An Occupancy Grid-Based Algorithm for Determining the Relationship Between Sensor Visibility and Foraging Efficiency in a Swarm Robotic System

Nitish Shirodkar
University of Minnesota
Department of Computer Science and Engineering
May 31, 2023

ABSTRACT

Swarm robotics is the study of decentralized multi-robot systems that involve robots with simple, limited functionalities such as local sensing, low processing power, and limited inter-robot communication. These limitations introduce the need for effective indirect communication and mapping algorithms, especially for foraging missions, in which robots are tasked with finding and collecting randomly distributed objects in unfamiliar environments. Previous studies have examined several methods of indirect swarm communication and navigation, such as random walk models and artificial pheromone stigmergy. This study explores the viability of occupancy grid mapping as a method of indirect communication for a swarm foraging task. Occupancy grid mapping is a method of generating maps of an environment by representing the occupancy of each location in an environment using a probability array. With its robustness to sensor measurement uncertainty, occupancy grid mapping addresses the issue of generating maps in noisy environments. This study extends this technique to the multi-robot case by introducing a novel occupancy grid-based swarm foraging algorithm and determining a relationship between robot sensor visibility and foraging efficiency. Results of simulation testing in the Webots simulation software for a swarm of five robots under a fixed time constraint reveal that this algorithm yields a 69-81% average object collection rate in low-density object distributions and a 66-75% average object collection rate in high-density object distributions. Additionally, as the robots' sensor noise levels are increased, object collection rates decrease only marginally, demonstrating the robustness of this algorithm to uncertainties in sensor measurements.

LITERATURE REVIEW

Swarm robotics is the study of decentralized multi-robot systems that involve robots with simple, limited functionalities. Such limited capabilities include local sensing, low processing power, and limited inter-robot communication [1]. The collective behavior of a swarm robotic system is a result of interactions between robots and their environment. Swarm robotic systems can partake in tasks such as pattern formation, foraging, and collective mapping [1]. Successful coordination in a swarm robotic system can have a variety of real world implications such as waste or post-disaster clean-up, search-and-rescue, mining, and sample collection [1,8].

A primary source of inspiration for swarm intelligence research is the collective behavior of natural swarms, such as ant colonies and fish schools [8]. These natural swarms show evidence of collective path planning, construction of nesting sites, task allocation, and other complex behavioral phenomena [8]. Tan et al. reviews the behavior of several types of natural swarms, including bacteria colonies, which exchange molecular signals as a form of communication, and ant colonies, which communicate indirectly through the use of pheromones [8]. Tan et al. also compares swarm robotic systems to other multi-agent systems and describes how these multi-agent systems are often centralized and heterogeneous, meaning that they often consist of a variety of different robots uniquely designed and programmed to complete separate tasks. The contrast between these multi-agent systems and swarm robotic systems, which are homogenous, only consisting of one type of robot, suggests that effective communication algorithms, either direct or indirect, are necessary to compensate for the loss in heterogeneity and centralization that is present in multi-agent systems [8]. Furthermore, examination of several communication algorithms for swarm systems, such as random walk, reinforcement learning, pheromone stigmergy, and occupancy mapping algorithms reveal that a major focus of current swarm research is the implementation of better swarm communication and coordination algorithms [1,3-7,11].

This paper addresses a subfield of swarm robotics that is concerned with coordination and communication algorithms. The goal of research in this subfield is to acquire knowledge that will lead to the development of effective communication algorithms and algorithmic innovations aimed at improving the efficiency and large-scale applicability of swarm systems, especially when robot capabilities are low. By discovering new forms of indirect communication for swarm systems, resources can be better allocated toward improving software and developing efficient swarm algorithms rather than being used for implementing advanced hardware aspects, such as advanced sensory and signal processing technologies. This will allow swarm systems to become more applicable to real world tasks, such as search-and-rescue and path planning, while remaining decentralized and retaining their limited functional capabilities [8].

