communication-efficient for multi-robot systems to accomplish path planning and exploration [4].

To perform relative localization, robots share additional information besides GMM maps to determine whether they have observed the same place. Previous works accomplish PR and map merging by sharing sensor data [5] or PR descriptors [6] among robots. If robots can perform relative localization barely based on GMM maps, without transmitting the descriptor at all, data transmission will be further reduced by 25% [7].

The performance of GMM mapping depends on the model selection strategy. Previous works construct a hierarchical GMM to provide multi-scale analysis. They either require point clouds [8] or are designed for specific applications [9], which limits their potential applicability for map updating and merging in the multi-robot system. When it comes to multi-robot exploration, the scale and characteristics of the environment can change dramatically. For complex environments, an insufficient number of Gaussian components results in the reduction of mapping precision. While for simple environments, excessive Gaussian components result in unnecessary data transmission and a high use of computational resources. Thus, determining the parameters in advance is unable to accomplish the efficient perception of the environment.

To address the challenges above, we propose a communication efficient **M**ulti-**R**obot **GMM**-based **mapping** (MR-GMMapping) system with the following contributions:

- We realize a GMM-submap-based Decentralized Simultaneous Localization and Mapping (SLAM) system with three applications: PR, RelPose, and local planner. Robots

Manuscript received September 7, 2021; accepted January 10, 2022. Date of publication January 25, 2022; date of current version February 8, 2022. This letter was recommended for publication by Associate Editor P. Biber and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grants U19B2019 and M-0248, in part by the Tsinghua-Meituan Joint Institute for Digital Life, in part by the Tsinghua EE Independent Research Project, in part by the Beijing National Research Center for Information Science and Technology (BNRist), and in part by the Beijing Innovation Center for Future Chips. *(Corresponding author: Yu Wang.)*

The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing 100083, China (e-mail: donghl17@mails.tsinghua.edu.cn; yjc16@mails.tsinghua.edu.cn; xuyf20@mails.tsinghua.edu.cn; xuzl18@mails.tsinghua.edu.cn; shen-cy20@mails.tsinghua.edu.cn; pppplustjh@gmail.com; shenyuan_ee@tsinghua.edu.cn; yu-wang@tsinghua.edu.cn).

Digital Object Identifier 10.1109/LRA.2022.3145059

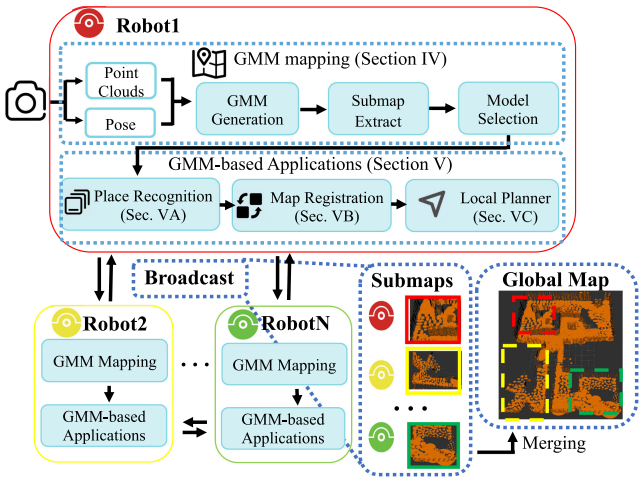


Fig. 1. Overview of MR-GMMapping framework. Different robots only transfer GMM submaps for PR, map registration, and local planner.

only transfer GMM submaps for relative localization, resulting in a 98% reduction in data transmission.

- We propose the GMM submap construction method, including the adaptive GMM model selection strategy. Our method improves 11% translation and 30% rotation precision in RelPose, making robots merge redundant Gaussian components dynamically.
- The open-source MR-GMMapping package, incorporating the two contributions above.

As illustrated in Fig. 1, the GMM mapping process consists of two steps (detailed in Section IV). Firstly, each robot first constructs GMM submaps using the subsequent point cloud frames. Secondly, the model selection module then chooses the GMM model with the least amount of data by merging similar Gaussian components. The GMM submaps are mainly used for three applications: 1) PR (detailed in Section V-A), 2) Map Registration (detailed in Section V-B), 3) Local Planner (detailed in Section V-C). In MR-GMMapping, robots only transfer GMM submaps used for global map construction after knowing their relative poses. The experimental results are presented in Section VI. Section VII gives the conclusion and future work.

II. Related Work

A. Multi-Robot Mapping System

SLAM is the most famous example of the multi-robot system. Previous multi-robot SLAM systems, either centralized [10], [11] or decentralized [4]–[6], [12], consist of two basic components: 1) PR, 2) RelPose. Different PR methods all calculate descriptors to determine whether robots have observed the same place, but use different data types. For example, NetVLAD [13] uses image sensor data, Bag of Words (BoW) [14] uses image feature points, and Segmatch [15] uses 3D laser sensor data. Similarly, different map registration methods with distinct data types are used to estimate and optimize relative positions. For example, Perspective-n-Point needs image feature points [5], and Iterative Closest Point (ICP) requires point clouds [10]. Therefore, currently proposed Decentralized-SLAM systems need to share the PR descriptors and the sensor data among robots; this incurs high transmission costs.

In communication-limited environments, it's not feasible to transfer PR descriptors or sensor data among robots. One possible solution is to let robots share their submaps [16] for PR [17] and RelPose [18]. Yu *et al.* [7] propose a submap-based multi-robot mapping system, where robots share 2D occupancy grid submaps and achieve high exploration efficiency. However, the discrete occupancy grid submaps for 3D mapping still suffer from the huge storage and communication consumption.

