Multi-robot Feature-based SLAM using Submap Joining

Shengduo Chen[1,2], Ruihua Han[1,3], Liang Zhao [2,*], Shoudong Hang[2] and Qi Hao[1,†‡§]**

Abstract

This paper considers the feature-based SLAM using multiple robots. To reduce the computational complexity and data storage, a distributed multi-robot feature-based SLAM algorithm under submap joining scheme is proposed. Each robot first independently builds a submap using the information collected by its sensors. Once the robots can observe each other, the submaps can then be fused together to obtain a global map. We implemented and tested the proposed algorithm in both simulation and real world environments. Both simulation and experimental results have validated the robustness and accuracy of the proposed algorithm.

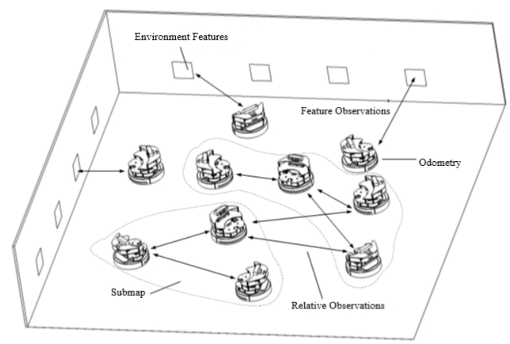


Figure 1: The illustration of multiple robot SLAM problem. Multiple robots move around in an environment with landmarks/features. The robot can observe each other from time to time. The SLAM problem is to use the odometry information, the feature observations, and the relative robot observations to estimate the feature positions and the trajectories of each robot.

1 INTRODUCTION and RELATED WORKS

Multi-robot systems (MRS) have been validated their efficiency in performing complex tasks in the real world such as rescue and disaster management, surveillance and monitoring, underwater exploration, etc. [Sajad-Saeedi *et al.*2016]. For multiple robots operating in an unknown environment, one important problem is building a map of the environment through Simultaneous Localization and Mapping (SLAM). SLAM research has been progressed significantly in the last two decades [Cadena *et al.*2016]. The feature based SLAM gains more and more attention as the growing trends of semantic SLAM. In many practical applications, it is necessary to utilize multiple robots to perform SLAM collaboratively [Tian *et al.*2018]. However, in a large-scale multi-robot SLAM, there are multiple technical challenges, including computational complexity and limited communication bandwidth. For large-scale multi-robot SLAM problem, it is in general very inefficient or impractical to directly solve the full SLAM problem using all the information available, especially when the inter-robot data transmissions are limited due to the limited bandwidth of unstable network [Cunningham and Dellaert2012]. Thus different research groups have proposed different strategies for distributed multi-robot SLAM. In [Andersson and Nygards2008], the authors introduced a base node constraint to connect the coordinate frames of the submaps. The base node constraint is obtained

*[1]Shengduo Chen, Ruihua Han and Qi Hao are with Department of Computer Science and Engineering, Southern University of Science and Technology, 518055 Shenzhen, China {11860006, Hanrh}@mail.sustech.edu.cn

†[2]Shengduo Chen, Shoudong Huang, and Liang Zhao are with the Robotics Institute of University of Technology Sydney, Australia.

‡[3]Ruihua Han is with the The Department of Computer Science, The University of Hong Kong.

§*Corresponding authors: *Liang Zhao*, *Qi Hao*. Liang.Zhao@uts.edu.au, haoq@mail.sustc.edu.cn

by non-linear optimization with relative observation of other robots (rendezvous-measurements). However, in this method, the submaps are required to be adequately linearized, which means it only works well with small submaps. Some researchers also considered factor graph representation with Gaussian elimination to generate condensed local graph for multi-robot SLAM [Cunningham *et al.*2010] [Cunningham *et al.*2012]. [Lazaro *et al.*2013] utilized condensed pose graph and the scan corresponds to those pose in condensed graph to represent submaps in multi-robot SLAM system. This framework also allows dynamically adding and removing robots from the system which improves the scalability of the system. However, these submap joining based multi-robot SLAM algorithms are all based on pose-graph SLAM where each submap only contains the robot poses, and the features in the environment are not explicitly estimated.

Feature based submap joining have been well studied and proven to have lower computational complexity than non-linear full least square optimization methods [Zhao *et al.*2014]. However, it is mostly used in single robot SLAM. [Huang *et al.*2008] leveraged the sparseness of SLAM information matrix and developed a Sparse Local Submap Joining Filter which utilize sparse information filter to merge submap efficiently. [Zhao *et al.*2013] further improved the efficiency by transforming the SLAM problem from traditional non-linear optimization problem to linear optimization problem with coordinate transformations. Although these algorithm can achieve good performance in large scale environment and guarantee of global optimum results, they are not designed for multiple robots.