Jeong et al. explores the concept of hitting time in swarm systems following a correlated random walk model [1]. This study quantitatively analyzes the average time for a subset of a swarm of robots to reach a target location, and simulation testing using the ARGoS software reveals that using larger swarm sizes yields greater scalability and smaller hitting times. In addition, this study finds that larger environment sizes yield lower overall margins of error in estimated hitting time distributions. This source also introduces the concept of representing a swarm's environment as a large-scale grid, allowing for discrete robot movement between adjacent grid cells. However, rather than utilizing a traditional square grid, Jeong et al. uses an equilateral triangular grid to represent the environment of the swarm system. By doing so, distances between neighboring nodes are all equal, unlike the case with diagonals in a square grid. This suggests that the allowed motion patterns of robots can impact swarm performance [1].

On the other hand, Yogeswaran et al. utilizes a reinforcement learning algorithm to address the problem of exploration vs. exploitation in a swarm foraging task [6]. In reinforcement learning, a robot learns through trial and error, along with reward and punishment, to maximize the total reward earned through the completion of a task. An effective algorithm will result in an optimal or nearly optimal sequence of states to minimize punishment and complete a task with the greatest efficiency and reward. The foraging task designed by the authors involves searching and retrieving target objects in a fixed time frame while maximizing object collection and environment exploration. Yogeswaran et al. designed a policy called FIFO-list, which maintains a list of recently visited states by each robot and initially prioritizes exploration of the environment to obtain a large amount of data regarding the unfamiliar environment. Once the exploration phase has been completed, the swarm gradually begins to prioritize "exploitation," which refers to the algorithm using the previously gathered data to make optimal decisions on the swarm's future actions. This study also introduced the idea of having multiple robot states; in this case, the four possible states were "Searching," "Grabbing," "Homing," or "Depositing." Each swarm robot completes the namesake task while in that state. Returning to the review by Tan et al., this idea of designating each robot into a different state allows for effective coordination in a homogeneous system, compared to the heterogeneous robot functionality required in multi-agent systems [6,8].

Additional studies utilized artificial pheromones or neural networks to implement stigmergic communication in their swarm simulations. In relation to swarm coordination and communication, a neural network is a method that allows robots to process sensory input much like the human brain. Stigmergy is a method of indirect communication in which the trace of an action left on a medium stimulates the performance of a subsequent action. Song et al. combines these two methods in their novel foraging algorithm based on virtual pheromones and a neural network chain, which uses previous actions of robots to determine subsequent motions of swarm robots, similar to the study conducted by Yogeswaran et al. [7,8]. However, Song et al. uses pheromone "markers," whereas the reinforcement learning model utilizes robot state values. Each node of the neural network outputs a value corresponding to the density of the pheromone "marker" at that location. Pheromones diffuse into adjacent nodes of the neural network chain depending on the value at that node, and regions of high density pheromones communicate to the swarm that the region has already been explored and suggest subsequent robots to forage in less explored areas of the environment [7]. In other words, pheromones and a neural network can act as markers on a grid to indirectly provide information to the swarm regarding the best possible routes that will lead to desired locations or target objects. Simulation testing resulted in accurate maps of pheromone trails, which led directly to target food sources. Additionally, the authors of this study showed that optimal foraging patterns could be derived using mathematical analysis of these maps [7]. Figure 1 below shows a sample pheromone concentration map generated from a simulation trial in this study. The blue and red lines indicate highly concentrated pheromone trails.

