Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Multi-Robot SLAM with Topological/Metric Maps

H. Jacky Chang, C. S. George Lee*, Y. Charlie Hu and Yung-Hsiang Lu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-2035
{chang26, csglee, ychu, yunglu}@purdue.edu

***Abstract*— In recent years, the success of single-robot SLAM has led to more multi-robot SLAM (MR-SLAM) research. A team of robots with MR-SLAM can explore an environment more efficiently and reliably; however, MR-SLAM also raises many challenging problems, including map fusion, unknown robot poses and scalability issues. The first two problems can be considered as an optimization problem of finding a consistent joint map based on robots' relative poses and sensory data. This optimization problem exhibits a similar property of a single-robot topological/metric mapping. To exploit this property, we propose a multi-robot SLAM (MR-SLAM) algorithm, which builds a graph-like topological map with vertices representing local metric maps and edges describing relative positions of adjacent local maps. In this MR-SLAM algorithm, the map fusion between two robots can be naturally done by adding an edge that connects two topological maps, and the estimation of relative robot pose is simply performed by optimizing this edge. For the third scalable problem, the proposed algorithm is also scalable to the number of robots and the size of an environment. Computer simulations with a public data set and experimental work on Pioneer 3-DX robots have been conducted to validate the performance of the proposed MR-SLAM algorithm.**

***Index Terms*— Mobile robotics, simultaneous localization and mapping, multi-robot systems.**

I. Introduction

When a robot is exploring an unknown environment, it usually needs to obtain two important information – a map of the environment and the robot's location in the map. Since mapping and localization are related to each other, these two problems are usually considered as a single problem called simultaneous localization and mapping (SLAM). Most of SLAM studies focus on addressing challenging problems associated with a single robot, including data association, loop closure, and complex computations. To overcome these difficulties, researchers have utilized various probabilistic techniques [1] to solve the SLAM problem. With some success of these techniques, it is natural to extend the SLAM problem from a single-robot system to a multi-robot system.

Although it seems straightforward to implement a multi-robot SLAM (MR-SLAM) algorithm from an existing single-robot SLAM algorithm, several distinctive problems between them must be addressed: 1) unknown relative robot poses, 2) map fusion, and 3) scalability issues in the number of robots and the map size. To deal with these problems, we first investigate a scalable single-robot SLAM algorithm called topological/metric SLAM approach (TM-SLAM), which has been realized in various single-robot systems [2]–[4]. We then derive the Bayesian formulation of the proposed algorithm, which shows that it divides a large SLAM problem into smaller SLAM problems, each with localization problems. The results show that the TM-SLAM algorithm is indeed scalable in map size. To take the advantage of scalability in MR-SLAM, we deploy the TM-SLAM algorithm in each robot. In the proposed MR-SLAM with TM-SLAM algorithm (MRTM-SLAM), the map fusion becomes connecting appropriate topological maps. The connection process is simply done by adding a link between two topological places utilizing the observations of relative robot poses. Hence, the MRTM-SLAM is also scalable to the number of robots due to the simplicity of map fusion.

To implement MRTM-SLAM, we propose an algorithm that each robot builds a topological map along with local metric maps. The local metric maps are fixed-sized, occupancy-grid maps, which are constructed by a Rao-Blackwellized particle filter. For the topological map, it is a graph-like map consisting of vertices and edges. Each vertex represents a topological place (i.e., a location visited by the robot) and includes a local metric map. If a robot is travelling between two vertices, an edge is inserted to connect these two vertices. Meanwhile, the edges also store transformation matrices and uncertainties to describe the relationship between connected vertices.

This paper is organized as follows. In Section II, we briefly review the related work of multi-robot SLAM and topological/metric SLAM. In Section III, we shall briefly introduce the probabilistic formulation of the SLAM problem, and then derive the Bayesian formulation of a single-robot TM-SLAM and show that MR-SLAM has the similar problem structure to TM-SLAM. In Section IV, we describe the implementation of MRTM-SLAM. In Section V, experimental results are presented and discussed, and conclusions are summarized in Section VI.

II. Related Work

The goal of a multi-robot SLAM algorithm is to build a global joint map and localize the robots in the map. Simmons

This work was supported in part by the National Science Foundation under Grant IIS-0329061. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

*This material is based upon work supported by (while serving at) the National Science Foundation.

et al. proposed a multi-robot SLAM approach based on a likelihood maximization to find maps that are maximally consistent with the sensor data and odometry [5]. Thrun et al. presented a multi-robot SLAM algorithm with sparse-extended-information filters without the constraint of relative initial positions [6]. Howard proposed a manifold map structure, and a maximum-likelihood algorithm was employed to merge overlapped maps [7]. Howard also employed a multi-robot SLAM with a Rao-Blackwellized particle filter [8]. Ko et al. proposed a method to merge maps through an adaptive particle filter [9]. The particle filter estimates the position of a robot in the other robots' partial maps.

