Multi-robot Exploration and Mapping Based on the Subdefinite Models*

Valery E. Karpov[1,2], Alexander S. Migalev[1], Anton D. Moscowsky[1], Maxim A. Rovbo[1], and Vitaly V. Vorobiev[1]

[1] National Research Centre "Kurchatov Insitute", Moscow, Russia
karpov.ve@gmail.com, alexander.migalyov@mail.ru, moscowskyad@gmail.com, rovboma@gmail.com, Vorobev_VV@nrcki.ru
[2] Moscow Institute of Physics and Technology, Moscow, Russia

Abstract. In this work we consider an environment exploration and mapping task for a group of heterogeneous mobile robots. We propose a range of methods tied together in a hierarchical two-level control system. The distinguished features of the proposed system are the use of subdefinite models for robot localization as well as an original mechanism for local interaction. We present both theoretical and experimental results. On the experimental side of the study we conduct both simulation experiments as well as a real robotic swarm system investigation.

Keywords: collective robotics · exploration · mapping · subdefinite models · multi-robot systems.

1 Introduction

Multi-robot systems have become a hot research topic due to the fact that many practical applications such as environment monitoring [1], technical inspection [2], search and surveillance [3] and several others can not be solved by a single robot. Many of these tasks can be formally expressed as various multi-robot exploration and mapping problems [3, 4, 5], which in turn can be decomposed into localization and mapping, path planning (obstacle avoidance) and multi-robot interaction (communication) problems.

Localization and mapping are traditionally viewed as a single coupled problem, e.g. SLAM problem, and various techniques and methods of solving this problem are known. Typically these methods rely on a Kalman filter [7], particle filter [8], graphs [9], etc. One should note that the vast majority of SLAM algorithms relies on accurate sensor measurements and meticulous odometry. This restricts the application of existing approaches to some well-defined classes of tasks and makes them unsuitable for other classes (for example, indoor SLAM methods are not good in outdoor navigation and vice versa). However, in general SLAM can perform well (or satisfactorily) if the

* The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-43955-6_18

SLAM pipeline is appropriately chosen in accordance with the features of specific tasks and robots.

In this paper we describe an approach of solving the multi-robot exploration and mapping problem which has the following distinctive features. Firstly, we rely on a well-established SLAM methods but consider the output (a robot location and a local map) to be not error-free (which is a realistic assumption) and we augment this output with a topological map of the environment constructed using subdefinite models. Secondly, we rely only on the local robot interaction and use inter-robot communication to guide the path planning. As for the environment, we consider it to be partially known to the robots, e.g. we consider particular distinctive markers to be present (although neither their absolute, nor relative position is known a priori).

The paper is organized as follows. In section 2 we present methods that are used to solve the multi-robot mapping and exploration task. In section 3.1 we present the results of the modeled experiments. Section 3.2 is devoted to the investigation of the real robot case in indoor mapping scenarios.

2 Models and Algorithms

We consider the control system of each mobile robot involved in group exploration and mapping task to be decomposed into two hierarchical levels: strategic (high) and tactical (low). Navigation modules of the strategic level perform fuzzy localization of the robot using predefined markers and characteristic objects. A schematic map is constructed on this level via recognition methods based on subdefinite models calculations. Modules residing on the tactical layer deal with obstacle avoidance, trajectory following, etc. In this work we examine only the strategic level of the system, e.g. the methods that are used to localize a robot (by using subdefinite calculations and fuzzy spatial models), methods of map refining, mapping methods for a single robot being part of the group and, finally, map merging techniques and strategies.

Problem statement. We consider a group of robots, each of which has the ability to determine its location and to mark obstacles on a map. Communication among the robots is strictly local — each one can talk only with its nearest neighbors. Therefore, the robots are equipped with transceivers with limited range. The goal of the robot group is to jointly construct a map of the environment.

As already mentioned, group or joint mapping tasks can be reduced to the problem of investigation of a map fragment by a single robot and global map construction from several such fragments obtained via communication between robots. The map fragment exchange task requires, as a rule, existence of a stable and broad communication channel. The communication system structure, in general, doesn't limit the robots' capability to transmit information to all accessible neighbors [10], [11]. However, the use of a fully connected topology is too bulky, despite its logical simplicity.

