Decentralized Strategy for Cooperative Multi-Robot Exploration and Mapping

Ana Batinović, Juraj Oršulić, Tamara Petrović, Stjepan Bogdan

University of Zagreb, Faculty of Electrical Engineering and Computing, Laboratory for Robotics and Intelligent Control Systems (LARICS), 10000 Zagreb, Croatia (e-mail: ana.batinovic@fer.hr).

Abstract: This work presents a novel approach to autonomous decentralized multi-robot frontier exploration and mapping of an unknown area. A mobile robot team simultaneously explores the environment, discovers frontier points (points on the border between explored and unexplored space), and shares information in order to become dispersed throughout the environment. During the exploration, information exchanged between the mobile robots is limited to data containing mobile robot positions and current mobile robot target points. The main goal of the approach is to allocate the mobile robots to target frontier points in a way which minimizes the overall exploration time. Moreover, a mobile robot team at the same time creates a common map of the environment. The proposed strategy has been implemented in a simulation environment and compared with a state-of-the-art exploration strategy. Simulation results demonstrate the advantages of the proposed decentralized multi-robot strategy.

Keywords: Mobile Robot, Multi-Robot System, Frontier Points, Target Point, Decentralized Strategy, Exploration, Mapping.

1. INTRODUCTION

Application of multi-robot systems to solving core robotics problems has drawn significant attention in the last few decades. One example is coordination of a mobile robot team for exploration of an unknown area, which is encountered in many applications, such as search and rescue (Murphy (2004)), cleaning (Endres et al. (1998)), (Pinheiro et al. (2015)), warehousing (Wurman et al. (2008)) or planetary exploration (Mataric and Sukhatme (2001)), to name a few. Due to the fact that autonomous multi-robot systems are entering society and as such will interact with people on a daily basis, development of efficient coordination algorithms becomes necessary.

Like in the human society, robots can be more effective when they work together. Moreover, a robot team can accomplish a predefined task much quicker than a single robot can (Dias and Stentz (2000)). Another advantage of mobile robot teams is the possibility of sensor fusion, which in turn can help to compensate for sensor uncertainty (Wurm et al. (2008)). If done properly, multi-robot coordination can lead to i) task accomplishment in shorter time, ii) increased robustness, iii) higher map quality, and finally iv) the completion of tasks impossible to be performed by a single robot (Dias et al. (2006)).

We consider the problem of autonomous multi-robot exploration and mapping using fast dense frontier detector and a novel decentralized exploration strategy. Frontier detection results in points on the border of the explored and unexplored space in the environment - *frontier points* shown in Fig. 1. The exploration strategy described in the paper assigns to each mobile robot a frontier point that needs to be explored. Strategy works in a decentralized manner (runs on each robot separately), while aiming to minimize total exploration time. It does so by sharing information between robots and taking into account the characteristics of the so-far explored environment, which enables the robots to become better dispersed throughout the unexplored environment. The considered multi-robot system uses a fully connected communication graph, which can however be relaxed to include a (not fully) connected communication graph together with implementation of multi-hop information sharing.

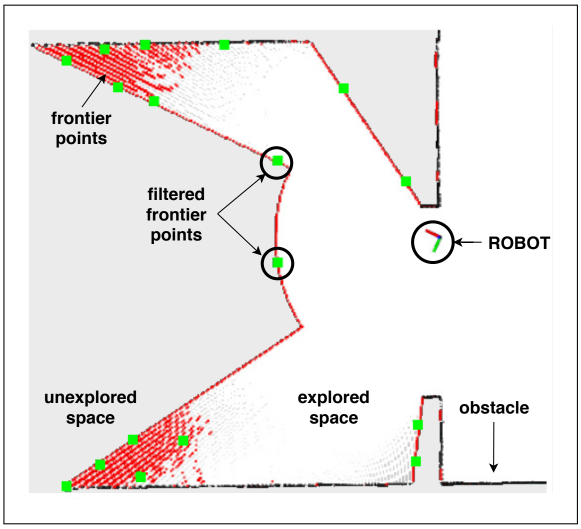


Fig. 1. The environment is represented by a 2D map, with an occupancy grid that divides the map into cells: white cells describe explored while grey cells unexplored space. Black cells define obstacles. Frontier points (red) and filtered frontier points (green).

