Research Article

A Fast Map Merging Algorithm in the Field of Multirobot SLAM

Yanli Liu,[1,2] Xiaoping Fan,[1] and Heng Zhang[2]

[1] *School of Information Science and Engineering, Central South University, Changsha 410075, China*
[2] *School of Information Engineering, East China Jiaotong University, Nanchang 330013, China*

Correspondence should be addressed to Heng Zhang; hbzhangheng@126.com

In recent years, the research on single-robot simultaneous localization and mapping (SLAM) has made a great success. However, multirobot SLAM faces many challenging problems, including unknown robot poses, unshared map, and unstable communication. In this paper, a map merging algorithm based on virtual robot motion is proposed for multi-robot SLAM. The thinning algorithm is used to construct the skeleton of the grid map's empty area, and a mobile robot is simulated in one map. The simulated data is used as information sources in the other map to do partial map Monte Carlo localization; if localization succeeds, the relative pose hypotheses between the two maps can be computed easily. We verify these hypotheses using the rendezvous technique and use them as initial values to optimize the estimation by a heuristic random search algorithm.

1. Introduction

Recent advances in mobile robotics have allowed autonomous systems to be involved in many successful applications including planetary exploration, search and rescue, surveillance, and other service scenarios [1].

For the purpose of successfully accomplishing a generic task, a main prerequisite for a mobile robot, deployed in an unknown area, is the ability to autonomously navigate, exploiting the information acquired through the joint estimation of its positions and a model of the surrounding environment. The problem of estimating both the robot pose and the environment representation is usually defined as simultaneous localization and mapping (SLAM) [2, 3]. While the maturity of SLAM in single robot scenarios is recognized in many recent works, a challenging issue is to extend these approaches to multirobot scenarios [4–7].

Recently, in order to cope with dynamic environment and multiple tasks, a lot of researches have been presented in multiagent and multiple distributed autonomous robotic systems. Each robot has its own task, such as building a map of local position. Moreover, they have to combine their data into shared maps. Using shared maps, robots coordinate their exploration strategies to maximize the efficiency of exploration. Multiple autonomous mobile robots can complete the task through cooperation and give a more accurate map by data fusion. The robots coordinate their exploration strategies to maximize the efficiency of their exploration using these shared maps.

Fenwick et al. extended single robot SLAM algorithm based on EKF to the multirobot [8–10]. Howard extended single-robot SLAM algorithm based on particle filter to multi-robot applications [11]. Thrun put forward multi-robot hybrid map building method that combines fast maximum likelihood map growing with a Monte Carlo localizer [12]. Thrun [10] points out that the key to multi-robot map building is to determine the initial position and orientation of each single robot. However, most current algorithms bypass the problem, either some algorithms assume that the road signs in the environment are identified, or some algorithms assume that each robot starting from a similar position, so that their initial map has large coverage area. Ko et al. [13] specialize in the methods by which the robots determine relative pose in environmental exploration. Each robot builds its own maps, and meanwhile, estimates the pose of other robots in its own map. If the estimation probability reaches the higher accuracy, two robots rendezvous [14] to confirm the pose between them, and this paper also adopts a similar approach. Another strategy is to assume that the robots can be cross-observed to determine the relative pose between them

[13, 15]. The deficiency of the method is that each robot in the process of movement must be "accidentally" encounter.

In this paper we present a novel fast and accurate algorithm for merging multiple maps represented as occupancy grids. The main idea is as follows: make a virtual mobile robot in one partial map and control its move and measure in the map. At the same time, these simulated data are used as information sources in the other map to do partial map Monte Carlo localization [16]; if localization succeeds, the relative pose hypotheses between the two maps can be computed easily. In order to speed up the process, we can constrain the virtual robot move along the local topological edge of the partial map, so in the process of Monte Carlo localization within the partial map, the initial random samples only are generated near the edges of the corresponding topology of the other partial map. Prior knowledge of the relative pose between maps or robots is not demanded in the method, which enhances autonomous exploration and multi-robot SLAM.

