Exploration and Navigation of GPS Denied Indoor Environments using Drone Swarm

Dr. Sathian D[1] ,Aaditya bagaddeo[2]

Abstract— **Indoor mapping using a swarm of drones is indispensable for a multitude of reasons. Firstly, drones excel at inspecting indoor spaces, navigating through challenging areas inaccessible to humans, and swiftly identifying structural or equipment issues for prompt maintenance. Additionally, these drones contribute significantly to architecture planning by generating precise 3D maps, enabling architects and designers to optimize space utilization for efficient building design. In the context of disaster relief, drones play a pivotal role in search and rescue operations, rapidly assessing indoor structures, locating survivors, and guiding rescue teams to affected areas. For large spaces like warehouses, drone mapping facilitates improved inventory management, optimized storage layouts, and streamlined logistics operations. Moreover, the detailed and exhaustive data collected by a swarm of drones proves invaluable for analytics, decision-making, and enhancing overall operational efficiency across diverse industries. In essence, the utilization of drone swarms for indoor mapping empowers organizations with unparalleled insights, efficiency gains, and informed decision-making in managing and optimizing indoor spaces.**

Index Terms— **Swarm Robotics, Drone Swarm, Navigation, LiDar , Computer Vision, Mapping , ROS , Publisher-Subscriber , Node**

I. INTRODUCTION

The background of the problem addressed by indoor mapping using a swarm of drones is rooted in the inefficiencies and challenges associated with traditional methods of surveying and inspecting complex indoor environments. Conventional approaches, often reliant on manual efforts or stationary sensors, struggle to comprehensively cover expansive or difficult-to-access areas. In response to this, industries recognize the pressing need for advanced, automated solutions. Challenges range from the intricate inspection of structures and equipment in industrial settings to optimizing architectural planning, responding to disasters, and streamlining operations in large warehouses. Additionally, the limitations of traditional methods have prompted a shift towards leveraging drone technology, equipped with precise sensors and mapping algorithms, for autonomous indoor mapping. This innovative approach not only addresses the shortcomings of existing methods but also empowers organizations with detailed, real-time data for analytics and decision-making, thereby enhancing overall operational efficiency across diverse sectors. Furthermore, it proves invaluable for conducting routine checks on old buildings and structures, where detailed inspections are essential but challenging to perform manually on a regular basis, contributing to the preservation and maintenance of architectural heritage and critical infrastructure.

Since all these problems can have very distinct environments of exploration and deployment, simulating the working of the swarm in such environments is very helpful to understand the effect and efficiency of these systems in such environments. Indoor environments lack a robust way of navigation given GPS data is very ineffective and inaccurate. Mapping of indoor environments requires precise navigation in unknown environments and thus , techniques like Visual SLAM need to be implemented. Further, given the need for effective implementation of swarms of drones in such diverse indoor environments, it is necessary to have a way to evaluate the effectiveness and usefulness of the algorithm and check the amount of data that can be extracted. This asks for an accurate simulation system that can track the working of these algorithms to come up with accurate navigation and mapping in GPS denied environments .

The research objective is to simulate the operation of a swarm of drones for indoor mapping using Airsim and Unreal Engine. To achieve this, several objectives have been outlined. Firstly, the aim is to enable seamless communication among the drones within the swarm to effectively cover the entirety of the indoor area. This entails establishing robust protocols for coordination and collaboration among the drones to optimize mapping efficiency. Secondly, the focus is on utilizing the 3D point cloud of the indoor environment. The goal is to obtain high-fidelity spatial data representing the surroundings from multiple perspectives. Subsequently, the research aims to implement SLAM (Simultaneous Localization and Mapping) techniques on the generated LiDAR data. This involves developing algorithms capable of accurately estimating the drones' positions and orientations within the mapped environment, facilitating precise navigation and spatial awareness. Additionally, emphasis is placed on obtaining precise distance measurements between different locations within the mapped area to enable accurate spatial referencing. Throughout these objectives, challenges such as setting up a custom simulation environment, generating data from the simulation, addressing localization without GPS, and monitoring battery levels across drones with potentially unequal task distributions must be overcome to realize the full potential of the research. Overall, the research seeks to advance the capabilities of drone swarms for indoor mapping applications, leveraging cutting-edge simulation technologies and algorithms to achieve comprehensive and accurate spatial mapping in diverse indoor environments.

II. LITERATURE SURVEY

The literature review covers a variety of papers discussing different aspects of drone swarm technology. Setup and configuration of an autonomous Multi-UAV Coverage Path Planning for the Inspection of Large and Complex Structures [1] and A Multi-agent Reinforcement Learning Method for Swarm Robots in Space Collaborative Exploration[2] both focus on how to efficiently cover an area with a swarm of drones. Meanwhile, Autonomous Aerial Mapping Using a Swarm of Unmanned Aerial Vehicles [3] uses a version of Hector SLAM that doesn't rely on odometry for SLAM,[4] introduces and compares two prominent architectures for coordinating communication within UAV swarms. Autonomous Drone Swarm Navigation and Multi-target Tracking in 3D Environments with Dynamic Obstacles [5] addresses challenges in artificial swarm systems and uses a policy-based deep reinforcement learning strategy for autonomous navigation. The second part of the review covers implementation of loop closure in slam under Real-Time Loop Closure in 2D LIDAR SLAM [6] and an energy-efficient navigation method for an autonomous swarm with adaptive consciousness [7]. Frontier Based Exploration [8] offers an enhanced version of the original frontier-based exploration method, while Swarm Crawler Robots Using Levy Flight for Targets Exploration in Large Environments[9] applies the Subsumption Architecture (SSA) to model the behavior of swarm robots. Lastly, Obstacle Avoidance for Swarm Robots Based on Self-Organizing Migrating Algorithm [10] presents an obstacle avoidance algorithm for swarm robots in unknown environments based on self-organizing migrating algorithm. The final part of the review mentions a paper discussing methods for scale factor estimation in monocular-camera-based SLAM [11] and another proposing an environment exploration method using multiple Unmanned Aerial Vehicles (UAVs) inspired by the construction behavior of bees in building hives [12]. SwarmLab: a MATLAB Drone Swarm Simulator [13] proposes a simulation software written entirely in MATLAB, addressing the lack of user-friendly simulation frameworks for drone swarm research. The final papers, Drone swarm strategy for the detection and tracking of occluded targets in complex environments [14] and Comparative analysis of ROS based 2D and 3D SLAM algorithms for Autonomous Ground Vehicles[15], deal with strategies for detecting and tracking occluded targets in complex environments and indoor mapping using graph SLAM, respectively.