To mimic the way how a human memorizes a map, Kuipers proposed a graph-like map called topological map [10]. However, a topological map lacks the details of an environment. To solve these problems, a hybrid map was proposed [11], and submap-based approaches can be considered a close approach to the hybrid SLAMs. They both have detailed local maps, but the submap-based approaches do not maintain a topological structure of an environment. Leonard and Fender decoupled a SLAM problem, which discards the topology of the environment, by dividing an environment into equal-sized submaps [12]. A more appropriate submap approach to our topological/metric algorithm is the relative submaps approach proposed by Chong et. al [2], and was improved by Williams with a constrained relative submap filter (CRSF) [3]. Later, Bosse et al. proposed an Atlas Framework for scalable single-robot mapping [4].

III. Multi-Robot SLAM with Topological/Metric Maps

A. The SLAM problem

Consider the sets of robot locations $X_k = \{x_1, x_2, \ldots, x_k\}$, motion commands $U_k = \{u_1, u_2, \ldots, u_k\}$, and observations $Z_k = \{z_1, z_2, \ldots, z_k\}$, where k is the discrete-time index, x_k is the robot pose, u_k is the motion command, and z_k is the observation, all at the k^{th} time instant. Given the motion commands U_k and the observations Z_k, the SLAM problem is to calculate the distribution $P(x_k, M|Z_k, U_k, x_0)$, where M is the map. The SLAM problem can be further derived as

$$P(x_k, M|Z_k, U_k, x_0) = \kappa \times P(z_k|x_k, M) \times \Lambda \quad (1)$$

where κ is a normalizing constant and

$$\Lambda = \int P(x_k|x_{k-1}, u_k) \times P(x_{k-1}, M|Z_{k-1}, U_{k-1}, x_0) dx_{k-1}.$$

If we assume all the variables are Gaussian distributed, we can utilize a Kalman Filter (KF) or an Extended Kalman Filter (EKF) to realize Eq. (1).

To show the computational burden in a KF/EKF, we assume that M contains landmarks labeled $l_1, \ldots, l_i$, where i is the landmark index. For simplicity, only the first four steps are shown in Fig. 1(a), where the robot observes landmarks l_1 and l_2 at x_1 and so on. If we perform the marginalization [13], then the Markov Random Field (MRF) becomes a fully connected graph (i.e., a clique) as shown in Fig. 1(b). Because each update operation needs to modify every vertex and edge, the update time is $O(n^2)$, where n is the number of landmarks.

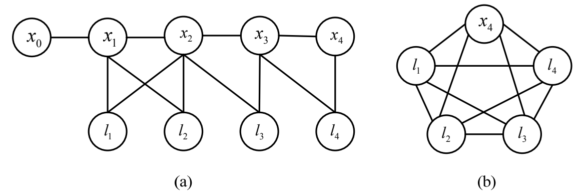


Fig. 1. (a) The MRF at first four time steps with four landmarks, l_1, l_2, l_3 and l_4. (b) The MRF after performing marginalization.

B. Bayesian Formulation of Topological/Metric SLAM

One way to avoid a fully connected MRF is to reserve all robot poses X_k. This approach keeps the original MRF structure in Fig. 1(a). This approach is called GraphSLAM [14] or smoothing SLAM [15] since it builds a map similar to building a graph incrementally and smoothing the whole robot trajectory. This results in changing the computational dominator from the number of landmarks to the number of recorded robot poses in the trajectory. Since a robot only observes a limited number of landmarks at each pose, the computation cost is $O(n_p)$, where n_p is the number of stored robot poses. In fact, one can consider that smoothing SLAM and EKF-SLAM are two extreme SLAM algorithms. Their efficiencies depend on the size of the environment (i.e., the number of landmarks) or the run-time of the robot (i.e., the number of stored robot poses).

