Multi-robot Decentralized Exploration using Weighted Random Selection

Abhijith N. Balan[a] and Asokan Thondiyath[b]

Department of Engineering Design, Indian Institute of Technology Madras, Chennai, India

Keywords: Decentralized, Frontier Allocation, Frontier-based Exploration, Multi-robot System.

Abstract: The exploration tasks using multi-robot systems require efficient coordination and information sharing between robots. The map creation is usually done through allocating frontiers,i.e., the boundary between explored and unknown regions of the map, to each robot. This paper introduces an efficient frontier allocation method based on weighted random selection for a decentralized multi-robot system. The weights are calculated based on the size of the frontiers and the distance between a robot and the frontiers. In this strategy, each robot identifies the available frontiers in a shared map and select the goal for exploration through a random selection. Even though a robot is randomly picking a frontier without coordination with other robots, a collective intelligence is developed at the swarm level. The proposed method is computationally efficient and uses minimal communication between the robots in a decentralized multi-robot system. As a result, the robots get allocated to frontier according to the calculated weight. A comparison with the nearest frontier exploration approach in computerized simulation demonstrates the efficiency of the proposed algorithm.

1 INTRODUCTION

Robots have come a long way from a mere fictional entity to a useful tool that can aid or even replace human effort in many situations. They can carry out tasks that are difficult or impossible for a human to perform. Introduction of Multi-Robot System(MRS) extends the capabilities of robots by many folds. These systems work by coordinating multiple robots to perform a single task or multiple tasks at the same time. These systems feature high levels of scalability, resistance to failure, and parallelization of tasks. Such systems are very efficient in tasks such as search and rescue in disasters, nuclear site exploration, surveillance, environment monitoring, and locating the sources of hazardous materials.

MRS is mainly classified into two categories, centralized and decentralized systems. A centralized system would have multiple robots being controlled and managed by a central entity. The central entity manages the system's decision making, communication, and data storage. Failure to this unit would disturb the entire system and the tasks can no longer be completed. A Centralized system has global information about all the robots and their tasks. Access to this information makes coordinating and controlling a centralized robot swarm easier and efficient. A decentralized system, however, lacks the central entity. In such systems, the robot swarm as a whole takes care of decision making through local communication alone. Since the robots lack the information of the state of the system(position of the individual robots and their task status), they take decisions based on its limited perception of the environment and its neighbors. This makes developing strategies for information sharing and exploration in the decentralized system difficult. Even though a decentralized system will not be as efficient as a centralized one, it offers scalability and redundancy, which plays a crucial role in robot swarms.

A robot swarm is an autonomous multi-robot system that can only perform local sensing and communication, with no access to information about other robot's position or task status (Brambilla et al., 2013). They are usually homogeneous, containing simple robots and resembles a decentralized MRS.

Since the exploration and mapping process is parallelizable, multiple robots can carry it out faster and more efficiently than a single advanced robot. A robot system performs this task through coordination, information sharing, and data fusion. Autonomous mapping and exploration can be done through exploring frontiers which are the boundary between mapped ar-

[a] https://orcid.org/0000-0003-4649-2429
[b] https://orcid.org/0000-0002-5474-3999

Balan, A. and Thondiyath, A.
Multi-robot Decentralized Exploration using Weighted Random Selection.
DOI: 10.5220/0010608405230529
In *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2021)*, pages 523-529
ISBN: 978-989-758-522-7

eas and unknown areas. A robot navigating to a frontier can disclose new regions and expand the map. In this paper, we focus on exploration using decentralized swarm robot systems. The most challenging problem in such systems is the decision making. Since global information is unavailable to an agent in a swarm, the agent chooses a local goal based on the information available. In a mapping process, it will be the selection of most promising frontier to explore next. A good strategy for optimizing the local goal can reduce redundant data collection and make the process faster and efficient.

This paper puts forward a decentralized algorithm for frontier selection which is computationally efficient and uses minimal communication between robots. This implicit frontier allocation algorithm allows each robot to take a decision with the local information it possesses without the need of coordination with other robots. The algorithm works better with multi-robot systems with large number of robots. Comparing to other implicit exploration algorithm, our method was 13.7% faster for a multi robot system of 7 robots without adding any extra communication effort. The independence of frontier selection on robot coordination enables the algorithm to scale easily as the number of robots increase and makes this algorithm the right choice for robot swarms in exploration and mapping.

2 PROBLEM STATEMENT

The following defines the problem addressed.