III. METHODOLOGY

The project involves simulating drone object modules within a simulated environment, utilizing hardware components such as the Quadrotor Drone, Lidar Sensor for obstacle avoidance and mapping, and Camera for image processing and perception. Various software modules are employed including Airsim, Unreal Engine, PyGame, Robot Operating System (ROS), Tensorflow, Keras, and OpenCV. Algorithms such as image segmentation, IMU-based motion tracking, LiDAR SLAM, and convolutional neural networks (CNN) for image classification are utilized for drone navigation and perception tasks. The protocol for implementation includes creating a world in Unreal Engine, establishing a database for door locations, defining agents and sensors for drones, setting up visual odometry-based path planning for drones, and integrating algorithms for data processing and analysis. Lidar and IMU data are collected and graphed to assess the effectiveness of the implemented algorithms and sensor fusion techniques in navigation and perception tasks within the simulated environment. The proposed exploration algorithm is designed to efficiently map multiple floors of a building using a coordinated swarm of drones, with a master drone orchestrating the exploration process. Each floor is assigned a set of drones, and the master drone receives instructions on which floor to explore next. To ensure precise navigation and mapping, the algorithm leverages an Attitude and Heading Reference System (AHRS) to monitor the position and orientation of the drones as they move through the floor space. As the drones explore, they autonomously navigate through doors and unexplored areas using advanced algorithms. The master drone dynamically adjusts the formation of the swarm based on the environment and the progress of exploration. This flexibility allows for optimal coverage of the floor space while adapting to obstacles and changing conditions.A crucial aspect of the algorithm is the monitoring of battery levels and safety considerations. If the battery levels of any drone drop below a predefined threshold or if the distance from the starting point becomes unsafe, the swarm returns to a designated charging station for recharging. This proactive approach ensures uninterrupted exploration while preserving the safety of the drones. The slave drones in the swarm are equipped with SLAM (Simultaneous Localization and Mapping) techniques, enabling them to efficiently explore doors and unexplored areas. This capability enhances the mapping accuracy and completeness of the floor space, providing valuable insights for various applications such as search and rescue operations, building inspection, or environmental monitoring. Figure 1 shows the system architecture for the proposed system.

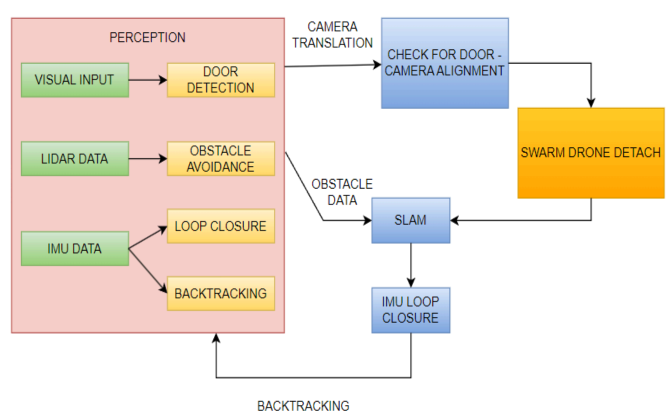

Figure 1 : Airsim Simulator Python Architecture

COMMUNICATION

The communication within the drone swarm utilizes a Publisher-Subscriber model, facilitated by the Robot Operating System (ROS), to exchange information between drones. ROS provides a robust framework for developing and simulating robotic systems, offering a communication infrastructure that allows nodes to publish messages on specific topics, which other nodes can then subscribe to and receive. In this model, each drone acts as both a publisher and a subscriber. The master drone publishes relevant information, such as floor assignments, exploration progress, and safety alerts, on designated topics. Meanwhile, the slave drones subscribe to these topics to receive instructions and updates from the master drone.ROS enables seamless communication between drones in the swarm, allowing for real-time coordination and synchronization of actions. For example, when the master drone detects a door or an unexplored area, it publishes a message indicating the location and instructs nearby slave drones to investigate. The slave drones, subscribed to this topic, receive the instructions and autonomously navigate towards the designated areas for exploration.Moreover, ROS facilitates the exchange of feedback and status updates between drones, enabling them to adapt their behavior dynamically based on changing conditions. For instance, if a slave drone encounters an obstacle or experiences a battery issue, it can publish a message indicating the situation. Other drones in the swarm, subscribed to this topic, can receive the update and adjust their actions accordingly, ensuring collective safety and efficiency. To simulate the communication between drones in the current scenario , there is no better tool than ROS . ROS allows definition of publisher and subscriber nodes and allows them to send data to respective nodes through a rosmaster .Figure 2 explains the role of ROS based communication in the proposed system.

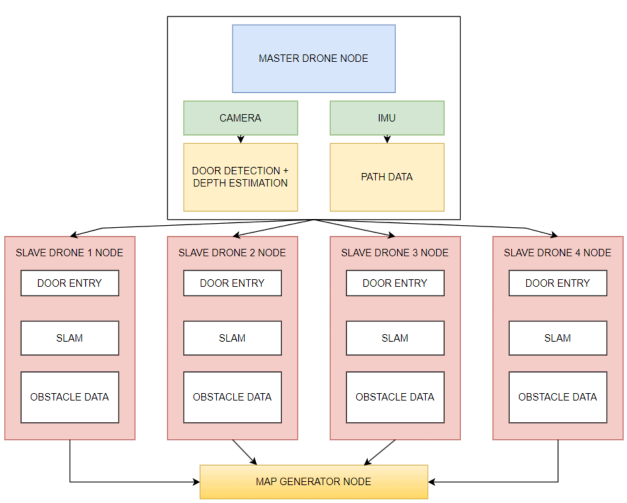

Figure 2 : ROS Communication Simulation Architecture

ALGORITHMS

*A. **Master Drone Navigation:***

The algorithm devised for the master drone within a swarm of five drones for indoor mapping employs a systematic approach to efficiently explore different floors of a building while prioritizing safety and battery conservation. Each floor is treated as a distinct case, allowing the algorithm to tailor its navigation strategy accordingly. Within the swarm, the master drone is positioned at the center, surrounded by four slave drones arranged in a plus pattern, forming a cohesive unit optimized for exploration. Upon receiving information about the target floor, the master drone coordinates the swarm's movements to navigate through the building.

As shown in Figure 3 , Once deployed on a floor, the swarm enters an exploration loop. Moving forward, the drones continuously monitor their position using an Inertial Measurement Unit (IMU) to track their x, y, and z coordinates. As the swarm progresses, adaptive separation occurs whenever a door is detected or the Lidar range fails to identify a wall. In such instances, one slave drone is directed towards the detected area, enabling exploration of branching paths and ensuring comprehensive coverage of the indoor environment.

