SwarmSense: Effective and Resilient Drone Swarm and Search for Disaster Response and Management Application

Cheolmin Jeon
(Chungnam National University, Daejeon, South Korea; cheolmin.jeon94@gmail.com)
Jeongsoo Ha
(SafegaurdAI, Kyung-ki, South Korea, jeongsoo@episyscience.com)
Hanbum Ko
(Chungnam National University, Daejeon, South Korea, hanbum.ko95@gmail.com)
Byeongman Lee
(Chungnam National University, Daejeon, South Korea; ssey0921@naver.com)
Bo Ryu
(EpiSci Science, Poway, CA, boryu@episyscience.com)

ABSTRACT

In this paper, we present SwarmSense, a novel collaborative navigation algorithm for a group of drones to effectively coordinate and share information for disaster response and management applications such as wildfires. Specifically, SwarmSense is aimed for efficient and resilient ways to autonomously manage complex unmanned aerial systems for search and rescue missions. SwarmSense does this by cooperatively mapping wildfire zones and detecting displaced survivors in need while fully adapting to dynamic changes. In other words, SwarmSense is designed to find areas where the disaster hotspots are, map and track the hotspots precisely in real time, and find and rescue the survivors spread across the disaster area. The performance of the proposed SwarmSense algorithm is extensively evaluated under ten realistic wildfire scenarios, each lasting 60 minutes, and is shown to exhibit a very high level of robustness and effectiveness to accomplish its missions.

Index Terms - Drone, Search, Swarm

1. INTRODUCTION

An effective drone swarm and search algorithm must address the following major challenges: (i) limited resources; (ii) limited information availability; (iii) extremely large area with highly challenging navigation conditions [1], [2], [3]. First, the algorithm must take into the limited resources in terms of the number of drones available and short battery life. Second, the algorithm must address the lack of information about the disaster in terms of the location and size of fire zones, continuously changing weather conditions, and the location of any survivors in need of the rescue. Finally, the algorithm must be able to adapt with robustness for challenging terrain conditions in combination with the dynamic weather conditions. Underestimating the terrain and weather effects lead to explosion/crash of the drones, while the conservative approach will lead to poor disaster response effectiveness. SwarmSense addresses these challenges with high performance measured in terms of the three metrics: (i) firezone detection ratio (percentage of the firezones detected by the drones with no prior knowledge of their locations); (ii) firezone mapping precision under changing constantly due to dynamic weather conditions; and (iii) drone mission completion ratio (the number of drones completing the entire scenario without being destroyed by fires or terrain).

We use the AMASE simulator developed by Air Force Research Laboratories (AFRL), available at https://github.com/afrl-rq/OpenAMASE, and evaluate the performance using ten 60-minute scenarios provided by the AMASE during the competition held at AFRL during March 29-31, 2019 (https://fire-hack.devpost .com/). Our extensive study shows that SwarmSense detects close to 100% of all the wildfires in the region within the first 40 minutes for all ten scenarios, achieves a high degree of precision (provided by the AMASE scoring mechanism proportional to the mapping precision), and maintains as high as 91% of the drones despite extremely challenging weather (wind) and terrain conditions.

2. SWARM AND SEARCH

2.1. AMASE

AMASE is a discrete event simulator for aircraft automation and autonomy analysis and is used to validate the algorithm as well as evaluate its performance under a wide range of disaster scenarios. A total of 10 scenarios were selected from over 30 scenarios provided during the 'Swarm and Search AI 2019 Fire Hack' event hosted by AFRL during March 29-31, 2019. A scenario consists of 9 to 18 drones, battery life for

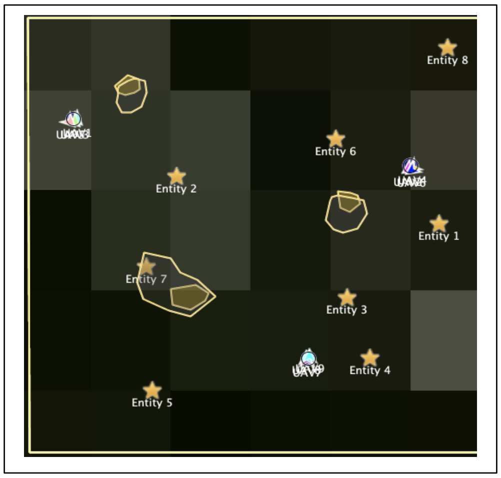

Figure 1. AMASE Example Scenario

each drone, designated battery recovery zone, size and location of fires, smoke zone of fire, locations of ground entities to represent survivors, terrain information of the entire disaster area, and so on.