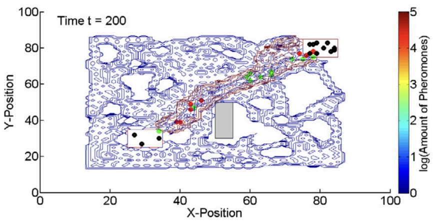

Figure 1

While the pheromone stigmergy model proves to be very useful for mapping the swarm's environment and determining optimal strategies, a study by Hunt et al. reveals that pheromone stigmergy is less effective in higher density food distributions and swarm sizes, as the average deviation in coverage time between the stigmergy-model and a traditional random walk model decreases as food density and swarm size increases [5]. A primary cause of this is due to over-deposition of pheromones, which leads to a decrease in useful information regarding the locations of target objects and an increase in overall sensor measurement uncertainty. As the number of objects and entities in a swarm foraging scenario increase, stigmergic mechanisms struggle to perform well unless the decay rate of pheromones is greatly accelerated. Doing so would prevent the pheromone map of the environment from piling with clusters of data that cannot be utilized. The authors also employed a unique method for comparing the two algorithms, as they used physical robots and ambient light levels to test their stigmergy-based foraging algorithm. Rather than only testing their algorithm on simulation, physical testing provided insights on real-world mechanisms for depositing such "pheromones" [5]. Nevertheless, the results of this study suggest the need for another communication model that can sustain a high level of foraging efficiency and success rate in noisy environments.

Regardless of the communication algorithm being employed, all swarm robots have several key functional aspects, one of them being sensors. Collins et al. describes how the ultrasonic distance sensor is one of the most useful sensors due to its lost cost and ease of use, yet they are still susceptible to a phenomenon known as specularity, which results in uncertainty in sensor data [18]. This specularity contributes heavily to sensor noise levels. From this phenomenon, however, arises a significant strength of occupancy grid mapping, as this family of computer algorithms was designed to "address the problem of generating maps from noisy and uncertain sensor measurement data" [15].

Alberto Elfes first introduced the idea of occupancy probability mapping [12]. This paper will explain this algorithm in greater depth in the next section, but occupancy grid mapping is essentially a method of generating maps of an environment by representing each location in the environment through an array. Elements in the array represent the certainty to which the occupancy of that cell is known, and as time progresses, the algorithm maps out the location of objects and obstacles throughout the environment [12,15].

Additionally, a study by Lajoie et al. explores a collaborative simultaneous localization and mapping algorithm for a swarm system, abbreviated C-SLAM [20]. While SLAM is a useful tool for mapping an environment, the authors of this study utilized an inter-robot loop closure prioritization approach, which allows the mapping algorithm to map out the boundaries of an environment and then work its way inwards. Using multiple robots to accomplish this SLAM task allows for the robots to perform as a single swarm, strengthening communication and coordination within the system [20]. Yet, while localization and mapping algorithms are a common focus in this field, there is yet to exist an effective foraging algorithm that can also map the surrounding environment using occupancy grids. Many past studies mentioned that a limitation of their model was the issue of noisy measurements and sensor data, and an occupancy-based foraging framework, given the robustness of occupancy maps to background noise, presents a viable solution to this specific gap in the field of swarm communication and object collection. This paper aims to explore the viability of occupancy grid mapping as it applies to swarm foraging and object aggregation.

In addition to introducing an occupancy-based swarm foraging algorithm, this study aims to determine a relationship between distance sensor visibility, or sensor noise levels, and swarm efficiency as a swarm system completes a foraging task under an arbitrary time constraint. In a real-world object-collection or cleanup task, an understanding of the relationship between sensor visibility and swarm efficiency, would ultimately provide key insights on the robust algorithms required for a swarm system to successfully complete a search or foraging task while simultaneously mapping its environment, especially when time constraints are a factor.

METHODS

Quantitative methods were employed in developing an occupancy grid-based algorithm for swarm foraging. This project consists of two major parts: creating an occupancy grid mapping-based swarm simulation algorithm and analyzing simulation data. The end result is a model that simulates a swarm robotic system as it completes an object foraging task and from this simulation data, a graphical representation of the relationship between sensor noise level and the number of objects successfully collected. This requires the use of quantitative methods, as with the majority of swarm robotics research. Since the ultimate goal is to examine the effect of changing sensor visibility on the number of objects successfully collected by the swarm system, an extensive quantitative analysis is necessary to accurately portray the results of this project. Variables such as standard deviation of noise levels and object distribution density will be varied, and the quantitative results will be graphically represented.

Additionally, in order to program the logic behind this occupancy grid-based algorithm, it is necessary to understand the mathematical concepts and equations behind the probability map, including Bayesian filters, the Markov assumption, and an algorithm to translate sensor data into an approximation of an object's overall position. As mentioned previously, some variables are manipulated, whereas other variables, such as swarm size and arena size, are kept constant, further supporting the need of a quantitative method. A primarily qualitative method would be insufficient in communicating the relationship between sensor visibility and foraging efficiency determined by this experiment, as numerical values are crucial to express the results and their significance.

