

Improvement of Obstacle Avoidance Algorithm in Navigation for Swarm Robotics using MATLAB

Girish Balasubramanian, Senthil Arumugam Muthukumaraswamy

Abstract— This paper discusses the recent developments of obstacle avoidance algorithms being used in the field of swarm robotics for various applications with a focus on navigation. This includes reviews of work done for algorithms in the area of object detection, multi robot navigation, obstacle avoidance among others. The paper also discusses how impactful the work done so far is to the field and what are the challenges being faced that hinder progress. The paper also showcases some simulation results obtained using MATLAB for obstacle avoidance in a multi robot environment. The paper then continues to discuss work that is expected to be done in the coming years.

Index Terms— AI, Artificial Intelligence, Swarm Robotics, Algorithms, Machine Learning, Obstacle Avoidance.

I. INTRODUCTION

Swarm robotics is a relatively under developed concept with tremendous potential for real world use. The ability to carry out dangerous and high risk missions or tasks without human intervention is an extremely beneficial capability for technology. Further development in the field of swarm robotics is necessary to practically apply these concepts. This paper showcases the development of algorithms that aim to improve swarm robots in areas of object detection and navigation. Most of the recent work has to do with improvements of algorithms for singular robots. These algorithms can be studied and extended to improve multi robot environments or in this case, swarm robots. Moreover, these swarm robots can be extended to real time applications that would make use of these algorithms to assist in many industries.

The first algorithm that was discussed involved object detection. Robust detection of objects is required for a robot system in order to perform optimally in various applications [1]. This usually involves at least a single camera connected to the robot along with some image processing algorithms and filters to process the data. This paper focuses on the aspect of object detection logic as opposed to the image processing. Recent work regarding object detection logic was researched and potential areas of improvement were identified. Lee et al. (2012) worked on a vision-based object detection system that was applied to underwater robots [2]. The conclusion arrived during the work highlighted that the quality of camera and image obtained is a big factor as this can skew results negatively. Ekvaal et al. (2007) used a mobile robot to navigate a domestic environment in which predefined objects

were identified and stored within a generated map [3]. This paper showcased very useful results which can be used to develop a mapping algorithm combined with object detection. One of the key findings during research on object detection algorithms is how there is a lack of published work regarding object detection in a swarm robotics environment. Details of further research are highlighted in the literature review section of this paper.

Obstacle avoidance was the next area of focus. All mobile robots need to feature a form of obstacle avoidance system [4]. This can be anywhere from simple stop and go systems to complex algorithms that can detour around any obstacles to reach the desired destination. In this paper, various implementations of obstacle avoidance techniques were studied. Bilbeisi et al. (2015) developed a multi robot path planning algorithm which incorporated particle swarm optimization (PSO) as the moving path planner and used an agoraphilic algorithm as the obstacle detection algorithm [5]. They tested their algorithm in swarms of various sizes while differing the difficulty of the environment. Their results demonstrated that the algorithm is scalable. In all their tests, the robots were able to successfully reach the goal. Marwah et al (2016) developed an obstacle avoidance system that used low cost IR sensors to detect and navigate around obstacles for mobile robotics systems [6]. Further details regarding this work is shown in the literature study section of this paper.

The next area of interest was in multi robot navigation systems. Swarm robots are a type of multi-robot system that work together as a team to carry out certain tasks. Various multi-robot systems were studied to see how it could be applied to swarm robotics. Reid et al. (2000) used multiple mobile robots to navigate and map areas. They were able to develop a form of co-ordination between robots by having the robots communicate with a central system [7]. The robots would estimate the area of travel while the central system would assign different tasks if more than one robot had too similar of an estimate. While the experiment was slightly outdated, it had some interesting methods which can be still used in the current day. Simon et al. (2016) developed a co-operative mapping, navigation and communication framework assuming unknown initial positions of robots [8]. They evaluated use of this framework in real-world scenarios with three robots. Details regarding multi navigation systems and its use in swarm robotics are discussed in the literature study section of this paper.

The rest of the paper is organized into 4 additional sections. Detailed review of previous work in this field is discussed in Section 2 (Literature Review). Simulation results are then shown and discussed in Section 3 (Results and Discussion) Following this, Section 4 (Future Scope) discusses the future of swarm robotics and how it can be applied to real life applications today along with any challenges that may arise. Lastly, the paper is concluded in Section 5 (Conclusion) with

Girish Balasubramanian, Department of Engineering and Physical Sciences, Heriot Watt University, Dubai, United Arab Emirates, +971509444328.