The TM-SLAM is a method between EKF-SLAM and smoothing SLAM. In a TM-SLAM algorithm, the SLAM problem is divided into several limited-sized SLAM problems along with localizing topological places (i.e., some selected robot poses). The limited-sized SLAM problem can be handled by a traditional SLAM algorithm such as EKF-SLAM or a particle-filter SLAM, and the localization of topological places can be considered as smoothing some selected robot poses in a robot trajectory. To show this structure, let us denote M_i as a local metric map, where i is the index, $M = \{M_1, M_2, \ldots, M_N\}$, and N is the number of local maps; ϕ_i denotes the topological place of M_i. The topological map Φ is a set of topological places $\Phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$. The sensory measurements and commands are combined as $\Psi = \{U_k, Z_k, x_0\}$. We also define $\mathbf{X} = \{X_k\} = \{\bar{X}_1, ..., \bar{X}_i, ...\bar{X}_N\}$ and$\Psi = \{Z_k, U_k, x_0\} = \{\bar{\Psi}_1, ...\bar{\Psi}_i, ..., \bar{\Psi}_N\}$ where $\bar{X}_i$ and $\bar{\Psi}_i$ are the corresponding trajectories and measurements when the robot builds M_i. Now, we write the Bayesian formulation of TM-SLAM as $P(X_k, \Phi, M|\Psi)$. If we factor out a local metric map M_i, the topological places ϕ_i and the trajectory $\bar{X}_i$, then we obtain

$$P(M_i, \bar{X}_i, \phi_i|\Psi, \hat{M}_i, \hat{\mathbf{X}}_i, \hat{\Phi}_i) P(\hat{M}_i, \hat{\mathbf{X}}_i, \hat{\Phi}_i|\Psi) \quad (2)$$

where $\hat{\mathbf{X}}_i = \mathbf{X} \backslash \bar{X}_i$, $\hat{\Phi}_i = \Phi \backslash \phi_i$ and $\hat{M}_i = M \backslash M_i$. The set operator "$\backslash$" on $\mathbf{X}$ and $\bar{X}_i$ is defined as $\mathbf{X} \backslash \bar{X}_i = \{x :$

$x \in \mathbf{X}$ and $x \notin \bar{X}_i\}$. The first term in Eq. (2) describes the local metric SLAM and its topological place. Because ϕ_i is determined by the relations to its adjacent maps, adjacent topological places and sensory data $\bar{\Psi}_i$, we can further derive this term by factorizing out ϕ_i as

$$\begin{aligned} &P(M_i, \bar{X}_i, \phi_i | \Psi, \hat{M}_i, \hat{\mathbf{X}}_i, \hat{\Phi}_i) \\ &= P(M_i, \bar{X}_i | \bar{\Psi}_i, \phi_i) P(\phi_i | \bar{\Psi}_i, \Phi_{\text{adj}(M_i)}, M_{\text{adj}(M_i)}) \\ &= P(M_i, \bar{X}_i | \bar{\Psi}_i, \phi_i) \prod_{\alpha \in \text{adj}(M_i)} P(T_{\{i,\alpha\}} | \phi_\alpha, M_\alpha, \bar{\Psi}_i) \end{aligned} \tag{3}$$

where $M_{\text{adj}(M_i)}$ are the adjacent maps of M_i, $\{\text{adj}(M_i)\}$ is the index set of adjacent maps of M_i, where $T_{\{i,\alpha\}}$ is the transformation matrices between the coordinate frame i and α. From Eq. (3), we observe that the local SLAM is partitioned into two parts: a traditional SLAM problem and a localization problem of localizing topological place in the adjacent maps. Substituting Eq. (3) into Eq. (2), we obtain

$$\begin{aligned} &P(\mathbf{X}, \Phi, M | \Psi) \\ &= \prod_{i=1}^{N} \underbrace{P(M_i, \bar{X}_i | \bar{\Psi}_i, \phi_i)}_{\text{SLAM}} \underbrace{P(\phi_i | \bar{\Psi}_i, \Phi_{\text{adj}(M_i)}, M_{\text{adj}(M_i)})}_{\text{Localization}}. \end{aligned} \tag{4}$$

From Eq. (4), it is clear that a TM-SLAM is a combination of SLAMs and localization processes. The MRF of topological/metric SLAM is shown in Fig. 2. The TM-SLAM decouples the graph into smaller connected cliques and chained by selected topological places.

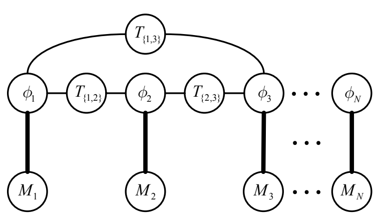


Fig. 2. The MRF of a single-robot TM-SLAM. A bold edge indicates full connections. For example, ϕ_1 is fully connected to every element in M_1 and therefore, $\{\phi_1, M_1\}$ forms a clique. The cliques are connected to their adjacent cliques through transformation vertices such as $T_{\{1,2\}}$, $T_{\{1,3\}}$ and $T_{\{2,3\}}$.