2. Concept of Map Merging

The research on map merging is limited, and the previous work focused on feature-based approaches [17–19], which rely on special landmarks that can be recognized through properly processing the data gathered by the robots. For example, the topological maps are graphs in which vertices represent recognizable places such as doorways, corridors, and different types of corner and edges represent the passage connecting two places. Merging map A and map B turns into searching subgraphs with the same structure in map A and map B. Our goal is to merge maps which are not based on features, but rather on occupancy grids. For sake of clearness, we formally define the several concepts [20].

Definition 1. Let N and M be two positive real numbers. A $N \times M$ map is a function. Consider

$$m : [1, N] \times [1, M] \longrightarrow \mathbf{R}, \tag{1}$$

where $I_{N \times M}$ denoting the set of $N \times M$ maps.

Discrimination is needed when a map is processed in the practical application, and the map is represented as a matrix with N rows and M columns. The function m is a confidence model in the map. For example, one could assume that if the value of $m(x, y)$ is positive, the point (x, y) in the map is free, while a negative value represents the point (x, y) as occupied. The absolute value represents the degree of belief. Especially, we assume that if the value of $m(x, y)$ is zero, the point (x, y) is unknown. The definition is consistent with the grid map. In the following, the planar transformation is defined, which will be used to find the best matching between two partial maps.

Definition 2. t_x, t_y, and t_θ are three real numbers; the pose transformation function consists of t_x, t_y, and t_θ is defined as follows:

$$T_{(t_x, t_y, t_\theta)}(x, y) : \mathbf{R}^2 \longrightarrow \mathbf{R}^2, \tag{2}$$

where t_x, t_y, and t_θ are three real numbers. It is defined as follows:

$$T_{(t_x, t_y, \theta)}(x, y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{3}$$

Definition 3. m_1 and m_2 are the two maps of $I_{N \times M}$; the coverage between m_1 and m_1 is defined as follows:

$$\omega(m_1, m_2) = \sum_{i=1}^{N} \sum_{j=1}^{M} e(m_1[i, j], m_2[i, j]), \tag{4}$$

where

$$e(a, b) = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The coverage function is used to measure the degree of the match between two maps. Ideally, the map, which the two robots build, is entirely consistent with the physical environment and covers the entire environment space. In this case, the coverage function is $\omega(m_1, m_2) = N \times M$; it is maximum.

According to the above definitions, we can formally make definitions on map merging.

Definition 4. Given two maps m_1 and m_2, pose transformation function $T_{(t_x, t_y, t_\theta)}(x, y)$ is searched to maximize the function $\omega(m_1, T_{(t_x, t_y, t_\theta)} m_2)$.

From the previous definition, the key to map merging is to search the optimal pose transformation $T_{(t_x, t_y, t_\theta)}$. The preliminary estimations of the transformation are got by the Monte Carlo localization in the partial map [16]. In order to improve computational efficiency, we adopt the sampling method with confined area. The heuristic random search algorithm is used to further optimize the pose transformation.

3. Fast Monte Carlo Global Localization

3.1. Topological Information of a Map. The topological information of a map is the abstraction of environmental information. Vertices represent discrete places in the environment, and edges represent paths connecting these places. The information can be obtained by generalized Voronoi graph (GVG) algorithm [21] or thinning algorithm [22, 23]. Useless boundaries and virtual intersections for navigation are generated in GVG algorithm, the thinning method does not generate this information because it is based on probabilistic framework. It is robust for the sensor noise and various environments.