In this paper, we propose a feature based submap joining approach for solving the distributed multi-robot SLAM problem. In the proposed approach, each robot first builds its own submap in its local coordinate independently, using the detected landmarks as features. The robots in the group communicate and detect each other to estimate their relative positions. Once needed, the submaps can be integrated using the relative information between the different local submaps through the observation between the robots or the observation of the same landmarks from different robots. The local submaps can be built online, the global map is built offline, and limited communication among the robots is needed for exchange the relative observations. The accuracy of the global map is similar to the map built using all the information through full least squares optimization. Both simulations and practical datasets with ground truth are used to validate the effectiveness of the proposed approach.

The contributions of this paper includes:

- Developing a distributed feature based multi-robot SLAM algorithm through submap joining.
- Evaluating the proposed multi-robot SLAM algorithm using both simulations and experimental datasets with ground truth.

The paper is organized as follows. Section 2 presents our multi-robot SLAM algorithm. Section 3 evaluates the algorithm using both simulation and experimental datasets. Section 4 concludes the paper.

2 MULTI-ROBOT SUBMAP JOINING BASED SLAM

This section presents the details of the proposed submap joining based multi-robot SLAM.

2.1 Generating and Maintaining Submap

In this paper, we focus on SLAM problem with multiple robots where each robot i ($i = 1 : s$) maintains a local submap[1]

$$\mathbf{S}^i = \{\mathbf{X}^i, \mathbf{X}^i_r, \mathbf{F}^i\} \tag{1}$$

which containing the poses of the current robot i: $\mathbf{X}^i = \{X^i_2, ..., X^i_t, ..., X^i_m\}$ at different timestep t ($t = 1 : m$); the poses of other robots observed by robot i: $\mathbf{X}^i_r = \{X^i_{j,t_j}, ..., X^i_{s,t_s}\}$, where t_j ($j \neq i$) is the timestep at which robot j was observed; and the positions of the observed features: $\mathbf{F}^i = \{F^i_1, ..., F^i_k, ...F^i_n\}$, where $k = 1 : n$ is the feature number; all in the local submap coordinate frame. Since the coordinate frame of submap $\mathbf{S}^i$ is defined by the robot pose at the first timestep, thus X^i_1 is not in the state vector.

During the SLAM process, at timestep t robots can obtain: the odometry information O^i_t from pose X^i_{t-1} to X^i_t with the proprioceptive sensors in the robot base; the observations/locations $\{Z^i_{F_k,t}\}$ of features $\{F^i_k\}$ in the coordinate frame of robot X^i_t; as well as the poses $\{Z^i_{X_j,t}\}$ of other robots ($j \neq i$) in limited range and field of view (FOV). Therefore, for each submap, we can solve the state vector in (1) with weighted non-linear least squares (NLLS) method by minimizing the following objective function:

$$\begin{aligned} f(\mathbf{S}^i) = &\sum_{t=1}^{m}\sum_{k=1}^{n} \left\| Z^i_{F_k,t} - H^{Z^i_{F_k,t}}(\mathbf{S}^i) \right\|^2_{P^{-1}_{Z^i_{F_k,t}}} \\ &+ \sum_{t=2}^{m} \left\| O^i_t \boxminus H^{O^i_t}(\mathbf{S}^i) \right\|^2_{P^{-1}_{O^i_t}} \\ &+ \sum_{t=1}^{m}\sum_{j=1}^{s} \left\| Z^i_{X_j,t} \boxminus H^{Z^i_{X_j,t}}(\mathbf{S}^i) \right\|^2_{P^{-1}_{Z^i_{X_j,t}}} . \end{aligned} \tag{2}$$

[1]To simplify the notations, the 'transpose's in the state vectors are sometimes omitted in this paper. For example, $\mathbf{S}^i, \mathbf{X}^i, \mathbf{X}^i_r, \mathbf{F}^i$ are all column vectors and the rigorous notation should be $\mathbf{S}^i = \{(\mathbf{X}^i)^T, (\mathbf{X}^i_r)^T, (\mathbf{F}^i)^T\}^T$.

In the objective function, $H^{Z^i_{F_k,t}}(\mathbf{S}^i)$ is the observation function of features

$$H^{Z^i_{F_k,t}}(\mathbf{S}^i) = R^i_t(F^i_k - T^i_t) \tag{3}$$

where R^i_t and T^i_t are the rotation matrix and translation vector of robot pose X^i_t.