B. GMM Mapping in Robotic Applications

A GMM map is a continuous probabilistic representation that models the environment as a series of Gaussian distributions rather than a set of discrete grids. O'Meadhra *et al.* [19] propose a memory-efficient GMM mapping method to generate variable resolution occupancy grid maps from GMMs. Due to its outstanding representation ability and small data volume, the GMM map outperforms other map types in some particular environments such as tunnels and mines [20], which in turn promotes the study of GMM-based applications. On the perception side, Tabib *et al.* [20] present a robust distribution-to-distribution registration method to improve GMM-based mapping and navigation performance in subterranean environments. Huang *et al.* [21] present a cross-modality method for visual localization, which tracks the camera in a priorly generated GMM map. On the exploration side, Dhawale *et al.* [22] leverage the geometric properties of GMM maps to accomplish real-time collision avoidance and safe flight of UAVs given a time-parameterized trajectory. Tabib *et al.* [3] propose a GMM-based framework for real-time perception and exploration in large and cluttered 3D environments. As for the multi-robot system with communication constraints, Corah *et al.* [4] employ the GMM map to describe complex environment geometries while maintaining a small memory footprint which enables distributed operation with a low volume of communication. However, if there is no external positioning method such as GPS, additional information such as sensor data and scene descriptors needs to be shared, which makes their work infeasible for application in communication-limited environments.

C. GMM Model Selection

Choosing a suitable number of components to model a GMM is an open area of research. Some methods intend to determine the elbow-point where adding additional components does not significantly add information. Work in this area has explored information-theoretic criteria such as the Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) or application evaluations such as mapping accuracy [19] and registration precision [9]. Others methods construct Hierarchical-GMM (HGMM) to achieve better performance. Eckart *et al.* [23] develop a top-down strategy that produces a GMM by successively partitioning the point clouds into Gaussian component leaf nodes. Sriviasta *et al.* [8], [24] propose a bottom-up strategy that successively merges components until a prior knee-point is reached. Thus, we agree with the idea that the number of Gaussian components should not be decided in advance [4].

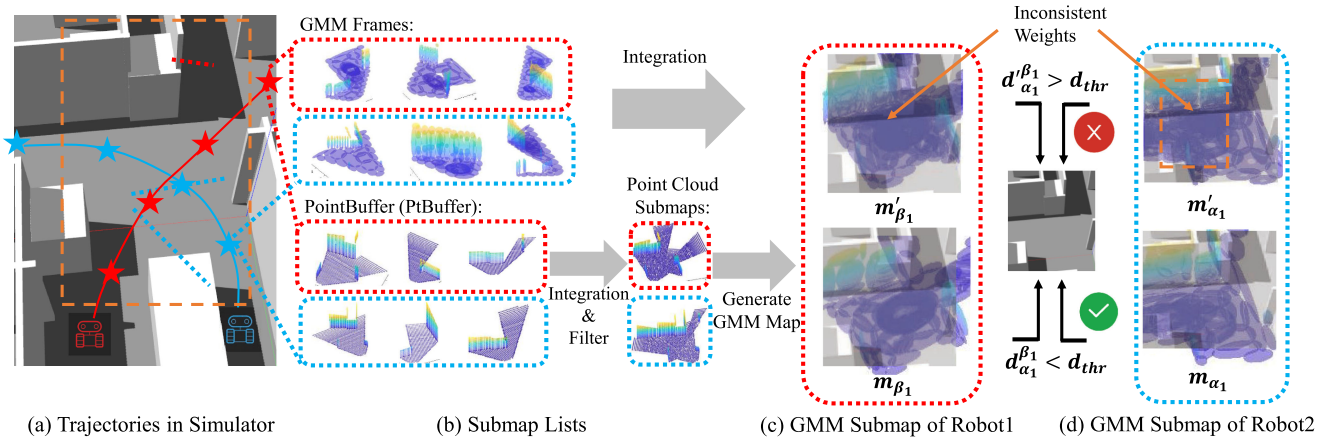


Fig. 2. Two robots observe the same area along different trajectories [Fig. 2(a)]. The previous GMM submap construction method generates GMM maps for each frame of point clouds and then integrates them (Fig. 2(b) top). In this way, the weight distribution of Gaussian components in the submap is affected by the observation trajectories. As a result, the GMM submaps of the same scene cannot be recognized correctly. Therefore, we temporarily store the point clouds in each submap, use voxel filtering to remove the duplicate information, and then generate the GMM submap one time (Fig. 2(b) bottom). By doing so, the GMM submaps of different trajectories are consistent in the Gaussian component weight, so the multi-robot RelPose can be correctly completed.

In communication-limited environments, we should adjust the model for different scenes on component level to reduce data transmission.

III. Mapping Via Gaussian Mixture Models

GMM is a probabilistic model that uses different weighted Gaussian distributions to represent certain probabilistic distributions. The probability density function of the GMM is defined as

$$p(x) = \sum_{k=1}^{N} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{1}$$

where each component is a Gaussian distribution $\mathcal{N}(x|\mu_k\Sigma_k)$ with a mean value μ_k and the co-variance matrix Σ_k. π_k is the probabilistic weights of each component which satisfies $\sum_{k=1}^{N} \pi_k = 1$. Through the EM algorithm [25], the parameter set $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{N}$ can be optimized to uniquely identify a GMM.