Thinning algorithm is the method that extracts the image skeleton in the digital image processing algorithm. Figure 1 illustrates the concept of thinning. The object on the left figure can be described well by the structure composed of connected lines (i.e., "T" shape design with thin lines on the right figure). Note that connectivity of the structure is still preserved even with thin lines. For mobile robots, the connecting lines can be used as collisionless paths of the robot navigation.
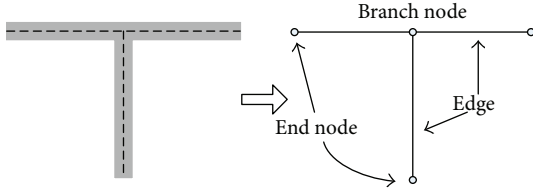
FIGURE 1: Concept of thinning.

FIGURE 2: Centre grid p_1 and its 8 neighbor grids.

In the paper, eight neighbour grid operators are used to solve the skeleton of free areas in grid map. Centre grid p_1 and 8 neighbor grids ($p_2 \sim p_9$) are described in Figure 2. "0" indicates that the grid is empty and "1" indicates the grid is occupied. For the occupied grid p_1, if its eight neighbour grids satisfy the thinning condition, grid p_1 is change into empty because it is not a part of the skeleton.

Thinning Condition 1. Consider

(1) $2 \leq N(p_1) \leq 6$,

(2) $S(p_1) = 1$,

(3) $p_2 \cdot p_4 \cdot p_6 = 0$,

(4) $p_4 \cdot p_6 \cdot p_8 = 0$.

Thinning Condition 2

(1) and (2) are the same as Thinning Condition 1,

(3) $p_2 \cdot p_4 \cdot p_8 = 0$,

(4) $p_2 \cdot p_6 \cdot p_8 = 0$.

Here,

$$p_i = \begin{cases} 0, & \text{grid } p_i \text{ is empty}, \\ 1, & \text{grid } p_i \text{ is occupied}. \end{cases} \tag{6}$$

$N(p_1)$ denotes the number of occupied grids of 8 neighbours, and it is defined as follows:

$$N(p_1) = \sum_{i=2}^{9} p_i. \tag{7}$$

$S(p_i)$ denotes times of the change from 0 to 1 in the sequence $p_2, p_3, \ldots, p_8, p_9$.

3.2. The Sample Space Limitation. The map skeleton is constructed for designating confined sample area and navigation path. If robot moves along bisector of the obstacles (approximately along the skeleton), it has more opportunities to collect more environment information. The performance of MCL is heavily dependent on density of particles if the other conditions are the same. The sample space is usually very large in the stages of MCL global localization and processing the kidnapped robot problem (the worst case is all free areas in the environmental map). A lot of particles are required in order to localize successfully, and the request of real time is not met. However, if we could acquire
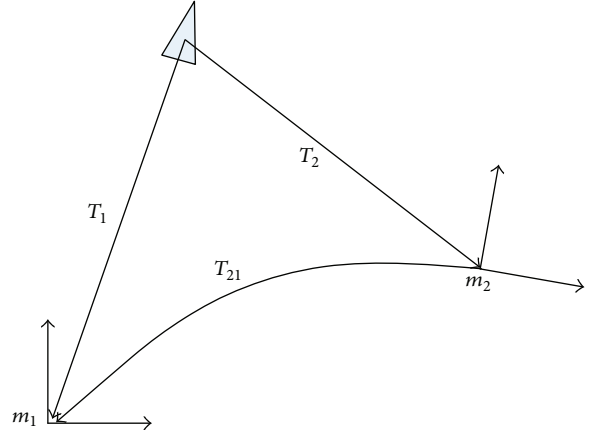
the relevant knowledge about pose of robot in advance, for example, we are sure that robot is located in the area that is much smaller than free area in the map, we can only sample in this area, which improves the speed of the algorithm.