Senthil Arumugam Muthukumaraswamy, Department of Engineering and Physical Sciences, Heriot Watt University, Dubai, United Arab Emirates, +971555518742

closing thoughts on swarm robotic algorithms and how they can impact society.

II. LITERATURE REVIEW

Jung et al. (2004) published a paper that focused on using a single camera on a mobile robot to detect moving objects in an outdoor environment. They highlighted that mobile robots have many practical applications in outdoor areas. This included street cleaning, surveillance as well as traffic policing. They also mentioned that it was fairly difficult for a robot to navigate in a populated outdoor area. This was due to many factors such as terrain, motion of people and objects among others. There are many objects can move even faster than the robot. Therefore, motion detection and ability to avoid collision are some of the most fundamental skills that a robot needs to effectively function [1].

In this paper, Jung et al. (2004) proposed to detect moving objects using one camera mounted on a mobile robot. The motion detection process is carried out in two steps. The first was the ego motion compensation of camera images and the second being the position estimation process that was performed using a Bayes filter. An adaptive particle filter was also utilized for iterative estimation. The process can be seen in Figure 1. Their algorithm was tested on different robotic platforms. The robots were successfully able to detect objects in motion in various situations.

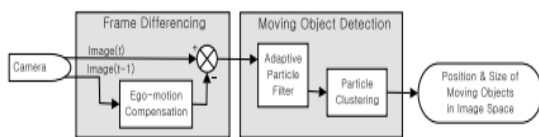


FIGURE 1. PROCESS OUTLINING MOVING OBJECT DETECTION ALGORITHM. [1]

Lee et al. (2012) published a paper that tackled object detection under water. They used the developed algorithm in autonomous underwater vehicles (AUVs). The applications of their work was to perform deep sea inspections for infrastructure, environmental monitoring as well as oceanographic mapping. One of the greatest challenges faced in this paper was restoring colour to underwater images obtained through the camera on the AUV. As the light travels in water, it is exponentially attenuated and the light reflected back from the object is a fraction of what is usually emitted. This causes images with poor contrast and haziness [2]. In this paper, Lee et al. (2012) proposed an algorithm to restore colour information and therefore the object detection results of the processing algorithm. They classified the image recovery methods into two types, image restoration and image enhancement. The focus of this paper was to perform the former. They were able to obtain many quantitative and qualitative results with their algorithm. However, they concluded that further improvement is needed for practical use. This paper highlighted that if robots are to be applied for underwater applications, it is very important to focus on improvements regarding underwater object detection algorithms. Ekvall et al. proposed a technique to perform object detection as well as mapping. They used an approach that involved identifying objects and adding them to a map generation by a simultaneous localization and mapping (SLAM) algorithm. They used a predefined set of objects to

train the robot by building up a database to improve detection results. Figure 2 shows a map generated by the test robot used in the paper. They used an object recognition algorithm to classify objects based on Scale Invariant Features (SIFT) methods [3].

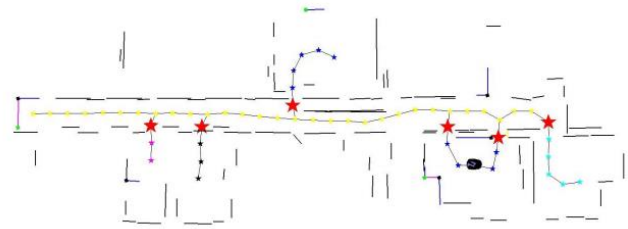


FIGURE 2. PROCESS OUTLINING MOVING OBJECT DETECTION ALGORITHM. [3]

There were some limitations to the algorithm they applied. Object features could only be detected reliably provided that the object in question occupies a large part of the image being examined by the algorithm. They used a pan-tilt and zoom camera to bypass this limitation. By zooming into multiple areas with potential objects, reliable object identification could be made. However, downsides to doing this include large processing times as every probable area needs to be checked for an object using the algorithm. This paper showcased some extremely useful results that can be improved upon for swarm robotics.