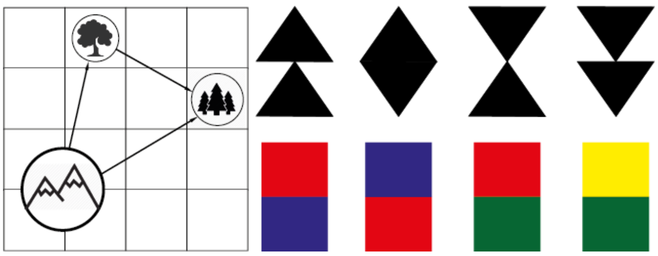
The global map construction can be reduced to the task of fragment integration, which consists of comparison of each part, similar areas detection and overlaying fragments correctly. Without the unified coordinates' binding it becomes a nontrivial task of searching for common subgraphs.

In the following section we describe the navigation task solution on the strategic level. Firstly, a mechanism that allows a robot to determine its location relying on a given schematic environmental map is described. After determining its approximate location, the robot should define the map more precisely. Then the movement trajectory planning mechanism is described, and finally the problem of group mapping and navigation is discussed.

2.1 Robot Localization

The robot localization system relies on data from a video camera (for landmark-based navigation) and from ultrasonic range sensors (for obstacle detection). Sub-definite procedures [12] are at the heart of the localization methods. These procedures are extensively used when the variables used to model some domain can not be determined precisely but only determined as probability interval estimates, which is our case. We represent robot's and markers' locations as sub-definite variables. It is also assumed within the approach under consideration that the interpolation functions that bind the variables' values with each other do exist. In addition, an iteration procedure is implemented that narrows the variables' domains at each step. Thus we work with a computational model which consequently narrows the area of indeterminacy of investigated variables' values, e.g. the robot's and markers' locations.

We use a well-known spatial model — a regular grid composed of square cells of identical size that represents the map. This representation is convenient and frequently used in such type of tasks, for example in [13]. As long as the localization system is based on landmarks (**Fig. 1**a), another approach is frequently encountered: along with registering cells on a map, objects and relations between them are also registered, for example [14] and [15]. We use two-component color markers and marine signs of definite shapes as the landmarks (**Fig. 1**b). A recognition system was developed that is capable of estimating such attributes of the objects as color, form, size and orientation in space. An identification of objects was implemented that relies on a composition of various attributes.



a) b)

Fig. 1. a) Combined map, objects and relations, b) Used markers

To identify the robot's location on a map we use a voting method. Suppose there exist several functions that determine possible locations of the robot based on several criteria: computer vision, sensor data, previous positions history, global navigation, etc.

The robot's location in this case is identified by a distribution of candidate-cells r_{ij} on the map.

$$f^m(X_t) = R_t{}^m = \{r_{ij}, r_{ik}, \ldots\} \ , \tag{1}$$

where X_t is the input data at a moment t. Let the method 1 give the distribution R_1, method 2 — the distribution R_2, etc. Then the robot location is determined as follows:

$$R_t = k_1 R_t^1 \oplus k_2 R_t^2 \oplus \ldots \oplus k_M R_t^M \ , \tag{2}$$

where the operation $\oplus$ is a superposition of two distributions that in a simple case match a value to the amount of times the cell is presented in sets R^m, k_m — the weighting coefficient utilized for increasing influence of a criteria against the remainder.

$$R_t = \{c_{ij} r_{ij}, c_{ik} r_{ij}, \ldots \} \ , \tag{3}$$

where c_{ij} is the amount of times the corresponding cell is encountered with their weighting coefficients in distributions R_m. We will denote the value c in the following as the vote of this cell, and the function f_m — as a voting method. The system has three methods of voting implemented:

1. The cells from which the current scene can be seen are determined. In other words, for every cell the following condition is examined:

$$\left|dist(r(O_i), r_0)L(O_i) - dist(r(O_j), r_0)L(O_j)\right| < \delta, \forall\, O_i, O_j \in O(t) \ , \tag{4}$$

   where O_i is a current scene object $O(t)$, $r(O_i)$ is a map cell, containing an object O_i, r_0 — a verified cell, $dist(r_1, r_2)$ — the distance between the r_1 and r_2 cells, δ — an established error, $L(O_1)$ — the O_1 object's height on the image. The cells fitting this condition receive a voice.
2. An approximate distance from the robot to an object is determined based on the information about the approximate sizes of objects for every object. All cells positioned at this distance also receive a voice.
3. We assume that the robot can not change its position rapidly between two sequential observations. Thus, the cell, in which the robot was located in the previous moment of time, as well as neighboring cells, receives a voice in every step of the algorithm.