Figure 1 shows a scenario and its elements: (i) drones are represented as colored arrows, battery recovery zone as small circles, wildfires as dark polygons, smoke zones (bigger than fire zone) as a light polygon, ground entities as stars, and the square border (keepin-zone). In particular, the number of firezones is more than one. For each and every scenario, a final score is produced that measures: how fast and how precisely a disaster area is mapped by drones, and how many entities are detected. At 1200, 2400, and 3600 seconds, the simulation calculates the rate of covering the firezone. (Calculate the ratio of cover area by 3, 2, and 1 times, respectively). Specifically, the scoring measures the mapping precision by comparing the areas of the mapped and true firezone areas. Both overshoot and undershoot of the area mapping is penalized. Entity perception is scored according to the number of detected entities (survivors) by the algorithm.

The important things are the firezone height is up to 1200m, and the smoke zone height is from 1000m up to 4000m from the terrain. So when the searching drones are into the firezone altitude, they are burned out. And when the searching drones are into the smoke zone altitude, they are not able to detect anything. It means that the drones couldn't find all of entity if they are into the smoke zone altitude. AMASE has the weather feature which is wind. The influence of the wind moves and resizes firezone and smokezone, and affect the direction of the drone. While the drones are moving the location or changing their body angle to curve, the sensor which is for detecting firezone and entity might look in any direction other than the desired direction. There is gimbal for controlling the drone's sensor, we can change angle of gimbal to the search area of the drone. Since the altitude between

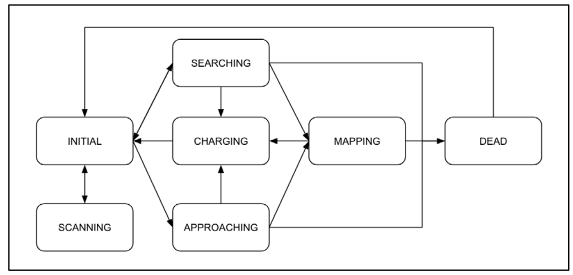

Figure 2. SwarmSense States and Possible Transitions

terrain and the drone also affect the search area of the drone, the length of the sensor is adjusted considering the height of the drone and the altitude of the terrain.

2.2. SwarmSense

We present SwarmSense to address existing algorithmic problems as mentioned in Introduction. SwarmSense allows drones to have their own state, collaborate with other UAVs to perform a given entry (the firezone discovery and destructor discovery). For a variety of situations that can occur in real life situations, SwarmsSense can organically resolve drones by changing the state of the drones.

SwarmSense allows drones to be a state and coordinate each drone. In SwarmSense, a drone may be at one of the 7 states which are 'SCANNING', 'INITIAL', 'SEARCHING', 'AP PROACHING', 'MAPPING', 'CHARGING' and 'DEAD', as described below:

1. INITIAL: Upon the beginning of the scenario, each and every drone makes an analysis of the current situation and decides the next action and the state.
2. SCANNING: During the INITIAL state, one drone per recovery zone is randomly selected to switch to SCANNING state. The selected drone then scans the entire disaster area with its on-board sensors and makes an initial estimation of the fire zones that are detected by the sensors. (note: not all fire zones are detected during the scanning due to sensors' sensing ranges).
3. SEARCHING: Drones begin to explore a subsection of the entire area to detect and arrive at the tagged fire zone.
4. APPROACHING: Drones move to the disaster area for helping to figure it out quickly.
5. MAPPING: Drones engage in mapping the fire zone by tracking/tracing the boundary of the zone.
6. CHARGING: Drones fly to the designated recovery zones to recharge their batteries.
7. DEAD: Drones are destroyed or become inoperational due to explosion (by fire), crash (by terrain), or depletion of the battery.

Governed by SwarmSense, each drone changes its state according to the algorithm's state transition conditions as

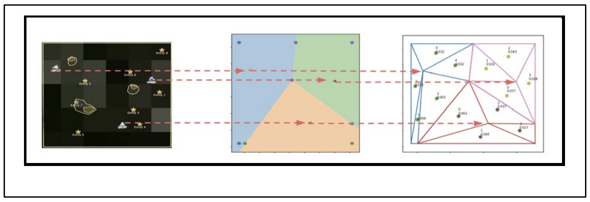

Figure 3. Illustration of the ‘Initial Search Module’ in action

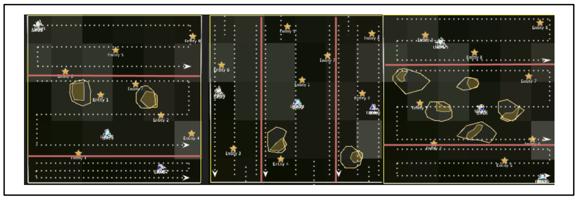

Figure 4. Another search pattern to replace Voronoi diagram

well as the shared information provided in real time by each and every drone during the scenario execution as shown in Figure 2.