Moreover, stringent thresholds for battery levels and safety are enforced to mitigate risks and optimize resource utilization. If a drone's battery falls below predefined thresholds, or if safety considerations deem the distance from the starting point unsafe, the algorithm triggers appropriate responses. Depending on the state of other drones and safety conditions, these responses may range from continuing exploration to orchestrating a coordinated return of the entire swarm for recharging.

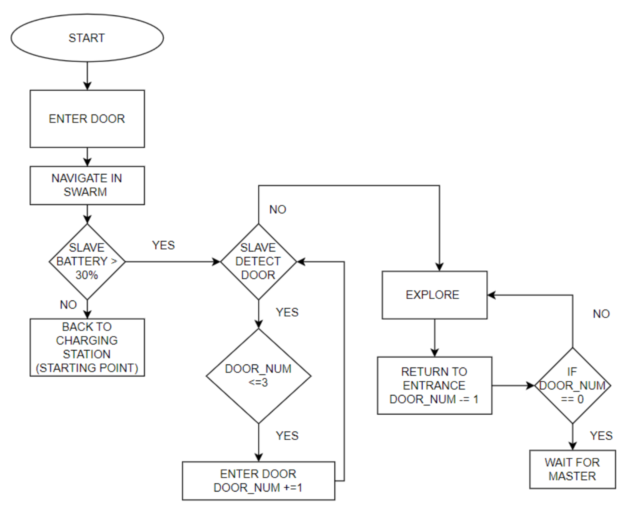

Figure 3 : Master Drone Flowchart

B. *Slave Drone Navigation:*

The algorithm designed for the slave drone within the swarm of drones for indoor mapping focuses on efficient exploration and battery management while responding to safety considerations. Initially, the algorithm checks if the drone's battery level is above 30 percent and if the distance from the starting point is considered safe. If these conditions are met, the drone proceeds with exploration. Upon encountering a door or an area where the Lidar fails to detect a wall, the drone separates from the master and moves towards the detected door or unexplored region. It then initiates the "EXPLORE" function, which involves systematically navigating through the detected doorways and exploring adjacent areas. While exploring, the drone keeps track of temporary door counts to ensure thorough coverage of the environment.

During exploration, if another door is detected within the explored area, the algorithm recursively calls the "EXPLORE" function to continue investigating new pathways. Otherwise, the drone performs Visual SLAM (Simultaneous Localization and Mapping) to map the surrounding environment accurately. Once the exploration is complete, the drone returns to the original door and waits for the master drone's signal to either land or hover, depending on the master drone's distance from the current location.

Alternatively, if the drone's battery falls below 30 percent or below 40 percent with the distance from the starting point deemed unsafe, the algorithm prompts the drone to return to the starting point for recharging. This proactive approach ensures that the drone conserves battery power and maintains safety protocols throughout the exploration process. By efficiently navigating through doors, exploring uncharted areas, and judiciously managing battery levels, the algorithm enables the slave drone to contribute effectively to the overall mapping objectives within the indoor environment.Figure 4 depicts this algorithm in the form of a flowchart.

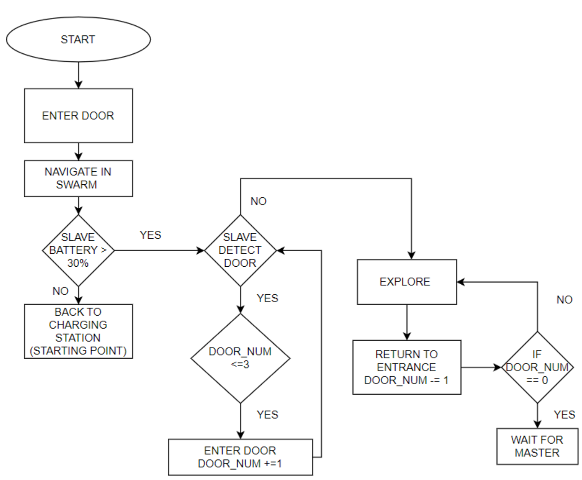

Figure 4 : Slave Drone Flowchart

IV. Result Analysis

A. ROS and PyGame

The ROS and PyGame setup is used to simulate the communication between the slave drones and the master drone. Further , it gives us the convenience of fusing obstacle data from each separate slave drone to generate the map of the environment. The working of the ros nodes as well as the communication can be seen as follows .

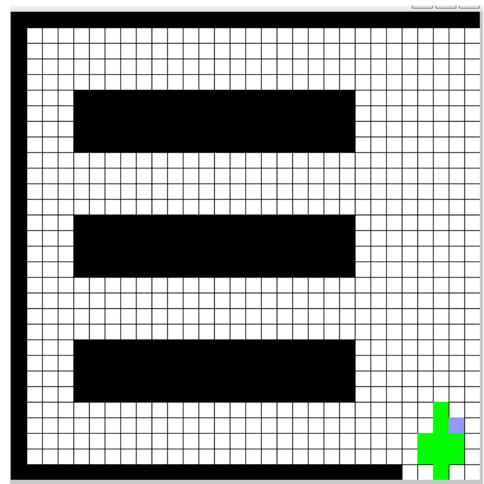

Figure 5 : Swarm Initiates Navigation

Figure 5 shows how the map occupancy grid map looks for the setup and Figure 7 shows the actual navigation taking place for the left drone after the master drone detects a door on the left leading to detachment of the left drone from the swarm . The RQT graph for the same can be seen in Figure 6.

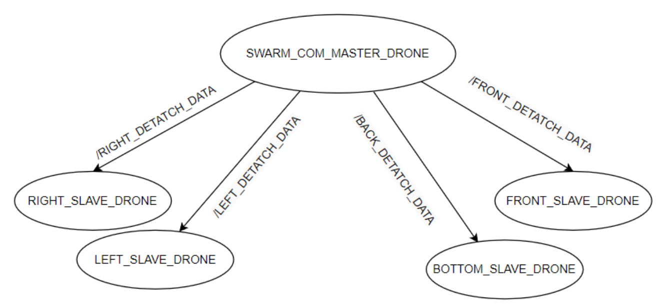

Figure 6 : rqt_graph for node communication in ROS

The communication sequence begins with the swarm navigating the environment until a door is detected. Upon detecting a door, the master drone sends a signal to the respective side slave drone, instructing it to separate from the swarm and begin exploration towards that door, while publishing its current local coordinates. All slave drones remain in subscribing mode until they receive the separation command. Once all drones receive the separation command, the master drone stops and publishes its current location to all the slave drones. After exploration, each slave drone looks for the master drone's published data about its current position and calculates the position difference to move towards the master's location. Once all four slave drones return to the swarm, they send their collected obstacle data to a separate Mapping node, which fuses these 2D map matrices to complete mapping of that area of the environment.