Consequently, we consider that the robot is located in the cell with the maximum amount of voices. A few cells could satisfy this criterion, so we choose only one using a mean value or using additional qualifying criteria. Let us describe the mechanism of voting in terms of subdefinite models (further SD-models). The robot position in this case is an H-value, i.e. some set, belonging to a whole range of definition. In this case the SD-expansion is the cells the map was divided into, and the range of definition accordingly is the whole set of map cells.

$$D_{NM} = \begin{Bmatrix} d_{11} & \ldots & d_{1M} \\ \ldots & \ldots & \ldots \\ d_{N1} & \ldots & d_{NM} \end{Bmatrix} \tag{5}$$

Let us use an SD-variable, determining a possible robot location as a cells set.

$$X = \{d_{ij} \ldots d_{kp}\} \tag{6}$$

The main idea of using SD-models is gradually decreasing the field of indeterminacy. The transformation of sets is performed using SD-operations.

$$f_i : X_1 \rightarrow X_2 \tag{7}$$

In case of the represented system, the SD-operations are voting methods, transferring all definition range to SD-values. Consequently, possessing a set of voting methods f_i and sequentially applying them to the initial set, we get a set of possible locations X_i for every method.

$$f_i : D \rightarrow X_i \tag{8}$$

After that a superposition operator is applied, which leaves only the cells with the maximum amount of votes.

$$F: X_1 \ldots X_i \rightarrow X_F \tag{9}$$

The set represents possible robot locations. In the following, the averaging operation can be applied and one location among all others can be selected.

$$\bar{f} : X_F \rightarrow x \tag{10}$$

2.2 Map Refinement

A map refinement procedure is triggered at each step of the navigation algorithm to update a partially known map in accordance with the new sensor measurements, e.g. measurements produced by the ultrasonic rangefinders. Each map cell is characterized by its weight *w*, which initially equals to zero and increases in case an obstacle is detected in that cell (so the cell is considered untraversable for the robot, see **Fig. 2**).

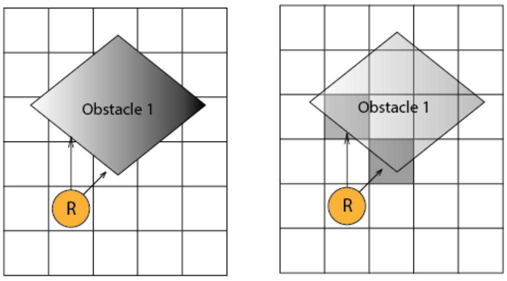


Fig. 2. Marking untraversable cells on a grid representation of the map.

As the location of the robot is subject to errors, the same is true for the obstacles. To handle this problem the following approach is adopted.

1. A degradation function F_D is introduced. This function decreases the weight of the cells over time in case these weights are below some predefined threshold. It means that cells with high probabilities of being blocked, e.g. the cells that are consequently reported as untraversable by the navigation system, will not lose their weight. At the same time, obstacles that were put on the grid map by mistake will tend to disappear. In the simplest case, a linear function can be used for cells weight degradation:

$$F_D(w) = w - k_D \tag{11}$$

   Here k_D is the degradation coefficient.
   In case the current weight value is lower than the initial value (zero) the degradation procedure is not performed for this cell.
2. All possible untraversable cells (taking into account the robot's localization error) are marked so, but the weight for each cell is calculated by the following formula:

$$w_t = w_{t-1} + \frac{k_0}{N^2}, \tag{12}$$

   where N is the number of possible robot's locations. Values of the coefficients k_0 and k_D are set in accordance with the following inequalities:

$$k_0 > \frac{k_D}{N^2},\ for\ large\ N, \tag{13}$$

$$k_0 < \frac{k_D}{N^2},\ for\ small\ N.$$

Thus, map refinement functions by receiving measurements of the rangefinders and a robot location on a grid map as the input and producing a marked grid as the output.

2.3 Individual Mapping and Map Merging

As written above, grids are used as environment models for each member of the group of mobile robots. Path planning for each robot is based on the following approach. Consider a grid area sized N×M:

$$F = (s_1, s_2, \ldots, s_{NxM}), \tag{14}$$

Each element of the working area (grid cell) s_i is assigned the value of a potential φ_I which defines the attractiveness of that map area for a robot. Negative values are considered to be attractive for the robot, while positive — distractive. Movement direction in this case is defined as the direction of the whole field:

$$\overline{E} = -\sum_i^{N\times M} \overline{h_i}, \tag{15}$$

Here h_i is the derivative of the potential φ_i which in the simplest case can be calculated in the following way:

$$|h_i| = \frac{\varphi_i}{{r_{i,robot}}^k}, \tag{16}$$

where $r_{i, robot}$ is the distance between the robot and the considered (*i*-th) cell.