With respect to the existing frontier-based strategies, our strategy uses different optimization functions and is decentralized and event-based. We implemented the approach in a realistic ROS-based simulation and compared its performance to a state-of-the art method in order to show its advantages.

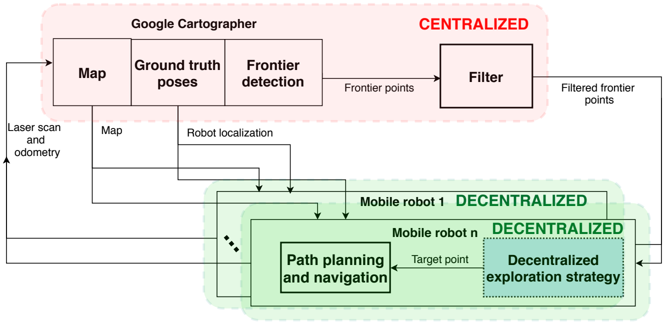


Fig. 2. Overall schematic diagram of the decentralized exploration and mapping process for n mobile robots in the simulator. Google Cartographer SLAM and filter module (highlighted in red) generate filtered frontier points that are (currently) the centralized part of exploration and mapping process. The exploration strategy, path planning and navigation module (highlighted green) are decentralized parts that generate n outputs and create a common map.

In the next section we describe more thoroughly different parts of the system, state-of-the-art for each part and algorithms used in this paper. In Section III we describe the proposed exploration strategy. Simulation results are presented in Section IV, and in the final section a conclusion is given.

2. EXPLORATION AND MAPPING OVERVIEW

Overview of the system is given in Fig. 2. The system consists of a centralized part (Simultaneous Localization and Mapping (SLAM), frontier detection and frontier points filter), which runs on a dedicated computer and a decentralized part (Exploration strategy and navigation), which runs on each robot. In the folloewing text each part is described separately.

2.1 SLAM

The laser scan and odometry sensor measurements of each mobile robot represent input data for a Simultaneous Localisation and Mapping (SLAM) module. Most exploration approaches in the literature use the ROS 'gmapping' package for generating the map and localizing mobile robots (Keidar and Kaminka (2012), Umari and Mukhopadhyay (2017)). The 'gmapping' package implements a SLAM algorithm that uses a Rao-Blackwellized particle filter (Grisetti et al. (2007)).

In this paper, we use a submap-based graph SLAM method - Google Cartographer (Hess et al. (2016)) for map building, which in our case generates the ground truth map. This map is used in simulations to obtain the mobile robot poses (*perfect localization*) from the Stage simulator (Vaughan et al. (2012)), allowing us to eliminate the SLAM algorithm uncertainty in simulations for algorithm comparison.

2.2 Frontier Detection

In order to determine frontier points we use frontier detection according to Orsulic et al. (2019). This method, which is an extension to Google Cartographer, has achieved good results in terms of wall-time per frontier update, which greatly speeds up our exploration and mapping process.

Another detection method, based on Rapidly Exploring Random Trees (RRTs) is given in Umari and Mukhopadhyay (2017). The RRT algorithm is biased towards unexplored regions and provides a general approach which can be extended to higher dimensional spaces. However, the method has proved not to be fast enough for a given scenario in instances when larger parts of the environments were explored, so we opted to use a dense frontier detection method from Orsulic et al. (2019).

2.3 Filter Module

The filter module receives frontier points from the frontier detector, clusters the points and stores only the centre of each cluster (Umari and Mukhopadhyay (2017)). The clustering process reduces the number of frontier points that are close to each other. In that manner we avoid unnecessary consumption of computational resources, without significant loss of information about the frontier. For clustering we use the Hierarchical Agglomerative Clustering algorithm (Scikit-learn (2019)), which does not require a predefined number of clusters. We only need to set a distance threshold parameter, above which clusters will not be merged. Taking into consideration a given scenario and a laser range, the distance threshold parameter is set to 1 meter.