$H^{O^i_t}(\mathbf{S}^i)$ and $H^{Z^i_{X_j,t}}(\mathbf{S}^i)$ are the functions corresponding to the odometry O^i_t and the observations to the other robots $Z^i_{F_k,t}$

$$O^i_t \boxminus H^{O^i_t}(\mathbf{S}^i) = \begin{bmatrix} O^{T,t}_i - (R^i_{t-1})^T(T^i_t - (T^i_{t-1}) \\ d_{SO}(O^{R,t}_i, (R^i_{t-1})^T R^i_t) \end{bmatrix} \tag{4}$$

and

$$Z^i_{X_j,t} \boxminus H^{Z^i_{X_j,t}}(\mathbf{S}^i) = \begin{bmatrix} Z^i_{T_j,t} - (R^i_t)^T(T^i_{j,t_j} - T^i_t) \\ d_{SO}(Z^i_{R_j,t}, (R^i_t)^T R^i_{j,t_j}) \end{bmatrix}, \tag{5}$$

where $(O^{T,t}_i, O^{R,t}_i)$, $(Z^i_{T_j,t}, Z^i_{R_j,t})$ and $(T^i_{j,t_j}, R^i_{j,t_j})$ are the rotation and translation components of O^i_t, $Z^i_{X_j,t}$ and X^i_{j,t_j}, respectively. $d_{SO}(\star, \bullet)$ means the distance function on the Lie group $SO(2)$ or $SO(3)$. One example is $\|\log(\star^\top \bullet)^\vee\|$ where $^\vee$ means the inverse of the skew-symmetric operator.

And $P_{Z^i_{F_k,t}}$, $P_{O^i_t}$ and $P_{Z^i_{X_j,t}}$ are the covariance matrices of the feature observation $Z^i_{F_k,t}$, odometry O^i_t and other robot observation $Z^i_{X_j,t}$, respectively.

If we write the NLLS problem (2) into the general formulation as

$$f(\mathbf{S}^i) = \left\|Z - H(\mathbf{S}^i)\right\|^2_{I_Z} \tag{6}$$

where $Z = \{..., Z^i_{F_k,t}, ..., O^i_t, ..., Z^i_{X_j,t}, ...\}$ combines all the observations and odometry, $H(\mathbf{S}^i)$ combines all the observation functions corresponding to Z and $I_Z = \mathrm{diag}(..., P_{Z^i_{F_k,t}}, ..., P_{O^i_t}, ..., P_{Z^i_{X_j,t}}, ...)^{-1}$ combines all the covariance matrices. The optimal solution $\hat{\mathbf{S}}_i$ which minimizing (6) can be obtained by using Gauss-Newton iteration method as shown in Algorithm 1. The observation function $H(\mathbf{S}_i)$ in (6) is first linearized using Taylor expansion. The Jacobean matrix J of $H(\mathbf{S}^i)$ can be obtain by calculating the derivation near the initial value $\mathbf{S}^i_{ini}$. The updated state vector $\mathbf{S}^i$ is then calculated by using (7) iteratively as

$$J^T I_Z J \Delta_{S^i} = J^T I_Z (Z - H(\mathbf{S}^i)) \tag{7}$$

$$\mathbf{S}^i = \mathbf{S}^i + \Delta_{S^i}. \tag{8}$$

After the optimal solution of the state $\hat{\mathbf{S}}_i$ is obtained, the corresponding information matrix can be calculated with:

$$I_{S^i} = J^T I_Z J. \tag{9}$$

Then, the optimal solution of the state vector together with the corresponding information matrix $\{\hat{\mathbf{S}}_i, I_{S^i}\}$ will be used to represent the submap built by robot i, and in the map joining of multiple robots in the next subsection.

Algorithm 1 Submap Building Process

Input: $\{O^i_t\}$, $\{Z^i_{F_k,t}\}$ and $\{Z^i_{X_j,t}\}$
Output: optimal submap $(\hat{\mathbf{S}}^i, I_{S^i})$

1: initialize $\mathbf{S}^i = \mathbf{S}^i_{ini}$;
2: **while** Δ_{S^i} is not small enough **do**
3: calculate error $Z - H(\mathbf{S}^i)$ at $\mathbf{S}^i$ using (3)-(5) ;
4: calculate Jacobin matrix J at $\mathbf{S}^i$ in (7);
5: calculate the update Δ_{S^i} using (7);
6: update $\mathbf{S}^i = \mathbf{S}^i + \Delta_{S^i}$ in (8);
7: **end while**
8: calculate information matrix I_{S^i} at $\hat{\mathbf{S}}^i$ using (9);
9: store $\hat{\mathbf{S}}^i$ and I_{S^i}.

2.2 Multi-robot Submap Joining

After robots obtained submaps that containing more than one pose of other robots or two common observations of landmarks, the map joining process can be performed to obtain the global map.