Suppose a robot obtains M frames of point clouds in total while perceiving the environment. Let $V = \{V_j\}_{j=1}^{M}$ be all of the point clouds and V_j be the j^{th} frame's data. Considering a map coordinate transformation, we extend the parameter set to

$$\Theta = \left\{ \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{N}, \{R_j, t_j\}_{j=1}^{M} \right\} \tag{2}$$

where $R_j \in \mathbb{R}^{3\times3}$is the rotation matrix and $t_j \in \mathbb{R}^3$ is the translation vector.

In multi-robot mapping systems, V expands by adding novel point clouds V_j into the set, Θ is updated by integrating novel GMM submaps. Robots then perform weight normalization to satisfy $\sum_{k=1}^{N} \pi_k = 1$.

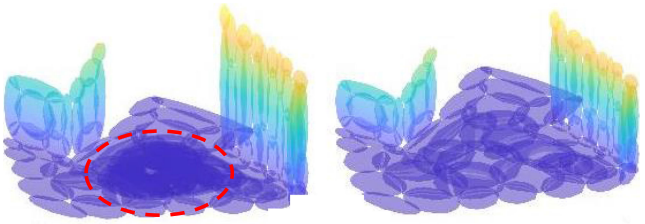
IV. Multi-Robot GMM Mapping

A. GMM Submap Construction

The construction of the GMM map directly impacts its performance in robotic applications. submap-based frameworks, such as SMMR-Explore [7], are favourable for multi-robot systems because robots only transfer submaps for collaborative exploration under communication-intensive scenarios. However, in multi-robot systems where robots use GMM submaps for relative localization, merging GMM frames directly to build submaps may cause the map registration to fail.

As illustrated in Fig. 2(a), two robots observe the same scene while moving along their respective trajectories. The previous submap extraction strategy [22] generates GMM frames for each sensor input and then integrates these GMM frames with weight normalizations [shown in Fig. 2(b)]. Although GMM submaps constructed by the method above describe the correct occupancy status of the scene, different observation paths lead to the inconsistency in Gaussian components' distribution and weights. The different weights of Gaussian components in submaps may make the GMM-based PR fail, and the different distribution of Gaussian components in submaps may make the GMM-based map registration fail [shown in the upper half of Fig. 2(c) and (d)]. In conclusion, two challenges of unsuitable GMM submaps are: 1) different weights of GMM components may make the PR descriptors of GMM submaps unable to match; 2) different distributions of GMM components may make the sum of squared L2 norm between all the Gaussian components in two GMM submaps difficult to converge. To solve this problem, we integrate the adjacent point clouds together instead of building GMM maps frame by frame. After one submap is determined, we conduct voxel filtering on the overall point clouds to remove duplicate information and then use EM algorithm to generate GMM submaps once [shown in the bottom half of Fig. 2(c) and (d)]. In this way, the weights of Gaussian components in the GMM submaps can correctly reflect the distribution of point clouds in the real environment, which improves the precision of scene matching and map registration at the same time (experimental results are in Section VI-B).

As for implementation, our GMM submap construction consists of three steps: Submap Construction, Submap Transmission, and Submap-based Trajectory Optimization. Each GMM submap goes through these steps sequentially, while these steps



(a) The original GMM map (b) After model selection

Fig. 3. Example of the original GMM map under global coordinate, colored by height [shown in Fig. 3(a)]. Fig. 3(b) shows the GMM map after model selection, where the overlapping components are merged into one. Transparency represents the number of Gaussian components. The fewer components, the more transparent the map will be.

are always active in three separate threads, processing successive map inputs.

Submap Consturction: each robot has a *PointBuffer (PtBuffer)* to store the point clouds of each submap temporarily. When a novel point cloud frame is received, the robot conducts voxel filtering in *PtBuffer* to remove duplicate points. We integrate 40 frames of point clouds for generating each GMM submap, which guarantees enough characteristics of the environment in each GMM submap for PR and the similarity of the GMM submaps describing the same scene for map registration. The robot generates GMM submap from the buffered points (introduced in Section IV-B) and then extracts scene descriptors locally for each GMM submap in preparation for the subsequent PR step.

Submap Transmission: when the robot receives the GMM submap published by other robots, it successively conducts PR and map registration steps (detailed in Section V).

Submap-based Trajectory Optimization: robots perform backend optimization using the results of RelPose from Submap Transmission. The optimized transformation (TF) is updated in the *TFGraph* which is the data structure we proposed to maintain TF trees for multiple robots. *TFGraph* can prevent the TF trees of different robots from forming a loop when the Robot Operating System (ROS) looks for the relative pose between two robots.

B. Adaptive Model Selection

The number of components in a GMM directly impacts the amount of data transmission among robots. As the point clouds distribute unevenly in the real world, directly generating GMM maps through the EM algorithm may result in the waste of its representation ability. For example, even with down-sampling, the number of points representing the floor is much higher than those representing the walls. Thus, without GMM model selection, overlapping Gaussian components can occur [Fig. 3(a)]. This results in redundant information transmission in the multi-robot system. In MR-GMMapping, robots conduct Adaptive Model Selection after one submap is determined inside Submap Construction thread to solve this problem.

If the number of Gaussian components is given in advance, submaps for small areas may have similar components in shape and position. From point clouds obtained by high-resolution RGBD camera in indoor scenarios, we observe that the point clouds are collected from the surface of obstacles, so most Gaussian components are flat. We define the direction of eigenvalues as the principal direction of each Gaussian component. Robots first classify Gaussian components representing planes in different directions according to their principal direction. Then, robots regard mean vectors as components' physical locations to divide each category into more subsets. Finally, robots use Kullback-Leibler (KL) divergence [26] to merge those components with high similarities in each subset. Due to the drastic changes in environmental features, robots may make some inevitable mismatchings. Thus, our GMM model selection strategy is a trade-off between data transmission and mapping accuracy, but it still can improve the utilization efficiency of GMM parameters.