2.4 Exploration strategy

The term exploration strategy considered here includes algorithms for assigning robots to target (frontier) points for environment exploration. Such algorithms can be grouped into centralized and decentralized algorithms. In the centralized approach, each mobile robot receives tasks from a single central *leader* which runs the overall planning algorithm, and afterwards the mobile robot sends its info back to the leader. Centralized assignment may be less practical due to communication limits (Dias and Stentz (2000)), robustness issues (Dias et al. (2006)), or time required for algorithm execution and scalability (Juliá et al. (2012)). An advantage of centralized approach is that optimal plans can be found (Z. Yan and Cherif (2011)).

In contrast to centralized approaches, in a decentralized approach, the mobile robots are completely independent throughout the exploration process. Each mobile robot has its own local knowledge of the world and can decide its future actions by taking into account its current context and tasks, its own capacities and the capacities of the other mobile robots, through a negotiation process (Yan et al. (2013)). Moreover, it typically has better reliability, flexibility, adaptability and robustness (Zlot et al. (2002)). Our approach is a hybrid one - the robots can independently decide towards which target point to navigate using an optimization procedure, while having common knowledge of all target frontier points and sharing information on their position and current goals.

Two frontier-based exploration approaches are introduced in Yamauchi (1998) and Singh and Fujimura (1993). Both approaches are uncoordinated in a sense that robots head to the closest frontier point. Approaches cover homogeneous and heterogeneous multi-robot systems, respectively. Authors in Zlot et al. (2002) use a market-based approach and allow robots to visit a set of goal points (a tour) while continuously negotiating with other robots. One approach similar to ours is given in Burgard et al. (2005), which is a centralized approach that takes into account the costs of reaching a target and the utility of reaching that target.

With respect to the mentioned approaches our approach is hybrid and uses slightly different objective functions for frontier points assignment, which are a combination of frontier point cost, utility of reaching the target point and *frontier occupancy function*. We also cluster frontier points to get a problem of manageable size and thus enable application of known optimization algorithms. Our approach is hybrid in a manner that target point assignment process and navigation are fully decentralized and event-based, that is, each robot team member makes an individual decision on the next target point each time it reaches the previous one. On the other side, SLAM extended with frontier detection and filter module are a centralized part of the exploration and mapping process (Fig. 2). There are several approaches that tackle decentralization of the SLAM for mobile robots (Jiménez et al. (2018), Atanasov et al. (2015)), but this is out of the scope of the paper. We assume that communication range is unlimited, however, one approach for dealing with limited communication range might be to take into account the last target points, such as in Burgard et al. (2005). Another workaround might be to implement a multi-hop communication network.

With respect to the previous approaches we conduct a realistic ROS-based simulation suitable for straightforward experimental deployment. We use state-of-the art frontier detection and map building software, and provide data to allow for future comparisons of different exploration strategies. Software architecture is modular, allowing components to be easily changed or extended, without affecting the remaining processes. Simulation results given at the end of the paper have shown that the proposed approach achieves better performance than the state-of-the art approach selected for comparison.

3. DECENTRALIZED EXPLORATION STRATEGY

At the core of our paper is a decentralized strategy for multi-robot exploration and mapping. The mobile robots exchange information about frontier points under the assumption of a fully connected graph and event-based communication. We define a *mission* as a process that starts by getting a target point and finishes by reaching the target point. Then, event-based communication is triggered by mission accomplishments, since all mobile robots communicate with each other in the moments when the mission for a single mobile robot s over. It means that the rest of mobile robots should send the data even though their missions are in progress and they are still navigating to the goal.

The exploration is performed by a team of n mobile robots $\mathscr{R} = \{1,2,...,N\}$, where the mobile robots do not have prior knowledge about the environment, i.e., the position of the boundaries and obstacles. Every mobile robot i gets the list of frontier points $\mathscr{Y} = \{1,2,...,M\}$ from the filter module (Fig. 2) in the moments when any of the mobile robots reaches the target point (when a mission is over). Each frontier point j is defined with its position in the environment, denoted as $y_j \in R^2$.