C. Bayesian Formulation of Multi-Robot SLAM

In the TM-SLAM structure, a new edge is inserted when a robot builds a new local map or re-visits an explored area. For a multi-robot SLAM with topological/metric maps, a new edge can also be created when robots meet with each other. Therefore, when we derive the Bayesian formulation of a multi-robot SLAM with topological/metric maps, we shall focus on the relative pose estimation since it is the major difference between a TM-SLAM and an MR-SLAM. For a two-robot case, the probabilistic form of a general multi-robot SLAM problem is

$$P({}^1M, {}^2M, {}^1X_k, {}^2X_k, {}^{\{1,2\}}R | {}^1\Psi, {}^2\Psi, {}^{\{1,2\}}\Delta) \tag{5}$$

where ${}^{\{1,2\}}R$ is the relative pose, the left superscript indicates the robot index, and Δ is a set of sensory measurements of relative poses. By factoring out ${}^{\{1,2\}}R$, we obtain

$$\begin{aligned} P({}^{\{1,2\}}R | {}^1\Psi, {}^2\Psi, {}^{\{1,2\}}\Delta, {}^1M, {}^2M, {}^1X_k, {}^2X_k) \times \\ P({}^1M, {}^2M, {}^1X_k, {}^2X_k | {}^1\Psi, {}^2\Psi, {}^{\{1,2\}}\Delta). \end{aligned} \tag{6}$$

The second term is factored into two independent SLAMs as

$$\underbrace{P({}^1M, {}^1X_k | {}^1\Psi, {}^{\{1,2\}}\Delta)}_{\text{SLAM for robot 1}} \underbrace{P({}^2M, {}^2X_k | {}^2\Psi, {}^{\{1,2\}}\Delta)}_{\text{SLAM for robot 2}}. \tag{7}$$

The first term in Eq. (6) represents the estimation of relative pose. The relative pose, ${}^{\{1,2\}}R$, can also be described by two transformation matrices as $\{{}^2_1\mathbf{R}, {}^1_2\mathbf{R}\}$, where ${}^2_1\mathbf{R}$ indicates a coordinate frame 2 is expressed with respect to a coordinate frame 1. For example, the pose of robot 2 expressed in the coordinate frame of robot 2 is written as ${}^2_2x_k$ and is transformed to the coordinate frame of robot 1 as

$${}^2_1x_k = {}^2_1\mathbf{R}\ {}^2_2x_k.$$

We rewrite the relative pose estimation as

$$\begin{aligned} &P({}^2_1\mathbf{R}, {}^1_2\mathbf{R} | {}^1\Psi, {}^2\Psi, {}^{\{1,2\}}\Delta, {}^1M, {}^2M, {}^1X_k, {}^2X_k) \\ &\approx \eta' P({}^2_1\mathbf{R} | {}^2\Psi, {}^2\Delta, {}^2X_k, {}^1M) P({}^1_2\mathbf{R} | {}^1\Psi, {}^1\Delta, {}^1X_k, {}^2M). \end{aligned} \tag{8}$$

$P({}^2_1\mathbf{R} | {}^2\Psi, {}^2\Delta, {}^2X_k, {}^1M)$ can be further approximated by

$$\eta'' \underbrace{P({}^2_1\mathbf{R} | {}^2\Psi, {}^1M)}_{\text{Localization}} \underbrace{P({}^2_1\mathbf{R} | {}^2\Delta, {}^2X_k)}_{\text{Direct Sensing Model}} \tag{9}$$

where η' and η'' are normalizing constants. In this form, we observe that the relative pose ${}^2_1\mathbf{R}$ is estimated from a direct sensing model and a localization process. Comparing with Eq. (4), a multi-robot SLAM is also a combination of SLAMs and localization processes, but we have an extra term from the direct sensing model. Therefore, if every robot has a topological/metric map structure, the map fusion will become extremely simple by connecting topological vertices and adding an edge by Eqs. (8) and (9).

IV. Implementation of the MRTM-SLAM

A. Map Structure

The topological map is a graph-like map consisting of vertices and edges. Each vertex represents a topological location, which has been visited by a robot, and the vertex also includes a metric map with its own coordinate frame. If two vertices are connected by an edge, it means that a robot can traverse between these two topological places. Since each vertex has its own coordinate frame, a connecting edge consists of two transformation matrices to describe the relationship of these two connected vertices. Also, the transformation matrix accompanies with a covariance matrix representing the uncertainty of the transformation.