Suppose N overlapping Gaussian components θ are similar in physical position (expressed by μ_n) and shape (expressed by Σ_n). n_k is the number of points represented by the k-th Gaussian component. We believe each Gaussian component can represent a large number of points, so each element represents the number of points $n_k \to \infty$. (3) shows the point set where x_i^k is the i-th point in the k-th Gaussian component.

$$\left\{x_i^k\right\}_{i=1}^{n_k\to\infty} \sim \mathcal{N}\left(\mu_k, \Sigma_k\right) \tag{3}$$

We adopt diagonal matrices Σ_k to reduce the delay of generating GMM maps [27], [28], while Σ_k may become non-diagonal after transformed to other robots' coordinate.

We assume the Gaussian component after merging as $\{\pi, \mu, \Sigma\}$ and the parameters can be calculated by (4)–(6), where $x_i^k \in \mathbb{R}^3, \mu_k \in \mathbb{R}^3, \Sigma_k = \mathrm{diag}\{\sigma_1, \sigma_2, \sigma_3\}$, and the square of a vector is defined by outerproduct.

$$\pi = \sum_{k=1}^{N} \pi_k \tag{4}$$

$$\mu = \lim_{\substack{n_k\to\infty \\ k=1\ldots N}} \frac{\sum_{k=1}^{N} n_k \mu_k}{\sum_{k=1}^{N} n_k} = \frac{\sum_{k=1}^{N} \pi_k \mu_k}{\sum_{k=1}^{N} \pi_k} \tag{5}$$

$$\begin{aligned}
\Sigma &= \lim_{\substack{n_k\to\infty \\ k=1\ldots N}} \frac{\sum_{k=1}^{N}\sum_{i=1}^{n_k} \left\|x_i^k - \mu\right\|^2}{\sum_{k=1}^{N} n_k - 2} = \lim_{\substack{n_k\to\infty \\ k=1\ldots N}} \frac{1}{\sum_{k=1}^{N} n_k - 2} \cdot \\
&\quad \left(\sum_{k=1}^{N}\left(\sum_{i=1}^{n_k}\left\|x_i^k - \mu_k\right\|^2 + n_k\left\|\mu_k - \mu\right\|^2 \right.\right. \\
&\quad \left.\left. + 2\sum_{i=1}^{n_k}\left(x_i^k - \mu_k\right)\left(\mu_k - \mu\right)\right)\right) \\
&= \lim_{\substack{n_k\to\infty \\ k=1\ldots N}} \sum_{k=1}^{N} \frac{\left(n_k - 1\right)\Sigma_k + n_k\left\|\mu_k - \mu\right\|^2}{\sum_{k=1}^{N} n_k - 2} \\
&= \sum_{k=1}^{N} \frac{\pi_k\left(\Sigma_k + \left\|\mu_k - \mu\right\|^2\right)}{\sum_{k=1}^{N} \pi_k}
\end{aligned} \tag{6}$$

As the weights of Gaussian components can express the relationship between the number of points described by different GMM maps, (5) and (6) can be simplified to the expression only containing GMM parameters rather than the point clouds information.

V. Applications With GMM Map

A. GMM Place Recognition

PR is the process that robots use to decide whether they observe the same scene. Previous work on multi-robot PR uses vision-based methods to extract scene descriptors, such as BoW [29] and NetVlad [13]. Other submap-based PR methods leverage the occupancy grid maps from different robots [7]. However, in communication-limited environments, it is not feasible to transfer sensor data or grid maps among robots. To make GMM-based PR possible, we design a GMM-based descriptor extractor $f(\cdot)$ to calculate the submap descriptor $f(m_{\alpha_i})$ from the GMM-submap list m_{α_i}.

As shown in Fig. 4, our $f(\cdot)$ consists of two components: 1) resampling 3D point clouds from m_{α_i} and 2) PointNetVLAD [30], which encodes 3D point clouds to a descriptor vector. We define $d_{\alpha_i}^{\beta_j}$ as the cosine distance [31] between vectors $f(m_{\alpha_i})$ and $f(m_{\beta_j})$. PointNetVLAD is trained to lower the $d_{\alpha_i}^{\beta_j}$ for the same scene m_{α_i} and m_{β_j}, while relatively increasing the $d_{\alpha_i}^{\beta_j}$ if m_{α_i} and m_{β_j} represent the different scene. In our MR-GMMapping, each robot computes scene descriptors locally instead of frequently sharing descriptors among robots. Once the robot receives a new submap, it selects the GMM submaps having distance below a given threshold d_{thr} for later RelPose; the others are filtered out.

B. GMM Map Registration

The map registration module estimates the relative transform of the candidate submap pairs provided by the PR module; it filters out the results with large errors. Previous work has proposed GMMreg [33], a robust distribution-to-distribution registration method to enable GMM mapping and navigation in subterranean environments. In GMMreg, an optimizer is trained to lower the errors between Gaussian components of two maps. To extend the idea of GMMreg, we make two improvements to make it favourable for the multi-robot system. First, we use map registration as a means to prevent PR mismatches as the registration results are vulnerable to incorrect place association. We regard the final loss of the GMMreg optimizer as the **Geometric Confidence** $C_{\alpha_i}^{\beta_j}$ of the relative pose $z_{\alpha_i}^{\beta_j}$. Then, the candidate pairs whose **Geometric Confidence** is larger than the threshold C_{thr} are filtered out. Second, in the multi-robot system, GMM submaps from different robots may have different numbers of Gaussian components. Thus, we adjust the loss function for GMM submaps with different component numbers.

