SMMR-Explore: SubMap-based Multi-Robot Exploration System with Multi-robot Multi-target Potential Field Exploration Method

Jincheng Yu[1*], Jianming Tong[2*], Yuanfan Xu[1], Zhilin Xu[1], Haolin Dong[1], Tianxiang Yang[1] and Yu Wang[1]

***Abstract*— Collaborative exploration in an unknown environment without external positioning under limited communication is an essential task for multi-robot applications. For inter-robot positioning, various Distributed Simultaneous Localization and Mapping (DSLAM) systems share the Place Recognition (PR) descriptors and sensor data to estimate the relative pose between robots and merge robots' maps. As maps are constantly shared among robots in exploration, we design a map-based DSLAM framework, which only shares the submaps, eliminating the transfer of PR descriptors and sensor data. Our framework saves 30% of total communication traffic. For exploration, each robot is assigned to get much unknown information about environments with paying little travel cost. As the number of sampled points increases, the goal would change back and forth among sampled frontiers, leading to the downgrade in exploration efficiency and the overlap of trajectories. We propose an exploration strategy based on Multi-robot Multi-target Potential Field (MMPF), which can eliminate goal's back-and-forth changes, boosting the exploration efficiency by $1.03\times\sim1.62\times$ with $3\,\%\sim40\,\%$ travel cost saved. Our SubMap-based Multi-robot Exploration method (SMMR-Explore) is evaluated on both Gazebo simulator and real robots. The simulator and the exploration framework are published as an open-source ROS project at `https://github.com/efc-robot/SMMR-Explore`.**

I. Introduction

Exploring the unknown environment is a fundamental task for autonomous robot systems. A typical exploration consists of two parts, including perception & location and decision. For **perception & location**, Distributed Simultaneous Localization and Mapping (DSLAM) systems can provide the inter-robot relative pose when different robots experience the same place. For **decision**, robots have to explore the environment individually before experiencing the same place, and explore cooperatively after relative poses are provided.

Present multi-robot SLAM systems [1]–[6] transfer two kinds of data: 1) Place descriptors, which is a compact code describing a place. Place Recognition (PR) extracts and matches the descriptors to find a same place between robots. 2) Sensor data, which is used to calculate the relative pose between robots based on the matched same place.

[1] Department of Electronic Engineering, Tsinghua University, Beijing, China. yjc16@mails.tsinghua.edu.cn, yu-wang@tsinghua.edu.cn

[2] This work is finished during his internship at Tsinghua University. He is now at Georgia Institute of Technology, GA, USA.

*These authors contributed equally.

This work is supported by National Natural Science Foundation of China (No. U20A20334, U19B2019, 61832007), National Key R&D Program of China (No. 2019YFF0301505). This work is also supported by Tsinghua EE Xilinx AI Research Fund, Tsinghua EE Independent Research Project, Beijing National Research Center for Information Science and Technology (BNRist), and Beijing Innovation Center for Future Chips.

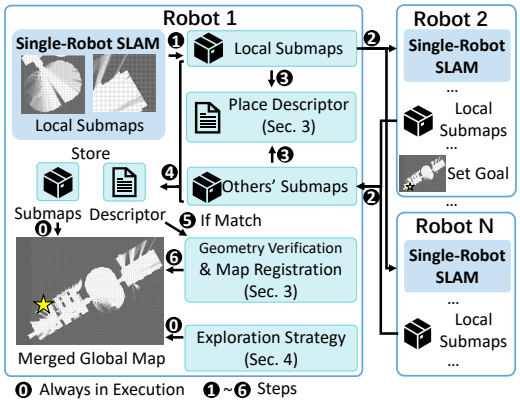


Fig. 1: Overview of the SMMR-Explore framework. Different robots only transfer submaps for PR, RelPose, and map generation & merging. The MMPF-based strategy improves the multi-robot exploration efficiency.

In the multi-robot exploration task, maps are indispensable and thus always shared among robots. Since the maps implicitly contains sensor and scene information, **maps themselves could be used for Place Recognition (PR) and relative pose estimation (RelPose) without redundant communication on place descriptors and sensor data**.

Frontier-based exploration strategies [7] are widely used in single or multi-robot exploration. Robots firstly detect dividing points between known and unknown area (a.k.a. frontiers) based on the current map. And then each robot selects the next viewpoint goal from detected frontiers considering the information gain, travel cost, and other factors.