Bilbeisi et al. (2015) published a paper discussing a multi-robot path planning and obstacle avoidance algorithm. They use particle swarm optimization (PSO) find the best path for the robots to take by communicating with each other. They said that path planning is categorized into two different types. This was online path planning and offline path planning. In offline path planning, the robots are assumed to know the environment and all obstacles present. This is unrealistic for real world applications as obstacles can be constantly changing. Online path planning assumes that no knowledge of the environment is known to the robots. This makes it a challenge for the robots to reach a certain target efficiently. The algorithm used for obstacle avoidance in this paper is called the Agoraphilic algorithm (AG). This algorithm is a reactive path planning algorithm which uses two main steps: [5]

- The robot measures the distances to the surrounding obstacles in order to generate a sectorized self-centered polar map that represents the distance profile of the surrounding environment, called the Free-Space Histogram (FSH)
- The initial forces generated by each sector in the FSH are directly proportional to the square of the distance. Then, using the weighting factors generated by a fuzzy logic controller, the forces in each sector are summed together to obtain the total resultant force towards the target location through the free space, which the robot will follow.

The amount of computations required to carry out this algorithm is directly proportional to the sectors used. It is also to keep in mind that more sectors also provide better detection of free space around a detected obstacle. Figure 3 showcases the steps outlined in the Agoraphilic algorithm.

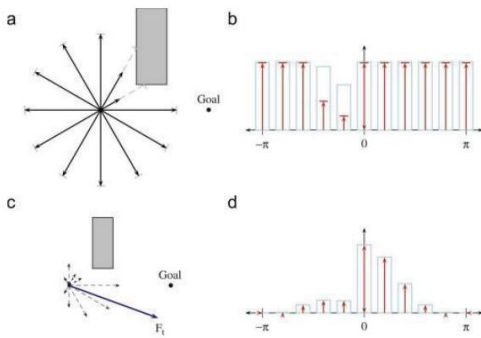


Figure 3. Steps carried out in the agoraphilic algorithm. (a) robot measuring distance to surrounding obstacles (b) free space histogram (c) weighting factors used to determine total resultant force which robot is to follow (d) adjusted histogram [5].

They carried out simulation experiments under various scenarios consisting of different obstacle amounts. They assumed that the robots need to reach a specified target location at the center of the test environment. The results they obtained showed that a larger number of iterations are required as more obstacles are added. The work showcased here successfully proved that PSO combined with AG can be used for obstacle avoidance and the algorithm is scalable for multiple robots and obstacles. However, the practical uses of the algorithm was not shown and is something that could be improved upon. Moreover, in a practical setting, the target area might not always be known.

Marwah et al. (2016) proposed a more low cost approach to obstacle avoidance by using infrared sensors (IR) which uses minimal calculations to be applied in real time applications. This is more applicable to the experiments that can be performed with the Elisa - 3 robot as they are also equipped with IR sensors. Their experiments consisted of a line following task while avoiding any obstacles in the path. The test robot they used was the e puck miniature robot which is similar to the Elisa- 3 robot used for this project. They employed IR sensors to detect both the ground as well as distance in front and around the robot. The inputs to these sensors are fed to the line following algorithm (LFA) and collision avoidance algorithm (CAA) respectively. both these algorithms work together to ensure that the robot follows the line while turning when necessary to avoid obstacles and returning to the path. Figure 4 showcases the block diagram of the experiment. [6]

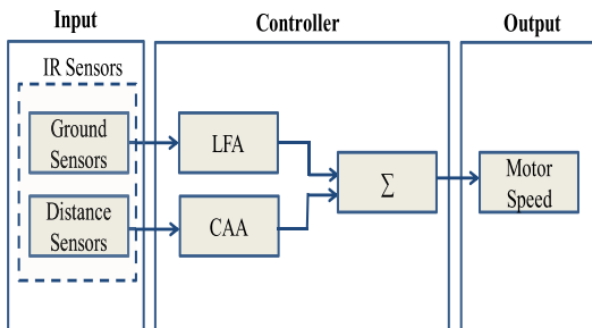


FIGURE 4. BLOCK DIAGRAM OF THE LINE FOLLOWING ALGORITHM [6]

The experiment involved using a Webots simulator tool to generate an oval racetrack type line for which the robot must follow. Two cylindrical obstacles were placed on the path for the robot to maneuver around. The distance sensors have a

range of 0 to 2000. For the purpose of the experiment, any value above 1000 was set to be an obstacle and hence must be avoided. The test environment that was used is shown in Figure 5.