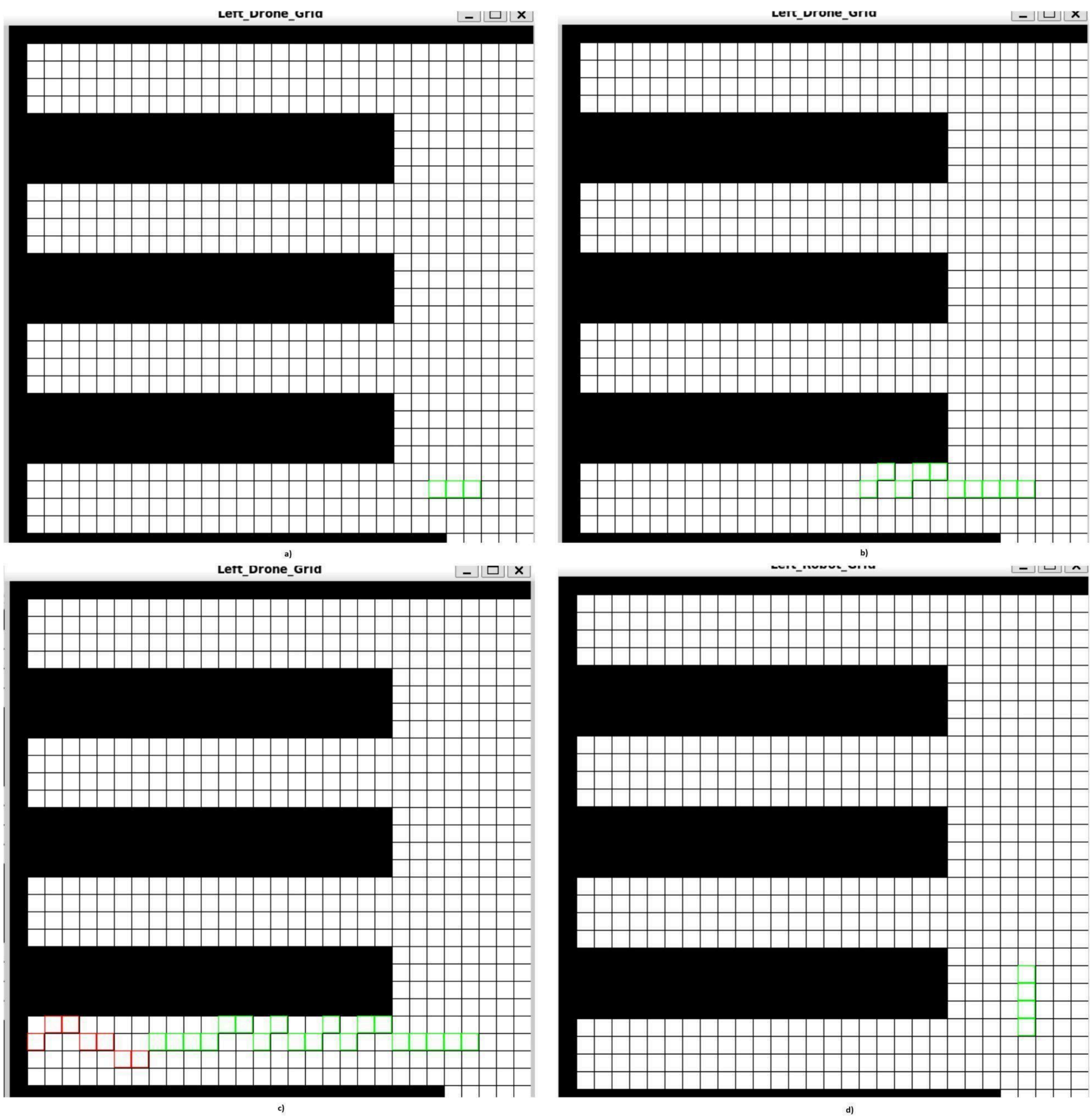

Figure 7: Sequence of communication leads to this kind of movement in the slave drone,(from top left to bottom right) - **a)** starting with it initiating motion towards that door ,**b)** separating from the swarm , initializing slam as it enters the door ,**c)** backtracking its initial path during slam to close the loop and reach its starting position and finally , **d)** moving back to master drone after receiving masters position from master node

The same can has been seen in working for the other 3 slave drones. As soon as the master finds a door , it calculates the distance of the door from the master as well as the direction in which the slave has to move to enter that door. Based on this, it makes drones separate from the swarm in a predefined sequence given as :

left drone —> back drone —> front drone —> right drone .Each drone subscribes to the master to get its initial position and direction as well as distance from the door, based on which , they start navigating towards the door , initialize slam after entering the door , backtrack and then return to master. This is depicted in figure 8 below.