We define the *weight* of a mobile robot performing a visit to frontier point W_{ij} as a function of cost C, utility U and frontier occupancy F. Cost function C: $(\mathscr{R} \times \mathscr{Y}) \to \mathbb{R}^+$ returns a positive real number. If i is the index of the mobile robot, and j is the index of the frontier point, then $C(i,j)$ (denoted in the remaining text as C_{ij}) describes the cost of ith robot to visit the jth frontier point. The cost can be a function of time, energy or, like in our case, the estimated distance travelled by mobile robot to reach the target frontier point. The estimated distance is approximated using Euclidean distance between the mobile robot position p_i and the frontier point position y_j:

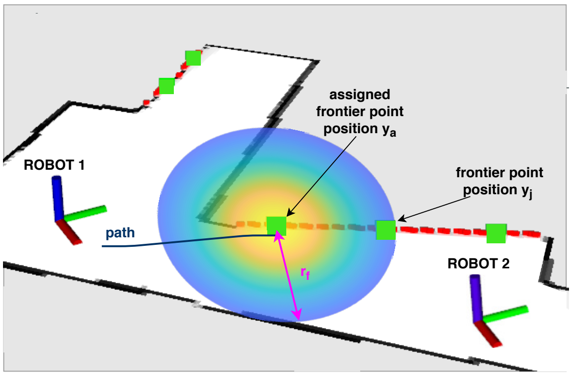


Fig. 3. Illustration of the value of the frontier occupancy function F being different from zero. The mobile robot 1 is assigned to visit a frontier point a (at position y_a), and follows the path to reach it. When mobile robot 2 calculates weight for all the current frontier points (green), F for the frontier point j is different from zero because j is inside the radius r_f.

$$C_{ij} = d(p_i, y_j) = \sqrt{(p_{ix} - y_{jx})^2 + (p_{iy} - y_{jy})^2}. \tag{1}$$

The utility function U_j: $(\mathscr{Y} \times \mathscr{M}) \to \mathbb{R}^+$ returns a positive real number. The cells of the occupancy grid $\mathscr{M}$ may be marked as explored space, unexplored space or obstacle. The utility function is proportional to the number of the unexplored cells k_j within a fixed distance from the frontier point j in the previously defined radius r:

$$U_j = \lambda_u k_j, \tag{2}$$

where λ_u is a constant determined experimentally. When the function U_j is taken into account, the mobile robot will prefer frontier points that are surrounded by more unexplored space even if they are a little bit further. It is assumed that the mobile robot will detect all unexplored cells around the assigned frontier point after reaching it.

Another important component is the frontier occupancy function F_{ij}: $(\mathscr{R} \times \mathscr{Y}) \to \mathbb{R}^+$, the 2-dimensional Gaussian function with the position of the mean in a frontier point and with the standard deviation $\sigma = [r_f \ \ r_f]^T$. If the frontier point j, for which the mobile robot i is calculating the weight, is in the range of radius r_f from the position of an another mobile robot assigned point (y_a); the value of the frontier occupancy function is calculated by Gaussian function, and zero otherwise:

$$F_{ij} = \begin{cases} \lambda_f e^{-\left[\frac{(y_{jx}-y_{ax})^2}{2\sigma_x^2} + \frac{(y_{jy}-y_{ay})^2}{2\sigma_y^2}\right]} & \text{if } d(y_j, y_a) < r_f, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where λ_f is an experimentally determined constant. An example of the frontier point position y_j, assigned point position y_a and Gaussian function values inside the radius r_f is shown in Fig. 3. The function F_{ij} is used to prevent assigning frontier point to the mobile robot B if that point is close to a point that is already assigned to mobile robot A.