For local metric maps, we adopt the occupancy-grid map representation that preserves the details of an environment. The shape of a local map is a fixed-size square with width $4d_s$, where d_s is the sensing range. When a robot creates a new metric map, the initial location of the robot is set to the center of the square. When a robot moves in the square with width $2d_s$ in the local map, all sensing measurements are preserved and processed. If the robot leaves the square with width $2d_s$, we say that it leaves the current metric map (vertex) and enters another metric map (vertex). To build local metric maps, we utilize a Rao-Blackwellized particle filter (RBPF) [16], which works well with the occupancy-grid map representation [17].

B. The MRTM-SLAM Algorithm

Since we do not have a centralized structure, each robot has to perform the MRTM-SLAM algorithm. The proposed multi-robot SLAM algorithm is described below.

MRTM-SLAM Algorithm: For a robot with index i, given sensory measurements ${}^i\Psi$ and relative pose measurements ${}^i\Delta$, the MRTM-SLAM algorithm builds a topological map, local metric maps and a jointed global map. The single-robot TM-SLAM algorithm is described in Steps **M3-M8**. In Steps **M9-M11**, the map fusion procedures are performed when robot i meets with other robots. When there is no further sensory measurements can be processed (i.e., the exploration task is finished), an optimization process (Step **M13**) is performed to retrieve a global map. All the variables in this algorithm are associated with robot i except Steps **M9-M11**, where the associated robot indices will be specified.

M1. [*Initialization.*] Create a new vertex V_0. Set time index $k \leftarrow 0$, where $\leftarrow$ stands for "assign with." Set current map index number $cv \leftarrow 0$. Set number of vertices $N_v \leftarrow 1$. Create and initialize a new local metric map M_{cv}. Initialize current robot position to $[\mathbf{0}]^T$, where superscript T indicates matrix transpose. Set new edge flag $FL_{new} \leftarrow$ true. Set the re-enter previous-map flag $FL_{re-enter} \leftarrow$ false.

M2. [*Data query.*] Obtain sensory data Ψ_k and Δ_k at time k.

M3. [*Build local metric map.*] If $FL_{re-enter} =$ true, generate a set of particles with variance Σ_{ini} and set $FL_{re-enter} \leftarrow$ false.
Given sensory measurements Ψ_k at time k, execute RBPF-SLAM algorithm to build local metric map M_{cv}, and obtain current robot pose x_k^{cv} and its variance Σ_k^{cv}, which is approximated by calculating the variance from particles.

M4. [*Determination of leaving current local map.*] Check if the robot stays in current local metric map.
IF current robot pose x_k^{cv} remains in map M_{cv} and inside the boundary $2d_s$, THEN go to Step **M9**.

M5. [*Edge update.*] If the edge $E_{cv,pv}$ is not a first-time created edge, the edge is updated through a covariance intersection where the index pv is a vertex index that the robot travels from V_{pv} to current vertex V_{cv}. The edge update is performed when a robot is leaving V_{cv} because the robot is well localized in V_{cv} and thus, it is the best opportunity to update $E_{cv,pv}$.
IF $FL_{new} =$ true, THEN go to Step **M6**, ELSE perform a covariance intersection.

$$\begin{aligned}(\Sigma_{cv,pv})^{-1} &\leftarrow \omega(\Sigma_{ini}^{cv})^{-1} + (1-\omega)(\Sigma_{cv,pv})^{-1} \\ T_{cv}^{pv} &\leftarrow \Sigma_{cv,pv}[\omega(\Sigma_{ini}^{cv})^{-1}T_{cv}^{pv,new} \\ &\qquad +(1-\omega)(\Sigma_{cv,pv})^{-1}T_{cv}^{pv}]\end{aligned}$$

where $T_{cv}^{pv,new}$ is a new transformation matrix obtained from the localized entering robot pose in V_{cv} and the leaving robot pose in V_{pv}. Σ_{ini}^{cv} is the covariance matrix of leaving robot pose with respect to the coordinate frame of V_{cv}. For simplicity, we set ω to 0.5.