The submap joining process can be done by joining one pair of submaps or joining multiple submaps at the same time.

Suppose the submap built from the robot i is denoted by $\{\hat{\mathbf{S}}_i, I_{S^i}\}$, $(i = 1 : s)$, in the map joining process, the state of the joined global map can be defined as

$$\mathbf{G} = \{\mathbf{X}^G_1, ..., \mathbf{X}^G_i, ..., \mathbf{X}^G_s, \mathbf{F}^G\} \tag{10}$$

where $\mathbf{X}^G_1 = \{X^G_{1,2}, ..., X^G_{1,t}, ..., X^G_{1,m}\}$ and $\mathbf{X}^G_i = \{X^G_{i,1}, ..., X^G_{i,t}, ..., X^G_{i,m}\}$ (for $i >= 2$) contain the poses of all the robots to be joint at all the timesteps $t = 1 : m$, $\mathbf{F}^G = \{F^G_1, ..., F^G_k, ...F^G_n\}$ $(k = 1 : n)$ contains all the features, all in the global coordinate frame defined by the pose of the first robot at the first timestep, thus $X^G_{1,1}$ is not included in the state vector $\mathbf{G}$.

The submap joining algorithm for multiple robots proposed in this paper can be formulated as a nonlinear optimization problem which minimizing the objective function:

$$g(\mathbf{G}) = \sum_{i=1}^{s} \left\|\mathbf{S}^i \boxminus H^{S_i}(\mathbf{G})\right\|^2_{I_i}. \tag{11}$$

Here each submap $\{\hat{\mathbf{S}}_i, I_{S^i}\}$ is used as an integrated observation in the map joining process. And the observation function $H^{S_i}(\mathbf{G}) = \{..., H^{F^i_k}(\mathbf{G}), ..., H^{X^i_t}(\mathbf{G}), ..., H^{X^i_{j,t_j}}(\mathbf{G}), ...\}$ is a combination of functions corresponding to the state of each submap $\hat{\mathbf{S}}_i$:

$$\hat{F}_k^i - H^{F_k^i}(\mathbf{G}) = \hat{F}_k^i - (R_{i,1}^G)^T(F_k^G - T_{i,1}^G) \tag{12}$$

compares the difference between the feature location $\hat{F}_k^i$ from the submap $\mathbf{S}_i$ and the feature location calculated from the global map $\mathbf{G}$, in the coordinate frame of submap $\mathbf{S}_i$. Here $R_{i,1}^G$ and $T_{i,1}^G$ are the rotation matrix and translation vector of robot pose $X_{i,1}^G$ in the global map $\mathbf{G}$.

$$\hat{X}_t^i \boxminus H^{X_t^i}(\mathbf{G}) = \begin{bmatrix} \hat{T}_t^i - (R_{i,1}^G)^T(T_{i,t}^G - T_{i,1}^G) \\ d_{SO}(\hat{R}_t^i, (R_{i,1}^G)^T R_{i,t}^G) \end{bmatrix} \tag{13}$$

calculates the difference between the estimated pose $\hat{X}_t^i$ of robot i at timestep t in the submap $\mathbf{S}_i$, and the corresponding one calculated using the state of the global map $\mathbf{G}$, again, in the submap coordinate frame. And

$$\hat{X}_{j,t_j}^i \boxminus H^{X_{j,t_j}^i}(\mathbf{G}) = \begin{bmatrix} \hat{T}_{j,t_j}^i - (R_{i,1}^G)^T(T_{j,t_j}^G - T_{i,1}^G) \\ d_{SO}(\hat{R}_{j,t_j}^i, (R_{i,1}^G)^T R_{j,t_j}^G) \end{bmatrix} \tag{14}$$

calculates the difference between the estimated pose $\hat{X}_{j,t_j}^i$ of other robot j at timestep t_j in the submap $\mathbf{S}_i$, and the corresponding one calculated using the state of the global map $\mathbf{G}$. Here $t_j \in (1 : m)$, $j \neq i$ is the timestep at which robot j was observed by robot i.

Then, the optimal solution of the global map together with the corresponding information matrix $\{\hat{\mathbf{G}}, I_G\}$ can be obtained using the similar way as described in Section 2.1.