C. GMM-Based Local Planner

The local planner is based on the gradient field of the GMM map. As noted in (1), $p(x_0, y_0)$ is the occupancy probability at a position (x_0, y_0), where the gradient is $\mathbf{g}(x_0, y_0) = \nabla p|_{x=x_0, y=y_0}$, which is a vector. The component of the vector $\mathbf{g}$ on a direction $\mathbf{r}$ is noted as $g(\mathbf{r})$. In the implementation, we select two nearby points on the direction $\mathbf{r}$ and calculate the gradient by $g(\mathbf{r}) = \lim_{\epsilon \to 0} \frac{f((x_0,y_0)+\epsilon\cdot\mathbf{r}) - f(x_0,y_0)}{\epsilon}$. Fig. 5(b) shows the gradient field of $\mathbf{g}$.

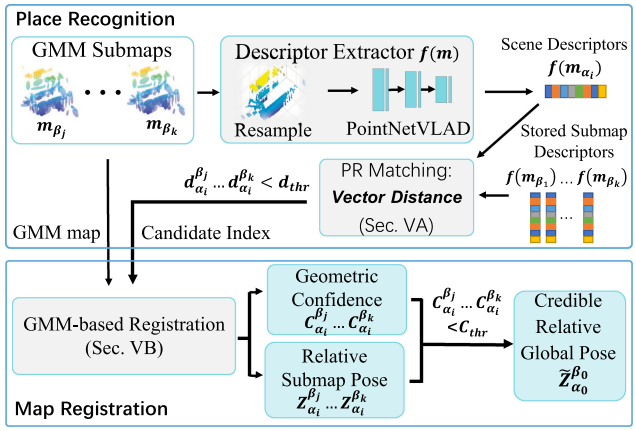
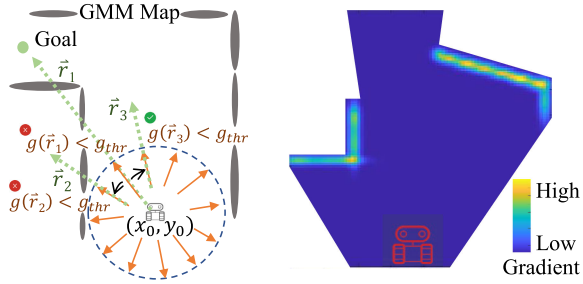


Fig. 4. Data flow of GMM-submap-based PR and map registration. When the robot receives a new GMM submap, it first resamples the GMM submap through Monte Carlo approximation [32], and then uses the PointNetVLAD [30] to extract scene descriptors. We use the cosine distance between vectors to filter out candidate GMM map pairs. Our GMM-based map registration regards the error of the transformed GMM maps as the geometric confidence to prevent scene mismatching, confirming the result of PR. The output of the map registration module is the credible relative global pose ($\tilde{Z}_{\alpha_0}^{\beta_0}$).



(a) GMM-based Local Planner (b) GMM Gradient Field

Fig. 5. GMM-based local planner on gradient field. The robot divides 360° into 12 discrete values and find a direction where the gradient is smaller than a threshold ($g(\mathbf{r}_3) < g_{thr}$).

Fig. 5(a) demonstrates the GMM-based local planner. We divide 360○ into 12 discrete directions and set the direction ($\mathbf{r}_1$) directly connecting the robot position and the goal as the candidate moving direction. If the gradient along $\mathbf{r}_1$ (noted as $g(\mathbf{r}_1)$) is smaller than a threshold g_{thr}, the robot will move in $\mathbf{r}_1$ direction. However, if $g(\mathbf{r}_1) > g_{thr}$, it indicates that obstacles will be encountered in this direction. We will then choose the peripheral direction as the candidate direction [$\mathbf{r}_2$ and $\mathbf{r}_3$ in Fig. 5(a)], and if $g(\mathbf{r}_2) < g_{thr}$ or $g(\mathbf{r}_3) < g_{thr}$, the robot will move in $\mathbf{r}_2$ or $\mathbf{r}_3$ direction. If it is still not satisfied, we will select the more peripheral direction as the candidate direction until the gradient in one direction is less than g_{thr}. In this method, we realize local planning using GMM maps instead of converting them to grid maps. It verifies the potential of using GMM directly for local navigation and expands the application scenario of GMM maps.

VI. Experiments

Previous work on GMM maps has reported extensive experimental results in representation ability [19]. We implement

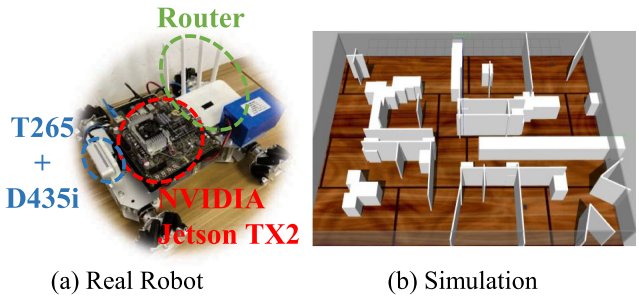


Fig. 6. Experiment setups.