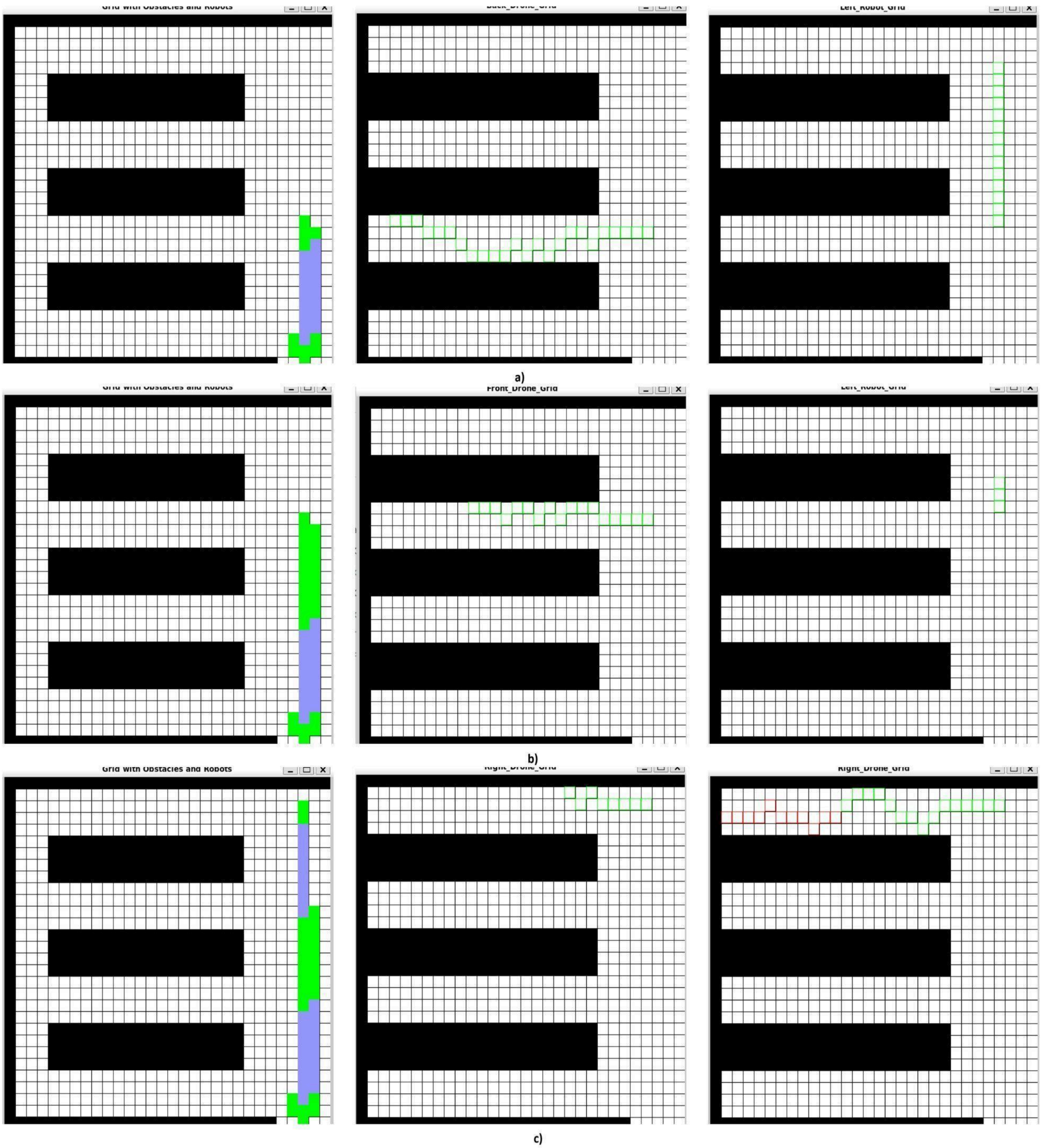

Figure 8: a) Back Drone Motion Sequence , b) Front Drone Motion Sequence , c) Right Drone Motion Sequence

This set of ros nodes shows the sequence of drone communication and the necessity of the master drone in the swarm to make sure that the local coordinate system is effective. Without the master drone acting as the source of all communication about the surroundings as well as being that mastermind that decides how to distribute the slave drones in the environment , the swarm would just collapse in such GPS denied environments. There is another very important role played by the master , which is to be able to gather the scattered drones back into one place. If the master drone is not publishing the data to the slave drones to be able to meet and arrange themselves again after the first separation, it would be extremely hard for these drones to regroup as the values of position and velocity published by these drones will be dynamic and as such , without the source of the data being stationary , there is a high chance the drones would keep trying to regroup infinitely. Finally , Figure 9 shows how the data from different swarms was fused to come up with a map of the area to be explored. It not only shows an extremely accurate portrayal of the environment , but also proves the fact that fusion of point clouds is definitely possible , given the high amount of complicated values these point clouds hold

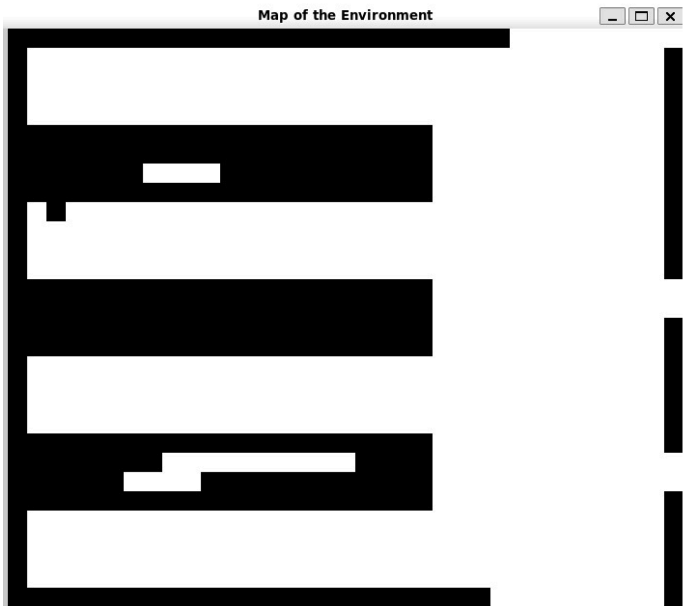

Figure 9 : Map Generated by fusing Lidar Data form all 4 individual drones

To further test the working of this algorithm , it was tested over 4 different complexities of the map , with increasing difficulty of exploration . The difficulty level in the maps was increased by adding a separate room to the map to see if different layouts of the map hinder the algorithm working. Furthermore , obstacles were added to the map to make it even harder for the Frontier SLAM algorithm to be able to explore the area efficiently. Figure 10 shows the different map environments used for testing.

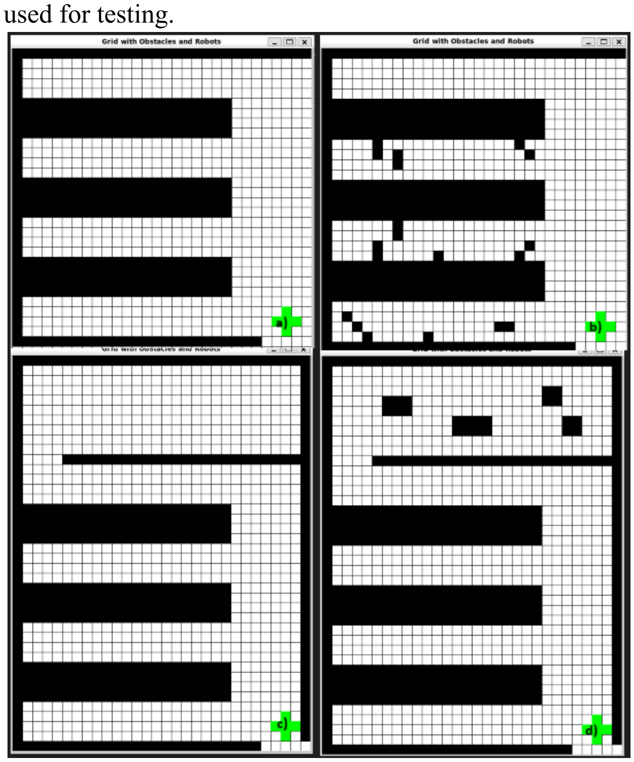

Figure 10 : Form a) to d) in increasing complexity of Map

The algorithm was implemented for all the four complexities to check the effectiveness of the algorithm. Given the stochastic nature of the Frontier SLAM algorithm , although the map generated every time was a little different , the maps were found to be up to 95 % accurate This accuracy was calculated based on the number of obstacles defined in the map and the number of obstacles generated in the map.