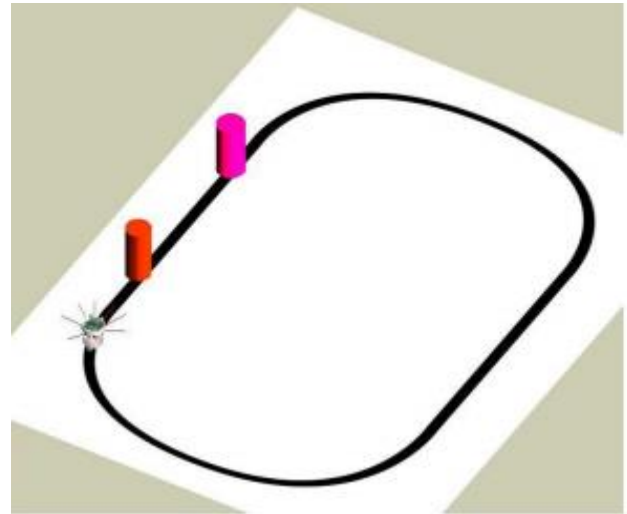


FIGURE 5. TEST ENVIRONMENT USED FOR THE WEBOTS SIMULATOR [6]

The results shown were able to successfully prove that the robot was able to follow the path, avoid the obstacles and return to the path. Further details of the simulation times can be found in [6]. However, their work did not showcase the algorithm in a real time robot as it was only a simulation. This work can be improved upon by testing it in real time robots as well as combining it with multiple robots for use in swarm robotics.

Reid et al. (2000) tried to address a problem regarding exploration and mapping in an unknown environment by multiple robots. The approach they took involved co-ordination between the robots by allowing each robot to compute a certain amount and send the results to a centralized system. This system would then integrate the results and tries to create a global map. The allows the central system to assign different tasks to robots that overlap by trying to explore the same area. The main purpose for using multiple robots was to perform the tasks in a shorter time frame. [7]

This approach can be further broken down in how it works. Each robot constructs an estimate of the information gained and cost of travel to perform calculated task. The master program will then receive these estimates and maximize the efficiency while minimizing any overlap between robots. The approach also makes two assumptions. First, the world is assumed to be static. This means that this algorithm does not work for dynamic environments. Second, it assumes that the robots all begin in each other's field of view. The robots then each receive a sequence it's own readings. Using these readings, the robots estimate it's own position and the position on a map.

Their experiment was successfully able to create a map using three robots. However, the experiment was slightly outdated due to having some limitations that cannot apply to modern day advancements such as a static environment. The paper was however successfully able to prove that multi robot navigation performs better than using a single robot to do the same task. Moreover, applying a more advanced algorithm for swarm would involve using a decentralized system where the

robots would be able to think for themselves and communicate with each other without a main system. Simon et al. (2017) took a different approach to multi robot navigation. They focused on co-operative mapping in a circular topology. Their experiments involved multiple robots with cameras facing towards the center of a circular area. The multiple robots view the center with a different perspective. The joint observation can be optimized to be able to interpret the best scene. One of the key ideas taken from this work is the fact that multiple robots can be used not only to perform one task much quicker or more efficiently but can be used to obtain better quality results with different field of views. [8] Miguel et al. (2016) used swarm robotics to carry out marine monitoring missions. They were able to use temperature monitoring systems incorporated into their swarm units to validate results obtained through other means. Their swarm robotics team consisted of 10 multiple decentralized robots capable of navigating under water. Each robot was controlled by a Raspberry Pi 2 controller and was equipped with a GPS receiver, digital compass and a water temperature sensor. The robots move with a differential-drive system that drive twin propellers. The swarm robots can be seen in Figure 6. [10]



FIGURE 6. MARINE SWARM ROBOTICS PLATFORM [10]

Each robot was controlled by an artificial neural network (ANN) which receives sensor data as input and outputs the heading and speed. The configuration of the ANNs were optimized using the NEAT neuro evolutionary algorithm. [11] NEAT optimizes both the networks' topology by adding neurons and connections as well as adjusting weights between them. The test experiment involved assigning each robot to a random position under water within the area. Once the targeted location was reached, monitoring behavior is started and ran for a certain period of time. Once the data collection was complete, the robots return to the base station. During the entire process, the robots were able to communicate with each other (upto 40m range) and were also aware of any boundaries set preemptively to avoid going out of range. This occurs while the temperature sensor's readings were recorded. Using a multi robot swarm also allowed for redundancy in data collection as each robot traverses a sub region multiple times. The results obtained from the swarm system was compared with results obtained from other standardized means. The results showed that the swarm was effective in covering areas underwater and was quickly able to identify the temperature gradients in the area. The accuracy of the results improved over time due to the redundancy of the results which allowed for average results to be obtained.