Currently, sample-based exploration methods [8]–[11] hit state-of-the-art because of low computation overhead. However, the exploration efficiency of such methods suffers downgrade from trajectory overlap [10] and small passages [12] because of two underlying defects: 1) new sampled frontiers make the next viewpoint goal jump among different frontier clusters. 2) the ratio of frontiers to number of map points decreases as map grows. Overall, only utilizing sampled information of map for exploration should be blamed. Therefore, a new exploration paradigm is called for, which could **utilize complete map information to detect and select frontiers without a sacrifice of more computation resources**.

To fulfill the requirements above, in this work, we propose:

- A fully submap-based DSLAM method. Both PR and RelPose are based on the submaps shared among robots, eliminating 30% redundant data transmission.
- A **M**ulti-robot **M**ulti-target **P**otential **F**ield (MMPF) exploration method. The MMPF utilizes complete in-

formation of frontiers to select the best frontier goal, thus eliminating the trajectory overlap and increasing the exploration efficiency by $1.03\times \sim 1.62\times$.

- The open-source **S**ub**M**ap-based **M**ulti-**R**obot exploration package (SMMR-Explore) incorporating the aforementioned two contributions.

As illustrated in Fig. 1, the single-robot SLAM method (we adopt Cartographer [13]) builds the submaps. Splicing the submaps generates the single-robot map and, with a given relative pose, the inter-robot merged map. Each submap is shared among robots and a NN-based method encodes each submap into a feature vector for further matching to detect inter-robot loop closures. The submap is also used to estimate the relative pose between robots according to the matched place (detailed in Section III). The MMPF-based exploration method guides the robot on its own single-robot map or on the merged global map (detailed in Section IV). Experimental results will be given in Section V.

II. Related Work

A. DSLAM system

Previous multi-robot SLAM systems, either centralized [1], [5] or decentralized [2]–[4], [6], consist of two basic components: 1) place recognition (PR), 2) relative pose estimation (RelPose). These methods all calculate place descriptors to figure out the same place, but use different types of data, such as NetVLAD [14] using image sensor data, Bag of Words (BoW) [15] based on image feature points and Segmatch [16] leveraging 3D laser sensor data. Geometric verification estimates and optimizes the relative position of two frames whose place descriptors are similar. Such progress is also called RelPose, which bases on the sensor data between two robots, such as Perspective-n-Point needs image feature points [3], and Iterative Closest Point (ICP) requires laser point cloud [1]. Therefore, present DSLAM systems need to share the place descriptors and matched frames' sensor data among robots.

However, previous DSLAM systems are designed for multi-robot positioning, ignoring maps sharing between robots [2], [3]. Thus, they are not feasible in multi-robot exploration with a high dependency on maps. Compared with sharing a complete single-robot map, incremental map construction and merging only need to share newly added parts of maps and put little pressure on the communication bandwidth. The Cartographer [13] is widely used in single-robot SLAM frameworks, which builds the new-coming laser scan into a submap, and concatenates submaps to construct the single-robot map incrementally. In this work, inter-robot maps are also constructed incrementally using submaps from robots. As the submap (built from a 2D laser) can be used for PR [17] and RelPose [18], there is no need to share the redundant place descriptors and the sensor data of matched frames to reduce communication transfer.

B. Multi-robots Space Exploration Strategy

Recently, sample-based exploration methods are widely used in practical exploration systems, mainly including RRT

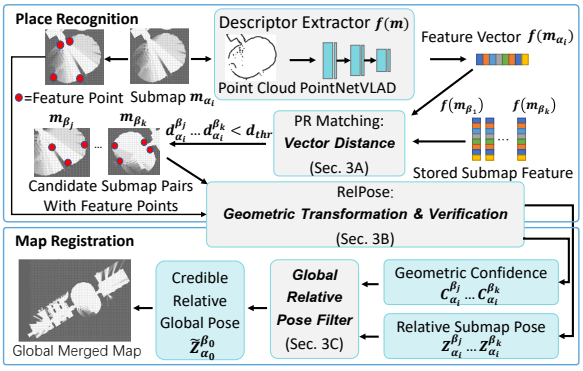


Fig. 2: An overview of submap-based PR and RelPose. We first use the distance of PR feature vectors to filter candidate match pairs. RelPose only processes the candidate match pairs selected by PR to reduce computation. The Global Relative Pose Filter gives a credible global relative pose ($\tilde{Z}_{\alpha_0}^{\beta_0}$) when multiple submap pairs are matched.

family [8]–[10], and random roadmap/graph family [19], [20]. The RRT continually picks random points in the map to grow itself and then publishes a goal when it reaches frontiers [10]. Lai etc. [12] propose disjointed trees to improve the sampling efficiency of the original RRT. Viseras etc. [21] cluster infinitely grown RRT to reduce the computation overhead. Unlike RRT, which performs only frontier detection, a random graph-based roadmap could also be used for path planning [20]. Wang etc. [20] reduce the sample region from the whole map to the free area and extended graph-based roadmap to support multiple queries for easy path planning. Although there are many improvements in sample efficiency and computation overhead for sample-based methods, frontiers detection and selection still use map samples, thus overlap just gets reduced instead of eliminated.