Unexplored regions of the working area are considered to be of high attractiveness for a robot initially and while it explores them they lose their attractiveness. During the exploration the robot always moves towards the area with the minimum sum of potentials. A merging procedure is triggered after all individual exploration tasks are finished. To merge maps, all robots must form a topology which we call a static swarm [16] and then start communicating with each other and exchanging maps and routes. To increase efficiency of the group exploration and mapping task we also suggest using local interaction.

Robots interaction. If the robots are planning their paths to stay far from each other, unnecessary duplication of mapping effort (two or more robots mapping the same area) can be avoided. Thus, it can be beneficial to generate an “avoid me” signal so that the other robot that receives the signal knows that the area he is moving to has already been explored. At the same time, when the configuration of obstacles is not trivial, it can be beneficial to perform multiple examinations of such working area by different robots to obtain a consistent map model. In such cases, a communication command “approach me” should be utilized.

In accordance with the path planning strategy presented above “avoid me” and “approach me” signals can be implemented as changing the potential values of the corresponding grid cells, e.g. grid cell that are occupied by the robots generating the signals. For a robot’s “avoid me” signal, the value of the cell’s potential is increased (repulsing other robots) and vice versa. It should be emphasized, that these changes of the potential field are registered by each robot on its local map, depending on the cell that the signal was registered in by the robot. On the other hand, interaction between robots is implemented using explicit communication: robots broadcast messages to all their neighbours.

3 Experimental Evaluation

3.1 Model Experiments

Both in simulation and in real robot experiments the map was divided into a 30 by 50 grid of cells. Grids used in the experiments had 5% to 30% filling rate, meaning that 5% – 30% of the cells were untraversable (occupied by the obstacles). Each experiment involved 16 robots. For each task, two communication profiles were used: with local communication while mapping and without such communication. As one can see from the **Table 1**, using the suggested local communication method significantly (an order of magnitude) increases the time efficiency of the mapping routine.

Table 1. Mapping time comparison

	Blockage percentage					
	5%	10%	15%	20%	25%	30%

Average mapping time (steps)	With commu-nication	29,58	30,28	31,48	32,47	33,75	35,46
	Without commu-nication	326,63	334,07	343,65	353,79	364,47	376,02

A step of simulation corresponds to the time a robot moves from a cell to an adjacent cell. One can note that the mapping time increases with the increase of the number of obstacles. It happens because a robot's path becomes more complicated when a larger number of obstacles is present. It could probably be possible to decrease the overall computation cost by using more advanced path planning techniques.

We would also like to note that, due to localization errors, the mapping method without local communication tends to detect more obstacles than actually exist. For example, for maps with 30% blocked cells, the number of detected obstacles is one third times as much again than it should be. Using local communication while mapping positively influences this problem. In this case the number of detected obstacles almost equals the actual number.

3.2 Field Experiments

Field experiments were carried out using a DrRobot X80 Pro platform [17]. This robot is a two-wheeled mobile platform with a differential drive equipped with ultrasonic and infrared (IR) rangefinders as well as a Wi-Fi camera (**Fig. 3**a).

The control system was implemented within the ROS framework: ROS Indigo [18] on a GNU/Linux Kubuntu 14.04 personal computer (installed on each robot). Control of a robot is based on automata techniques. All commands — both low-level (movement forward, backward etc.) and high-level (exploration, mapping etc.) are modeled as Mealy machines. CV system based on the OpenCV library is used for marker-based robot localization while ultrasonic range-finders data is used to detect obstacles and construct a local map. The global map is updated each time local maps are shared by the group.

The local communication was implemented on IR connection based on RC5 protocol. The aim of the local communication is the identification of the nearest robots that are considered neighbours and their relative position. Four transmitters and receivers are located on a robot facing different directions at a 90° angle. The robot transmitted its own unique ID number in four directions simultaneously through a standard RC5 command using four IR diodes.

Experiments were carried out on a polygon (**Fig. 3**) sized 10x20 meters with four robots involved. A virtual model of the polygon was created beforehand using Gazebo simulation software [19] in the ROS framework.

a) b) c)

Fig. 3. a) DrRobot X80Pro Platform used in the experiments, b) Fragment of the polygon used in experiments, c) Virtual model in Gazebo

Twelve markers were placed in different locations on a polygon. Each marker is a cylinder with a marine navigation sign (unique for each marker) painted on the surface. The highest precision of localization was achieved when different markers were detected in a single image of the CV video stream. However due to the polygon large size, such situations occurred rarely. In this case, robots behaved in the following way. First robot performed a full rotation scanning all the markers and trying to localize itself by narrowing the ambiguity region. If this region still remained too large, then robot started to move in a random direction just following a trajectory created by the path planner and relying on encoders' measurements.