2.3. SwarmSense in AMASE

SwarmSense performs performance evaluation in AMASE simulation. SwarmSense controls the behavior of the drones and the collaboration among them in AMASE, which has realistic characteristics and is to the actual environment. It is able to address and prove performance with weather changes, drones' batteries, and their physical defects (broken or missing) through various scenarios in AMASE. Like the real world, in AMASE, wind direction and intensity are changed from time to time, but SwarmSense can solve wind problems flexibly. SwarmSense allows to define and operate a method that works in AMASE through 4 modules which are ‘Initial Searching Module’, ‘Firezone Scanning Module’, ‘Terrain Following Module’, ‘Firezone Mapping Module’. We will cover these later.

2.3.1 Initial Searching Module using patterns

To cover the entire disaster area using the available resources (number of drones and battery lives) as efficiently and fast as possible, the entire disaster area should be divided with similar interval based on drone’s start point [4], [5], [6]. So ‘Initial Searching Module (ISM)’ uses Voronoi diagram that is a partitioning of a plane into regions based on distances to points [7], [8]. Upon the beginning of the scenario, each drone activates ISM. To find the most resource- and time-efficient path for each drone, the entire area is divided into smaller triangle areas by the Voronoi diagram using the entire area’s edges and the recovery zone centers, and each triangle area is assigned a drone. This drone then flies towards the center of the assigned triangle area.

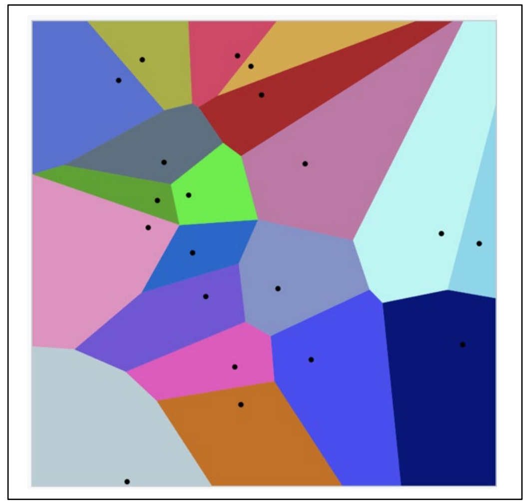

Figure 5. Voronoi diagrams of 20 points

Figure 3 illustrates this process. From the left, the center of each recovery zone is mapped to the center of a region determined by the Voronoi algorithm (If the positions of the recovery zones are parallel or vertical, the Voronoi diagram cannot be applied and a different pattern is applied like Figure 4). This is the concept of the Voronoi diagram we apply. Within the keep in zone, each recovery zone must efficiently distribute the area surveyed by the drones. We assumed that the points of Figure 5 are recovery zones and that the wide rectangle surrounding all points is a keep in zone. If a polygon assigned to each point (recovery zone) is searched appropriately by the drones in the recovery zone, the big problem can be a small problem. The drone's wide area search problem is broken down into smaller problems, which can effectively shorten the time for the drones to find the firezone. From this, a group of triangle regions are created.

2.3.2 Firezone Scanning Module

Searching for an extremely large disaster area with a small number of battery-powered drones without knowing the hotspot locations can lead to significant waste of the resources, and therefore, predicting the direction of likely fire zones using the onboard sensor is highly effective. So, a single drone per recovery zone is randomly selected to execute the ‘Firezone Scanning Module (FSM)’. The FSM is needed to utilize the onboard sensor capable of scanning the entire disaster area up to its sensing range. Any firezone detected during this scanning is stored and shared with all other drones for reducing the time a designated drone takes to arrive at the zone for zone mapping. All drones in each scenario have EO and HazardzoneDetect sensor. EO is a camera sensor, HazardzoneDetect sensor can detect firezone and smokezone which are defined in AMASE. FSM uses a HazardzoneDetect sensor.

