MR-GMMExplore: Multi-Robot Exploration System in Unknown Environments based on Gaussian Mixture Model

Yichun Wu[1*], Qiuyi Gu[1*], Jincheng Yu[1], Guangjun Ge[1], Jian Wang[1], Qingmin Liao[2], Chun Zhang[3] and Yu Wang[1]

***Abstract*— Collaborative exploration in an unknown environment is an essential task for mobile robotic systems. Without external positioning, multi-robot mapping methods have relied on the transfer of place descriptors and sensor data for relative pose estimation, which is not feasible in communication-limited environments. In addition, existing frontier-based exploration strategies are mostly designed for occupancy grid maps, thus failing to use surface information of obstacles in complex three-dimensional scenes. To address these limitations, we utilize Gaussian Mixture Model (GMM) as the map form for both mapping and exploration. We extend our previous mapping work to exploration setting by introducing MR-GMMExplore, a Multi-Robot GMM-based Exploration system in which robots transfer GMM submaps to reduce data transmission and perform exploration directly using the generated GMM map. Specifically, we propose a GMM spatial information extraction strategy that efficiently extracts obstacle probability information from GMM submaps. Then we present a goal selection method that allows robots to explore different areas, and a GMM-based local planner that realizes local planning using GMM maps instead of converting them into grid maps. Simulation results show that the transmission of GMM submaps reduces approximately 96% communication load compared with point clouds and our mean-based extraction strategy is 4 times faster than the traversal-based one. We also conduct comparative experiments to demonstrate the effectiveness of our approach in reducing backtracking paths and enhancing cooperation. MR-GMMExplore is published as an open-source ROS package at https://github.com/efc-robot/gmm_explore.**

I. Introduction

Autonomous exploration is critical in unknown environments such as disaster search and rescue (SaR) and the subterranean environment [1], where the communication is limited and external positioning is not accessible. Compared with single-robot system, multi-robot system allows robots to cooperatively map the 3D environment, thus improving exploration efficiency. However, in the absence of external positioning, robots have to transfer place descriptors and sensor data for relative pose estimation (RelPose) [2] [3], which requires high communication volume. The occupancy grid maps shared among robots for map merging also put pressure on the communication bandwidth [4]. In addition, complex environments bring challenges to the multi-robot exploration task as the existing exploration strategies often rely on the discrete grid map that is unable to precisely describe the geometry information of obstacles [5].

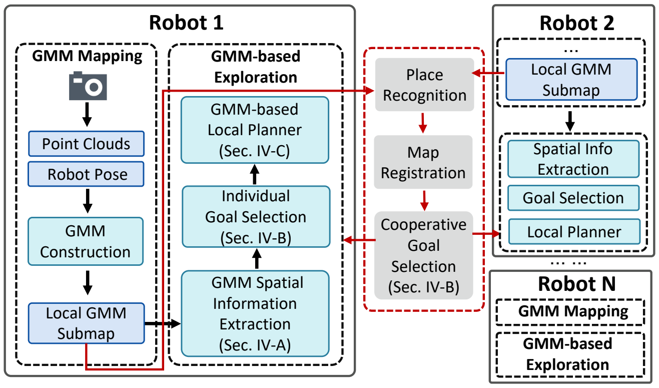


Fig. 1. GMM-based Multi-Robot Exploration System Architecture.

To address the above challenges, we use Gaussian mixture model (GMM) as the map form for both mapping and exploration. The GMM map is suitable for scenarios with communication constraint due to its small data volume, reducing transmission data volume by approximately 200× that of occupancy grid maps (from 1 MB/s to 5 KB/s) [6]. Meanwhile, its continuous probability representation is capable of capturing the occupied surface and free space information in the 3D environment [5] and improves the efficiency for real-time collision avoidance [7].

As illustrated in Fig. 1, our multi-robot exploration system is composed of two parts: GMM mapping and GMM-based exploration. **For GMM mapping**, the robots construct GMM submaps from point clouds, in which an adaptive model selection method [8] is applied to combine overlapped Gaussian components. Then to further reduce the data transmission, we adopt the submap-based framework [9] where only GMM submaps are transmitted among robots. The place recognition module extracts descriptors from GMM submaps to decide whether different robots experience the same place, and the inter-robot relative pose is estimated in map registration module for incremental map merging.