Bakhsipour et al.(2017) proposed a new meta-heuristic optimization algorithm called Swarm Robotics Search & Rescue (SRSR) inspired by artificial intelligence in swarm robotics. Problems related to applications of swarm robotics can be categorized into two classes:

- Pattern based problems which include examples such as self organizing grids, mapping, migration, aggregation and covering a range of area.
- Object existence problems which include examples such as searching for ore veins in mining applications, locating an unknown target in an unknown environment, foraging and the key application of search and rescue for a victim in a post-disaster environment.

They claimed that swarm robotics systems are ideal for tasks requiring a large area of space. One example given was searching and collecting multiple targets in an open area. The large swarm can take advantage of parallel searching with smaller grouping within sensing ranges of the robots. A flowchart with the proposed SRSR algorithm can be seen in Figure 7. [12]

Robots are to be distributed in post-disaster areas. These robots are to be equipped with sensors capable of detecting vital signs resembling human life as the target. The robots can then maneuver around these areas to identify any human life. They were able to achieve results that surpass other types of similar algorithms in almost all cases. They claimed that this does not mean it is universally superior but it can be used to set new benchmarks.

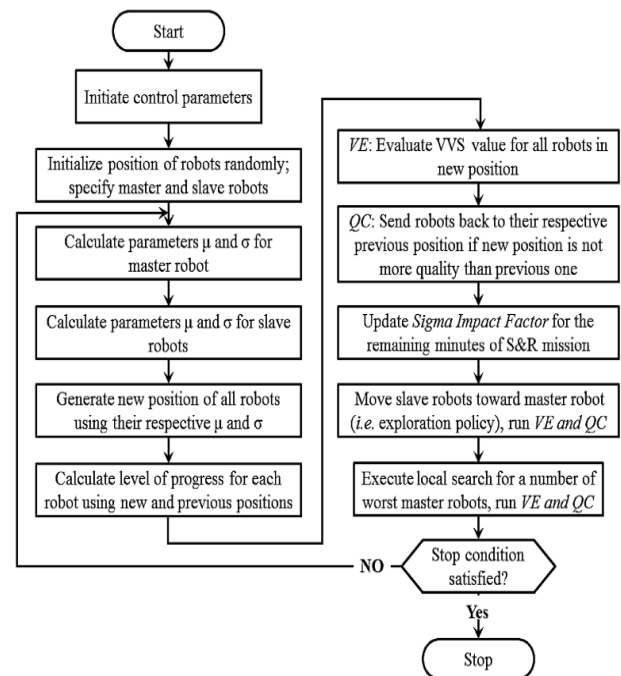


FIGURE 7. FLOW CHART OF SRSR ALGORITHM [12]

Trianni et al. proposes a way to use swarm robotics in agricultural applications. They are planning to use a swarm of unmanned aerial vehicles (UAVs) to provide automatic weed monitoring. The swarm will patrol the field, identify the presence of weeds and use mapping algorithms to generate a heat map highlighting the areas with the biggest urgency of weed infestation. This was to be done using a decentralized

control system so as to be flexible and scalable. Figure 8 shows the weed to be identified on the left and the UAV on the right. [13]



FIGURE 8. WEED TO BE IDENTIFIED AND THE UAV USED [13]

III. RESULTS AND DISCUSSION

The simulation carried out in MATLAB involved a basic mapping algorithm using N number of robots to navigate a 2-D grid. This simulation involved randomly placing robots within an area where the robots work together to map the area out. The collision detection system developed was made to move to every possible area in the grid while detecting for obstructions. In this example the area is a 2-D grid. Figure 9 shows an example of the starting conditions of a randomly generated 2-D area.