The task is to reduce the overall duration of exploration by a fleet of autonomous robots while keeping the communication between the robots minimal. The robot swarm is considered homogeneous and is equipped with exteroceptive sensors, which allow them to create a map of the environment. The robots also contain devices for communication which are crucial for cooperative exploration. The range of communication is assumed to be less than the mapping range and may not cover the whole robot fleet, i.e., all robots are not connected to each other always. The environment is unknown, and all the robots may not start at nearby locations. One robot might explore a frontier even if another robot had already explored it.

Due to the limited communication range, one robot may not be able to communicate with all the robots in the fleet. A robot should make a decision based on the local information available at that time. The algorithm should be efficient in computation so that the system would not be affected as the number of robots in the fleets is increased.

To limit the scope of the problem, the details on robot localization, mapping, and map sharing are not discussed in this paper. However, they are discussed in the work by (Balan, 2019) in which complete multi-robot SLAM(Simultaneous Localization And Mapping) simulations are addressed in both centralized and decentralized systems.

After identifying all the available frontiers to a robot, the robot has to be allocated to one frontier for further exploration. The inability of one robot to know the position and status of all other robots in the fleet results in selecting a sub-optimal frontier. This problem is inherent in a decentralized multi-robot system.

The following notations are used:

- $\mathcal{R}$ is the set of all robots participating in exploration. $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3 ...... \mathcal{R}_n\}$, where n is the total number of robots in the system.
- $\mathcal{F}^i$ is the set of all frontiers available to the $\mathcal{R}_i$. $\mathcal{F}^i = \{\mathcal{F}_1{}^i, \mathcal{F}_2{}^i, \mathcal{F}_3{}^i ...... \mathcal{F}_m{}^i\}$, where m is the total number of frontiers available to $\mathcal{R}_i$.
- α^i is the assignment vector for $\mathcal{R}_i$.

$$\alpha^i{}_j = \begin{cases} 1 & \text{if } \mathcal{F}^i{}_j \text{ is assigned to } \mathcal{R}_i \\ 0 & \text{if otherwise} \end{cases} \tag{1}$$

Since this is a decentralized approach, the available frontiers and the assignment vectors are different for each robot. Now the problem has simplified to finding the assignment vector α for each robot using $\mathcal{F}$ at any time.

3 STATE OF THE ART

In this section, we discuss the previously proposed methods in multi-robot exploration.

3.1 Nearest Frontier Exploration

Nearest frontier exploration was a popular frontier based exploration strategy formulated by (Yamauchi, 1997). Yamauchi recognized frontiers as the boundary between the known area and the unexplored area of the map. Navigating a robot to a frontier will enable the robot to identify unexplored area that consequently increases the size of the explored area. In a system with more than one robot, each robot has to be navigated to the nearest frontier and broadcast the map information to all other robots. The robot can stop exploration when there are no frontiers left to explore in the map. This ensures coordination between the robots. The area in the map which was already covered would not be covered again by another robot. This is an implicit method, i.e., after receiving the information from other robots, a robot does not need to communicate with other robots for selecting the best frontier to navigate. Nearest frontier exploration is a distributed approach and requires less computation for frontier selection. The algorithm for nearest frontier exploration is given in Algorithm 1.

Algorithm 1: Nearest Frontier Exploration $O(m)$.

Input: $\mathcal{F}_i$
Output: Assignment vector α_i of $\mathcal{R}_i$
$\alpha^i_j = 1$ with $j = \underset{\forall \mathcal{F}^i_j \in \mathcal{F}_i}{\operatorname{argmin}}$ distance to $\mathcal{F}^i_j$;

Nearest frontier exploration can be reduced to $O(1)$ using Breadth-First Search from robot position and taking the first frontier we encounter. However, it is possible that multiple robots may get allocated to the same frontier if their maps are similar (when they are close to each other. This is common in a robot swarm). This keeps the fleet from utilizing its full potential to explore the map.

3.2 MinPos

MinPos is a decentralized frontier allocation algorithm for multi-robot exploration by (Bautin et al., 2012). This algorithm strategically allocates best frontiers to robots.each robot evaluates its relative rank among the other robots in term of travel distance to each frontier. Accordingly, robots are assigned to the frontier for which it has the lowest rank. To evaluate this criterion, a wavefront propagation is computed from each frontier giving an alternative to path planning from robot to frontiers. This method not only considers the distance to each frontier but also the number of robots closer to that frontier. The algorithm builds a local minimum free artificial potential field from each group of frontier using the wavefront propagation algorithm. This also builds the cost matrix, which is used to allocate frontiers. The algorithm for MinPos is given in Algorithm 2.