I. Occupancy Grid Mapping Algorithm

The foundational concept behind this research is that of occupancy grid mapping, which was first explored by Alberto Elfes using Bayesian filters [12]. Occupancy grid mapping is a set of probabilistic computer algorithms that addresses the problem of generating maps of the environment from noisy sensor data [15]. Since this research is concerned with foraging objects, it makes sense to use occupancy grid mapping to determine the certainty to which objects are located in certain grid locations in an environment, especially when sensor noise is prevalent. In the simulation, all cells of the occupancy mapping array initialize at 0.5, signifying that there is a 50% of an object being located at that cell. As a grid location is observed more frequently by the robots, the occupancy probability value of that location in the map increases toward 1. Once a threshold probability is met, in this case, 0.8, the occupancy grid map will indirectly communicate to all nearby swarm robots that an object is located in that grid square, and nearby robots will collect objects in those high-probability locations.

The *posterior probability* over the map describes the overall occupancy probability values in the map by updating previous probability information gathered from sensor data [15]. The posterior probability over a map *m* is given by

$$p(m|z_{1:t}, x_{1:t}),$$

where $z_{1:t}$ represents a grid cell $x_{1:t}$ being occupied, signified by the number "1" in the subscript. However, since a map can contain a large number of cells, it is more practical to determine individual posterior probabilities over each cell and take the product of the cells to express the posterior probability over the map, as described by the equation

$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(m_i|z_{1:t}, x_{1:t}) .$$

A Bayesian filter and the Markov assumption are used in the probability updating algorithm. The Bayesian filter essentially uses the prior probability of a cell to update the probability given new sensor data about the occupancy of that cell. In the occupancy map updating algorithm, the Markov assumption is used in combination with the Bayesian filter to allow the new probability value of a cell to depend solely on its prior state, thus simplifying computations [12].

II. Simulation Logic

At all times, the robots in the swarm system are either in two implicit states: EXPLORER or FORAGING. When the group of robots begins to move throughout the environment, they are initially in the EXPLORER state, as exploring the environment is necessary to determine the possible locations of target objects. Returning to the study by Yogeswaran et al., as more and more values in the occupancy map pass the threshold value for being observed, robots will begin to move toward those locations and those objects will be "collected," or removed from the environment [6]. These objects will be marked as collected and a counter will keep track of the number of objects collected. This will mark the gradual transition into the "exploitation" phase, or, in the case of this algorithm, the FORAGING state. Once a robot collects an object, it will return to the EXPLORER state and repeat this sequence of transitions until the time limit is reached. At this point, data on the number of objects collected is achieved. Additionally, at the end of each trial, a digital occupancy map of the swarm's environment can be generated, which visually depicts the locations of objects in the environment. One can examine these maps throughout the course of a simulation trial, but since the method utilized is a quantitative method, qualitative analysis of these maps will not be particularly useful for the purpose of this study.

III. Simulation Software and Specifications

The Webots software was used to implement and simulate the swarm foraging algorithm. Each robot has a robot controller, which controls the actions of the robot during the simulation. Since swarm systems are homogenous, each robot in the swarm used an identical robot controller. The controller consists of the occupancy map updating and line drawing algorithms, as well as general robot movement logistics. In order to prevent collisions between robots, a simple method of detecting oncoming robots was implemented into the algorithm. Lastly, for objects to be physically removed from the simulation once "foraged," each robot controller required the use of supervisor actions in Webots, as this allows robots to modify their environment in a programmatic way.

The E-puck robot is the robot utilized in this simulation, for several important reasons. Firstly, the E-puck robot consists of proximity sensors and position sensors, which are necessary for collecting object location data. Secondly, the E-puck robot is a small and compact robot, so it is on a more similar size scale to the target objects. The E-puck robot in Webots was based on the physical E-puck robot created for experimental and teaching purposes, since it has unlimited extension possibilities, but for the sake of limited functionality, and since this study is simulated digitally, no physical E-puck robots are used in this experiment.