For GMM-based exploration, upon the generation of a

[1] Department of Electronic Engineering, Tsinghua University, Beijing, China. {wuyc21, gqy19}@mails.tsinghua.edu.cn {yu-jc, jian-wang, yu-wang}@tsinghua.edu.cn
[2] Shenzhen International Graduate School, Tsinghua University, Beijing, China. {liaoqm}@tsinghua.edu.cn
[3] School of Integrated Circuits, Tsinghua University, Beijing, China. {zhangchun}@tsinghua.edu.cn

[*] These authors contributed equally.

This research was supported by National Key Research and Development Program of China (No. 2019YFF0301500), National Natural Science Foundation of China (No. U19B2019, M-0248), Tsinghua-Meituan Joint Institute for Digital Life, Tsinghua EE Independent Research Project, Beijing National Research Center for Information Science and Technology (BNRist), and Beijing Innovation Center for Future Chips.

GMM submap, robots extract the mean values of Gaussian components to update the set of areas that have been explored. Then, taking into account the height of the robot, we sample the occupancy probabilities at different heights for each newly explored area. In the goal selection module, the robot chooses candidate target goals from the frontier of the explored area with a maximum obstacle probability constraint. At the beginning, robots haven't observed the same scene. Therefore, they select goals individually and then plan paths based on the gradient of occupancy probability. When robots successfully realize map merging, a cooperative exploration strategy is performed to choose different goals and avoid inter-robot collision during local path planning. The contributions are as follows:

- A GMM spatial information extraction strategy that efficiently extract exploration statement information and obstacle probability information from GMM components of each generated submaps.
- A goal selection method including individual goal selection and cooperative goal selection.
- A GMM-based local planner which utilizes the gradient of obstacle probability to plan paths.

The structure of the paper is as follows. We first present current works on GMM mapping in robotic applications and multi-robot exploration (Section II). Then we demonstrate the representation of GMM map and the GMM submap construction method (Section III). The GMM-based exploration is composed of three modules mentioned before: a. GMM Spatial Information Extraction (Section IV-A), b. Goal Selection (Section IV-B) and c. GMM-based Local Planner (Section IV-C). Afterwards, we conduct simulation experiments to evaluate the performance of our GMM-based exploration system (Section V). Finally, we conclude this work (Section VI).

II. Related work

A. GMM Mapping in Robotic Applications

A GMM map is a continuous probabilistic representation that models the environment as a finite set of Gaussian distributions rather than discrete cells. Meadhra *et al.* [5] propose a memory-efficient GMM mapping method for deriving occupancy grid maps at arbitrary resolutions without the need to store a grid map of the entire environment. To meet the requirement of real-time GMM mapping, Xu *et al.* [4] present an acceleration engine called GAME on embedded FPGA. Various GMM model selection methods [8] [10] are proposed to choose a suitable number of GMM components. In terms of perception, Tabib *et al.* [11] present a real-time distribution-to-distribution GMM registration methodology to enable robust mapping and navigation in subterranean environments. GMM is also employed in Huang *et al.* [12] as a prior map for visual localization of a moving camera.

In terms of exploration, Dhawale *et al.* [7] leverage geometric properties of GMM maps to efficiently avoid collision given an arbitrary time-parameterized trajectory. Tabib *et al.* [6] develop a single-robot GMM-based framework for perception and exploration in communication-constrained environments. Corah *et al.* [13] and Goel *et al.* [14] use GMM for multi-robot mapping and local occupancy grid maps are generated from GMM for use in planning. However, if external positioning methods fail, additional information such as sensor data needs to be shared in Corah *et al.* [13], which is infeasible in communication-limited areas. Additionally, Goel *et al.* [14] makes the assumption that relative initial transforms between robots are known in advance.

B. Multi-Robot Exploration in Unknown Environments

Frontier-based exploration strategies [15] are widely used for multi-robot exploration. Robots separate explored and unexplored areas based on the current map, and then each robot selects a goal from detected frontiers according to different criteria, e.g., cost-based [16] [17], sample-based [18] [19], and potential field-based [9]. Robots always move towards the nearest frontier [20] in cost-based methods while sample-based methods detect frontiers using multiple Rapidly-exploring Random Trees (RRTs) [18]. Yu *et al.* [9] design a potential field-based method to reduce back-and-forth changes of goals for 2D exploration.