M6. [*Determination of re-entering a built map.*] The current robot pose x_k^{cv} is projected to all other vertices, and then the projected poses are examined if they are in any previous built maps. If there are more than one projected poses that can describe current robot pose, we will only select the best one. This procedure consists of two processes: 1) find the projection pose for all vertices, and 2) find the best projected robot pose if it exists.
M6a. For each vertex V_n, $n = 1, ..., N_v$, perform Dijkstra's shortest path algorithm to find the shortest path $Ph(V_{cv}, V_n)$. Calculate the projection matrix $T_{Ph(V_{cv},V_n)}$ by concatenating the transformation matrices along $Ph(V_{cv}, V_n)$. Calculate the projected robot pose $x_k^n = T_{Ph(V_{cv},V_n)} \times x_k^{cv}$. Similarly, calculate the Jacobian matrix $JT_{Ph(V_{cv},V_n)}$ of $T_{Ph(V_{cv},V_n)}$. Project current robot pose's uncertainty with respect to vertex V_n's coordinate frame as $\Sigma_k^n = JT_{Ph(V_{cv},V_n)} \times \Sigma_k^{cv} \times [JT_{Ph(V_{cv},V_n)}]^T + \Sigma_{Ph(V_{cv},V_n)}$, where $\Sigma_{Ph(V_{cv},V_n)}$ is the accumulated uncertainty of transformation matrices along the path. Set $pv \leftarrow cv$.
M6b. For each projected pose x_k^n, $n = 1, ..., N_v$, if x_k^n is inside the local metric map M_n and within boundary $2d_s$, store x_k^n in a candidate set X_{cs}.
IF X_{cs} is not empty, THEN select $x_k^{V_{bv}}$ that $\det(\Sigma^{bv})$ is minimum in X_{cs} where bv is the index of the selected vertex.
IF $x_k^{V_{bv}}$ exists, THEN set $FL_{re-enter} \leftarrow$ true, set $\Sigma_{ini} \leftarrow \Sigma_k^{bv}$ and set $FL_{new} \leftarrow$ false.
IF $x_k^{V_{bv}}$ exists and there is no edge between V_{bv} and V_{cv}, THEN go to Step **M8**.
ELSE go to Step **M9**.

M7. [*Vertex creation.*] A new vertex is created when a robot leaves current metric map and explores a new area. The new vertex creation procedure consists of two processes: 1) initialize a new vertex and a new local metric map, and 2) insert an edge between V_{cv} and the new created vertex.
M7a. Set $N_v \leftarrow N_v + 1$. Create a new vertex V_{N_v}. Create and initialize a new local metric map M_{N_v}. Initialize current robot position to $[\mathbf{0}]^T$.
M7b. Insert an edge E_{cv,N_v} between V_{cv} and V_{N_v}. The transformation matrix on the edge is obtained from $T_{N_v}^{cv} \leftarrow f(x_k^{cv}, [\mathbf{0}])$ where $f(\cdot)$ is a function to obtain a transformation matrix from a robot pose with respect to two coordinate frames. Because the uncertainty of the

transformation comes from x_k^{cv}, the variance $\Sigma_{N_v,cv}$ is set to Σ_k^{cv}. Set current map index number $cv \leftarrow N_v$. Set $FL_{new} \leftarrow$ true. Go to Step **M9**.

M8. [*Loop edge insertion.*] When the robot enters a non-adjacent vertex V_{bv} from V_{cv}, then a loop edge $E_{cv,bv}$ is inserted between V_{cv} and V_{bv}.
Set $T_{bv}^{cv} \leftarrow T_{Ph(V_{cv},V_{bv})}$. Set $\Sigma_{bv,cv} \leftarrow \Sigma_k^{bv}$. Set current robot pose to x_k^{bv}. Set $cv \leftarrow bv$. Set $FL_{new} \leftarrow$ true.

M9. [*Robot detection in vicinity* .] `IF` ${}^i\Delta_k$ is empty, `THEN` go to Step **M12**, `ELSE` create a set RV containing all robot indices in current vicinity, and go to Step **M10**.

M10. [*Map exchanging process.*] Exchange topological and local metric maps with robots where indices are in RV. The exchanged maps are the maps that have been updated recently or have never been exchanged before.

M11. [*Fusing edge insertion.*] When a robot observes other robots, a fusing edge is inserted between two exchanged topological maps. For all $j \in RS$, if vertex ${}^iV_{cv}$ and vertex ${}^jV_{cv}$ is not connected, add an edge $E_{{}^jcv,{}^icv}$. Set $T_{{}^jcv}^{{}^icv} \leftarrow f({}^ix_k^{cv}, {}^jx_k^{cv})$ and set $\Sigma_{{}^jcv,{}^icv} \leftarrow \Sigma_{ob}$ where the covariance matrix Σ_{ob} is obtained from the sensing model. Note that this process only applies the direct sensing model in Eq. (9), and finding optimized transformation matrices (i.e., localization process) is performed in Step **M13**.