RESULTS

Table 1 below describes the six different rounds of testing that were conducted in this experiment. Each trial in each round stopped at $t = 100$ s and involved 5 swarm robots. In Rounds 1-3, the standard deviation in sensor noise level, or uncertainty in sensor measurement was varied from 5% to 10% to 15% for low-density object distributions (20 objects). This was repeated in Rounds 4-6, except for high-density object distributions (40 objects).

Round	Standard Deviation of Noise	Number of Objects
1	0.05	20
2	0.10	20
3	0.15	20
4	0.05	40
5	0.10	40
6	0.15	40

Table 1

Figure 2 below depicts a sample low-density object distribution, containing 20 cuboidal target objects in a 16 meter by 16 meter grid arena. Figure 3 below depicts a sample high-density object distribution, containing 40 objects. In both figures, the five E-puck robots are located at the four corners and at the center of the grid arena. Pseudorandom object distributions were used throughout all rounds, meaning that the objects were always randomly distributed, but the same random distributions were used in the corresponding trials of each of the first three rounds and each of the last three rounds. As a result, ten unique low-density distributions of objects were used throughout Rounds 1-3, and ten unique high-density distributions of objects were used throughout Rounds 4-6.

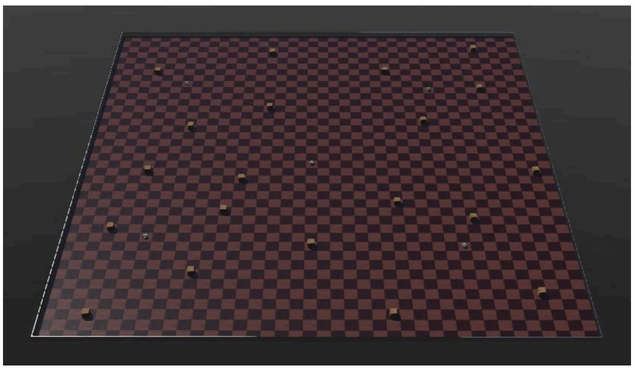

Figure 2

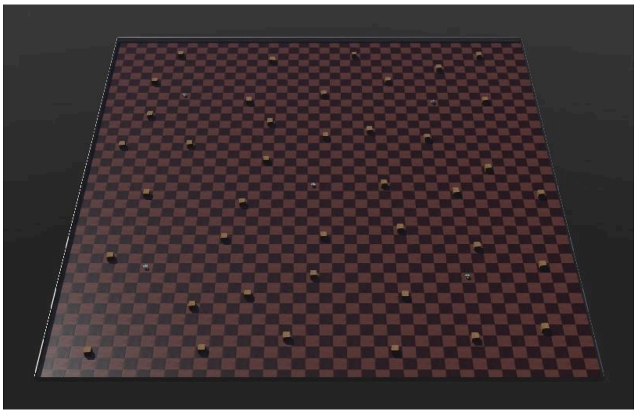

Figure 3

For each round, 10 trial simulations were run, and the averages and standard deviations in the number of objects collected in each round are numerically and graphically depicted in Table 2 and Figures 4 and 5 on the right.

Round	Average Number of Objects Collected	σ
1	16.1	1.5
2	14.7	1.6
3	13.7	1.3
4	30.1	2.3
5	28.4	2.1
6	26.2	2.0