The Artificial Potential Field (APF) is proposed to utilize full map information to perform single target path planning [22]. In such a potential field, obstacles contribute to high potentials, and the designated target offer lower potential. Therefore, a robot under the APF will fall towards targets in the fastest gradient descent path, but sometimes it gets trapped in local minimums. Although some tactics to escape from local minimums are proposed [23], APF using them still wastes lots of time in local minimums escape. Recently, the ability to utilize the map information, travel cost, and multi-robot interference [24] plus the feature to avoid dynamic features [25] make APF suitable for multiple robot exploration. But APF still serves as the single-target path planner and will introduce numerous local minimums when deploying in scenarios with multiple targets. Therefore, we propose a new Multi-robot Multi-target Potential Field (MMPF) to eliminate local minimums and deploy it directly in multiple target selection. Besides, previous multi-robot exploration methods need external positioning system to provide the initial inter-robot relative pose. We combine exploration with DSLAM to do exploration without any external positioning.

III. Submap-based DSLAM

As illustrated in Fig. 1, there are two main components for processing inter-robot data in DSLAM system: 1) place

recognition and matching (PR). 2) inter-robot relative pose estimation (RelPose). In this section, we implement both components based on submaps. The submap-based DSLAM flow is illustrated in Fig. 2. m_{α_i} is the i^{th} submap of robot α. The PR method matches the descriptor of m_{α_i} to find the same place, and the RelPose method outputs the geometric confidence ($C_{\alpha_i}^{\beta_j}$) and the geometric transformation ($Z_{\alpha_i}^{\beta_j}$) between two submaps, m_{α_i} and m_{β_j}, from robot α and β.

A. Submap-Based Place Recognition

Robots judge whether they observe the same scene based on the shared submaps. Intuitively, a submap can be turned into an image and the PR could use vision-based methods like BoW [15] and NetVlad [14] to process the submap images. However, compared with images from the camera, the information in submaps is sparse and spatially structured, usually describing the shape of obstacles. Therefore, the full-image descriptors suffer performance degradation, as shown in Section V. Based on the fact that a submap is built with a few consecutive scans [13] and the 2D scan data contain the same geometric information with 2D point cloud, we design a submap-based descriptor extractor $f(.)$ to calculate a feature vector $f(m_{\alpha_i})$.

Our $f(.)$ consists of two components: 1) reconstructing the 2D point cloud from m_{α_i} and 2) 2D-PointNetVLAD, which encodes a 2D point cloud to a global descriptor vector. 2D-PointNetVLAD is modified from PointNetVLAD [26] to take 2D rather than 3D points as input and is trained to lower the $d_{\alpha_i}^{\beta_j}$ if m_{α_i} and m_{β_j} represent the same scene. $d_{\alpha_i}^{\beta_j}$ is the distance between $f(m_{\alpha_i})$ and $f(m_{\beta_j})$, e.g. cosine distance in our implementation. Each robot locally computes and compares PointNetVLAD descriptors of all stored submaps, including its own submaps and others' submaps. By doing so, a robot selects candidate pairs of submaps having distance below a given threshold d_{thr}, and filters out other pairs. The candidate pairs then go through the geometric consistency verification based on RelPose in order to achieve credible recognition results.

B. Submap-Based Relative Pose Estimation

The RelPose module estimates the relative transform of candidate pair provided by the PR module.

There are 2 stages in relative transform estimation: 1) Feature-point extraction. 2) Feature-point matching and transformation calculation. We use the $AKAZE$ feature detection method in OpenCV [27] to extract the feature-point, and the $AffineBestOf2NearestMatcher$ function to match the feature-points and calculate the image transformation $z_{\alpha_i}^{\beta_j}$, scale factor $s_{\alpha_i}^{\beta_j}$, and image consistency confidence $c_{\alpha_i}^{\beta_j}$, as [18] does. In this work, we consider that the geometric transformation between two submaps should be the same scale, thus, $z_{\alpha_i}^{\beta_j}$ is normalized by $s_{\alpha_i}^{\beta_j}$ to get geometric transformation $Z_{\alpha_i}^{\beta_j}$. The final geometric confidence C_{α_i} is also influenced by the scale factor. $C_{\alpha_i}^{\beta_j} = \min(s_{\alpha_i}^{\beta_j}, 1/s_{\alpha_i}^{\beta_j}) \cdot c_{\alpha_i}^{\beta_j}$. If $C_{\alpha_i}^{\beta_j}$ is above the threshold c_{thr} ($c_{thr} = 0.5$ in experiments), the candidate matched pair between m_{α_i} and m_{β_j} is verified as the same place.