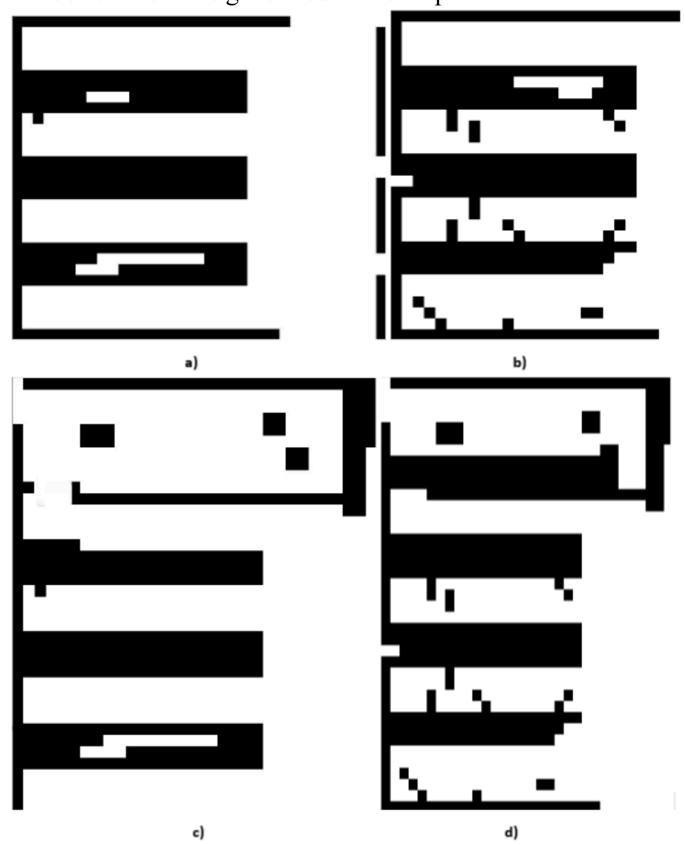

Figure 11 : Maps Generated after exploratory SLAM

Figure 11 shows how the algorithm generated maps based on exploration done by the drone swarm in the environments shown in Figure 10. Further , the table in Figure 12 shows the accuracies as in ratio of obstacles correctly detected to the number of obstacles in the map.

Iterations	Iter1	Iter 2	Iter 3	Average
Level 1	0.9881	0.9691	0.9763	0.9778
Level 2	0.9887	0.9503	0.9683	0.9691
Level 3	0.9902	0.9731	0.9828	0.9820
Level 4	0.9372	0.9641	0.9170	0.9394

Figure 12 : Map Accuracy Attained for Multiple Iterations

Now that the effectiveness of the method has been proved , it is further necessary to compare the working of this SLAM algorithm with other exploration algorithms to check how the conditions in the map affect the working of the algorithms and how it can be flexible with what slam algorithm is to be implemented. To do this , the Frontier SLAM implemented here was compared with Exploratory RRT as well as Potential Field based Navigation , comparing the distance traveled as well as time of flight for multiple iterations done on the map due to the stochastic nature of the algorithms.