Table 2

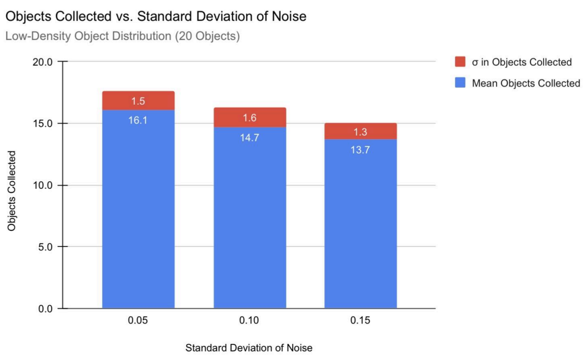

Figure 4

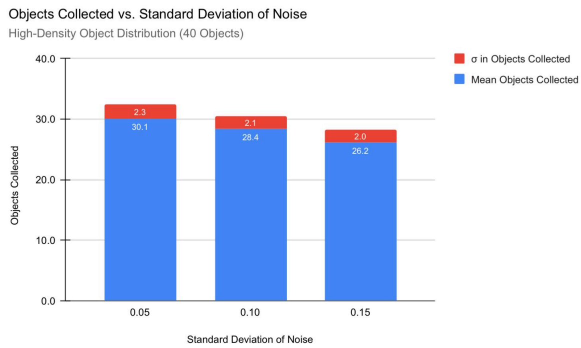

Figure 5

DISCUSSION

I. Sensor Visibility vs. Number of Objects Collected - Rounds 1-3

According to Table 2 and Figure 4, there was a direct relationship between distance sensor visibility and number of objects collected by the swarm during the 100 s time period. The swarm system with a 5% noise level in an environment of 20 target objects will collect 14-18 objects 68% of the time. These numbers decrease to 13-17 objects and 12-15 objects for Round 2 and Round 3, respectively. The 70-90% object collection rate in Round 1 establishes that the occupancy grid-based algorithm is effective even with low sensor functionality and relatively large measurement uncertainty, and the slight decrease in collection rate from Round 1 to Round 3 proves that the rate at which the swarm generates the occupancy map of its environment decreases steadily with decreasing visibility. Therefore, it can be concluded that the efficiency of the swarm system arises primarily from its ability to quickly generate and update the occupancy map. When a relatively high proportion of the occupancy map contains values that are past the threshold values, indicating that most of the objects have been located, the remaining simulation time is spent primarily foraging and collecting the objects. Furthermore, since object aggregation time for a single object remains relatively constant, the time constraint has a greater impact while robots are in the EXPLORER rather than the FORAGING state, introducing the need for future research that focuses on developing faster occupancy grid mapping algorithms. Lastly, while there was a negative relationship between noise level and number of objects collected, this relationship was nonlinear, as doubling the noise level did not yield half the number of objects collected. This results from the nonlinear effect of noise on measurement uncertainties as distance increases.

II. Sensor Visibility vs. Number of Objects Collected - Rounds 4-6

According to Table 2 and Figure 5, there was a direct relationship between distance sensor visibility and number of objects collected by the swarm during the 100 s time period. The swarm system with a 5% noise level in an environment of 40 target objects will collect 27-33 objects 68% of the time. These numbers decrease to 26-31 objects and 24-29 objects for Round 5 and Round 6, respectively. The 68-83% object collection rate in Round 4 establishes that the occupancy grid-based algorithm is effective even with low sensor functionality and relatively large measurement uncertainty, and the slight decrease in collection rate from Round 4 to Round 6, similar to Rounds 1-3, proves that the rate at which the swarm generates the occupancy map of its environment decreases steadily with decreasing visibility. Once again, it can be concluded that the efficiency of the swarm system arises primarily from its ability to quickly generate and update the occupancy map. Additionally, while there was a negative relationship between noise level and number of objects collected, this relationship was nonlinear, as doubling the noise level did not yield half the number of objects collected. This results from the nonlinear effect of noise on measurement uncertainties as distance increases. The slight decrease in object collection rate in a higher density object distribution proves a result by Hunt et al. in their analysis of pheromone stigmergy in high-density environments [5]. The natural increase in noise and the clustered environment in a high-density object distribution prevents as high of a foraging rate as in Rounds 1-3. However, increasing swarm size may solve this issue.

CONCLUSION

This study accomplishes two main tasks. The results of this project demonstrate that occupancy grids are a viable tool in developing an efficient swarm foraging system. The use of Bayesian filters and the Markov assumption proved to be effective in approximating object locations in noisy environments, as seen by the 70-90% object collection rate in Round 1 of simulation testing and 68-83% object collection in Round 4. Especially under an arbitrary time constraint of 100 s, which would be a relatively small time limit for a real-life object aggregation task, occupancy grids prove very effective at utilizing occupancy maps of the environment to make decisions on which objects to collect.