C. Relative Pose Estimation with Pose Filter

$Z_{\alpha_i}^{\beta_j}$ can be easily converted to the relative pose of the initial robots' poses, $Z_{\alpha_0}^{\beta_0}$ (the initial pose is aligned to the 0^{th} submap of each robot). Different verified pairs may produce different $Z_{\alpha_0}^{\beta_0}$s. As DSLAM are vulnerable to incorrect place association and error $Z_{\alpha_0}^{\beta_0}s$ [3], we design a relative pose filter to eliminate mismatchings. When three or more submap pairs are verified between two robots, we calculate the mean value and standard deviation (σ) of all of these $Z_{\alpha_0}^{\beta_0}$s. The outliers which deviate more than $1{\cdot}\sigma$ from the mean value are filtered out. We calculate the average value of the remaining as the credible inter-robot relative pose $\tilde{Z}_{\alpha_0}^{\beta_0}$. $\tilde{Z}_{\alpha_0}^{\beta_0}$ is further used to merge the inter-robot maps.

IV. MMPF Exploration

As mentioned in Section II, current sample-based exploration strategies fail to solve the trajectory overlap because of random frontier detection and sample-info-based goal selection [8], [9]. Though Artificial Potential Field (APF) can fully utilize the global information, current works [22], [24], [25] use APF to plan a path towards a single target goal and waste time on escaping local minimums, as shown in Fig. 3a. We introduce **M**ulti-robot **M**ulti-target **P**otential **F**ield (MMPF) to utilize global information for frontier selection with three innovations: 1) a new potential field function of frontier incorporating obstacles' information and thus eliminating local minimums as shown in Fig. 3b, 2) a fast frontier cluster algorithm for reducing computation overhead, 3) a new potential field function of robots to dispatch them to different frontier clusters.

A. Potential Function of Frontier

In APF, tremendous local minimums come from interference of multiple targets and obstacles for two critical issues.

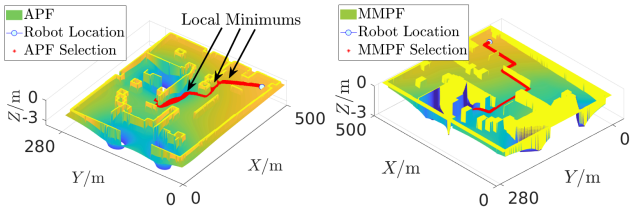
Firstly, the impact of a target's potential field decreases quadratically [23], [24] with the distance to the target increasing. The potential value will decrease to almost 0 in areas far from target, which is easily affected by potential fields of other targets, leading to the local minimums. Therefore we slow down the decrease in potential's impact by replacing square of distance r in APF into linear of distance as shown in Equ. (1), which is also the expression of the physical electrical potential. In physics, charged particles always tend to reach the place with the lowest potential in electric potential fields [28], which guarantees the the functionality of MMPF.

All frontiers are clustered into several groups, and the centroid of each group represents all frontiers of the group. The value in the potential field of the centroid i at point j is

$$P(i,j) = -\frac{kQ_i}{r_w(i,j)} \tag{1}$$

where Q_i stands for total number of frontiers in the cluster that centroid i belongs to, $r_w(i,j)$ represents the wave-front distance [23] from point j to centroid i.

Secondly, the euclidean distance of previous APF methods could not measure the length of the robot's actual moving



(a) APF finish in 6667 steps (b) MMPF finish in 512 steps

Fig. 3: MMPF is 10× faster then APF for eliminating local minimums (from 6667 to 512 search steps).

path, because obstacles are ignored. For example, robots must avoid obstacles lying between the departure and the target, yet the straight line between the departure and the target is not passable. Such difference between real passable length and measured euclidean length renders the local minimums. To precisely measure the length of path, we adopt wave-front distance [23] instead, to eliminate local minimums above. For each centroid of the frontier cluster, the wave-front distances from all non-obstacle points to it should be generated first and then stored as distance map for querying in following frontier selection. The distance map of each centroid is generated only once so the progress does not introduce much computation overhead. What's better, in this way, there is no need to calculate the potential from obstacles because the information of moving and static obstacles has already been encoded in wave-front distance. Thus, the obstacles' potential computing overhead in APF is also eliminated in our MMPF.