Algorithm 2 Submap Joining Process

Input: submaps $\{(\hat{\mathbf{S}}^i, I_{S^i})\}$
Output: optimal global map $(\hat{\mathbf{G}}, I_G)$

initialize $\mathbf{G} = \mathbf{G}_{ini}$;
while Δ_G is not small enough **do**
    $I = 0$, $E = 0$;
    for robot $i \in s$ **do**
        calculate $e_i = \mathbf{S}^i \boxminus H^{S_i}(\mathbf{G})$ and Jacobian J_i;
        calculate $E = E + J_i^T I_{S^i} e_i$;
        calculate $I = I + J_i^T I_{S^i} J_i$;
    end for
    calculate the update from $I\Delta_G = E$;
    update $\mathbf{G} = \mathbf{G} + \Delta_G$;
end while
calculate information matrix I at $\hat{\mathbf{G}}$;
share the optimal global map $(\hat{\mathbf{G}}, I_G)$.

3 EXPERIMENTAL EVALUATION

In this section, we evaluate the proposed multi-robot SLAM algorithm using both simulation data and practical experimental data.

3.1 Simulation

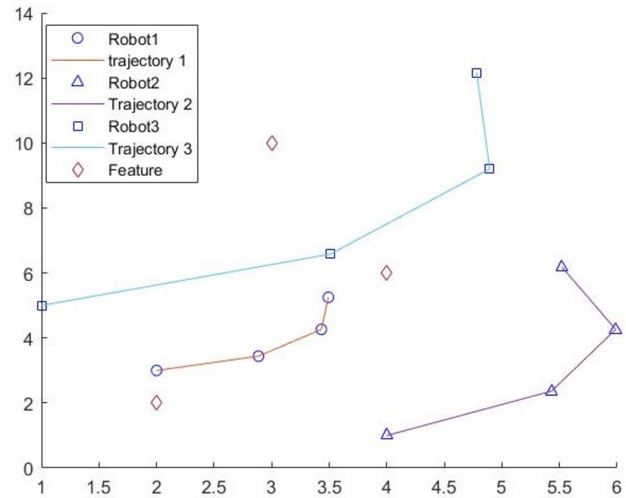


Figure 2: The simulation scenario involve three robots and three features. The robots move in 2D $x - y$ plane and the z values of the three features are all $2m$.

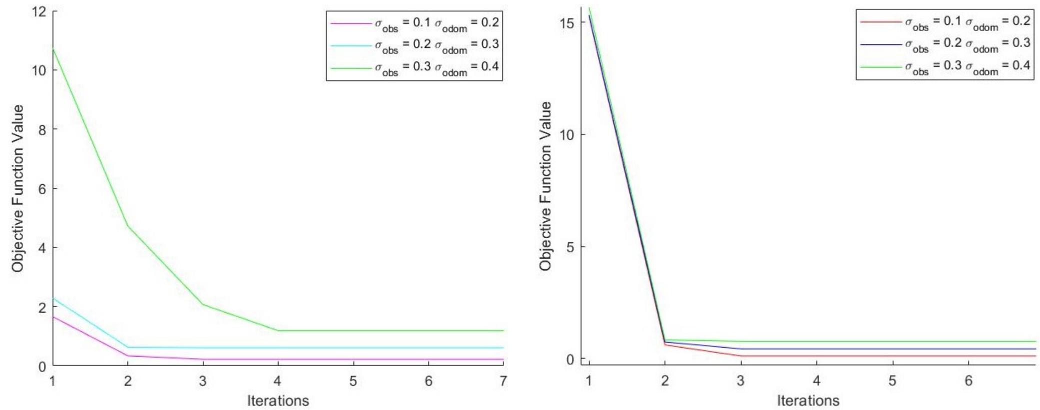
For the simulation, we use MATLAB to simulate an experiment with three robots and three features/landmarks. Fig. 2 shows the trajectories of the three robots and the (x, y) location of the three features. The robots are moving in 2D $x-y$ plane and the z value (the height) of the features are the same $z_f = 2m$.

Simulation Setup
In order to generate simulation data as close to the data obtained in the real experiments as possible, in the simulation, we control the robots with linear velocity v and angular velocity ω under differential drive kinematic model

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} v/\omega\cos(\theta_t) + v/\omega\sin(\omega\delta t + \theta_t) \\ v/\omega\sin(\theta_t) - v/\omega\cos(\omega\delta t + \theta_t) \\ \omega\delta t \end{bmatrix} \tag{15}$$