For each frontier point j, the weight W_{ij} of the i-th mobile robot is calculated as:

$$W_{ij} = C_{ij} - U_j + F_{ij}. \tag{4}$$

The weight matrix W $(N \times M)$ is formed for N mobile robots and M frontier points:

Algorithm 1: Decentralized strategy for mobile robot i

1 **while** *Unexplored* **do**
2   **if** *Request* **then**
3   Send position and current target point to the other mobile robots;
4   **if** *Mobile robot i has reached the previous target point* **then**
5   Request positions and current target points from other mobile robots;
6   Calculate the weight matrix W;
7   Hungarian algorithm (W);
8   **return** Mobile robot i is assigned to frontier point;

$$W = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1j} & \dots & W_{1M} \\ W_{21} & \ddots & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ W_{i1} & & & \ddots & & \vdots \\ \vdots & & & & \ddots & \vdots \\ W_{N1} & \dots & \dots & \dots & \dots & W_{NM} \end{bmatrix}. \tag{5}$$

Since mobile robots have the same set of frontier points, the only information mobile robots share is their position and current target point, needed to calculate (1)-(3). The amount of exchanged data is thus reduced, which enables easier and faster communication. The weight matrix W represents the input into the Hungarian algorithm that finds an optimal assignment solution in polynomial time. The Hungarian algorithm is described in Kuhn (1955) and tested in Kulich et al. (2015). Initially, the Hungarian algorithm assumes that the number of frontier points is the same as the number of mobile robots. Due to the fact that there are usually fewer mobile robots than frontier points, virtual mobile robots are added and then skipped during the process of assignment and exploration.

Let the matrix X be the matrix of zeros and ones, where $X_{ij} = 1$ iff the mobile robot i is assigned to the frontier point j. Then the optimal task assignment has weight:

$$\min_X \left(\sum_i \sum_j W_{ij} X_{ij} \right), \tag{6}$$

anticipating that minimisation of sum will ensure the dispersion of the mobile robots in the environment.

All frontier points are visible to all mobile robots. When mobile robot i is assigned to a frontier point according to line 8 in Algorithm 1, the mobile robot starts to follow the planned path and navigates to the target frontier point. At the moment when the mobile robot i reaches the target point (mission is over), a request is sent to other mobile robots to get their positions and current target points, and to fill in the weight matrix W, which is an input to Hungarian algorithm. The described process executes until the whole environment is explored and a complete map of the environment is generated. The Fig. 4 illustrates the described steps and algorithm lines for a mobile robot team.

To summarize with respect to the coordinated strategy from Burgard et al. (2005), the cost function described in (1) is the same (probability is taken into account through frontier points) and the utility of a frontier point, in contrast to (2), depends on the number of mobile robots that are moving to that point. Component (3) is introduced in this paper. Event-based approach that we use results in less frequent target changes, thus minimizing the need for re-planning.

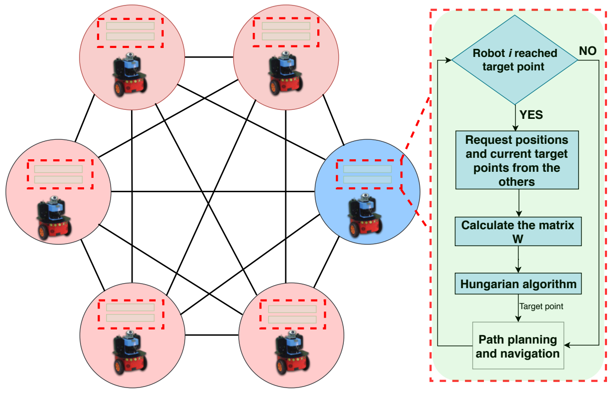
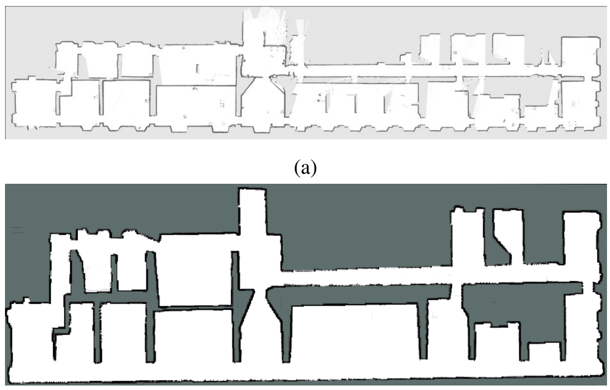


Fig. 4. Execution of the decentralized part of the Fig. 2. Every mobile robot explores the environment following the steps from Algorithm 1. The blue robot accomplished the mission and requests weights from other team members (red). The algorithm is the same for all mobile robots and always executes in the moment when a robot reaches a target point.