4. Relative Pose Estimation of Maps

4.1. Hypothesis Generation. The simplest situation is considered: two complete maps m_1 and m_2 (their coordinates are different in global coordinates) in the same closed environment Ω save two parts of information: grid occupancy information and skeleton of corresponding free area. If a robot constantly moves along the skeleton of free area in environment Ω and observes the environment at the same time, MCL global localization is used to localize the robot in map m_1 and map m_2. Suppose the location estimations of robot in two maps are $\xi_t^{(1)}$ and $\xi_t^{(2)}$, respectively, at time t, as shown in Figure 3, the following formula can be concluded:

$$T_1 = T_{21} \cdot T_2, \tag{8}$$

where T_1 and T_2 are corresponding homogeneous transformation matrixes to $\xi_t^{(1)}$ and $\xi_t^{(2)}$, respectively. T_{21} is homogeneous transformation matrix that the coordinates of map m_2 is relative to one of map m_1.

According to (4), the relative pose between m_1 and m_2 is as follows:

$$T_{21} = T_1 \cdot T_2^{-1}. \tag{9}$$

The specific approach is put forward in the paper. A robot is simulated to move along the skeleton of the partial map m_1 and constantly observe the map, which is obtained by

FIGURE 3: Map merging coordinate diagram.

a robot using the single-robot-SLAM algorithm. At the same time, these simulated data are used as information sources to do Monte Carlo localization for partial map in the map m_2. The localization process is carried out until the particles converge to one or more clusters. Because the simulation data is obtained near the map skeleton, the initial particles only need to be generated near the skeleton of m_2 in the process of MCL in the map m_2. In this way the initial pose estimation between m_1 and m_2 is got quickly. Figure 4 shows the model of the method.

4.2. Hypothesis Verification.

From the previous method, we get the preliminary estimation to the relative pose between two partial maps m_1 and m_2 which were constructed by Robot1 and Robot2, respectively. The estimation may have one or more peak values, which are supposed to be as relative pose hypotheses between m_1 and m_2. Suppose that homogeneous transformation matrix T is one of these hypotheses, as show in Figure 5, the homogeneous transformation matrix T_3 of the two robots' relative pose can be calculated with T, T_1, and T_2 as follows:

$$T_3 = T_1^{-1} \cdot T \cdot T_2. \tag{10}$$

We can use T_3 to path planning two robots' paths in the map m_1 and make them meet as far as possible. The plan is operated on Robot1, and the results are sent to Robot2 in order to collaborate with each other. If the two robots can meet according to the path planning, the relative pose hypothesis T is true, and we go to the stage of estimation optimization; otherwise we test the next hypothesis. If all hypotheses are false, the robot still constructs map independently (single-robot SLAM), waiting for the next operation of map merging.

5. Estimation Optimization

After the previous steps, the preliminary estimation to the relative pose of map is got. Based on the research of literature [24], the relative pose is further optimized. The concrete optimization process is as follows.

5.1. Dissimilarity Measurement Function of Map.

From the definition of map merging, the map merging problem can be seen as an optimization problem, whose optimization function is $\omega()$. It is the problem that $\omega()$ is directly used as optimization function. Because the values of $\omega(m_1, T_{(t_x, t_y, t_\theta)}(m_2))$ are arbitrarily leap with continuous changes of variables (t_x, t_y, t_θ), the function $\omega()$ delivers no effective gradients to do optimization like hill-climbing algorithm. The dissimilarity function is as follows:

$$\psi(m_1, m_2) = \sum_{c \in C} [d(m_1, m_2, c) + d(m_1, m_2, c)] \tag{11}$$

with

$$d(m_1, m_2, c) = \frac{\sum_{m_1[p_1]=c} \min\{md(p_1, p_2) \mid m_2[p_2] = c\}}{\#_c(m_1)}, \tag{12}$$
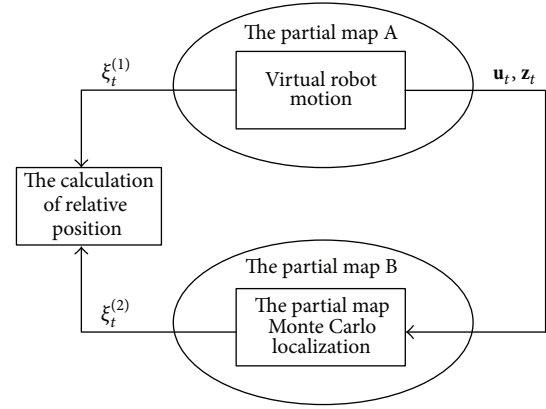