MinPos is a decentralized frontier allocation algorithm. It does not require any central unit to function like in MOARSLAM (Morrison et al., 2016). However MinPos algorithm requires the communication range to be large enough so that it covers the whole map,i.e., all robots should be in constant communication with all other robots at all times. MinPos requires information about all the robots participating in the cooperative mapping and their task status for the frontier allocation to work. This would not be available in a robot swarm always. Even though MinPos

Algorithm 2: MinPos $O(nm)$.

Input: $\mathcal{R}, \mathcal{F}^i, \mathcal{C}$ cost matrix
Output: Assignment vector α^i of $\mathcal{R}_i$
for $\mathcal{F}^i_j \in \mathcal{F}^i$ **do**
$\quad \mathcal{P}_i j = \sum_{\forall \mathcal{R}^k \in \mathcal{R}, k \neq 1, C^k_i < C^i_j} 1$
end
$\alpha_{\mathrm{i}\,j} = 1$ with $j = \underset{\forall \mathcal{F}^i_j \in \mathcal{F}^i}{\operatorname{argmin}} \mathcal{P}_{\mathrm{i}} j$;

requires global information, the decisions are taken by individual robots. It is also interesting to note that the algorithm complexity is $O(mn)$,i.e., the algorithm becomes slower to allocate frontiers as the number of robots in the swarm increases. This can cause MRS to scale poorly.

3.3 Centralized and Decentralized

A centralized exploration strategy will have a central server which allocates frontiers to each robot for the maximum efficiency. In MOARSLAM (Morrison et al., 2016), the server stores the map created by autonomous robots. The server guide the robots to each location for exploration and the robots transmits the data back to the server. In (Simmons et al., 2000), the robots perform maximum likelihood estimation of the map using odometry and observations. These maps are transmitted to a server which develops the global map. A similar strategy is implemented in (Gil et al., 2010), in which the robot sends the virtual descriptors along with the odometry information to the server. The server adds this information to the global map. All these strategies represent a single point which is in constant bi-directional communication with individual robots. The performance of the system depends on the capabilities of this central entity. There are also limitations on the extent of area that can be mapped (communication range of the central unit) and the number of robots that can communicate to the central unit at a time (communication bandwidth limitation). This inhibits the scalability of the system. To make the systems decentralized, one will have to remove the central server and make the robots capable of making decisions. Nearest frontier exploration 3.1 and Minpos 3.2, which were discussed before, are examples for this. Another approach was proposed in (Yan et al., 2011) in which a trade based scheme is implemented in a decentralized multi-robot system so that each robot bids on a frontier based on the limited information available. This method can reduce computation cost through parallel computation but induces a communication overhead of $O(mn)$, which does not facilitate the scalability factor.

We need a frontier allocation method which is $O(m)$ so that the system becomes easily scalable (computation cost is independent of number of robots) and should work in the absence of global information.

4 PROPOSED METHOD

We propose an approach to the frontier allocation problem based on weighted random selection. In this approach, we calculate a weight for each frontier encountered based on its size and distance and allocate a frontier to a robot according to it. This is a distributed frontier allocation method that will divide the robot swarm according to the size of the discovered frontier (number of frontier points) and the distance to it. This algorithm is computationally efficient and does not require robot communication to select the frontier for exploration. The algorithm performance is independent of the number of robots in the fleet, which makes the system to be scaled easily to any number of robots.
In the proposed method, each robot performs the following steps. (More details on the robot system modules and their interaction are given in the flowchart (Figure 1))

1. Each robot explores freely in the environment.
2. At robot rendezvous, the i^{th} robot broadcasts its local map data to other robots.
3. Also, the i^{th} robot receives map data from other robots.
4. The robot R_i creates an updated map.
5. R_i performs weighted random selection for frontier selection.
6. The R_i navigates to the selected frontier

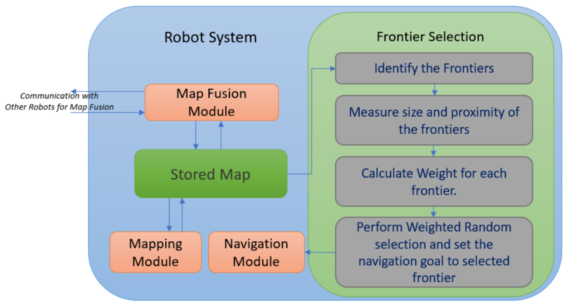


Figure 1: A schematic diagram showing the robot systems module and their interactions. Note that the communication is only required for the Map Fusion, and the frontier selection is performed within the robot system.