The second goal of this study was to determine a relationship between sensor visibility and number of objects collected by the swarm. The results of extensive simulation testing with three different noise levels (5%, 10%, and 15%) as well as two object densities (20 objects and 40 objects) revealed that there was a positive relationship between visibility and object collection, albeit the relationship was nonlinear due to the nonlinearly scaling nature of sensor noise with increasing visibility distance. In addition, throughout all rounds of testing, it was observed that the rate at which occupancy maps were generated heavily impacted the performance of the swarm foraging algorithm as a whole when subjected to time constraints. In other words, a major limitation of this study is the computational and grid-updating speed of the occupancy mapping algorithm. Hence, future research in this field should evaluate the effectiveness, and more precisely, the speed and efficiency, of different occupancy mapping algorithms, including those that do not utilize Bayesian filters. In its analysis of different mapping techniques, Collins et al. presents a comparative study of past occupancy grid mapping algorithms, including the one developed by Elfes using a Bayesian framework, the one developed by Thrun using a neural network, and two others developed in subsequent years utilizing an enhanced Bayesian framework and a Forward Modeling approach [14,15,18,19]. While this comparative study analyzes the accuracy of several occupancy mapping frameworks in generating maps of an environment, additional research is needed to develop faster environment mapping algorithms, especially since this article only analyzes models prior to 2003.

The unique model presented in this paper, being one that combines swarm foraging mechanics with an occupancy grid mapping framework, contributes toward unlocking new possibilities for future research. While previous literature suggests that occupancy mapping has not been typically used for foraging purposes, the idea of simply locating multiple objects throughout an environment using probabilities and sensor data has other applications in different tasks. For instance, occupancy maps could play an even more important role in foraging or search-and-rescue tasks in cluttered environments, such as environments in which obstacles are present. Furthermore, a possible direction for future research would be the development of an algorithm capable of distinguishing between objects that need to be retrieved and obstacles; this would ultimately lead to an intelligent swarm system that could mark the locations of permanent obstructions while also providing information about the location of target objects to the rest of the swarm. In most real-world scenarios, such as post-disaster cleanup, not all objects are retrievable, and especially in obstructed environments, knowing which objects are of importance and which objects are obstacles would strengthen the role of environment mapping in such scenarios [1,8]. Additionally, converting occupancy array data into images and quantitatively analyzing error in occupancy map cells would provide insights on more efficient and effective ways of updating occupancy maps to obtain more accurate depictions of a swarm's environment.

Ultimately, the results of this study conclude that there is a clear relationship between sensor visibility and swarm efficiency for a swarm system that utilizes environment mapping while completing a foraging task. The existence of such a relationship sets up future field research dedicated toward finding the optimal sensing capabilities and faster mapping algorithms required for a swarm system to complete a search or foraging task in the most efficient manner possible. Benefits of determining these optimal functional capabilities include lower costs of building multi-robot systems, smarter allocation of resources toward software aspects of swarm robots rather than hardware aspects (as swarm robotics is defined by its focus on robots with limited functionality), and most importantly, swarm systems that are better equipped to handle real-world time constrained tasks.

REFERENCES

[1] Jeong, M., Harwell, J., & Gini, M. (2021). Analysis of exploration in swarm robotic systems. *International Conference on Intelligent Autonomous Systems*, *16*, 445-457. https://doi.org/10.1007/978-3-030-95892-3_34

[2] Patel, M., Bandopadhyay, A., & Ahmad, A. (2021). Collaborative mapping of archaeological sites using multiple UAVs. *International Conference on Intelligent Autonomous Systems*, *16*, 54-70. https://doi.org/10.1007/978-3-030-95892-3_5

[3] Luo, Y., Guo, J., Ye, G., Wang, Y., Xie, L., Wang, X., Zhang, S., & Yan, X. (2020). Toward target search approach of swarm robotics in limited communication environment based on robot chains with elimination mechanism. *International Journal of Advanced Robotic Systems*, *17*(3). https://doi.org/10.1177/1729881420919954