B. Continuity-based Faster Cluster for Frontiers

In Equ. (1), only centroid of each frontier cluster instead of all frontiers in this cluster is processed to reduce computation overhead. To avoid information lose, the number of frontiers in cluster i is denoted as coefficient Q_i. Popular clustering algorithms [29] suffer from great computation overhead because they need to generate the total number of clusters and classify all frontiers instead of just performing a classifying under specific number of groups. For example, the computation overhead of Meanshift [30] even exceeds the benefit of clustering feature of frontiers. We found that frontiers are always distributed in the form of continuous lines or curves. Leveraging such prior knowledge of frontiers' continuous layout (Line 7), our cluster algorithm reduces overhead to generate the total number of clusters to make it almost a classification algorithm, as shown in Algorithm 1.

C. Potential Function of Robots

Multiple robots should explore different places to increase the overall time efficiency, after the robots' maps are merged. Therefore, we propose a new repulsive potential and add it to every robot. The value in repulsive potential field of robot m at point j is shown in Equ. (2).

$$P_r(m,j) = \begin{cases} k_r \cdot (d_s - r_d(m,j)) & r_d(m,j) < d_s \\ 0 & r_d(m,j) > d_s \end{cases} \quad (2)$$

Algorithm 1 Continuity-based Cluster Algorithm

Require: all detected frontiers: $Frontiers$
1: initialize a frontier cluster list as $Clusters$
2: **while** $Frontiers$ is not empty **do**
3: initialize a cluster as $NewCluster$
4: copy the first one of $Froniters$ into $NewCluster$ and remove it from $Frontiers$
5: **repeat**
6: **for** $item$ in $Frontiers$ **do**
7: **if** $item$ is a neighbor of any frontier in $NewCluster$ **then**
8: copy $item$ into $NewCluster$ and delete $item$ in $Frontiers$
9: **end if**
10: **end for**
11: **until** no new frontier added into $NewCluster$
12: Put $NewCluster$ into $Clusters$
13: **end while**

where d_s stands for the sensors range, $r_d(m,j)$ is the distance between point j and robot m, k of Equ. (1) and k_r of Equ. (2) are coefficients to bring potential of targets and robots to the same magnitude.

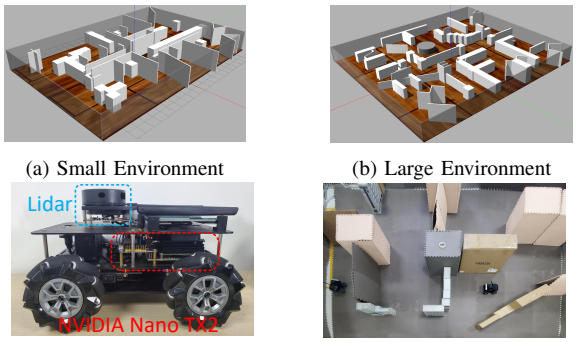
The potential field of robots only affect the area within its sensor range, because we only need to separate robots when they meet each other, and thus the separated robots will tend to explore different places in MMPF if there are multiple frontier clusters. Due to the small valid range of robots' potential fields, the absolute value of coordinates' difference could measure the real length of path between robots and then is used to calculate $r_d(m,j)$ to simplify computation. To strengthen the repulse among robots, we choose the repulsive potential to be linear with distance.

The root cause of trajectory overlap in sample-based methods could get solved in MMPF. Firstly, among all frontiers, only the best viewpoint and always this one is selected by MMPF, because utilized information does not change unless frontiers and map change. Secondly, the robot continues to explore one side without turning back until it has finished exploring this side, because the impact of distance is larger than the number of frontiers (the inverse proportional function decreases faster than the increase of proportional function).