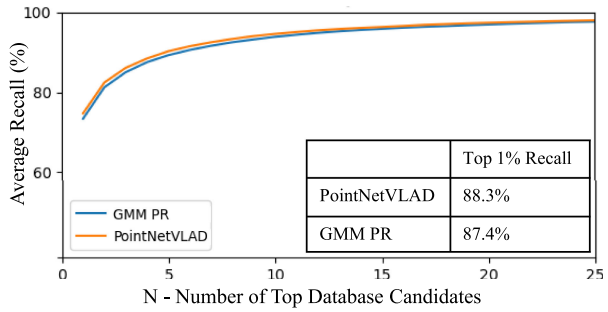


Fig. 7. Average recall of the PointNetVLAD and our GMM-based place recognition (GMM PR). The same model, trained on the Oxford RobotCar dataset [36], is used for drawing two average recall curves. The table at lower right also shows that the performance of the two methods is similar.

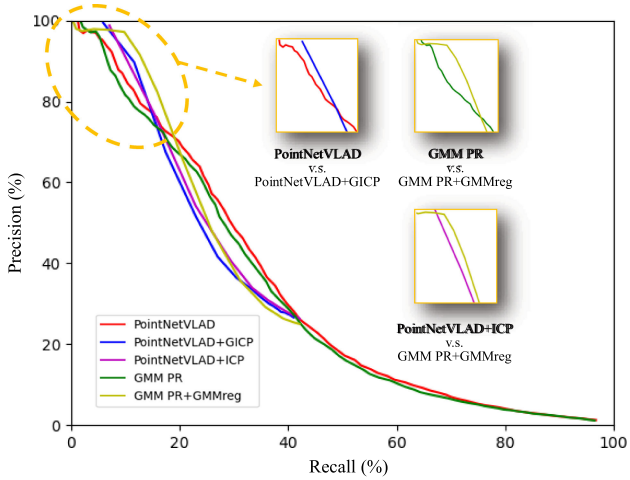


Fig. 8. Compared with the point-cloud-based PointNetVLAD with ICP, our GMM PR with GMMreg [33] has a higher recall in the range of high-precision, to find as many PR matched places as possible with less mismatching. The two boxes on the upper-right corner compare the map registration's improvement on PR between point-cloud-based and GMM-based methods. The box on the lower-right corner shows that our combination of GMM PR with GMMreg has a better performance than PointNetVLAD with ICP.

our system in both different scales Gazebo simulators [7] and real robots (shown in Fig. 6). Our robot is equipped with Intel RealSense T265 [34] which uses visual SLAM technology to accomplish the real-time acquisition of the robot pose, and depth camera D435i [35] which obtains point clouds from the environment. For computation, we use the NVIDIA Jetson TX2 with a 4-core ARM CPU and an embedded GPU. To guarantee the stable and real-time data transmission under low communication bandwidth, we design each robot with a separate router instead of using the TX2 network port. Our robots use the LAN to share GMM submaps. Our experimental results prove that our proposed GMM submap approach is able to preform PR (Section VI-A), map registration (Section VI-B), and local path planning (Section VI-C). On the system level, MR-GMMapping's data transmission is much smaller (Section VI-D).

A. Place Recognition

We compare our GMM-based PR with PointNetVLAD [30] on the Oxford RobotCar dataset [36]. Oxford RobotCar dataset has point clouds downsampled to a fixed number of points with zero coordinate means. We use the same model trained from PointNetVLAD for both GMM-based PR and point-cloud-based PR. Fig. 7 shows the recall curves of each model for the top 25 matches from each database pair in the dataset. Our method has similar performance with PointNetVLAD. As for the Top 1% recall, GMM-based PR can reach 87.4%, while point-cloud-based PR can reach 88.3%. In practice, the small difference is not anticipated to affect the performance of PR on real robots.

We also test the improvement of the precision-recall curve when robots use the confidence of map registration to further confirm the PR results (shown in Fig. 8). However, multi-robot SLAM systems are vulnerable to PR mismatching and consume additional computing and communication resources to handle the PR mismatching [5]. Thus, we are more concern about the higher recall under less mismatching (the recall on the high-precision range of precision-recall curve). We combine the PR with map registration and select the baseline as the point-cloud-based PointNetVLAD with ICP. We select the baseline corresponding to the 70% precision rate of scene recognition and compare the results between the GMM-based PR with GMMreg and the point-cloud-based PR with ICP. Two ICP methods are used to perform point cloud registration: primitive ICP and GICP [37], whose results only have slight differences. We utilize Point Cloud Library (PCL) [38] version 1.8 to calculate ICP and GICP. We use GMMreg as the registration step after GMM-based PR, and ICP as the registration step after point-cloud-based PR. Registration step can improve the recall of PR in the range of high precision for both GMM-based and point-cloud-based PR (shown in the upper two boxes in Fig. 8). As for the overall improvement of PR, our GMM-based method achieves higher recall compared to the point-cloud-based method at the same high precision (shown in the lower box in Fig. 8).