(a)

(b)

Fig. 5. Maps used for simulation experiments. (a) Original map: Map from Haehnel (2014). (b) Google Cartographer map: The map generated using Google Cartographer SLAM algorithm.

4. SIMULATION RESULTS

The proposed strategy is implemented and compared with the coordinated strategy (Burgard et al. (2005)), one of the first coordinated approaches which achieved enviable results and is easy to implement. The frontier detector and filter module are the same for both algorithm implementations - coordinated and our decentralized.

4.1 Simulation Setup

Simulations were carried out in well-known robot operating system (ROS) using the Stage 2D simulator (Vaughan et al. (2012)), which simulates robot movement and lidar perception inside a loaded environment map. The ROS Navigation stack is used to control and direct the mobile robots towards exploration goals.

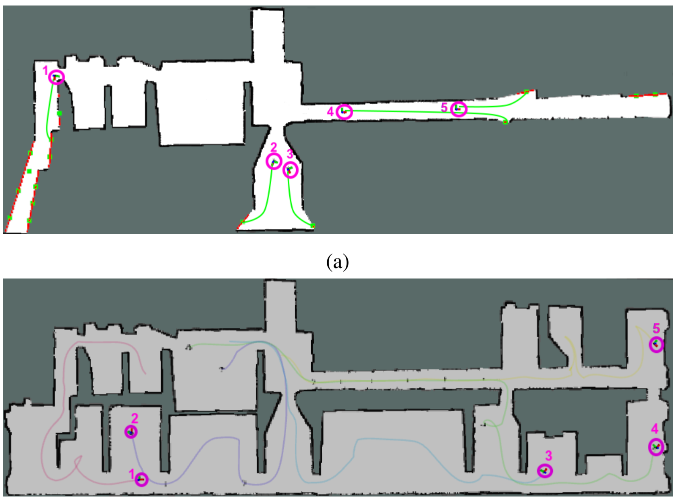
The scenario used in the simulation is the Belgioioso Castle, available in Haehnel (2014) and shown in Fig. 5. It is a challenging and a typical office-like scenario with a free space area of approximately 225 m^2. For the simulation, we use a model of Pioneer P3-DX with maximum speed of 1.3 m/s, laser range 20 m and 360° laser scan window. The algorithms were tested with teams of two, three and five mobile robots. In our case, the number of mobile robots was limited because of the complex simulation setup and computational requirements.

The robots started off in random positions within this world, but the initial positions are the same for both algorithms to allow for comparison. The results are presented as averages of 10 runs for each set. Constants λ_u, λ_f, r and r_f were experimentally determined considering map dimensions and mobile robot laser ranges and set to 0.9, 1.2, 0.5, 3.0 respectively. The constant values are set to give more importance to the frontier occupancy function than the utility function.

4.2 Simulation Results

The exploration process is visualized using the RViz visualization tool shown in Fig. 6. During the exploration and mapping process, the covered area as a function of time was recorded. The comparison of the coordinated and our decentralized strategy is shown using the *Coverage Ratio (CR)* indicator which shows the percentage of the accessible terrain covered by the mobile robot team. It is calculated as: $\frac{\text{explored cells}\cdot 100}{\text{accessible cells}}$, where accessible cells represent free cells. We report the time it took to cover 50, 75, 90 and 98 percent of the environment. Box plots for coverage ratio in time measured during exploration are shown in Fig. 7. As can be observed, the decentralized strategy has an advantage over the coordinated strategy.