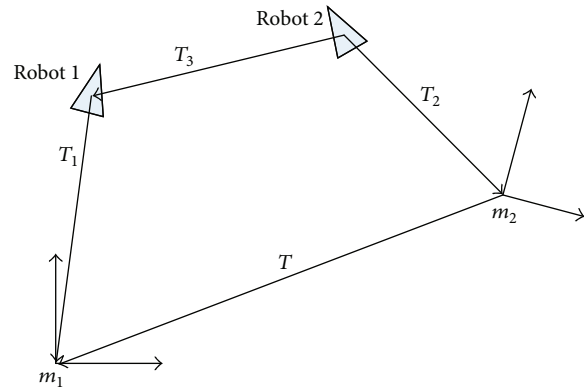Figure 4: Calculation of relative pose of maps based on virtual robot motion.

Figure 5: Relative pose of robots.

where C denotes grid range of map m_1 and map m_2, $m_1[p_1]$ denotes the value of grid p_1 in map m_1, $md(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$ denotes the Manhattan-distance between points p_1 and p_2, and $\#_c(m_1) = \#\{p_1 \mid m_1[p_1] = c\}$ denotes the number of grids with value c in map m_1.

To simplify the calculation, the grids' value of the map is marked as "free" or "occupied" or "unknown" according to the predefined threshold. Only occupied and free grids are considered for computing dissimilarity function, so $C = \{occ, free\}$. In order to compute the dissimilarity function in linear time, a so called distance-map is introduced. Distance-map $dmap_c[x_1][y_1]$ denotes the Manhattan distance between the grid $p_1 = (x_1, y_1)$ in map m_1 and a grid which is the nearest point to p_1 with value c in map m_2:

$$dmap_c[x_1][y_1] = \min\{md(p_1, p_2) \mid m_2[p_2] = c\}. \tag{13}$$

The concrete calculation process for $dmap_c$ is shown in Algorithm 1.

Using the distance-map $dmap_c$, we can calculate $d(m_1, m_2, c)$ with Algorithm 2.

Input: the grid map m and grid value c
Output: distance-map $dmap_c$
1 **for** $y = 0$ **to** $n + 1$ **do**
2   **for** $x = 0$ **to** $n + 1$ **do**
3   **if** $m(x, y) == c$ **then**
4           $dmap_c[x][y] = 0$;
5   **else**
6           $dmap_c[x][y] = \infty$;
7   **end**
8   **end**
9 **end**
10 **for** $y = 1$ **to** n **do**
11   **for** $x = 1$ **to** n **do**
12       $h = \min(dmap_c[n - x + 1][n - y] + 1,$
           $dmap_c[n - x][n - y + 1] + 1)$;
13       $dmap_c[n - x][n - y] = \min(dmap_c[n - x][n - y], h)$;
14   **end**
15 **end**
16 **for** $y = 1$ **to** n **do**
17   **for** $x = 1$ **to** n **do**
18       $h = \min(dmap_c[x - 1][y] + 1, dmap_c[x][y - 1] + 1)$;
19       $dmap_c[x][y] = \min(dmap_c[x][y], h)$;
20   **end**
21 **end**
22 **return** $dmap_c$;

ALGORITHM 1: Calculation process of distance-map $dmap_c$ in grid map.

Input: the grid map m_1, m_2 and grid value c
Output: $d(m_1, m_2, c)$
1 calculate the $dmap_c$ of m_2 according to Algorithm 1;
2 $d(m_1, m_2, c) = 0$;
3 **for** $y = 0$ **to** $n + 1$ **do**
4   **for** $x = 0$ **to** $n + 1$ **do**
5   **if** $m_1(x, y) == c$ **then**
6          $d(m_1, m_2, c) = d(m_1, m_2, c) + dmap_c[x][y]$;
7   **end**
8   **end**
9 **end**
10 **return** $d(m_1, m_2, c)$;

ALGORITHM 2: Calculation process of $d(m_1, m_2, c)$.