Although this is an implicit method, i.e., robots do not coordinate with each other at the frontier selection stage, the fleet as a whole takes a meaningful decision by dividing and allocating itself to different frontiers in the map. This technique is similar to the swarm intelligence present int the nature (e.g. Ants finding the shortest path to food and Termites building their nests). These systems also contain various forms of decision making with the help of limited local information. Our technique enables a single robot following simple local rules to give rise to a collective intelligence in the swarm level.

4.1 Map Sharing

The robots upon rendezvous, share their map data and update their local map. After the map update, all the robot's local maps share a common set of features. Since the communication range is assumed to be less than the mapping range, the complete map of the environment has been developed using these common map features. Each robot fuses the received maps to its local map to obtain the updated map result. The image-based map merging algorithm, as illustrated in (Hörner, 2016) has been used for map sharing.

4.2 Frontier Classification

After the robot develops the updated map, the robot searches the map for frontier points. Frontier points in the occupancy grid map are found by searching for the points in between the explored and unknown parts of the map. After every frontier points are identified, they are clustered into individual frontiers (set of frontier points) using a clustering algorithm based on their position. After finding the frontiers, the shortest distance Euclidean distance ($\mathcal{D}^i_j$) from the robot $\mathcal{R}_i$ to the frontier $\mathcal{F}^i_j$, and the frontier size ($\mathcal{S}^i_j$, the number of frontier points in the frontier $\mathcal{F}^i_j$) are calculated for each of the frontiers. These parameters will be used to select the frontier for exploration.

4.3 Frontier Allocation

In this stage, the robot is left with the frontiers and their parameters (distance and size). We create the weight of each frontier from these to parameters as

$$\mathcal{W}^i_j = \frac{\mathcal{S}^i_j}{\mathcal{D}^i_j} \tag{2}$$

The weight indicates the tangent of the angle created by a frontier in robot's vision. Refer Figure 2. An imaginary frontier is considered at the frontier point

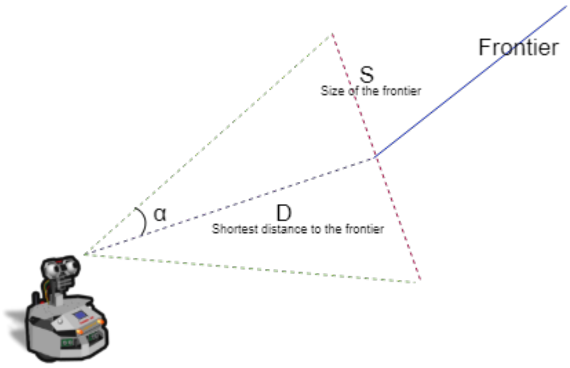


Figure 2: Diagram showing how the weight corresponds to the angle made by the frontier in robot vision.

closest to the robot, aligned perpendicular to the sensing direction, with the same frontier size. This frontier makes an angle α in the view of the robot. Note that tangent of the angle α is the weight that frontier has. A large frontier located far away from the robot gets lower weight. So the weights indicate the angles each frontier makes in the vision of the robot. Refer Figure 3 for example of such frontier selection.

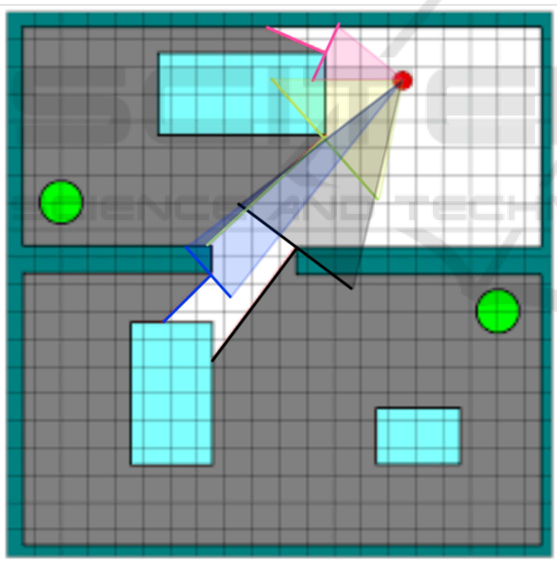


Figure 3: A schematic diagram showing the concept of weight used in the proposed approach. Each color (magenta, green, blue, black) indicate an available frontier, and the red circle indicates the robot R_i. Here note that the angle made by the frontier colored green makes the biggest angle in the robot's view. This frontier will have the largest weight among all frontiers. Image credits: (Virtual Labs, IIIT Hyderabad).