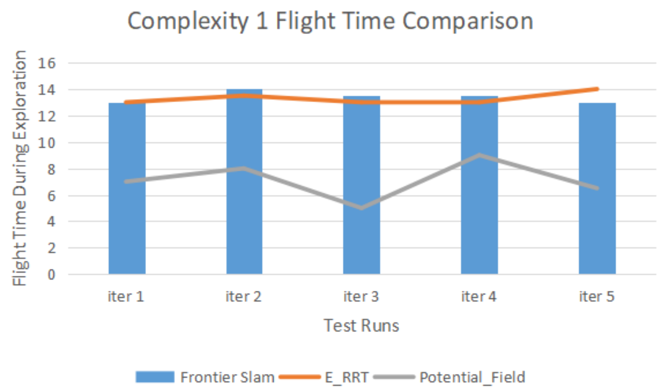

Figure 13 : Flight Time Comparison for Complexity 1

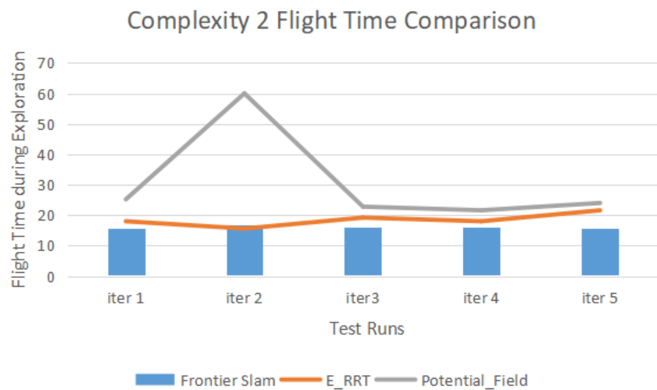

Figure 14 :Flight Time Comparison for Complexity 2

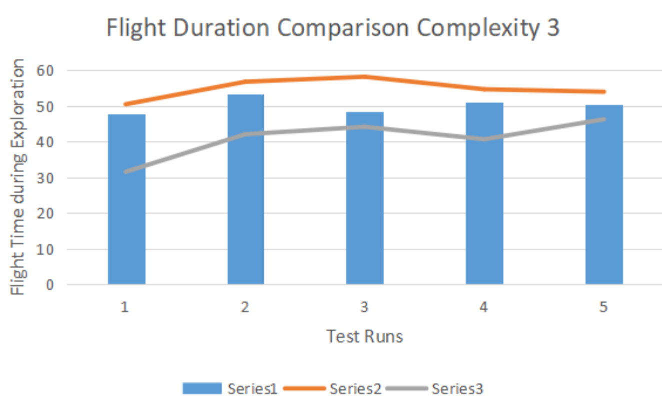

Figure 15 : Flight Time Comparison for Complexity 3

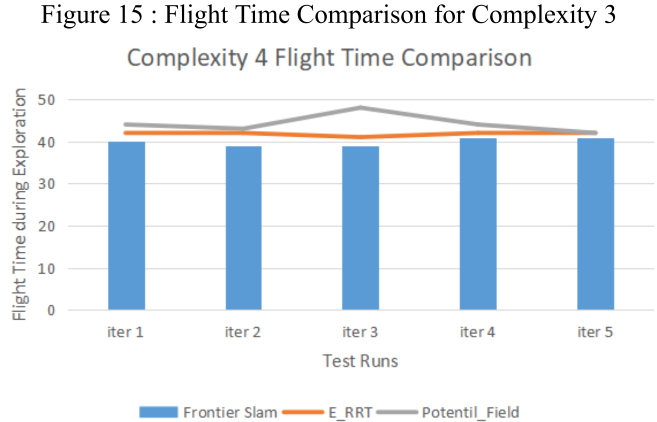

Figure 16 : Flight Time Comparison for Complexity 4

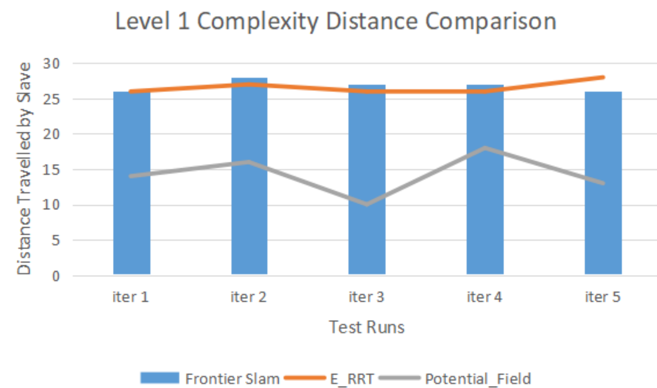

Figure 17 : Distance Traveled Comparison for Complexity 1

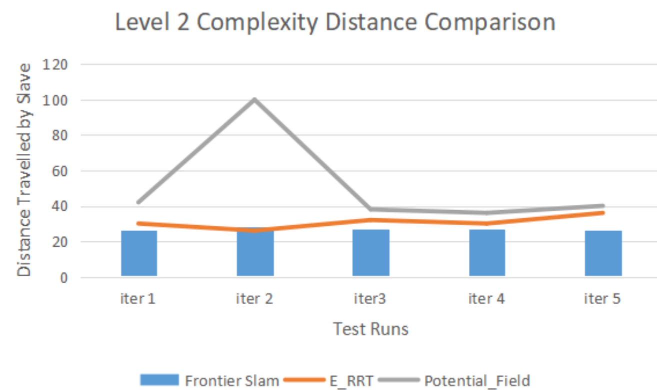

Figure 18 : Distance Traveled Comparison for Complexity 2

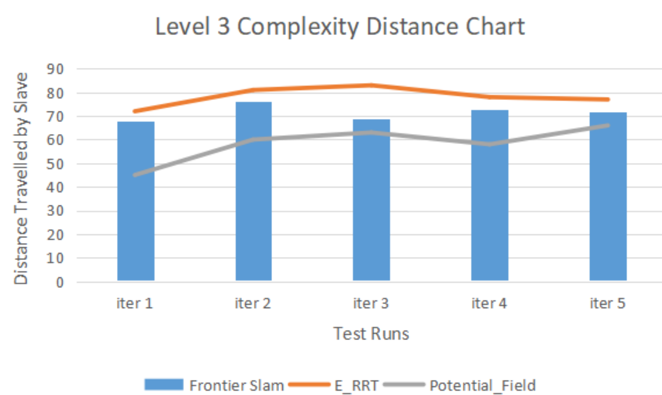

Figure 19 : Distance Traveled Comparison for Complexity 3

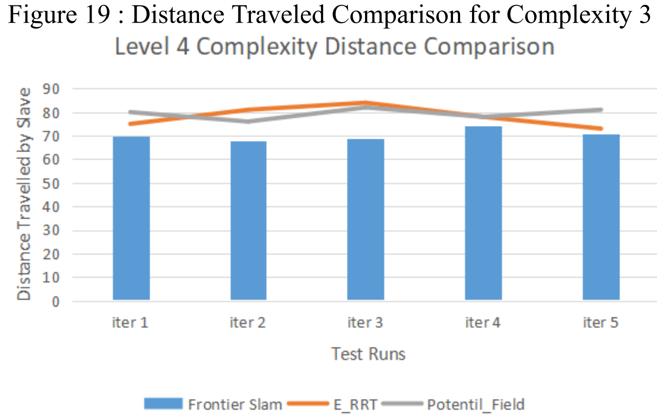

Figure 20 : Distance Traveled Comparison for Complexity 4

Figures 13 - 16 compare the distance traveled by the slave drones for different map complexities for 3 exploration algorithms. Figures 17 - 20 compare the time of flight for the same.

B. Airsim and Unreal Engine

Figures 21 - 24 are pictures taken in Airsim simulation to show how the proposed algorithm would work in a real world indoor 3D environment.

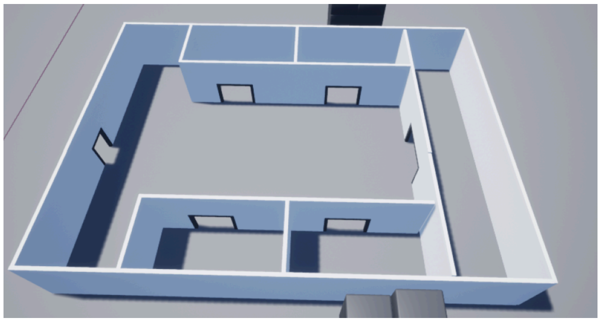

Figure 21 : Top View of Indoor Simulation Environment setup in Airsim

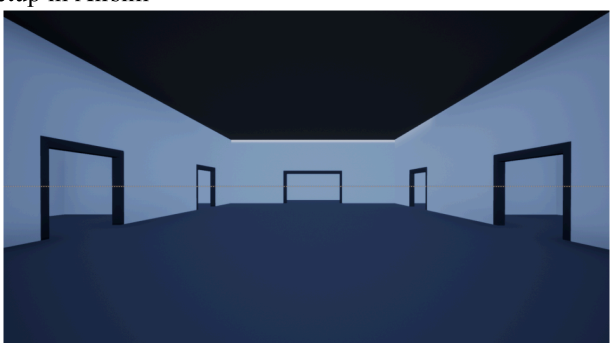

Figure 22 : Front View of Environment from inside

The Airsim simulation environment consists of a hallway with rooms and doors connecting to these rooms. The aim of the simulation is to show how the drones will move in the environment under the proposed navigation and exploration algorithm. As seen in the above pictures, a simple indoor environment is considered so as to allow testing of multiple edge cases and scenarios in the algorithm. The drone swarm is made to spawn outside the area and the algorithm is tested to check if the drones are able to separate from the swarm and sensor data is able to demonstrate the working in such an environment without the existence of GPS data. The simulator was able to show the following results as door detection was carried out to make the drone stop , separate from the master and enter the room for exploration.