When we compare the average exploration time for both strategies, we conclude that adding mobile robots to exploration team



(a)

(b)

Fig. 6. Decentralized exploration with five mobile robots. In (a), the mobile robots focused on different frontier points according to the exploration strategy. For instance, instead of choosing the closest frontier point, the mobile robot 1 navigated to the frontier point that minimized equation (4). Following the same reasoning as mobile robot 1, other mobile robots chose their target points. In (b), the exploration is completed and paths traversed by each mobile robot are shown.

reduces exploration time. Using our decentralized strategy, it took **7.8%, 15.3% and 32.6%** less time for two, three and five mobile robots respectively to explore the environment compared to coordinated strategy. This result shows that our decentralized strategy (within described setup) performs significantly better than coordinated, especially for five mobile robots. However, depending on the area size and configuration, adding of more mobile robots might not pay off in terms of shorter exploration time due to overcrowding. Additionally, openly available source code of this work can be found on Github [1] . It includes developed and implemented decentralized multi-robot exploration strategy as well as our implementation of the coordinated algorithm. Video recordings for simulation with multiple mobile robots can be found on YouTube [2] .

5. CONCLUSION AND FUTURE WORK

In this paper, a modular approach to autonomous decentralized multi-robot exploration and mapping was presented. This approach is not only restricted to Google Cartographer SLAM and dense frontier detection, but may also be applied to different multi-robot systems. This strategy has resulted in improved behaviour in terms of exploration time compared to a state-of-the-art strategy in terms of exploration time.

Even though the goals of this paper were shown to be achieved, the algorithm is open towards improving. Future research should consider decentralized map creating. Also, a significant improvement to this strategy would be a simpler simulator, that would allow for simulation of more robots.

Another research direction can be an extension to the algorithm to cope with a limited communication range of the mobile robots. Future work should also consider a multi-robot system which uses a (not fully) connected communication graph. Finally, we would like to take into consideration scenarios in which the robots may fail as well as time-varying environment scenarios.

REFERENCES

Atanasov, N., Ny, J.L., Daniilidis, K., and Pappas, G.J. (2015). Decentralized active information acquisition: Theory and application to multi-robot SLAM. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3), 376–386.

Dias, M.B. and Stentz, A. (2000). A free market architecture for distributed control of a multirobot system. *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, 115–122.

Dias, M., Zlot, R., Kalra, N., and Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7), 1257–1270.

Endres, H., Feiten, W., and Lawitzky, G. (1998). Field test of a navigation system: autonomous cleaning in supermarkets. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*. IEEE.

Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1), 34–46.

[1] https://github.com/larics/decentralized_multi_robot_exploration
[2] https://youtu.be/vPEIYrDiT-0