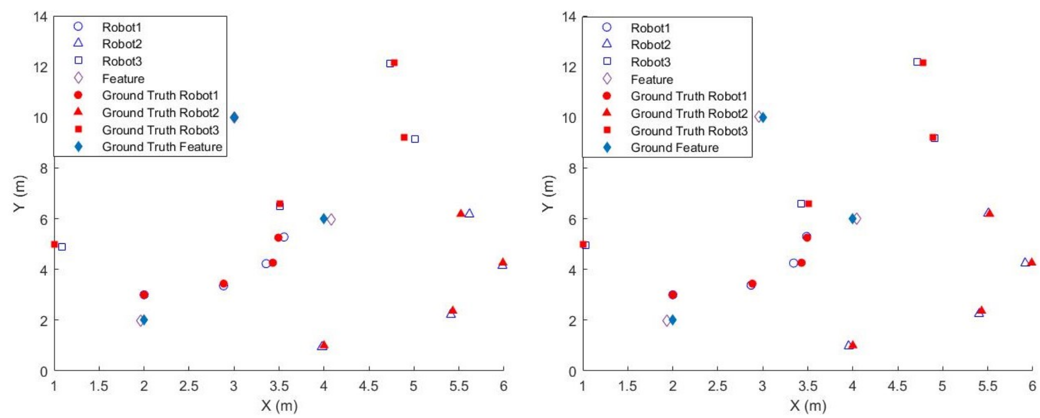
where $[x_t, y_t, \theta_t]^T$ is the robot pose at time t and δt is the time increment in simulation. Each robot is assumed to be able to observe features and other robots within a certain range (3m) at each time step with the following feature observation model:

$$\begin{bmatrix} \delta x_f \\ \delta y_f \\ \delta z_f \end{bmatrix} = \begin{bmatrix} \cos(\theta_{i,t}) & -\sin(\theta_{i,t}) & 0 \\ \sin(\theta_{i,t}) & \cos(\theta_{i,t}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{f,i} - x_{i,t} \\ y_{f,i} - y_{i,t} \\ z_{f,i} - h \end{bmatrix}. \tag{16}$$



(a) NLLS (b) Submap Joining

Figure 3: The changes of objective function value in different iteration steps in simulation under different noise levels $\sigma_{obs} = 0.1, 0.2, 0.3$ and $\sigma_{odom} = 0.2, 0.3, 0.4$. The objective function values of (a) NLLS, (b) the objective function value of submap joining.



(a) Submap Joining (b) NLLS

Figure 4: The simulation results. This results are obtained with observation noise $\sigma_{obs} = 0.3$ and odometry noise $\sigma_{odom} = 0.4$.

where h is the height of the sensor on the robot that set to 1m in the simulation. And for the relative pose observations of robot j in robot i coordinate:

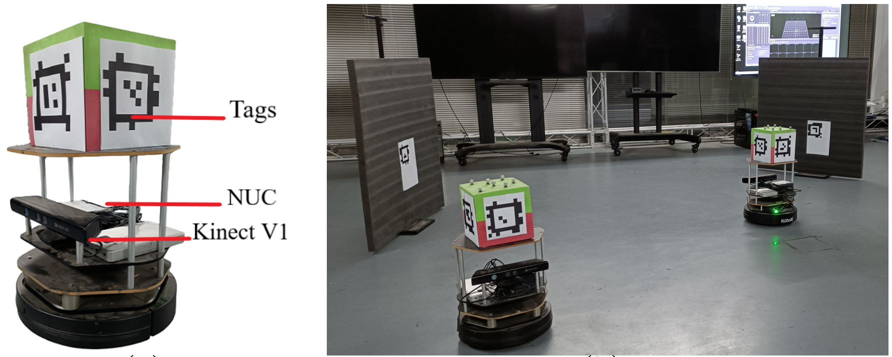
$$\begin{bmatrix} \delta x^i_{j,t} \\ \delta y^i_{j,t} \\ \delta \theta^i_{j,t} \end{bmatrix} = \begin{bmatrix} \cos(\theta^G_{i,t}) & -\sin(\theta^G_{i,t}) & 0 \\ \sin(\theta^G_{i,t}) & \cos(\theta^G_{i,t}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^G_{j,t} - x^G_{i,t} \\ y^G_{j,t} - y^G_{i,t} \\ \theta^G_{j,t} - \theta^G_{i,t} \end{bmatrix}. \tag{17}$$

where the $X^G_{j,t} = \{x^G_{j,t}, y^G_{j,t}, \theta^G_{j,t}\}$ is the pose of robot j in global coordinate G at time t.

In order to evaluate the robustness of the proposed algorithm. We have added different zero-mean Gaussian observation noises and odometry noises. The covariance matrix of the observation noises is $\Sigma_{obs} = \text{diag}\{\sigma^2_{obs}, \sigma^2_{obs}, \sigma^2_{obs}\}$ and the covariance matrix of the odometry noises is $\Sigma_{odom} = \text{diag}\{\sigma^2_{odom}, \sigma^2_{odom}, \sigma^2_{odom}\}$.