Now we select a frontier (or calculate α^i_j) from the set of frontiers through weighted random selection using the weights we calculated.

$$\alpha^i_j = \begin{cases} 1 & \text{if } \mathcal{F}^i_j = \text{Random Choice}(\mathcal{F}^i) \text{ according to } \mathcal{W}^i \\ 0 & \text{if otherwise} \end{cases} \tag{3}$$

Algorithm for the proposed method is given below

Algorithm 3: Proposed Method $O(m)$.

Input: $\mathcal{F}^i$
Output: Assignment vector α^i of $\mathcal{R}_i$
for $\mathcal{F}^i_j \in \mathcal{F}^i$ **do**
  $\mathcal{W}^j_i = \frac{\mathcal{S}^i_j}{\mathcal{D}^i_j}$
end
SelectedFrontier $\leftarrow$
  Random Choice($\mathcal{F}^i$) with weights $\mathcal{W}^i$
$\alpha_{ij} = 1$ where $\mathcal{F}^i_j =$ SelectedFrontier

Through individual robots performing a weighted random selection for choosing frontier, we can divide a robot swarm to multiple frontiers according to its proximity and size. It is interesting to note that this division is happening without explicit coordination between the robots.

5 EXPERIMENTS AND ANALYSIS

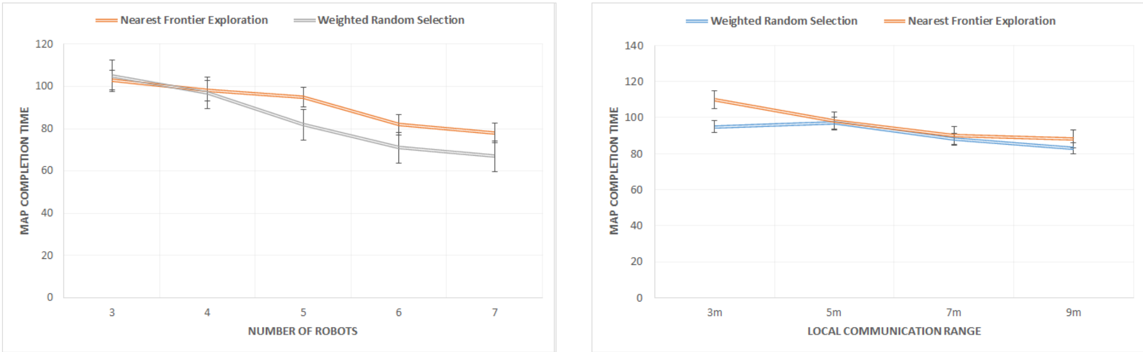
The experiments are done in a computer simulation using Robot operating system. The main concentration is on the time taken for thoroughly exploring the environment. The proposed approach is compared with the other implicit frontier allocation algorithm Nearest Frontier Exploration.

5.1 Simulation

For simulation stage simulator (Figure 5) is used with cave map accommodating all the robots used for exploration. The robots have 180° vision for mapping with parameterized communication range. The map is scaled appropriately (30m X 30m) to simulate a decentralized decision-making process with all the robots. The robot is approximately the size of a 0.4m cube.

The Figure 4a shows the comparison between the map exploration time and the number of robots for both the methods, Nearest Frontier Exploration, and the proposed method. As both algorithms are distributed, with implicit frontier allocation, they can be compared. The results shown are the average of 10 iterations on each robot count in both the algorithms.

Figure 4b compares the map exploration time while varying the local communication range. The local communication range denotes the distance to



(a) The trend in exploration completion time with respect to the number of robots. (Local communication range = 5m)

(b) The dependence of the algorithms to the local communication range. (For five robot fleet)

Figure 4: Simulation result using the cave environment. The figure shows a comparison between Map completion time and Number of robots(a) and Local communication range(b). The proposed algorithm is compared with Nearest Frontier Exploration Algorithm.