V. Experiment Result

A. Dataset, Simulation Environment and Real Robot

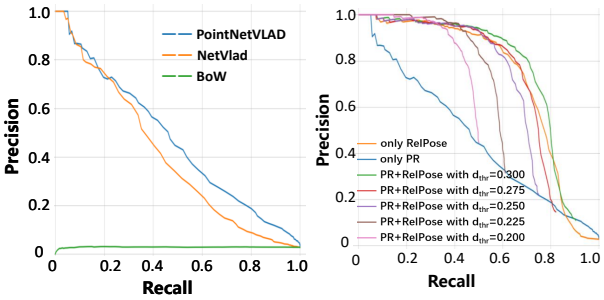
1) Deutsches Museum Dataset: To evaluate the performance of the submap-based place recognition (PR) and relative pose estimation (RelPose), we use the 2D-Lidar dataset provided with Cartographer [13]. Cartographer builds the submaps and calculates each submap frame's position, taken as the ground-truth position. The submaps, the distance of whose centroid is less than 6 m, are considered as the same place in training and evaluation. There are tens of trajectories in the Museum Dataset. We evaluate our work on



(a) Small Environment (b) Large Environment

(c) Real car with Lidar (d) Real Environment

Fig. 4: Experiment environments



(a) PointNetVLAD wins on PRC (b) d_{thr}'s impact on PR+RelPose

Fig. 5: The precision-recall curves of different methods

two trajectories (Museum Long/Short) and train the Neural Network (NN) on the rest.

2) Simulation Environment: We use Gazebo [31] to build the simulation environments, as shown in Fig. 4(a,b). The simulation robot has an 180° laser scanner, whose distance range is $7\,\mathrm{m}$. Submaps are considered the same place if the distance between centroids of them is less than $1\,\mathrm{m}$ in simulation and real robot evaluation.

3) Real Robot: The SMMR-explore is also deployed and evaluated in a real two-robot environment, as shown in Fig. 4(c,d). For perception, a laser scan, an IMU sensor, and the platform's odometer data are provided. For computation, an NVIDIA Jetson Nano board, with 4-core ARM CPU and an embedded GPU, is deployed on the robot. Cartographer for map construction and the RelPose run on the ARM CPU, while the NN-based PR method runs on the GPU.

B. Place Recognition Evaluation

1) Evaluation on Deutsches Museum Dataset: The point cloud converted from submap is downsampled to a fixed number of points and rescaled to be zero mean, as [26] does. The precision-recall curves (PRC) of different PR methods on submap, Bag of words (BoW), NetVLAD, PointNetVLAD, on the Museum test set are illustrated in Fig. 5(a). We use the AKAZE feature point extracted by RelPose to train the BoW model on the Museum training set. For lack of enough feature-points in the 2D submap, the BoW cannot distinguish different maps. The precision of BoW is almost zero. Although NetVLAD is more computationally expensive than PointNetVLAD, its performance is slightly worse than PointNetVLAD. The reason may be that NetVLAD is not designed for 2D maps and suffer from the rotation of maps, which is a common problem of CNN-based methods [32].

The PRC of the PointNetVLAD's results under different d_{thr} after RelPose geometric verification is illustrated in Fig. 5(b). The orange curve is the PRC of geometric verification without a PR filter. The blue curve is the PointNetVLAD method without geometric verification. To balance the accuracy and computation time, we set $d_{thr} = 0.275$.

2) Evaluation on Simulation and Real Robot: We collect submaps from the simulation environment for training and test the trained model in the real-world environment. The accuracy results and the speed results on Desktop PC (CPU: Intel I7-7920, GPU: Nvidia 1080TI) and Nano are listed in Table I. Our pre-trained model has a good ability of domain adaptation from simulation to a real environment. RelPose consumes most of the computation. PR can eliminate some unmatched scenes and reduce the computation of RelPose. The better the PR performance is, the more computation gets reduced, and the shorter the overall PR+RelPose time is. Compared with only geometric verification without PR filter, our method achieves comparable accuracy and $2\times\sim3\times$ acceleration. Note that the Cartographer generates a submap every several seconds. Our PR+RelPose takes $1\,\mathrm{s}\sim2\,\mathrm{s}$ to finish, which can meet the real-time requirements. As the scene size expands, the time of PR+RelPose will increase. For much larger scale scenarios, we need to design a more efficient PR method in the future.

TABLE I: Speed evaluation on desktop PC and the real robot.

			RelPose Without PR	RelPose with PR filter		
				PR	RelPose	Total
Train in Sim Test on Real Robot	PC (ms)	Worst.	991	29.5	418.5	448
		Ave.	912.75	13.6	394.1	407.7
	Nano (ms)	Worst.	3843	51.9	1579.1	1631
		Ave.	1854	43.4	686.6	730
	Acc. (%)	Recall	0.537	0.521		
		Precs	0.809	0.822		

[1] The column *RelPose Without PR* records the matching and estimation time when we directly apply RelPose w/o PR to the pairs of the current received submap and every stored submap. The column *RelPose with PR filter* records the time of different phases using our method.
[2] *Worst.* and *Avg.* are the longest and average matching time, separately.