M12. [*Termination of on-line SLAM process.*] While there remains sensory data to be processed, go to Step **M2** and set $k \leftarrow k+1$; otherwise go to Step **M13**.

M13. [*Final global-map retrieving procedure.*] After the exploration task is done, retrieve an optimized topological map and a global metric map. The global metric map is fused from all local maps with respect to one coordinate frame.
Set initial vertex V_0 as the root and its coordinate frame as the global coordinate frame. From V_0, construct a tree by using Dijkstra's algorithm, which gives the shortest path from the root to all vertices. Use this tree as an initial solution of a global fused map.
In local metric maps, there are overlapped regions between connected vertices; utilize these regions to align adjacent metric maps and the transformation matrices. This alignment is considered as an optimization problem as follow

$$\begin{aligned}\hat{\mathbf{T}} &= \arg\max_{\mathbf{T}} \prod_{m,n} P(M_m, M_n, \phi_n | \phi_m, T_{ini}) \\ &= \arg\max_{\mathbf{T}} \sum_{m,n} L(M_m, M_n, \phi_n | \phi_m, T_{ini})\end{aligned}$$

where m, n are vertex indices in the topological maps, $\mathbf{T}$ is a set of all transformation matrices $\forall T_{\{m,n\}} \in \mathbf{T}$, T_{ini} is the initial transformation matrices extracted from the tree, and $L(\cdot)$ is the log-probability form, which is used in the occupancy-grid maps. The optimization process is an Expect Maximization method described in [1]. Utilizing the optimized $\hat{\mathbf{T}}$, fuse all metric maps in the global coordinate frame and obtain a global metric map.

END MRTM-SLAM Algorithm.

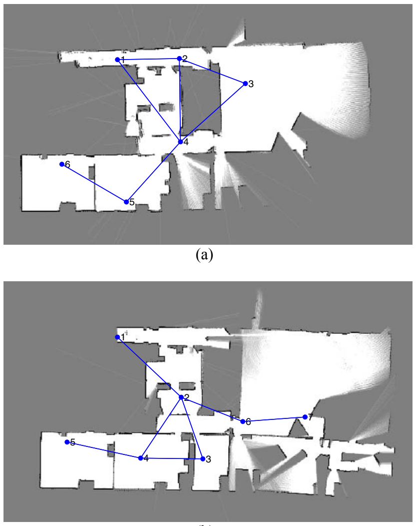
V. Experimental Results

Two experiments were conducted to verify the performance of the proposed MRTM-SLAM algorithm. The first experiment was performed by running data sets downloaded from the Robotics Data Set Repository (Radish) [18], and the second experiment was conducted on the first floor of MSEE building at Purdue University. In these experiments, Pioneer 3-DX robots equipped with a SICK LMS-200 laser ranger were used. The effective sensing range is set to 10m, and the size of a local metric map is $20\text{m} \times 20\text{m}$. To identify robots and estimate relative robot poses, a retro-reflective tag was attached to every robot. The number of particles in the RBPF is set to 40 for local metric-map building. In all experiments, most of the on-line computation power is used for local metric map building, which is performed by a RBPF. Since the number of particles and the size of the map are small, each RBPF iteration is about 4 times faster than a RBPF with 200 particles that is usually used in RBPF-based SLAMs. Moreover, the map fusion on a topological map is also very efficient because adding an edge is an $O(1)$ operation.

The first experiment was a public data set of 45 minutes runtime. The test environment was the Fort AP Hill. The mapping result of robot 1 and robot 2 are shown in Fig. 3(a) and 3(b), respectively. The final fused map before global optimization is shown in Fig. 4(a). The optimization process is done off-line in MATLAB and took about 12 seconds, and the optimized map is shown in Fig. 4(b). During the fusion process, we have several vertices describing the same areas. For example, vertex 6 in Fig. 3(a) and vertex 5 in Fig. 3(b) are two vertices representing the bottom-left room. Both vertices contain high-quality local metric maps as shown in Fig. 5. Hence, the map fusion process does not bring in too much benefits in the overlapped areas. In fact, the major advantage of the map fusion is in the non-overlapped area, which is the bottom right of the map. In addition, the bottom-right local-metric maps were improved and aligned after the global optimization process. The second experiment were performed with three robots. The final fused maps are shown in Fig. 6. In these experiments, we coordinated the robots to explore different areas, and the overlapped areas are limited. Hence, we do not have different topologies of the same area, and the topological maps are correctly representing the topologies of the environment.