B. Map Registration

We adopt two different submap extraction strategies and conduct experiments on *freiburg3 long office household* RGBD sequence in the TUM dataset [39]. Two submap construction methods are **A.** splicing adjacent GMM map frames (GMM+GMM submap), and **B.** storing point clouds and generating the GMM submap one time (Point-to-GMM submap in proposed method). We use GMMreg [33] for both types of submaps and GICP for the point-cloud-based baseline to perform map registration. We preprocess the point clouds by integrating a sequence of 5 frames, so the full 2509 frames of the sequence are divided into approximately 500 GMM submaps. 5 frames for one submap

TABLE I
MAP REGISTRATION RESULTS ON THE TUM DATASET

	t_{error} (m)	t_{RMSE} (m)	r_{error} $(^\circ)$	r_{RMSE} $(^\circ)$
Point Cloud (GICP)	0.0155	0.0168	0.489	0.532
No Submap	0.0158	0.0171	0.485	0.553
GMM + GMM Submap	0.0152	0.0175	0.524	0.752
Point-to-GMM Submap	**0.0140**	**0.0156**	**0.339**	**0.401**

TABLE II
NAVIGATION TRAJECTORY LENGTH IN THE SIMULATOR

	Move Base (m)	GMM-based Local Planner (m)
Regular Corner	5.1	5.0
Straight Passageway	4.0	4.1
Clustered Environment	4.8	5.0
Front Obstacles	3.8	3.2

is feasible because it makes submaps able to describe the same scene while sharing a certain degree of overlapping for map registration evaluation. Besides, our experiments show that submaps with more or fewer frames do not provide higher registration accuracy.

Table I shows the results of average pose error and Root Mean Squared Error (RMSE) of the trajectory. We use the evaluation toolkit provided by TUM dataset [39] to calculate these results. The results prove that map registration with Point-to-GMM submaps outperforms GMM+GMM submaps. We believe this is because the overall probability description of the environment is destroyed after the GMM frame integration and the weight normalizations. This affects the accuracy, especially for those submaps with large angles of rotation. Although the frame-by-frame map registration's (no submap) accuracy only decreases slightly compared to the Point Cloud-to-GMM submap registration, it results in significant data transmission, which is infeasible for multi-robot systems with communication constraints.

C. GMM-Based Local Planner

We compare the trajectory length of our GMM-based local planner and ROS Move Base [40] (shown in Table II) in the four different simulation scenarios: regular corner (Scene 1), straight passageway (Scene 2), clustered environment (Scene 3), and front obstacles (Scene 4). Our GMM-based local planner and the Move Base plan paths with similar lengths. In scenes where the robot has to turn at a large angle, such as Scene 1 and Scene 4, our method's trajectory length is less than Move Base's. We believe this is because our GMM-based local planner prefers to take the shortest path, while Move Base guarantees robots' safety using Costmaps. The GMM-based local planner is sensitive to the obstacles in the moving direction. It makes the robot stay away from the obstacles by conducting real-time path replanning. However, when the robot finds no obstacles in its moving direction, it directly moves toward the local goal along the shortest path without considering the distance to the obstacles.

In this work, we realize local planning using GMM maps instead of converting them to grid maps, and verify it in some simple scenarios. The experiments show the potential of using GMM directly for local navigation and expand the application scenario of GMM maps.

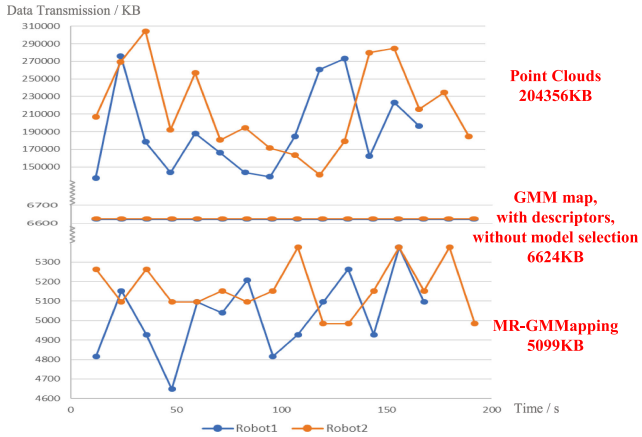


Fig. 9. Data transmission between two robots. We construct a submap using 20 adjacent point cloud frames and initialize each submap with 100 components. In the simulation environment, two robots construct 13 and 16 submaps, respectively. The average data transmission of submap data is shown in the red caption on the right. For each frame of the point cloud map, the 0.1 * 0.1 * 0.1 voxel downsampling is used for sensor data. If the place descriptors are transferred and the model selection method is not applicable, the map data volume per frame is constant. After using the model selection method, the amount of map data are reduced by approximately 10%, and the fluctuation of the amount of data is similar to that of point cloud maps.

D. Data Transmission in Multi-Robot System

Fig. 9 shows the data transmission of submaps for two robots in our simulation environment. After downsampling and novel information checking, the rate of adding novel point clouds into the submap is approximately 0.6 seconds per frame. We determine a submap for GMM generation using 20 frames of point clouds to ensure each submap contains enough environment information for place recognition. In 200 seconds' exploration, the two robots construct 13 and 16 frames of submaps, respectively. If robots transfer point clouds, the average submap data volume is 204356 KB. If robots transfer GMM submaps with 100 Gaussian components and a 1024 KB scene descriptor for place recognition, the average submap data volume is 6624 KB. While in MR-GMMapping, robots transfer GMM submaps without descriptors after the model selection, which reduces the average data transmission to 5099 KB. In addition, we can see that the trend of data volume between point clouds and GMM submap in MR-GMMapping is similar, proving that our model selection method can effectively reduce redundant information according to the scale of the environment.

In summary, compared with the point clouds map, MR-GMMapping reduces the data transmission by approximately 98%. In comparison to the GMM map with descriptors, MR-GMMapping reduces the data transmission by approximately 23%. On the one hand, we eliminate descriptor transmission among robots, contributing to 15% (1024 KB) reduction in data transmission. On the other hand, our model selection method can improve the expression ability of GMM elements, reducing data transmission by approximately 8% (∼500 KB). We select the optimal number of components for the environment by merging redundant components. These two technologies can be used for

multi-robot GMM SLAM in different environments to reduce data transmission.