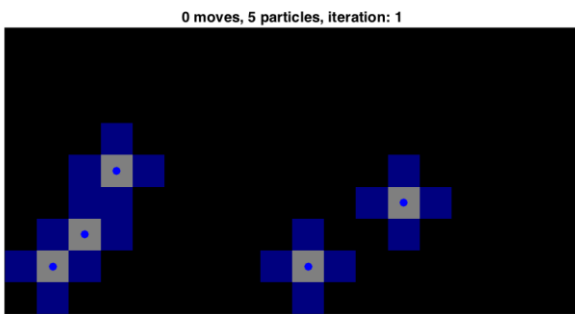


FIGURE 9. RANDOMLY GENERATED 2-D SPACE CREATED IN MATLAB

The grey square with a blue dot is the placeholder for a robot. The blue squares around the robot indicate possible moving locations and the black areas indicate either obstructed area or unexplored area. The robots in this experiment have four movement limitations: up, down, left and right. Using a movement algorithm, they move in these directions while also using a collision detection system to prevent them from hitting a wall or into each other. Figure 10 shows the mapping in progress.

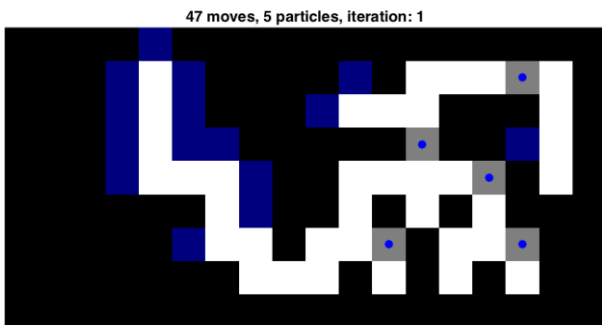


FIGURE 10. MAPPING OF 2-D GRID USING SWARM ROBOT PARTICLES IN PROGRESS

It can be seen that the white areas indicate explored and unobstructed areas while the swarm moves to map areas. One of the downsides of this design is that the swarm moves together, meaning that in areas of very few obstruction areas, this algorithm can take time to completely map an area out as the swarm ends up moving in the same directions for every move. Figure 11 shows the complete mapped out area.

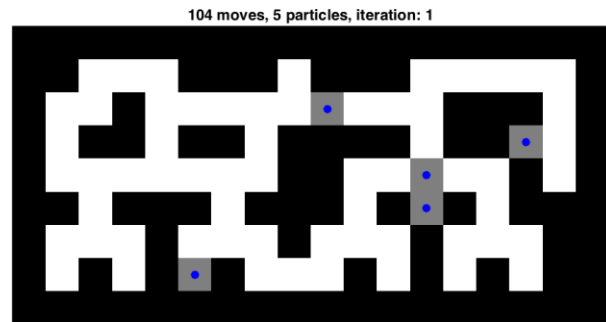


FIGURE 11. MAPPING OF GRID COMPLETED AFTER 104 MOVES

Once the mapping is completed the swarm robots stop as all possible areas have been identified. This algorithm took a total of 104 moves for the entire swarm to map. This experiment was carried out for multiple iterations, some of the more notable results of which can be seen in Figure 12.

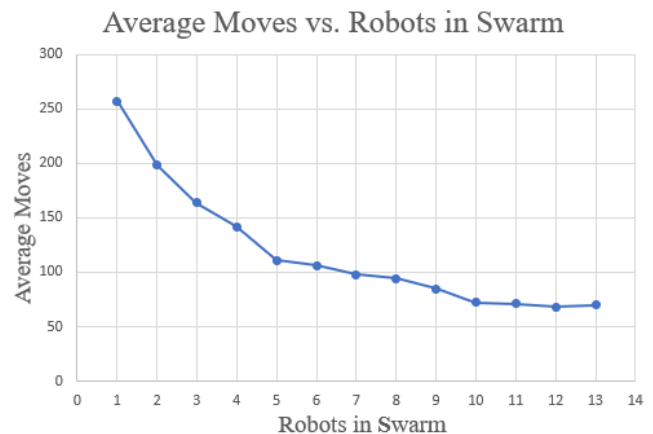


FIGURE 12. RESULTS OBTAINED FOR AVERAGE MOVES WITH RESPECT TO NUMBER OF ROBOTS IN SWARM

It can be observed that when the number of robots is between the range of 1 to 5, the number of moves vary by a substantial number. Since the algorithm places the robots in a random location each time, and there is several spaces to cover, having more robots can substantially reduce the number of moves required to explore the entire grid. There can also be random errors within each iteration due to clumping of robots or other factors. Therefore, an average value is taken across all iterations. The effect of this can be shown to a lesser extent when 6 to 13 robots are used, even though the number of robots were increased, the space available can cause clumping of robots to occur and hence only reduce the number of average moves required by a smaller amount. This makes it a key factor to ensure that a right number of robots are chosen depending on the size of the area that needs to be mapped. Any more can be inefficient. The number of moves can also be reduced by using a more improved collision detection algorithm