C. RelPose Evaluation

We choose submap pairs, which are already verified with our PR and geometry verification modules, to calculate the map registration error, on Museum, simulation, and real-world datasets. We compare the translation error t_{error} and rotation error r_{error} of our submap-based method, which directly registers the submap image (Image-based) and the ICP method, which registers the converted point cloud. The results are shown in Table II. Image-based method is much better than ICP. Because ICP is an optimization-based method, which may lead to local optimum in sparse 2D point clouds.

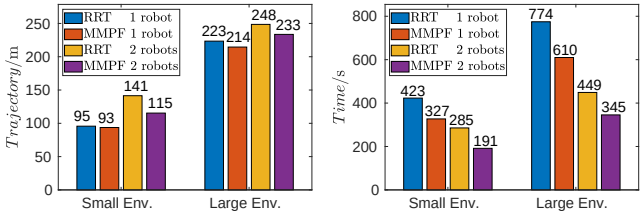
D. MMPF Exploration Evaluation

1) Baseline, Criteria and Finishing Condition: APF [24], [25] is not incorporated in experiments because it fails to complete exploration for the huge cost to handle local minimums. The exploration efficiency of the previous RRT work

TABLE II: Accuracy comparison between ICP and our method

	Image-based		ICP	
	t_{error}(m)	r_{error}(°)	t_{error}(m)	r_{error}(°)
Museum (Long)	0.55	1.41	6.89	1.51
Museum (Short)	0.13	0.28	8.48	1.74
Simulation	0.45	4.87	0.96	1.96
Real-world	0.23	2.71	0.71	2.24

t_{error}: average translation error (m) r_{error}: average rotation error (°).



(a) Trajectory Path Comparison (b) Time Comparison

Fig. 6: Comparison of MMPF and RRT

[10] suffers from the high latency between goal detection and selection. Thus, we improve the RRT work with synchronous goal selection, and take the improved method as the baseline.

We compare our MMPF and RRT with a variant number of homogeneous robots in a small environment ($391\,\mathrm{m}^2$, Fig. 4a) and a large environment ($661\,\mathrm{m}^2$, Fig. 4b). The exploration time of a robot refers to the total time from the robot's departure to the completion of exploring all frontiers in the environment. For multiple robots, the exploration time is the average of all robots' exploration time, and the trajectory length is the accumulation of length of all robots' trajectories.

2) Result: As shown in Fig. 6, compared with SOTA RRT, a single robot explores $1.18\times\sim1.27\times$ faster with $15.7\,\%$ length saved in small places under MMPF and $1.03\times\sim1.62\times$ faster with $19\,\%\sim40\,\%$ length saved in a large place. All improvement comes from 1) overlaps between two frontiers are eliminated by MMPF, which means robots will not move back and forth frequently. 2) robots will directly go to the frontiers without waiting for sampled points to fall on one of the frontiers, enabling MMPF to pass small passages rapidly.

When it comes to multiple robots, MMPF performs even better with $1.17\times\sim1.50\times$ efficiency with $10\,\%\sim23\,\%$ less travel cost in small place and $1.30\times\sim1.45\times$ efficiency with $3\,\%\sim6\,\%$ less travel cost. Apart from the overlap and the sample efficiency, such an efficiency boost in multi-robot scenarios also comes from collaborations. MMPF tends to dispatch robots to different frontier clusters to explore multiple places simultaneously, which RRT can not offer. Such improvement becomes more significant as the map's size becomes larger since low ratios of frontiers in maps make frontiers harder to detect.

3) Deep Dive in MMPF: As we observe, the map of MMPF will have more corner details than that of RRT because MMPF will always exhaust all unexplored small places, which are hard for sample-based methods to detect. Such a feature could be critical for rescue scenarios since trapped people usually stay just at corners.

As for overhead, distance maps consume the most memory in MMRF, which cost tens of MB on average. Although the computation overhead of MMPF per iteration is more than that of RRT per iteration, the average computation of MMPF stays almost at the same level with RRT because MMPF only needs to run at 1 Hz to satisfy the real-time demand. In contrast, RRT needs to run as frequently as possible, otherwise, it leads to low probability of sampling out a frontier. (100 Hz in our experiments).