5.2. Random Walk Optimization. The basic idea of random walk optimization is to search in the given solution space using the way of random walk. At each step a random solution is generated and the corresponding heuristic rule of next step

is computed. The concrete process is shown in Algorithm 3. And a more detailed description is in the literature [24].

6. Experimental Results

We perform experiment based on the map merging algorithm proposed in this paper. The process of experiment is as follows. Firstly, grid map of the identical simulation environment (Figure 6(a)) is constructed by using SLAM algorithm twice, as shown in Figures 6(b) and 6(c). Then, the relative pose between two partial maps is calculated using the method proposed in the paper. Finally, the results of map merging are illustrated in Figures 6(d) and 6(e). Figure 6(d) is the result based on calculation of relative pose of maps using virtual robot motion, and its optimized result using random walk optimization is showed in Figure 6(e).

7. Conclusions

In the paper, map merging method based on virtual robot motion is proposed in the field of multirobot SLAM. For multi-robot SLAM, there are four kinds of interaction effect between two robots. The first kind is no interaction between two robots. The second kind is hypothesis generation because

(a) Simulation environment

(b) Partial map m_1

(c) Partial map m_2

(d) Map merging based on preliminary estimation

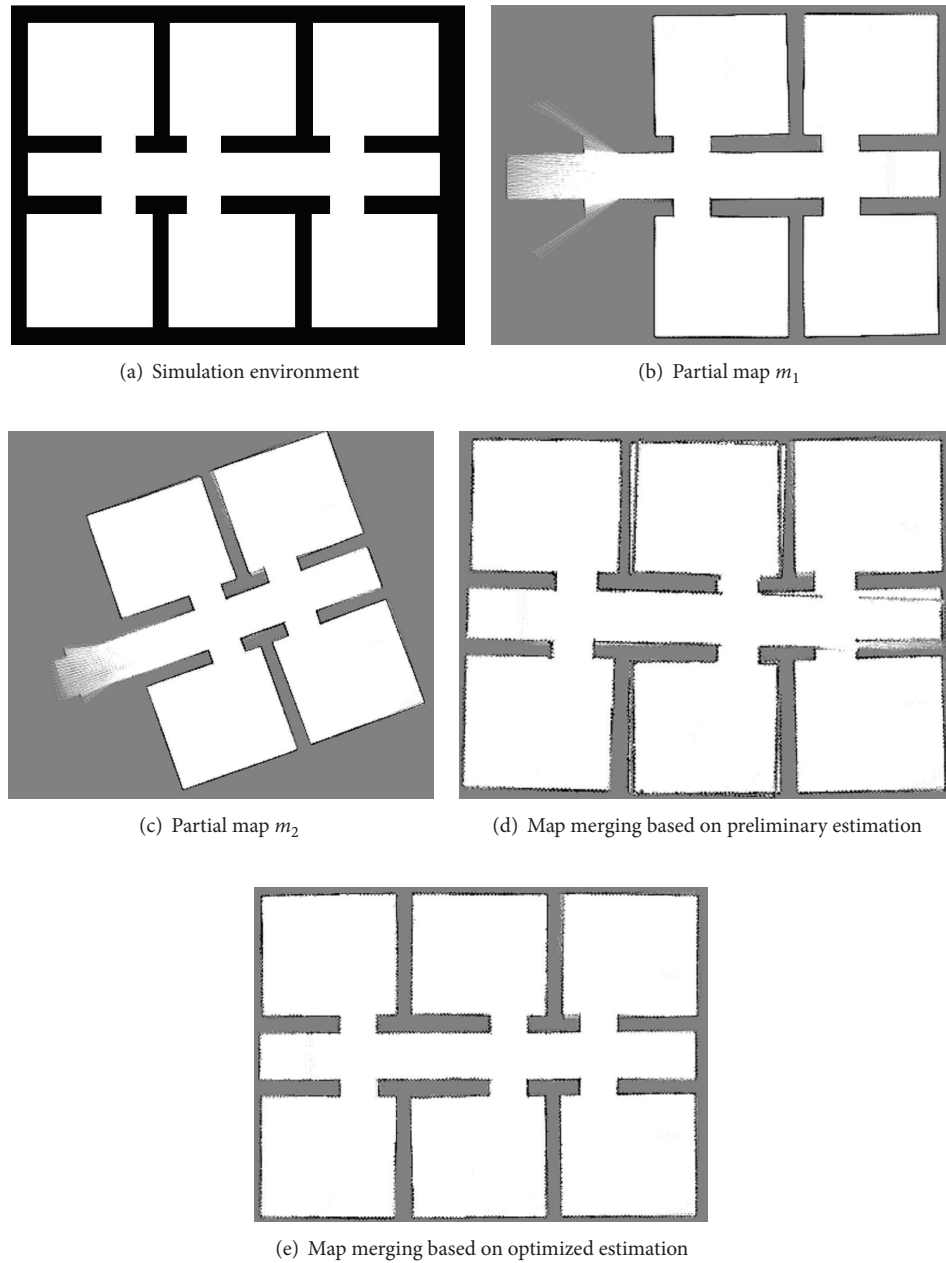(e) Map merging based on optimized estimation