However, since frontier can be easily determined by grid maps, these exploration methods are usually combined with the occupancy grid, integrating new sensor measurements straightforwardly [17]. When the 3D environment is described by Gaussian mixture model, existing strategies cannot directly make decisions based on detailed surface information of objects. Even in GMM-based exploration frameworks [13] [14], local occupancy grid maps are sampled from GMM maps using Monte Carlo ray tracing for planning, while it has been proven that using GMM directly for local navigation is time-efficient and safety-guaranteed [8].

III. The GMM Map Representation

A. Gaussian Mixture Model

GMM is a probabilistic model using a set of weighted Gaussian distributions. Its probability density function is defined as

$$p(\boldsymbol{x})=\sum_{i=1}^{N} \pi_i \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \tag{1}$$

where each component $\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)$ is a Gaussian distribution parameterized by a mean value $\boldsymbol{\mu}_i$ and a covariance matrix $\boldsymbol{\Sigma}_i$. π_i represents the probabilistic weight of selecting the ith component which satisfies$\sum_{i=1}^{N} \pi_i = 1$. The GMM can be estimated by Expectation Maximization (EM) algorithm [21] to achieve optimal parameters.

B. GMM Map

A GMM map $\mathcal{G} = p(\boldsymbol{x})$ provides the probability of spatial occupancy at each location $\boldsymbol{x}$ [10]. In our GMM-based exploration system, robots represent the explored environment in the form of GMM and share GMM submaps for map merging and collaborative exploration. The GMM submap construction method is realized in our previous work

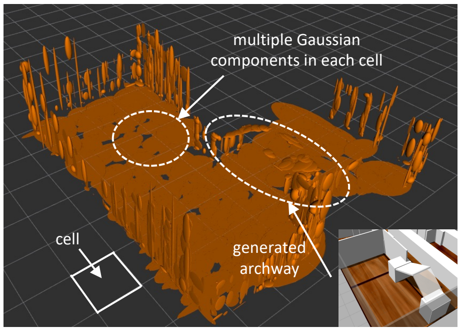


Fig. 2. Constructed GMM submap.

[8]. Briefly speaking, for generating each GMM submap, we first integrate 20 frames of adjacent point clouds and remove duplicate points through voxel filtering. Then we construct the GMM submap via the EM algorithm. However, due to the uneven distribution of point clouds in the real environment, the generated GMM submap has overlapped Gaussian components. Therefore, we apply an adaptive model selection method on the GMM submap to combine components with similar positions and shapes.

IV. Exploration Based On GMM Map

The GMM map captures the surface information of obstacles, which is helpful for robot exploration. Using the real-time GMM submaps as the robots move, each robot constantly chooses its next goal and plans paths until the whole environment is explored. The process of GMM-based multi-robot exploration is shown in Fig. 3.

A. GMM Spatial Information Extraction

The GMM map models the 3D environment observed by robots as continuous probability, so we extract the spatial information that is necessary for the exploration task. Specifically, we define an exploration statement matrix $\mathcal{M} = \{\mathcal{M}_{xy} | 0 \leq x < W, 0 \leq y < L\}$ to distinguish explored and unexplored areas. $\mathcal{M}_{xy} = 1$ means the area $[x*l, (x+1)*l] \times [y*l, (y+1)*l]$ (called as one cell in Fig. 2) has been explored, while $\mathcal{M}_{xy} = 0$ means it has not. The size length of cell l can be adjusted to meet the required accuracy for exploration. $\mathcal{M}$ is updated every time the robot generates a new GMM submap or receives a submap from other robots. According to Eq. (1), we describe the GMM submap as a parameter set $\Theta_{gmm} = \{\{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{N}\}$. We do not traverse each cell and calculate the occupancy probability at every height to determine $\mathcal{M}$. Such traversal-based strategy is time-consuming and inaccurate as unexplored areas also have small probability values. Instead, we present a mean-based extraction strategy. For each component in Θ_{gmm}, the exploration statement matrix updates as follows:

$$\mathcal{M}_{xy} = 1 \tag{2}$$

where $x = \lfloor \frac{\mu_{ix}}{l} \rfloor$, $y = \lfloor \frac{\mu_{iy}}{l} \rfloor$ and $\boldsymbol{\mu}_i = [\mu_{ix}, \mu_{iy}, \mu_{iz}]$.