VII. Conclusion and Future Work

This letter proposes a communication efficient Multi-robot GMM Mapping System with three submap-based robotic applications: place recognition, map registration, and local planner. Robots can accomplish all three robotic applications after the submap extraction. By using our component-level model selection method, robots can eliminate redundant Gaussian components within and between submaps. Robots in MR-GMMapping only transfer GMM submaps, which not only reduces the total amount of data transmissions by 98% but improves the accuracy of RelPose by 11% in average translation error and 30% in average rotation error.

Since collaborative exploration is an essential task for multi-robot systems, we plan to propose novel multi-robot exploration methods in the future that can be integrated into the GMMapping approach. In addition, we also plan to expand the MR-GMMapping to Unmanned Aerial Vehicle (UAV) applications that utilize 3D maps.

References

[1] Mars science laboratory data rates/returns, 2022. [Online]. Available: https://mars.nasa.gov/msl/mission/communications/

[2] A. Souza, R. S. Maia, R. V. Aroca, and L. M. G. Gonçalves, "Probabilistic robotic grid mapping based on occupancy and elevation information," in *Proc. 16th Int. Conf. Adv. Robot.*, 2013, pp. 1–6.

[3] 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. Syst.*, 2019, pp. 1–9.

[4] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1715–1721, Apr. 2019.

[5] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "DOOR-SLAM: Distributed, online, and outlier resilient SLAM for robotic teams," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1656–1663, Apr. 2020.

[6] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2466–2473.

[7] J. Yu *et al.*, "SMMR-explore: SubMap-based multi-robot exploration system with multi-robot multi-target potential field exploration method," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 1–6.

[8] S. Srivastava and N. Michael, "Efficient, multifidelity perceptual representations via hierarchical Gaussian mixture models," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 248–260, Feb. 2019.

[9] B. Eckart, K. Kim, and J. Kautz, "Fast and accurate point cloud registration using trees of Gaussian mixtures," 2018, *arXiv:1807.02587*.

[10] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot SLAM system for 3D LiDARs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1004–1011.

[11] K. Ebadi *et al.*, "Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 80–86.

[12] P. Schmuck and M. Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *J. Field Robot.*, vol. 36, no. 4, pp. 763–781, 2019.

[13] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.

[14] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[15] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5266–5272.

[16] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1271–1278.

[17] E. Liu and K. Tanaka, "Discriminative map retrieval using view-dependent map descriptor," 2015, *arXiv:1509.07615*.

[18] J. Hörner, "Map-merging for multi-robot system," Charles Univ. Prague, Faculty of Mathematics and Physics, pp. 6–13, 2016. [Online]. Available: https://is.cuni.cz/webapps/zzp/detail/174125/

[19] C. O'Meadhra, W. Tabib, and N. Michael, "Variable resolution occupancy mapping using Gaussian mixture models," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 2015–2022, Apr. 2019.

[20] W. Tabib, C. O'Meadhra, and N. Michael, "On-manifold GMM registration," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3805–3812, Oct. 2018.

[21] H. Huang, H. Ye, Y. Sun, and M. Liu, "GMMLoc: Structure consistent visual localization with Gaussian mixture models," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5043–5050, Oct. 2020.

[22] A. Dhawale, X. Yang, and N. Michael, "Reactive collision avoidance using real-time local Gaussian mixture model maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3545–3550.

[23] B. Eckart, K. Kim, A. Troccoli, A. Kelly, and J. Kautz, "Accelerated generative models for 3D point cloud data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5497–5505.

[24] S. Srivastava and N. Michael, "Approximate continuous belief distributions for precise autonomous inspection," in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, 2016, pp. 74–80.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via theEMAlgorithm," *J. Roy. Stat. Soc.: Ser. B. (Methodological)*, vol. 39, no. 1, pp. 1–22, Sep. 1977. [Online]. Available: https://doi.org/10.1111%2Fj.2517-6161.1977.tb01600.x

[26] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[27] Y. Xu *et al.*, "Game: Gaussian mixture model mapping and navigation engine on embedded FPGA," in *Proc. IEEE 29th Annu. Int. Symp. Field-Programmable Custom Comput. Mach.*, 2021, pp. 60–68.

[28] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. IEEE 17th Int. Conf. Pattern Recognit.*, 2004, vol. 2, pp. 28–31.

[29] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[30] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4470–4479.

[31] "Cosine similarity," 2022. [Online]. Available: https://en.wikipedia.org/wiki/Cosine_similarity

[32] R. M. Karp, M. Luby, and N. Madras, "Monte-Carlo approximation algorithms for enumeration problems," *J. Algorithms*, vol. 10, no. 3, pp. 429–448, 1989.

[33] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.

[34] Intel RealSense tracking camera T265, 2022. [Online]. Available: https://www.intelrealsense.com/tracking-camera-t265

[35] Intel RealSense D435, 2022. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435

[36] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.

[37] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robot.: Sci. Syst.*, vol. 2, no. 4, pp. 1–8, 2009.

[38] Point cloud library, 2022. [Online]. Available: https://pointclouds.org/

[39] J. Sturm, W. Burgard, and D. Cremers, "Evaluating egomotion and structure-from-motion approaches using the tum RGB-D benchmark," in *Proc. Workshop Color-Depth Camera Fusion Robot. IEEE/RJS Int. Conf. Intell. Robot Syst.*, 2012, pp. 1–6.

[40] Move base local planner, 2022. [Online]. Available: http://wiki.ros.org/base_local_planner