Simulation Results

We consider two different scenarios. One is that only minimum feature observations are available. One is sufficient feature observations are available. Table 1 shows the root mean square error (RMSE) results of the proposed algorithm and full non-linear least squares SLAM algorithm (NLLS) in the first scenario with different σ_{obs} and σ_{odom}. Table 2 shows the RMSE results of the proposed algorithm compared with NLLS in the second scenario under different observation noises. In both tables, the RMSE of robot pose and state includes the robot orientations and ignore the feature orientations. The tables show that the algorithm is robust to the noises. Fig. 3 shows the convergence of the NLLS and submap joining algorithms. Fig. 4 shows the estimation results of the two scenarios under one noise level.



(a) (b)

Figure 5: (a) The hardware setup of experiment robot. (b) The experimental environment with 2 Turtlebot2 and four landmarks.

Table 1: RMSE of the NLLS and Submap Joining Algorithms in Simulation with Minimum Observations

Method	RMSE (State)	RMSE (Robot Pose)	RMSE (Feature Position)	Noise
NLLS	0.0216	0.0224	0.0184	$\sigma_{obs} = 0.1$
Submap Joining	0.0494	0.0505	0.0448	$\sigma_{odom} = 0.2$
NLLS	0.1184	0.1202	0.1110	$\sigma_{obs} = 0.2$
Submap Joining	0.1405	0.1467	0.1157	$\sigma_{odom} = 0.3$
NLLS	0.1767	0.1940	0.1073	$\sigma_{obs} = 0.3$
Submap Joining	0.4998	0.5690	0.2228	$\sigma_{odom} = 0.4$

Table 2: RMSE of the NLLS and Submap Joining Algorithms in Simulation with Sufficient Observations

Method	RMSE (State)	RMSE (Robot Pose)	RMSE (Feature Position)	Noise
NLLS	0.0138	0.0160	0.0051	$\sigma_{obs} = 0.1$
Submap Joining	0.0170	0.0197	0.0065	$\sigma_{odom} = 0.2$
NLLS	0.0162	0.0200	0.0192	$\sigma_{obs} = 0.2$
Submap Joining	0.0217	0.0234	0.0149	$\sigma_{odom} = 0.3$
NLLS	0.0496	0.0536	0.0334	$\sigma_{obs} = 0.3$
Submap Joining	0.0564	0.0659	0.0183	$\sigma_{odom} = 0.4$

3.2 Real World Experiment

To further verify the performance of the proposed multi-robot SLAM algorithm, a real world experiment is adopted. We carried out an experiment with 2 robots and 4 features in a 5×5 m^2 area where the ground truth positions of robots and features can be obtained with motion capture system to verify the accuracy of our algorithm.

Experimental Setup

As shown in Fig. 5, for the real world experiments, we use Turtlbot2 ground robot with Kinect V1 RGB-D sensor to obtain odometry, relative observations between robots as well as observations of features. Each robot in the experiment is tagged with a $14 \times 14 \times 14$ cm^3 cube with Apriltag [Wang and Olson2016] on each sides together with motion capture tags. The landmarks are installed at fixed location who's positions can be obtained by motion capture system. In this experiment, we utilize motion capture system to obtain ground truth position of the robot to verify the accuracy of the proposed algorithm. For the data synchronization, we equipped each robot with a NUC micro computer which runs ROS operation system [Stanford Artificial Intelligence Laboratory et al.] with message filter to obtain and synchronize data from multiple robots and motion capture system.

Experimental Results

Table 3: Experiment Results.

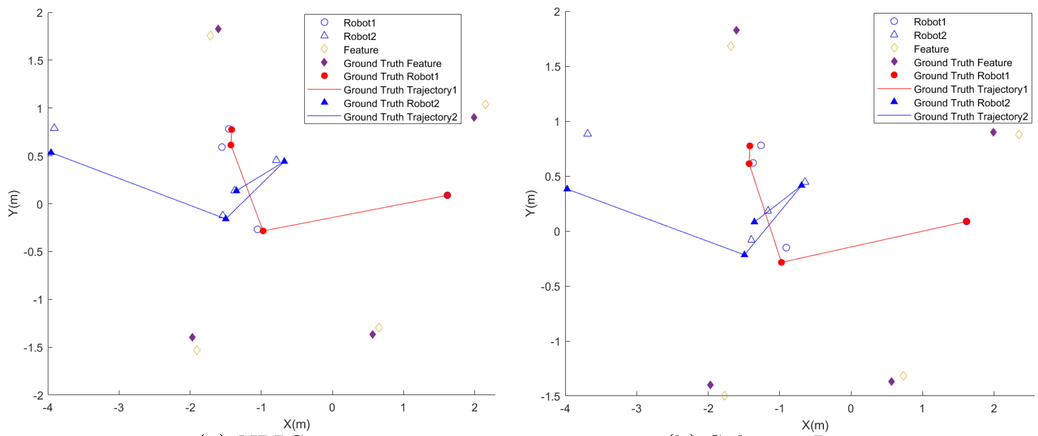
	Robot Position Error (m)	Feature Position Error (m)
NLLS	0.1107	0.0664
Map Joining	0.1309	0.0750

Fig. 6 shows the result from experimental data. As shown in Table 3, our approach can achieve similar accuracy compared with the non-linear least squares SLAM method. The converge of our approach and that of the NLLS method are shown in Fig. 7.