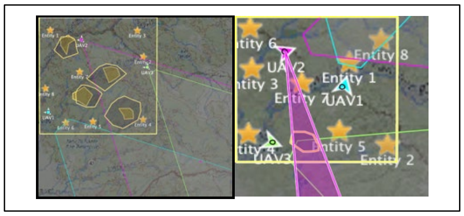

Figure 6. Problem of sensor

Figure 7. Refinement of the shared information

A Hazardzone Detect sensor is used to detect smokezone which could be a hint of firezone. And the sensor's range for smokezone is longer than for firezone. Using this feature, we can estimate initial firezone location. As explained before, FSM selects one drone randomly in each recovery zone. The selected drones are raised up to 3500m altitude to scan as far as possible and set gimbal's elevation value to -3. The values are selected by experiments. And the selected drones start to scan the entire disaster area. While scanning, if smokezone is detected, drones store the detected points. The points are not actual smokezone points, but sensor's center point. Figure 6 shows a polygon of the detected points. We can know the direction of smokezone but can't know exactly where it is. So, the selected drones need to refine the polygon by sharing the collected information to each other. More overlapped and predicted smokezone tend to have higher accuracy, so in terms of the number of overlapped drones estimate an initial firezone location Figure 7. According to Figure 8, we can check the performance of FSM. On average, the firezone detection ratio is increased 37% more. In wildfire where the initial suppression is important, the fact that half of firezones is detected in 20min would be good news.

2.3.3 Terrain Following Module using stair algorithm

AMASE can apply the actual terrain elevation with data form. When the drones are performing a search or scanning mission, they recognize that the drones hit the terrain when the height of the drones is less than the current altitude. As soon as the drones collide with the terrain, the drones disappear in the AMASE simulation and lose one drones. In order to maintain a safe distance between the drone and the terrain (to avoid the crash), we employ 'Terrain Following Module (TFM)' [9], [10]. The TFM is used prevent itself from crash due to the challenging terrain conditions such as

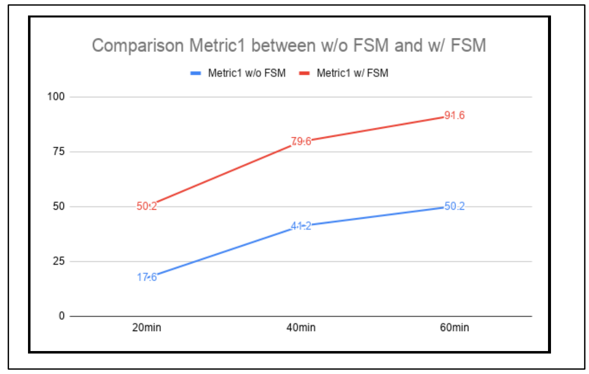

Figure 8. Comparison Metric1 between w/o FSM and w/ FSM

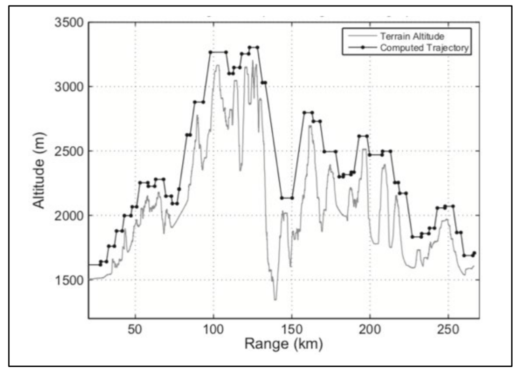

Figure 9. calculated points from stair algorithm

steep and sudden ascending and descending slopes [11].

To address this challenge, the TFM divides the drone's planned path into short segments and calculates the starting and ending points of the drone's ascending and descending as well as its slopes. We used stair algorithm to calculate these points. Stair algorithm can be used to obtain an efficient route that prevents the drone from colliding with the terrain (Figure.9). This algorithm is divided into 6 steps. 1) Divide the entire route into section of equal length {deltaR}; 2) For the ascent case, select the first point of the next section as a node; for descent select the end point of the current section as the node; 3) For ascent, join the node with the current section by a line with a slope equal to the rate of climb vehicle; 4) For descent, find the intersection point of a line passing through the descent node point with the next section; 5) In a Valley where the terrain elevation decreases and then increases again, set specified gap between descent and the next ascent; 6) For the last node point which could be the landing point, find the intersection of a line joining the last node to the previous section.