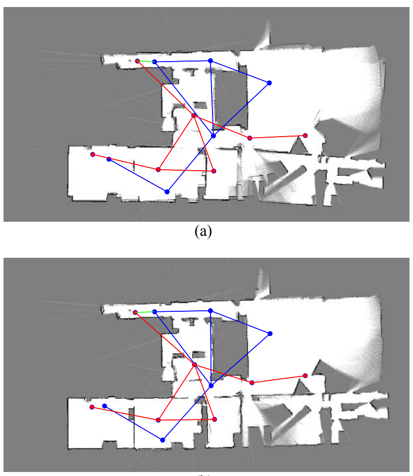
VI. Conclusions

In this paper, we have proposed a multi-robot SLAM algorithm with topological/metric maps. With the proposed algorithm, a team of robots can explore an environment and build a consistent joint map. In addition, we have derived its Bayesian formulation, which indicates that a SLAM problem can be decoupled into several smaller SLAM problems along with their localization problems. This decoupled structure allows us to design a scalable multi-robot SLAM. The proposed algorithm also has some minor limitations. Since



(a) (b)

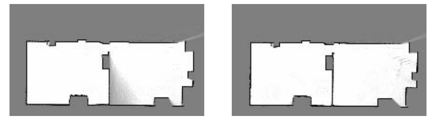
Fig. 3. (a) The first robot mapping result. (b) The second robot mapping result.



(a) (b)

Fig. 4. (a) The fused map before global optimization from robot 1 and robot 2. (b) The map after global optimization.

we perform off-line optimization in topological maps, in a large exploration task, the robots sometimes may need to stop the exploration and perform optimization process before they resume the exploration task. Thus, one of the important future work is to develop an efficient algorithm for on-line topological map optimization.



(a) (b)

Fig. 5. (a) The local metric map of vertex 6 in robot 1. (b) The local metric map of vertex 5 in robot 2.

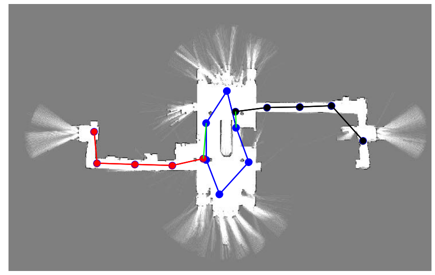


Fig. 6. Mapping result of the first floor of MSEE building. Three robots construct three topological maps in red, blue and black.

REFERENCES

[1] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[2] K. Chong and L. Kleeman, "Feature-based mapping in real, large scale environments using an ultrasonic array," *International Journal Robotics Research*, vol. 18, pp. 3–19, Jan. 1999.

[3] S. Williams, "Efficient solutions to autonomous mapping and navigation problems," Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, 2001.

[4] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1899–1906.

[5] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *Proc. of the AAAI National Conference on Artificial Intelligence*. Austin, TX: AAAI, 2000, pp. 852–858.

[6] S. Thrun and Y. Liu, "Multi-robot SLAM with sparse extended information filers," in *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*, 2003.

[7] A. Howard, "Multi-robot mapping using manifold representations," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2004, pp. 4198–4203.

[8] ——, "Multi-robot simultaneous localization and mapping using particle filters," in *Proc. of Robotics: Science and Systems*, Cambridge, USA, June 2005, pp. 201–208.

[9] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, "A practical, decision-theoretic approach to multi-robot mapping and exploration," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003)*, 2003, pp. 3232–3238.

[10] B. Kuipers, "Modeling spatial knowledge," in *International Joint Conference on Artificial Intelligence*, 1978, pp. 292–298.

[11] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Journal of Robotics and Autonomous Systems*, vol. 8, pp. 47–63, 1991.

[12] J. Leonard and H. Feder, "Decoupled stochastic mapping," *IEEE J. of Ocean Engineering*, vol. 26, pp. 561–571, Oct. 2001.

[13] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter, *Probabilistic Networks and Expert Systems*. Springer, 1999.

[14] S. Thrun and M. Montemerlo, "The graph slam algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[15] F. Dellaert and M. Kaess, "Square root sam: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[16] K. P. Murphy, "Bayesian map learning in dynamic environments," in *Advances in Neural Information Processing Systems*. MIT Press, 2000, pp. 1015–1021.

[17] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *IEEE International Conference on Robotics and Automation*, April 2005, pp. 2443–2448.

[18] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: http://radish.sourceforge.net/