IV. FUTURE SCOPE

From the review of recent work in swarm robotics, this field has tremendous potential for growth in many areas including industrial fields and military sectors. Real time applications of swarm robotics can soon be developed and used for many applications such as search and rescue, fire fighting, defense systems, warehouse management and many more. Many industries nowadays are already using similar multi-robot systems to carry out automated tasks.

These types of systems can be studied further and swarm robotics techniques can be implemented to improve automation and reliability of these robots.

Some of the key difficulties with development of swarm robotic algorithms are the costs involved with developing specialized robots on a large scale that are scalable. Once the technology for the required components improve, swarm robotics will be more accessible for study and improvement in various fields.

V. CONCLUSION

Swarm robotics is the future when it comes to automation for various teamwork oriented fields such as construction, military, firefighting, etc. They provide a safer and more robust alternatives to human labor especially in performing dangerous and high precision tasks. The recent developments to swarm robotics and the algorithms that can be incorporated to them make this technology ever closer to be realized in everyday tasks.

From the obtained simulations, it can be seen that with more development and real time testing of the algorithms, greater results can be obtained to prove the efficacy of the obstacle avoidance algorithms applied to swarm robotics.

REFERENCES

- [1] Jung, B. and Sukhatme, G.S., 2004, March. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In International Conference on Intelligent Autonomous Systems (pp. 980-987).
- [2] Lee, D., Kim, G., Kim, D., Myung, H. and Choi, H.T., 2012. Vision-based object detection and tracking for autonomous navigation of underwater robots. *Ocean Engineering*, 48, pp.59-68.
- [3] Ekvall, S., Kragic, D. and Jensfelt, P., 2007. Object detection and mapping for service robot tasks. *Robotica*, 25(2), pp.175-187.
- [4] Borenstein, J. and Koren, Y., 1989. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on systems, Man, and Cybernetics*, 19(5), pp.1179-1187.
- [5] Bilbeisi, G., Al-Madi, N. and Awad, F., 2015, November. PSO-AG: A Multi-Robot Path Planning and obstacle avoidance algorithm. In Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on (pp. 1-6). IEEE.
- [6] Almasri, M.M., Alajlan, A.M. and Elleithy, K.M., 2016. Trajectory planning and collision avoidance algorithm for mobile robotics system. *IEEE Sensors Journal*, 16(12), pp.5021-5028.
- [7] Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S. and Younes, H., 2000, July. Coordination for multi-robot exploration and mapping. In *Aaai/Iaai* (pp. 852-858).
- [8] Simon Bultmann, Laëtitia Matignon, Olivier Simonin. Multi-Robot Navigation and Cooperative Mapping in a Circular Topology. [Research Report] INSA Lyon. 2017. <hal-01526089>
- [9] Couceiro, M.S., Rocha, R.P. and Ferreira, N.M., 2011, November. A novel multi-robot exploration approach based on particle swarm optimization algorithms. In Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on (pp. 327-332). IEEE
- [10] Duarte, M., Gomes, J., Costa, V., Rodrigues, T., Silva, F., Lobo, V., Marques, M.M., Oliveira, S.M. and Christensen, A.L., 2016, April. Application of swarm robotics systems to marine environmental monitoring. In *OCEANS 2016-Shanghai* (pp. 1-8). IEEE.
- [11] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks Through Augmenting Topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99-127, 2002.
- [12] Bakhshipour, M., Ghadi, M.J. and Namdari, F., 2017. Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach. *Applied Soft Computing*, 57, pp.708-726.
- [13] Trianni, V., Ijsselmuiden, J. and Haken, R., 2016. The SAGA concept: Swarm Robotics for Agricultural Applications. Technical Report, <http://laral.istc.cnrit/saga/wp-content/uploads/2016/09/saga-dars2016.pdf>.