FIGURE 6: Experimental results.

communication is permitted between robots, but relative pose of other robot is unknown. The third kind is hypothesis verification because communication is permitted between robots, and relative localization hypothesis is generated in the process of hypothesis generation. The forth kind is coordinated exploration because robots have relative pose and can share map and explore environment.

In this paper, a mobile robot is simulated in one map; it moves along the map's skeleton and measures the virtual environment. At the same time, these simulated data are used as information sources in the other map to do partial map Monte Carlo localization; if localization succeeds, the

relative pose hypotheses between the two maps can be computed easily. Then, they actively verify one hypothesis using a rendezvous technique. If successful, using the hypothesis as initial value, the estimation is optimized by a heuristic random search algorithm. The algorithm is not only for grid maps but also other types of map. The experimental results have verified the algorithm.

In the future, the corresponding problems, such as network transmission and collaboration of robots, are required to be considered. Cloud robotics is considered to be the next great-leap-forward development of robotics. The method will be improved to apply to cloud robotics.

Input: the grid map m_1, m_2, initial relative pose estimation t_{start}
Output: the relative pose t between map m_1 and m_2
1 $k = 0$; $t_k = t_{\text{start}}$;
2 $\Sigma_0 = \Sigma_{\text{init}}$; $\mu_0 = \mu_{\text{init}}$;
3 $c_0 = \psi(m_1, T_{t_{\text{start}}}(m_2))$;
4 **while** $k < numsteps$ **do**
5     $\nu_k \sim N(u_k, \Sigma_k)$;
6     $s = t_k + \nu_k$;
7     $c_s = \psi(m_1, T_s(m_2))$;
8   **if** $c_s < c_k$ **or** $RS(t_k, s) = s$ **then**
9         $k = k + 1$; $t_k = s$; $c_k = c_s$;
10        $\Sigma_{k+1} = (t_k, t_{k-1}, \ldots, t_{k-M})$;
11        $\mu_{k+1} = (t_k, t_{k-1}, \ldots, t_{k-M})$;
12   **end**
13 **end**
14 **return** s;

ALGORITHM 3: Random walk optimization.

Acknowledgments

References

[1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, New York, NY, USA, 2008.

[2] S. Thrun, "Simultaneous localization and mapping," in *Springer Tracts in Advanced Robotics*, M. Jefferies and W. Yeap, Eds., vol. 38, pp. 13–41, Springer, Berlin, Germany, 2008.