4 CONCLUSIONS

This paper proposed a novel submap joining based approach for distributed multi-robot SLAM. The local submap of each robot consists of the poses of this robot as well as the position of the observed features and the poses of the other robots that can be observed. The submaps are joining together treating each local submap as a virtual integrated observation. Simulation and experimental results show that the proposed distributed multi-robot can efficiently generate accurate localisation and mapping results.

In our future work, we will conduct larger-scale experiments with more robots and more landmarks. We will also extend the algorithm such that the robot can move



(a) NLLS (b) Submap Joining

Figure 6: The experiment results. The red circles, blue triangles and the purple diamonds represent the ground truth of the robot positions and feature positions respectively. The hollow circles and hollow triangles represent the estimated robot positions of robot 1 and robot 2. The hollow diamonds represent the estimated feature positions.

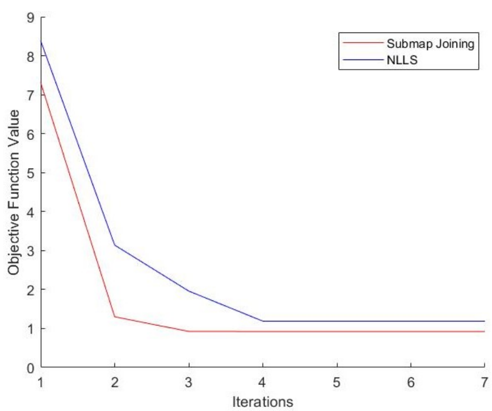


Figure 7: The objective function value of NLLS (blue line) and map joining (red line) in the real world experiment.

freely in the 3D space and different types of robots can be used.

References

[Andersson and Nygards, 2008] Lars AA Andersson and Jonas Nygards. C-sam: Multi-robot slam using square root information smoothing. In *2008 IEEE International Conference on Robotics and Automation*, pages 2798–2805. IEEE, 2008.

[Cadena *et al.*, 2016] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian D. Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32:1309–1332, 2016.

[Cunningham and Dellaert, 2012] Alexander G. Cunningham and Frank Dellaert. Large-scale experimental design for decentralized slam. In *Defense, Security, and Sensing*, 2012.

[Cunningham *et al.*, 2010] Alexander G. Cunningham, Balamanohar Paluri, and F. Dellaert. Ddf-sam: Fully distributed slam using constrained factor graphs. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3025–3030, 2010.

[Cunningham *et al.*, 2012] Alexander G. Cunningham, Kai M. Wurm, W. Burgard, and F. Dellaert. Fully distributed scalable smoothing and mapping with robust multi-robot data association. *2012 IEEE International Conference on Robotics and Automation*, pages 1093–1100, 2012.

[Huang *et al.*, 2008] Shoudong Huang, Zhan Wang, and G. Dissanayake. Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 24:1121–1130, 2008.

[Lazaro *et al.*, 2013] Maria Teresa Lazaro, Lina María Paz, Pedro Pinies, Jose A Castellanos, and Giorgio Grisetti. Multi-robot slam using condensed measurements. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1069–1076. IEEE, 2013.

[SajadSaeedi *et al.*, 2016] G. SajadSaeedi, Michael Trentini, Mae L. Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *J. Field Robotics*, 33:3–46, 2016.

[Stanford Artificial Intelligence Laboratory et al.,] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.

[Tian *et al.*, 2018] Yulun Tian, Katherine Liu, Kyel Ok, L. D. Tran, Danette Allen, Nicholas Roy, and Jonathan P. How. Search and rescue under the forest canopy using multiple uas. In *ISER*, 2018.

[Wang and Olson, 2016] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[Zhao *et al.*, 2013] Liang Zhao, Shoudong Huang, and G. Dissanayake. Linear slam: A linear solution to the feature-based and pose graph slam based on submap joining. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 24–30, 2013.

[Zhao *et al.*, 2014] Liang Zhao, Shoudong Huang, Lei Yan, and Gamini Dissanayake. A new feature parametrization for monocular slam using line features. *Robotica*, 33:513 – 536, 2014.