It is sufficient to use only the mean value information of every component because: 1) When generating a GMM submap, we set a lower limit on the number of components so that the mean value of at least one component falls into the explored cell, illustrated in Fig. 2. 2) Even if there are oversights, such as near the corner and other view blind areas, our goal selection method in Section IV-B is fault-tolerant and will not select these mislabeled points as goals. 3) As the camera's field of view changes during further exploration, it is possible that the mislabeled points will eventually be successfully marked as explored. Section V-B compares the computation efficiency of two strategies.

In addition, we define an obstacle probability matrix $\mathcal{P}$ that has the same size as $\mathcal{M}$. $\mathcal{P}_{xy}$ refers to the probability for a certain vehicle to encounter obstacle in the cell. Given a GMM submap, the corresponding $\mathcal{P}$ varies depending on the height of the robot. For example, suppose there is a box on the group, the obstacle probability in this area for a ground vehicle is high. However, a drone may has a low obstacle probability since it may fly directly over the box. In this paper, we consider the unmanned ground vehicle (UGV) with height h. It can travel on flat ground and pass through the archway shown in Fig. 2, so we need to know if there are obstacles in the altitude range from 0 to h at each location. The obstacle probability matrix is obtained through the following formula:

$$\mathcal{P}_{xy} = \underset{j=1,\dots,n}{max} \sum_{i=1}^{N} \pi_i \mathcal{N}\left((x, y, jz_0) \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \tag{3}$$

where $z_0 = \frac{h}{n}$ is the sampling interval in height direction and n is the number of samples. Each time a new GMM submap is added, we update $\mathcal{M}$ and then calculate the obstacle probability of areas that have just been marked as explored.

B. Goal Selection

Robots continuously make decisions on which area to explore and plan paths based on the acquired information about the surrounding environment. We call the former step as goal selection, where the robot selects a goal area from the boundary between explored and unexplored areas. The selection of goal needs to meet the three requirements: 1) The goal has been explored and is a free space. 2) It is next to unexplored areas. 3) For safety consideration, the area around the goal is as free as possible of obstructions. We choose a goal from the following candidate set $\mathcal{L}$:

$$\begin{aligned} \mathcal{L} = \{(x,y) | \mathcal{M}_{xy} = 1, \mathcal{P}_{xy} < 0.01, \\ \exists\ (a,b) \in U_{xy},\ s.t.\ \mathcal{M}_{ab} = 0, \\ \forall\ (a,b) \in U_{xy},\ \mathcal{M}_{ab}\mathcal{P}_{ab} < 0.1\} \end{aligned} \tag{4}$$

where U_{xy} is the set of eight regions adjacent to the goal. Each point (x, y) in $\mathcal{L}$ refers to a corresponding cell. As the robot moves, it first determines whether the points in the candidate set still satisfy the conditions of goal selection, and removes them if they do not. Then the robot simply judges whether the latest cells marked as explored are on the boundary and adds points that meet the requirements.