[4] Pang, B., Song, Y., Zhang, C., & Yang, R. (2021). Effect of random walk methods on searching efficiency in swarm robots for area exploration. *Applied Intelligence*, *51*. https://doi.org/10.1007/s10489-020-02060-0

[5] Hunt, E. R., Jones, S., & Hauert, S. (2019). Testing the limits of pheromone stigmergy in high-density robot swarms. *Royal Society Open Science*, *6*(11). https://doi.org/10.1098/rsos.190225

[6] Yogeswaran, M., Ponnambalam, S. G., & Kanagaraj, G. (2013). Reinforcement learning in swarm-robotics for multi-agent foraging-task domain. *2013 IEEE Symposium on Swarm Intelligence (SIS)*, 15-21. https://doi.org/10.1109/SIS.2013.6615154

[7] Song, Y., Fang, X., Liu, B., Li, C., Li, Y., & Yang, S. X. (2020). A novel foraging algorithm for swarm robotics based on virtual pheromones and neural network. *Applied Soft Computing*, *90*. https://doi.org/10.1016/j.asoc.2020.106156

[8] Tan, Y. & Zheng, Z. (2013). Research advance in swarm robotics. *Defence Technology*, *9*(1), 18-39. https://doi.org/10.1016/j.dt.2013.03.001

[9] Alers, S., Ranjbar-Sahraei, B., May, S., Tuyls, K., & Weiss, G. (2013). Evaluation of an experimental framework for exploiting vision in swarm robotics. *ECAL 2013: The Twelfth European Conference on Artificial Life*, 775-782. https://doi.org/10.7551/978-0-262-31709-2-ch111

[10] White, N., Harwell, J., & Gini, M. (2019). Socially inspired communication in swarm robotics. *2019 AAMAS Conference*. https://doi.org/10.48550/arXiv.1906.01108

[11] Türkler, L., Akkan, T., & Akkan, L. O. (2022). Usage of evolutionary algorithms in swarm robotics and design problems. *Sensors*, *22*(12). https://doi.org/10.3390/s22124437

[12] Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, *22*(6), 46-57. https://doi.org/10.1109/2.30720

[13] Mu, X., Yue, G., Zhou, N., & Chen, C. (2022). Occupancy grid-based AUV SLAM method with forward-looking sonar. *Journal of Marine Science and Engineering*, *10*(8), 1056-1070. https://doi.org/10.3390/jmse10081056

[14] Zhang, J., Wang, X., Xu, L., & Zhang, X. (2022). An occupancy information grid model for path planning of intelligent robots. *International Journal of Geo-Information*, *11*(4), 231-252. https://doi.org/10.3390/ijgi11040231

[15] Bagnell, D. Occupancy maps. Carnegie Mellon University. https://www.cs.cmu.edu/~16831-f14/notes/F14/16831_lecture06_agiri_dmcconac_kumarsha_nbhakta.pdf

[16] Chaves, R., Rezeck, P., & Chaimowicz, L. (2019). SwarMap: Occupancy grid mapping with a robotic swarm. *2019 19th International Conference on Advanced Robotics (ICAR)*, 727-732. https://doi.org/10.1109/ICAR46387.2019.8981640

[17] Marconi, P. L. (2020). Simple foraging and random aggregation strategy for swarm robotics without communication.

[18] Collins, T., Collins, J. J., & Ryan, D. (2007). Occupancy grid mapping: An empirical evaluation. *2007 Mediterranean Conference on Control and Automation*, 1-6. https://doi.org/10.1109/MED.2007.4433772

[19] Ramachandran, R. K., Kakish, Z., & Berman S. (2020). Information correlated Lévy walk exploration and distributed mapping using a swarm of robots. *IEEE Transactions on Robotics*, *36*(5), 1422-1441. https://doi.org/10.1109/TRO.2020.2991612

[20] Lajoie, P. & Beltrame, G. (2023). Swarm-SLAM: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. https://doi.org/10.48550/arXiv.2301.06230