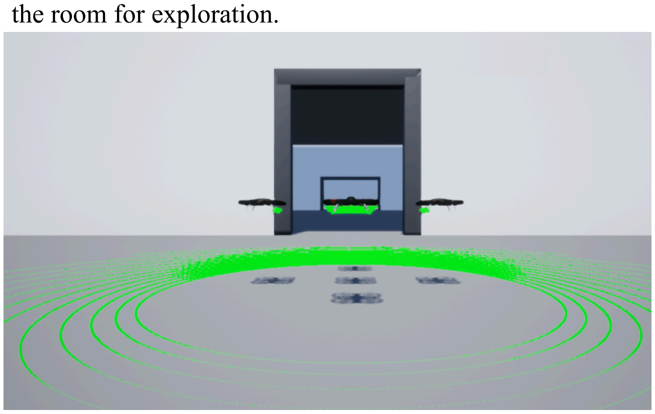

Figure 23 : Initializing of the Simulation

The swarm is initialized to take off and start moving forward . While the drone is moving, the LiDAR attached to the master drone acts as the LiDAR of the swarm as a whole to allow easy navigation and obstacle avoidance. The swarm moves forward and the drone cameras keep scanning for doors in the surroundings, As soon as the door is detected and it is found that the door is in the center of the frame , the swarm stops, the slave drones detach and move towards the rooms for exploration.

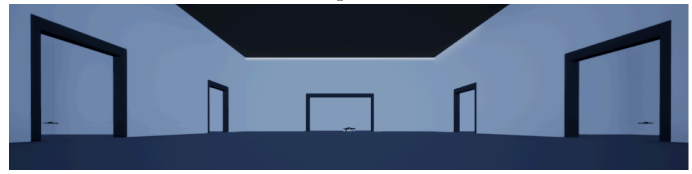

Figure 24 : Slave Drones separate to enter room for exploration

V. CONCLUSION

Airsim simulation for this project is limited to navigation of the environment given the need for fusing 3D point clouds from all different drones . This is very tricky in case of real LIDAR 3D point clouds as fusing point clouds from multiple lidars requires an extremely high amount of synchronous data including positional data as well as accurate time stamped point clouds . Using the 2D local coordinate system which is implemented in the ROS with Pygame simulation allows us to see just how much is possible with the correct combination of the currently available technology in the field of autonomous drones.
The simulations act as proof of concept for this algorithm as they show the combinations of sensor readings effectively make the drones explore the area in an efficient manner. Although the algorithm still has a lot of scope for improvement and a lot more possible testing , this marks a beginning to solve one of the biggest problems in the field of autonomous drone swarms , where the algorithm is able to explore and map areas without using a GPS or doing a trial run .

VI. REFERENCES

[1] W. Jing, D. Deng, Y. Wu and K. Shimada, "Multi-UAV Coverage Path Planning for the Inspection of Large and Complex Structures," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 1480-1486

[2] Y. Huang, S. Wu, Z. Mu, X. Long, S. Chu and G. Zhao, "A Multi-agent Reinforcement Learning Method for Swarm Robots in Space Collaborative Exploration," *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, Singapore, 2020, pp. 139-144

[3] Alsayed, Ahmad & Nabawy, Mostafa & Arvin, Farshad. (2022). Autonomous Aerial Mapping Using a Swarm of Unmanned Aerial Vehicles. 10.2514/6.2022-4062.

[4] MitchCampion, PrakashRanganathan, and SalehFaruque. 2019. UAV swarm communication and control architectures: a review. *Journal of Unmanned Vehicle Systems*. 7(2): 93-106.

[5] Qamar S, Khan SH, Arshad MA, Qamar M, Gwak J, Khan A. Autonomous drone swarm navigation and multitarget tracking with island policy-based optimization framework. IEEE Access. 2022;10:91073-91091.

[6]W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-time loop

closure in 2D LIDAR SLAM," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016, pp. 1271-1278

[7] Yasin JN, Mahboob H, Haghbayan M-H, Yasin MM, Plosila J. Energy-Efficient Navigation of an Autonomous Swarm with Adaptive Consciousness. *Remote Sensing*. 2021; 13(6):1059

[8] Topiwala A, Inani P, Kathpal A. Frontier based exploration for autonomous robot. Published online 2018. arXiv preprint. arXiv:1806.03581..

[9] Katada Y, Hasegawa S, Yamashita K, Okazaki N, Ohkura K. Swarm Crawler Robots Using Lévy Flight for Targets Exploration in Large Environments. *Robotics*. 2022; 11(4):76.

[10]Bao DQ, Zelinka I. Obstacle avoidance for swarm robot based on self-organizing migrating algorithm. Procedia Computer Science. 2019;150:425-432. doi:10.1016/j.procs.2019.02.073.

[11] Karam S, Nex F, Chidura BT, Kerle N. Microdrone-Based Indoor Mapping with Graph SLAM. *Drones*. 2022

[12] da Rosa R, Aurelio Wehrmeister M, Brito T, Lima JL, Pereira AIPN. Honeycomb Map: A Bioinspired Topological Map for Indoor Search and Rescue Unmanned Aerial Vehicles. *Sensors*. 2020; 20(3):907.

[13] Soria E, Schiano F, Floreano D. SwarmLab: a Matlab drone swarm simulator. Published online 2020. arXiv preprint. arXiv:2005.02769.

[14] Amala Arokia Nathan, R.J., Kurmi, I. & Bimber, O. Drone swarm strategy for the detection and tracking of occluded targets in complex environments. *Commun Eng* 2, 55 (2023)

[15] P. Sankalprajan, T. Sharma, H. D. Perur and P. Sekhar Pagala, "Comparative analysis of ROS based 2D and 3D SLAM algorithms for Autonomous Ground Vehicles," *2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, 2020, pp. 1-6