In case the markers are not unique, the localization error becomes much higher as the ambiguity region significantly enlarges. At this point, we would like to mention that a robot's location is composed of two components: a grid cell (x and y coordinates) and a heading angle (orientation). When the size of the grid cell was large (more than the diameter of a robot footprint), most of the errors occurred in the heading angle estimation. By latter we mean that the x and y coordinates' estimation error was much lower than the heading estimation error. This is due to the geometrical aspects of calculations as well as the low level of orientation discretization (8 values).

4 Summary

Conducted experiments showed the applicability of the proposed methods in exploration and mapping tasks. At the same time, the results of the simulated experiments only partially correspond to the real world experiments. This is due to the observation methodology of the experiments that allows giving only qualitative estimates.

Directions of the future work include but are not limited to the following. Firstly, the environment should become more complex by adding additional obstacles and active objects — intruders. In addition, a system of precise tracking of a robot's location on the polygon is needed. Existence of such a system will allow to obtain not only qualitative estimates but quantitative measurements, too.

5 Acknowledgements

The project was partially supported by RSF 16-11-00018, RFBR 15-01-07900 and RFBR 15-07-07483 grants.

References

1. Marino, A., Antonelli, G.: Experiments on sampling/patrolling with two Autonomous Underwater Vehicles. Rob. Auton. Syst. 67, 61–71 (2015).
2. Lopez, J., Perez, D., Paz, E., Santana, A.: WatchBot: A building maintenance and surveillance system based on autonomous robots. Rob. Auton. Syst. 61, 1559–1571 (2013).
3. Portugal, D., Rocha, R.P.: Cooperative multi-robot patrol with Bayesian learning. Auton. Robots. (2015).
4. Shvets, E.: Stochastic multi-agent patrolling using social potential. In: 29th European Conference on Modelling and Simulation. pp. 521–526 (2015).
5. Portugal, D., Rocha, R.P.: Multi-robot patrolling algorithms: examining performance and scalability. Adv. Robot. 27, 325–336 (2013).
6. Elmaliach, Y., Agmon, N., Kaminka, G.A.: Multi-robot area patrol under frequency constraints. Ann. Math. Artif. Intell. 57, 293–320 (2009).
7. Castellanos, J.A., Neira, J., Tardos, J.D.: Limits to the consistency of EKF-based SLAM. Robotics. 45, 1878–1881 (2004).
8. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: A factored solution to the simultaneous localization and mapping problem. Proc. 8th Natl. Conf. Artif. Intell. Conf. Innov. Appl. Artif. Intell. 68, 593–598 (2002).
9. De, A., Lee, J., Keller, N., Cowan, N.J.: Toward SLAM on Graphs. In: Workshop on the Algorithmic Foundations of Robotics. pp. 1–16 (2008).
10. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Trans. Robot. 21, 376–386 (2005).
11. León, A., Barea, R., Bergasa, L.M., López, E., Ocaña, M., Schleicher, D.: SLAM and map merging. J. Phys. Agents. 3, 13–23 (2009).
12. Narin'yani, A.S., Borde, S.B., Ivanov, D.A.: Sub-Definite Mathematics and Novel Scheduling Technology Programs. Artif. Intell. Eng. 11, (1997).
13. Elfes, A.: Using Occupancy Grids for Mobile Robot Perceeption and Navigation. Computer (Long. Beach. Calif). 22, 46–57 (1989).
14. Castle, R.O., Klein, G., Murray, D.W.: Combining monoSLAM with object recognition for scene augmentation using a wearable camera. Image Vis. Comput. 28, 1548–1556 (2010).
15. Jensfelt, P., Ekvall, S., Kragic, D., Aarno, D.: Augmenting SLAM with object detection in a service robot framework. Proc. - IEEE Int. Work. Robot Hum. Interact. Commun. 741–746 (2006).
16. Karpov, V., Karpova, I.: Leader election algorithms for static swarms. Biol. Inspired Cogn. Archit. 12, 54–64 (2015).
17. Dr Robot Inc: X80pro, http://www.drrobot.com/products_item.asp?itemNumber=X80pro.
18. ROS Indigo Igloo, http://wiki.ros.org/indigo.
19. Gazebo: Gazebo Simulator Official Site, http://gazebosim.org/.