[3] H. Zhang, X.-P. Fan, and Y.-L. Liu, "Simultaneous localization and mapping for mobile robots," *Journal of Data Acquisition and Processing*, vol. 20, no. 4, pp. 458–465, 2005.

[4] H. C. Lee, Y. J. Cho, and B. H. Lee, "Accurate map merging with virtual emphasis for multi-robot systems," *Electronics Letters*, vol. 49, no. 15, pp. 932–934, 2013.

[5] D. Saitov, K. Han, and S. Lee, "Diffusing method for unknown environment exploration in multi robot systems," in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and K. Tan, Eds., vol. 6145 of *Lecture Notes in Computer Science*, pp. 683–690, Springer, Berlin, Germany, 2010.

[6] L. Carlone, M. Kaouk Ng, J. Du, B. Bona, and M. Indri, "Simultaneous localization and mapping using rao-blackwellized particle filters in multi robot systems," *Journal of Intelligent and Robotic Systems*, vol. 63, no. 2, pp. 283–307, 2011.

[7] S. N. Givigi and H. M. Schwartz, "Map merging of Multi-Robot SLAM using Reinforcement Learning," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 53–60, Seoul, Republic of Korea, 2012.

[8] J. W. Fenwick, P. M. Newman, and J. J. Leonard, "Cooperative concurrent mapping and localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1810–1817, Washington, DC, USA, May 2002.

[9] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, "Towards multi-vehicle simultaneous localisation and mapping," in *Proceedings of the IEEE International Conference on Robotics adn Automation*, pp. 2743–2748, Washington, DC, USA, May 2002.

[10] S. Thrun and Y. Liu, "Multi-robot SLAM with sparse extended information filers," *Robotics Research*, vol. 15, no. 5, pp. 254–266, 2005.

[11] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.

[12] S. Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.

[13] 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 the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3232–3238, Las Vegas, Nev, USA, October 2003.

[14] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: algorithms, performance bounds and observations," *Autonomous Robots*, vol. 11, no. 2, pp. 117–136, 2001.

[15] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431–447, 2006.

[16] H. Zhang, Y. L. Liu, and J. Sun, "Mobile robot Monte Carlo localization in incomplete map," in *Proceedings of the 10th International Conference on Intelligent Technologies*, pp. 202–205, Guilin, China, December, 2009.

[17] W. H. Huang and K. R. Beevers, "Topological map merging," *International Journal of Robotics Research*, vol. 24, no. 8, pp. 601–613, 2005.

[18] G. Dedeoglu and G. Sukhatme, "Landmark-based matching algorithm for cooperative mapping by autonomous robots," in *Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems (DARS '00)*, pp. 251–260, Knoxville, Tenn, USA, 2000.

[19] D. Benedettelli, A. Garulli, and A. Giannitrapani, "Cooperative SLAM using M-space representation of linear features," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1267–1278, 2012.

[20] S. Carpin, A. Birk, and V. Jucikas, "On map merging," *Robotics and Autonomous Systems*, vol. 53, no. 1, pp. 1–14, 2005.

[21] H. Choset and J. Burdick, "Sensor based planning, Part I: the generalized Voronoi graph," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation*, pp. 1649–1655, May, 1995.

[22] T.-B. Kwon and J.-B. Song, "Real-time building of a thinning-based topological map," *Intelligent Service Robotics*, vol. 1, no. 3, pp. 211–220, 2008.

[23] D. Portugal and R. P. Rocha, "Extracting topological information from grid maps for robot navigation," in *Proceedings of the International Conference on Agents and Artificial Intelligence*, pp. 137–143, Algrave, Portugal, February, 2012.

[24] S. Carpin, "Fast and accurate map merging for multi-robot systems," *Autonomous Robots*, vol. 25, no. 3, pp. 305–316, 2008.