TABLE III: Data transmission between two robots

	Method	Data Transmission (KB)			
		PR	RelPose	Submap	Total
Small	ScanPerSubmap	56.3	79.2	357.5	493
	ScanPerMatch	56.3	20.2	357.5	434
	OnlySubmap	0	0	357.5	357.5
Large	ScanPerSubmap	163.5	250.2	990	1403.7
	ScanPerMatch	163.5	155.4	990	1308.9
	OnlySubmap	0	0	990	990

E. Communication Evaluation

As described in Section III, the information shared between robots in our system is only the submap. In some centralized multi-robot SLAM systems like [1], every robot needs to share the PR descriptors of each submap and the scan data associated with the submap (Sharing scan data together with each submap, *ScanPerSubmap*). In previous DSLAM systems like [2], [3], they need to transmit PR descriptors of all keyframes to detect the same scene. When a match occurs, these methods need to send the keypoint or the raw data associated with the matched keyframe (Sharing raw/scan data when keyframe/submap matched, *ScanPerMatch*). Table III reports the data transmission between two robots on the trajectories from the small and large simulation environments. We share the submaps in compressed PNG format. Compared with the above two information sharing paradigms, our method (*OnlySubmap*) reduces the communication cost by roughly $30\,\%$. In fact, a robot extract PR descriptors not only for its own places, but also for other robots from the shared submaps. It is an exchange of computation for communication cost.

Although results of only two robots are given in this section, our framework supports three or more robots. More videos and results can be found on our website.

VI. Conclusion & Future Work

In this paper, we proposed 1) a submap-based map merge method which saves communication overhead caused by sensor scans, 2) a Multi-robot Multi-target Potential Field (MMPF) exploration method, which designates single robot or multiple robots to explore small or large places with better efficiency and less travel cost than sample-based methods.

As place recognition module has a significant influence on the computation cost and mapping quality, other place recognition methods for 2D range scans, such as [33], [34], can be involved in future. We will also try to expand the SMMR-Explore framework to 3D scene in future.

REFERENCES

[1] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot slam system for 3d lidars," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1004–1011.

[2] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual slam," in *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] 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.

[5] 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.

[6] P. Schmuck and M. Chli, "Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[7] 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.

[8] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[9] A. Q. Li, "Exploration and mapping with groups of robots: Recent trends," *Current Robotics Reports*, pp. 1–11, 2020.

[10] 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 2017, Vancouver, BC, Canada, September 24-28, 2017*. IEEE, 2017, pp. 1396–1402. [Online]. Available: https://doi.org/10.1109/IROS.2017.8202319

[11] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3d," *Advanced Robotics*, vol. 27, no. 6, pp. 459–468, 2013.

[12] T. Lai, F. Ramos, and G. Francis, "Balancing global exploration and local-connectivity exploitation with rapidly-exploring random disjointed-trees," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5537–5543.

[13] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[14] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *CVPR*, 2016, pp. 5297–5307.

[15] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, June 2010, pp. 3304–3311.

[16] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5266–5272.

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

[18] J. Hörner, "Map-merging for multi-robot system," Prague, 2016. [Online]. Available: https://is.cuni.cz/webapps/zzp/detail/174125/

[19] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3105–3112.

[20] C. Wang, W. Chi, Y. Sun, and M. Q. . Meng, "Autonomous robotic exploration by incremental road map construction," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1720–1731, 2019.

[21] A. Viseras, Z. Xu, and L. Merino, "Distributed multi-robot cooperation for information gathering under communication constraints," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1267–1272.

[22] C. W. Warren, "Global path planning using artificial potential fields," in *1989 IEEE International Conference on Robotics and Automation*. IEEE Computer Society, 1989, pp. 316–317.

[23] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, S. Thrun, and R. C. Arkin, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[24] C. Wang, L. Meng, T. Li, C. W. De Silva, and M. Q. . Meng, "Towards autonomous exploration with information potential field in 3d environments," in *2017 18th International Conference on Advanced Robotics (ICAR)*, 2017, pp. 340–345.

[25] C. W. Warren, "Multiple robot path coordination using artificial potential fields," in *Proceedings., IEEE International Conference on Robotics and Automation*, 1990, pp. 500–505 vol.1.

[26] M. Angelina Uy and G. Hee Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4470–4479.

[27] OpenCV, "Opencv," [EB/OL], https://opencv.org/ Accessed Oct. 20 , 2020.

[28] B. S. Guru and H. R. Hiziroglu, *Electromagnetic field theory fundamentals*. Cambridge university press, 2009.

[29] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[30] Yizong Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[31] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[32] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2017–2025. [Online]. Available: http://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf

[33] G. D. Tipaldi, L. Spinello, and W. Burgard, "Geometrical flirt phrases for large scale place recognition in 2d range data," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2693–2698.

[34] M. Himstedt, J. Frost, S. Hellbach, H.-J. Böhme, and E. Maehle, "Large scale place recognition in 2d lidar scans using geometrical landmark relations," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 5030–5035.