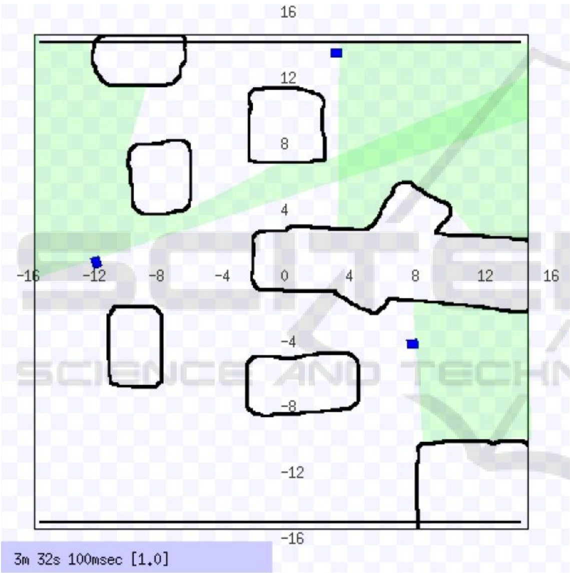


Figure 5: Simulation of 3 robots (blue boxes) in Cave map loaded in stage simulator.

which the communication between the robots is possible.

5.2 Analysis

From the simulation results, we can see that the proposed method, on an average, is 13.7% faster compared to the other algorithm on high robot count. Although both algorithms have similar results in low number of robots, the proposed method shows a clear advantage as the number of robots increases. We can also see that the algorithm works better when the local communication range is small. This indicates the advantage our algorithm has in covering large areas compared to the robot size and its sensing range. We could observe that the robots tend to move together in Nearest Frontier Exploration when they were spawned very close to each other. Having similar maps and frontier distances were forcing the robots to flock together. However, in the proposed method, the robot system was allocated across the frontiers even when they were outside the communication range. A larger frontier was getting assigned more robots even if the robots are outside each other's communication range. This shows that our method developed a collective intelligence in choosing the frontiers for exploration without the need for any communication between the robots. This proves that the proposed algorithm can work better in a swarm system containing a large number of robots.

6 CONCLUSION

In this paper, we addressed the problem of exploring an unknown environment with the decentralized multi-robot system. An algorithm was proposed for this problem, which allocates the frontiers to the robots efficiently so that the exploration time can be reduced. The algorithm addresses the limitations of other previous approaches and attains good results with reducing the computation load on the robots and also using less communication between the robots.

The proposed algorithm uses the concept of weighted random selection for assigning frontiers to robots. In the discussed scenario, i.e., mapping an unknown environment, the weights are calculated according to the size and proximity of frontiers. Other scenarios might require different parameters for weight calculation (e.g., likelihood of finding a target in search and rescue operations), but the underlying concept of weighted random selection remains the same.

The performance of this algorithm is compared with nearest frontier allocation through computer simulation. The proposed algorithm appeared to be significantly more efficient in robot systems with higher number of robots. Furthermore, our algorithm has lower complexity than other multi-robot exploration algorithms and is independent of the number of robots in the fleet. These properties make our approach a good option in robot swarms for explorations.

REFERENCES

Balan, A. (2019). Strategies for multi-robot slam using robot swarms.

Bautin, A., Simonin, O., and Charpillet, F. (2012). Minpos : A novel frontier allocation algorithm for multi-robot exploration. In Su, C.-Y., Rakheja, S., and Liu, H., editors, *Intelligent Robotics and Applications*, pages 496–508, Berlin, Heidelberg. Springer Berlin Heidelberg.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7:1–41.

Gil, A., Reinoso, Ó., Ballesta, M., and Juliá, M. (2010). Multi-robot visual slam using a rao-blackwellized particle filter. *Robotics and Autonomous Systems*, 58:68–80.

Hörner, J. (2016). Map-merging for multi-robot system.

Morrison, J. G., Gavez-Lopez, D., and Sibley, G. (2016). Scalable multirobot localization and mapping with relative maps: Introducing moarslam. *IEEE Control Systems Magazine*, 36(2):75–85.

Simmons, R. G., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., and Younes, H. L. S. (2000). Coordination for multi-robot exploration and mapping. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 852–858. AAAI Press.

Virtual Labs, IIIT Hyderabad. *Frontiers*. Virtual Labs.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *CIRA*.

Yan, Z., Jouandeau, N., and Chérif, A. A. (2011). Multi-robot decentralized exploration using a trade-based approach. In *ICINCO*.