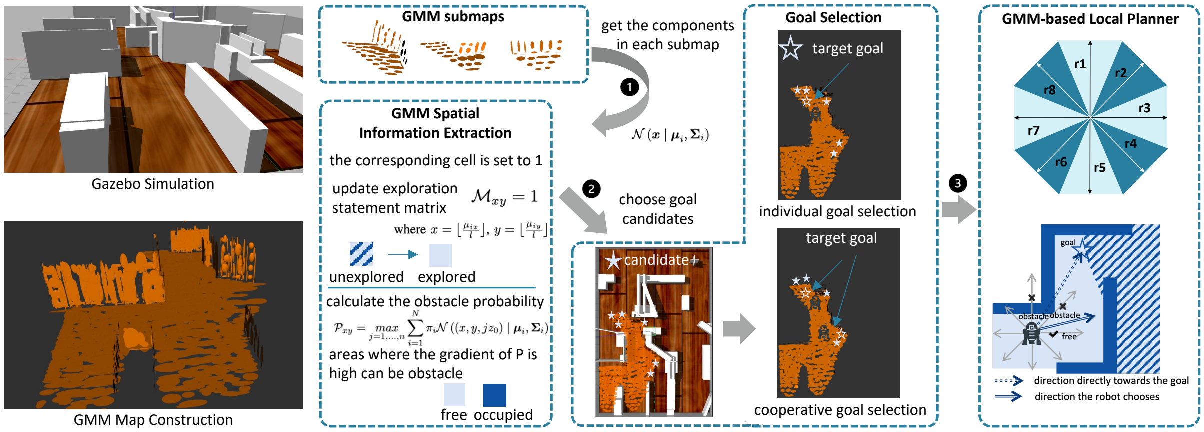

Gazebo Simulation
GMM Map Construction
GMM submaps
get the components in each submap
1
$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
GMM Spatial Information Extraction
the corresponding cell is set to 1
update exploration statement matrix $\mathcal{M}_{xy} = 1$
where $x = \lfloor \frac{\mu_{ix}}{l} \rfloor$, $y = \lfloor \frac{\mu_{iy}}{l} \rfloor$
unexplored
explored
calculate the obstacle probability
$\mathcal{P}_{xy} = \max_{j=1,\dots,n} \sum_{i=1}^{N} \pi_i \mathcal{N}((x,y,jz_0) \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
areas where the gradient of P is high can be obstacle
free
occupied
2
choose goal candidates
candidate
Goal Selection
target goal
individual goal selection
target goal
cooperative goal selection
3
GMM-based Local Planner
r1
r2
r3
r4
r5
r6
r7
r8
goal
obstacle
obstacle
free
direction directly towards the goal
direction the robot chooses

Fig. 3. The Process of Multi-Robot Exploration.

1) Individual Goal Selection: In the beginning, robots perform individual mapping and exploration. Each robot selects the point nearest to it from $\mathcal{L}$ as its goal. Since robots update $\mathcal{M}$ and $\mathcal{P}$ based on their own collected GMM submaps, their generated candidate sets are different. Therefore, robots are unlikely to choose the same goal.

2) Cooperative Goal Selection: When robots recognize that they have explored the same scene by means of place recognition (PR), they perform relative pose estimation (RelPose) and map merging. After that, robots use the merged map and shared location for collaborative exploration. We expect the robot to choose a goal that is close to itself but far from other robots. So our distributed strategy is that, the robot in $\boldsymbol{p}_k = (x, y)$ calculate the scores of each goal candidate in $\mathcal{L}$ and select the one with the highest score as the next goal to be explored. The score s is defined as:

$$s = \frac{\sum_{m=1, m \neq k}^{M} ||\boldsymbol{p}_L - \boldsymbol{p}_m||}{||\boldsymbol{p}_L - \boldsymbol{p}_k||} \tag{5}$$

where $\boldsymbol{p}_L \in \mathcal{L}$ is the candidate goal and M is the number of robots.

C. GMM-based Local Planner

As the next goal is determined, robots perform local path planning based on the obstacle probability of surrounding areas extracted from the GMM map. The obstacle probability matrix $\mathcal{P}$ demonstrates the environment as discrete probability. Therefore, we make modifications on the gradient-based local planner [8] by using difference to approximate the gradient field. As noted in Eq. (3), $\mathcal{P}_{x_0 y_0}$ is an obstacle probability at a location (x_0, y_0). Its gradient at a direction $\boldsymbol{r} = (r_x, r_y)$ is calculated by $g(\boldsymbol{r}) = \mathcal{P}_{(x_0+r_x)(y_0+r_y)} - \mathcal{P}_{x_0 y_0}$, where $r_x, r_y \in \{\pm 1, 0\}$ and $(r_x, r_y) \neq (0, 0)$. Then, we get the gradient of eight directions $\boldsymbol{r}_i,\ i = 1, .., 8$. The robots can move in any directions, with its direction falling into one of the eight zones shown in Fig. 3. For a zone containing direction $\boldsymbol{r}_i$, $g(\boldsymbol{r}_i)$ represents the gradient of the whole zone. For each robot, the obstacle probabilities at other robots' positions are set to 1 so as to avoid inter-robot collisions.