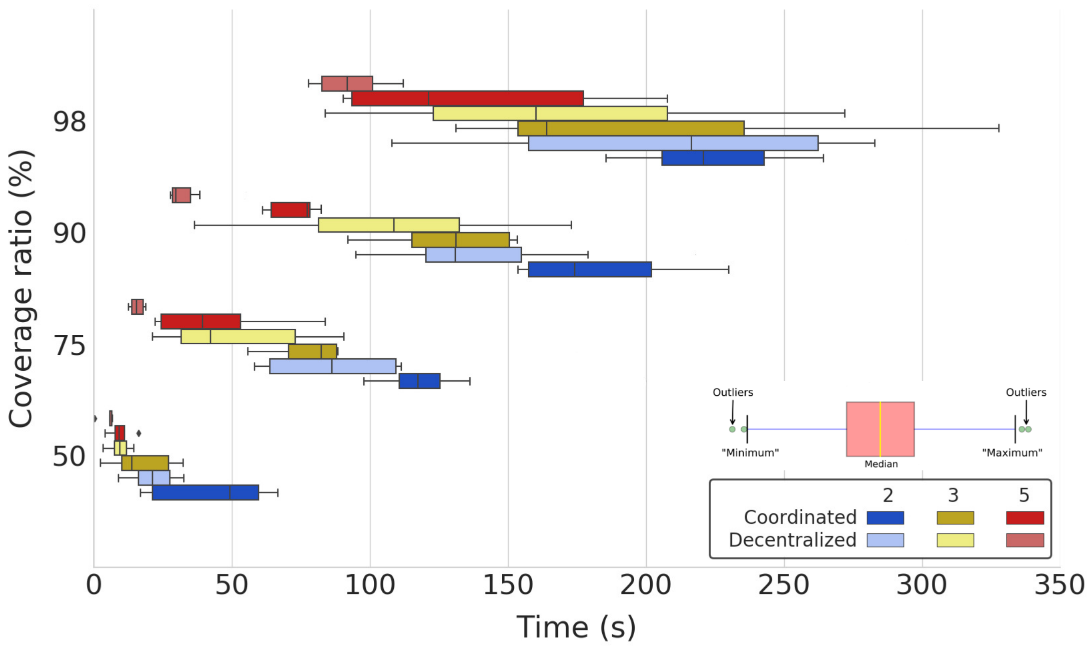


Fig. 7. Statistical comparison of coordinated and decentralized strategies tested on two, three and five mobile robots. The box plot compares time for obtaining different coverage ratios for both strategies using the median, quartiles and outliers.

Haehnel, D. (2014). Robotics Datasets. `http://www2.informatik.uni-freiburg.de/~stachnis/datasets.html`. Accessed: 2019-08-28.

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

Jiménez, A., García-Díaz, V., González-Crespo, R., and Bolaños, S. (2018). Decentralized online simultaneous localization and mapping for multi-agent systems. *Sensors*, 18(8), 2612.

Juliá, M., Gil, A., and Reinoso, O. (2012). A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33(4), 427–444.

Keidar, M. and Kaminka, G.A. (2012). Robot exploration with fast frontier detection: Theory and experiments. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, 113–120. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC.

Kuhn, H.W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), 83–97.

Kulich, M., Juchelka, T., and Přeučil, L. (2015). Comparison of exploration strategies for multi-robot search. *Acta Polytechnica*, 55(3), 162–168.

Mataric, M.J. and Sukhatme, G.S. (2001). Task-allocation and coordination of multiple robots for planetary exploration. In *In Proceedings of the 10th International Conference on Advanced Robotics*, 61–70.

Murphy, R. (2004). Human–robot interaction in rescue robotics. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 34(2), 138–153.

Orsulic, J., Miklic, D., and Kovacic, Z. (2019). Efficient dense frontier detection for 2d graph SLAM based on occupancy grid submaps. *IEEE Robotics and Automation Letters*.

Pinheiro, P., Cardozo, E., Wainer, J., and Rohmer, E. (2015). Cleaning task planning for an autonomous robot in indoor places with multiples rooms. *International Journal of Machine Learning and Computing*, 5(2).

Scikit-learn (2019). Agglomerative Clustering. `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html`. Accessed: 2019-09-15.

Singh, K. and Fujimura, K. (1993). Map making by cooperating mobile robots. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 254–259. IEEE.

Umari, H. and Mukhopadhyay, S. (2017). Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Vaughan, R., Gerkeyand, B., et al. (2012). Stage v4.1.1. `http://wiki.ros.org/stage`. Accessed: 2019-07-18.

Wurm, K., Stachniss, C., and Burgard, W. (2008). Coordinated multi-robot exploration using a segmentation of the environment. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE.

Wurman, P.R., D'Andrea, R., and Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29(1), 9–20.

Yamauchi, B. (1998). Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, 47–53. ACM.

Yan, Z., Jouandeau, N., and Cherif, A.A. (2013). A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12), 399.

Z. Yan, N.J. and Cherif, A.A. (2011). Multi-Robot Decentralized Exploration Using a Trade-Based Approach. In *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics*. SciTePress - Science and and Technology Publications.

Zlot, R., Stentz, A., Dias, M., and Thayer, S. (2002). Multi-robot exploration controlled by a market economy. In *Proceedings 2002 IEEE International Conference on Robotics and Automation*. IEEE.