Fig. 3 illustrates the online GMM-based path planner. Given the current robot position and goal, the robot chooses the direction directly towards the goal as the candidate direction $\boldsymbol{r}_{cand}$. Then its gradient $g(\boldsymbol{r}_2)$ is calculated to check whether the robot will encounter obstacles. If $g(\boldsymbol{r}_2)$ is smaller than the threshold g_{thr}, it implies that this direction is safe, and the robot moves towards $\boldsymbol{r}_{cand}$. Otherwise, the robot selects the peripheral direction $\boldsymbol{r}_{rand} + 45°$ or $\boldsymbol{r}_{rand} - 45°$ as the candidate direction. If its corresponding gradient $g(\boldsymbol{r}_1) < g_{thr}$ or $g(\boldsymbol{r}_3) < g_{thr}$, the robot moves in $\boldsymbol{r}_{rand} + 45°$ or $\boldsymbol{r}_{rand} - 45°$. If the candidate direction constraint is still unsatisfied, we continue to choose its peripheral direction until the gradient of a direction is less than g_{thr}.

V. Experimental Results

A. Setup

1) Simulation Environment: As shown in Fig. 4, we create the simulation environment in Gazebo, which is 24m × 16m. Such cluttered environment includes straight passageways, archway, regular corners and front obstacles etc. The simulation robot is Turtlebot3 Waffle pi with a 180°laser scanner, whose distance range is 5m. The height of robot is 0.141(m). Each generated GMM submap is made up of 80 components. We set the resolution for exploration to l = 1m and the sampling interval for z-axis is 0.1m. We implement the multi-robot GMM mapping utilizing the ROS package *gmm_map_python* [22] and our GMM-based exploration implementation is packaged in the ROS package *gmm_explore*.

All simulations and evaluations are performed on a Desktop PC with an Intel Core i7-11700 processor and an NVIDIA GeForce RTX 3050 GPU.

2) Exploration Strategies for Comparison: Our proposed exploration strategy utilizes the 3D information extracted from GMM submaps, which is called **GMM 3dEnv Explore**. We perform comparison experiments by introducing **GMM 2dEnv Explore**, **GMM rand Explore** and **GMM non-cooper Explore**. In GMM 2dEnv Explore strategy, robots directly use the mean value of components, once the component satisfies $\mu_{iz} > 0$, the grid where $x = \lfloor \frac{\mu_{ix}}{l} \rfloor$, $y = \lfloor \frac{\mu_{iy}}{l} \rfloor$ is judged to

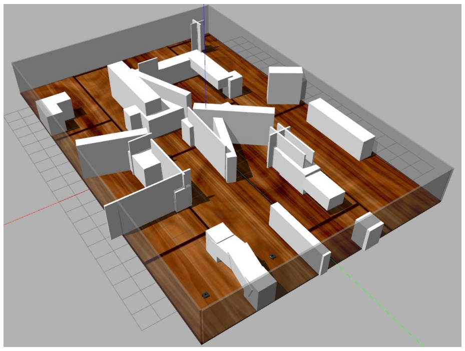


Fig. 4. Simulation Environment. (This is the map for multi-robot exploration while single-robot exploration adds a wall on it.)

be an obstacle area. GMM rand Explore strategy randomly chooses the target goal from candidate set, while robots using GMM non-cooper Explore strategy choose their own nearest target goals.

TABLE I
Computation Efficiency

Method	Traversal-based	Mean-based(ours)
Computation Time(s)	0.0628	0.0156

TABLE II
Data Transmission between two Robots (Byte)

	Robot1	Robot2	Average of 2 robots
Point Cloud	85460	106212	95836
GMM	4032	4244	4138

B. *Computation Efficiency for GMM Info Extraction*

The exploration statement matrix $\mathcal{M}$ and the obstacle probability matrix $\mathcal{P}$ are updated when a new GMM submap is generated. We calculate the average computation time for each update and the results in Tab. I show that our mean-based extraction method is 4 times faster than the traversal-based method.

C. *Data Transmission in Multi-Robot System*

Tab. II illustrates the average data transmission volume of each submap transmitted between two robots. The rate of adding novel point clouds into the submap is about 0.6 seconds per frame and each GMM submap is generated using 20 frames of point clouds. We can see that if robots transfer point clouds, the average submap data volume is 95836 byte. If robots transfer GMM submaps with 80 Gaussian components, the average data transmission is reduced to 4138 byte, thus saving approximately 96% communication load.

D. *GMM-based Exploration Evaluation*

We firstly evaluate the performance of GMM-based exploration on a single robot. Results in Fig. 5 show that our method performs better in both exploration time and traveled path length. The exploration time required to reach 95% coverage using GMM 3dEnv Explore, GMM 2dEnv Explore and GMM rand Explore is 350.7s, 385.5s and 464.2s respectively, while the path length is 60m, 70.5m and 83.5m. As illustrated in Fig. 6, compared with GMM rand Explore, our method can explore nearby areas in time, thus avoiding backtracking and repeated trajectory. Additionally, the GMM 3dEnv Explore makes better use of the information extracted from the 3D environment. As we can see from the marked red circle in Fig. 6, the robot using GMM 3dEnv Explore identifies an archway and chooses to pass through it, which consumes less time than driving around it in GMM 2dEnv Explore.

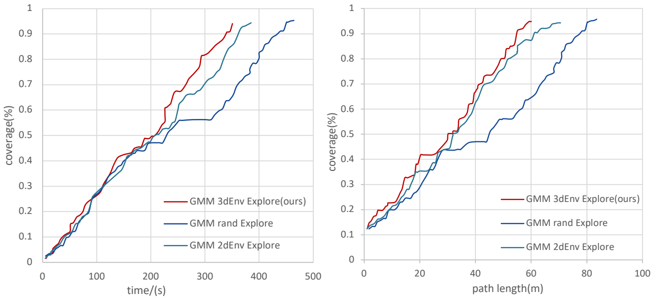
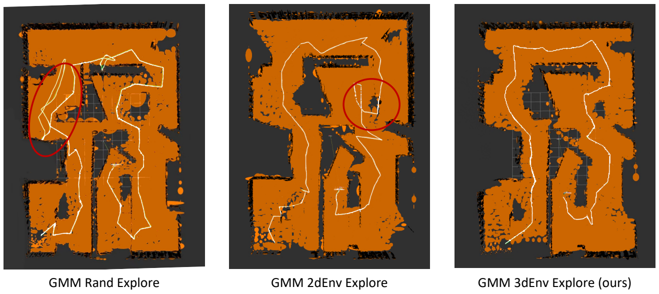


Fig. 5. Single-Robot Exploration Time and Path Length.



GMM Rand Explore

GMM 2dEnv Explore

GMM 3dEnv Explore (ours)

Fig. 6. Single-Robot Exploration Trajectory.

In order to clearly show the differences in exploration performance of the multi-robot system compared to the single-robot system, we find that two robots are most appropriate for our simulation environment.Thus we deploy two robots in the experiment with their exploration trajectories depicted in Fig. 8. In the GMM non-cooper Explore strategy, two robots repeatedly explore the same area, while in our method the two robots explore different places with almost no repeated path. Fig. 7 shows that in the first 50s, two robots perform individual exploration since they have not mapped the same area and estimated their relative pose. Therefore, the coverage rates of GMM non-cooper Explore and GMM 3dEnv Explore are similar. Afterwards, robots transfer GMM submaps with each other, merging them into a global map. We can see that the coverage rate of GMM 3dEnv Explore increases much faster than GMM non-cooper Explore. As a result, the exploration time when the coverage rate reaches 95% is reduced from 253.85s to 177.61s. Unnecessary paths have also been reduced seen in Tab. III. In addition, the exploration time using two robots is approximately half of the time using one robot.

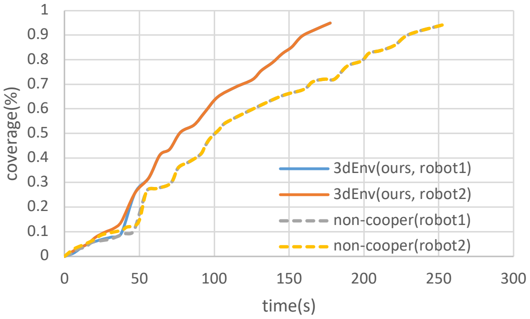


Fig. 7. Multi-Robot Exploration Time.

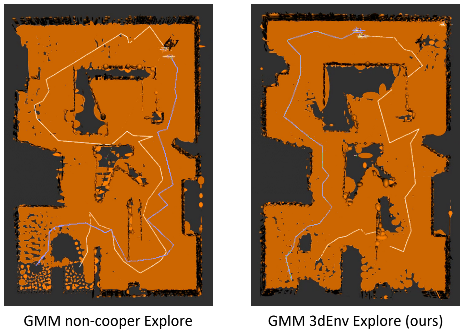


Fig. 8. Multi-Robot Exploration Trajectories.

VI. Conclusion and Future Work

In a communication-limited environment, it is important for robots to efficiently explore the environment while maintaining low data transmission. Thus, this work proposes a Gaussian mixture model-based multi-robot exploration system called MR-GMMExplore. The proposed GMM-based exploration involves 1) a GMM spatial information extraction strategy, 2) a goal selection method, 3) a GMM-based local planner.

Future work will consider the air-ground exploration in the communication-constrained environment. We expect to take the respective characteristics of heterogeneous robots into account and design more time-efficient and communication-efficient mapping and exploration strategy.

TABLE III

Multi-Robot Exploration Time and Path Length

	Path Length (m)			Time (s)
	Robot1	Robot2	Average	
3dEnv Explore (ours)	31.75	36.29	34.02	177.61
non-cooper Explore	43.41	38.74	41.08	253.85

References

[1] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood *et al.*, "Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 80–86.

[2] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual slam," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2466–2473.

[3] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "Door-slam: Distributed, online, and outlier resilient slam for robotic teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, 2020.

[4] Y. Xu, Z. Zhang, J. Yu, J. Cao, H. Dong, Z. Huang, Y. Wang, and H. Yang, "Game: Gaussian mixture model mapping and navigation engine on embedded fpga," in *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2021, pp. 60–68.

[5] C. O'Meadhra, W. Tabib, and N. Michael, "Variable resolution occupancy mapping using gaussian mixture models," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2015–2022, 2018.

[6] W. Tabib, K. Goel, J. Yao, M. Dabhi, C. Boirum, and N. Michael, "Real-time information-theoretic exploration with gaussian mixture model maps." in *Robotics: Science and Systems*, 2019, pp. 1–9.

[7] A. Dhawale, X. Yang, and N. Michael, "Reactive collision avoidance using real-time local gaussian mixture model maps," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3545–3550.

[8] H. Dong, J. Yu, Y. Xu, Z. Xu, Z. Shen, J. Tang, Y. Shen, and Y. Wang, "Mr-gmmapping: Communication efficient multi-robot mapping system via gaussian mixture model," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3294–3301, 2022.

[9] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, "Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8779–8785.

[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. O'Meadhra, and N. Michael, "On-manifold gmm registration," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3805–3812, 2018.

[12] H. Huang, H. Ye, Y. Sun, and M. Liu, "Gmmloc: Structure consistent visual localization with gaussian mixture models," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5043–5050, 2020.

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

[14] K. Goel, W. Tabib, and N. Michael, "Rapid and high-fidelity subsurface exploration with multiple aerial robots," in *International Symposium on Experimental Robotics*. Springer, 2020, pp. 436–448.

[15] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[16] S. Oßwald, M. Bennewitz, W. Burgard, and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 716–723, 2016.

[17] J. Faigl and M. Kulich, "On determination of goal candidates in frontier-based multi-robot exploration," in *2013 European Conference on Mobile Robots*. IEEE, 2013, pp. 210–215.

[18] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1396–1402.

[19] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.

[20] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*, 1998, pp. 47–53.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[22] "ROS package: gmm_map_python," 2022. [Online]. Available: https://github.com/efc-robot/gmm_map_python