With this algorithm, we calculated the waypoints the drones would follow, and as a result, the drone that traveled along that waypoint significantly reduced the probability of being destroyed due to the terrain altitude. As a shown Figure 10, we have demonstrated that no drone is destroyed due to terrain.

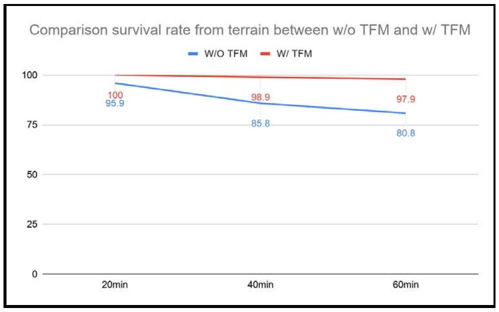

Figure 10. Number of Survival Drones by Time

< Before > < After >

Figure 11. The result of ‘Area MAPPING Module’

2.3.4 Firezone Mapping Module

It required score to represent the performance of the algorithm in objective value. The AMASE simulation scales according to the ratio of how much the drones cover the area of the firezone. The drones search the entire map area and firezone Mapping Module is run. ‘Firezone Mapping Module (FTM)’ is used to approximate the current firezone area. If drones detect the same firezone while searching the assigned area, they start MAPPING the boundary of the zone using the onboard sensor by changing its state to ‘MAPPING’. If a drone stays inside the zone over 5 seconds, it is destroyed by the fire to expedite the zone mapping (MAPPING), the drones at the ‘MAPPING’ state requests drones at ‘INITIAL’ and ‘SEARCHING’ states to join them for assisting the mapping (MAPPING). Those drones selected to participate in the ongoing mapping change the state to ‘APPROACHING’ and fly towards the zone. Figure 11 shows before and after FTM.

3. RESULT

We have developed SwarmSense to improve three specific Metrics. (i) firezone detection ratio (percentage of the firezones detected by the drones with no prior knowledge of their locations); (ii) firezone mapping precision under changing constantly due to dynamic weather conditions; and (iii) drone mission completion ratio (the number of drones

Scenario ID	Metric (i) 20min / 40min / 60min	Metric (ii) 20min / 40min / 60min	Metric (iii) 20min / 40min / 60min
1	33% / 100% / 100%	0% / 26% / 96%	100% / 100% / 78%
2	50% / 100% / 100%	8% / 82% / 66%	100% / 100% / 56%
3	33% / 66% / 66%	1% / 92% / 93%	100% / 89% / 56%
4	50% / 100% / 100%	49% / 75% / 80%	100% / 100% / 33%
5	100% / 100% / 100%	3% / 96% / 70%	100% / 78% / 56%
6	66% / 100% / 100%	21% / 86% / 77%	100% / 78% / 44%
7	50% / 50% / 100%	1% / 50% / 92%	100% / 100% / 56%
8	0% / 0% / 50%	0% / 0% / 11%	100% / 100% / 67%
9	60% / 80% / 100%	15% / 88% / 74%	100% / 78% / 22%
10	60% / 100% / 100%	14% / 63% / 75%	100% / 89% / 44%
Average	50.2% / 79.6% / 91.6%	11.2% / 66% / 73.4%	100% / 91.2% / 45.9%

Figure 12. Total result

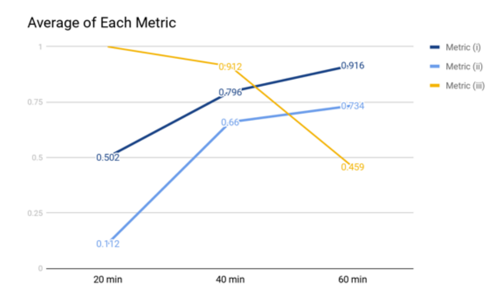

Figure 13. Total result

completing the entire scenario without being destroyed by fires or terrain). Figure 12 and Figure 13 are a summary of the results collected from the experiments conducted with the ten scenarios.

4. DISCUSS

Defining a lot of rules and state of drone, the drones share their information and we can check doing coordination and collaboration automatically to solve problems appropriately based on that information. We solved this issue by using rule-based strategy through current state information. In addition to this, defining reward according to action to figure out disaster and entity is our remaining problem. Solving these kinds of problems by setting discovering of hazard zone and entity to positive reward and time cost and damage to drones to negative reward is also our remaining problems.

5. CONCLUSION

This work presented a new collaborative navigation algorithm for a group of drones so that we control drones to effectively coordinate and share information for disaster response and management applications such as wildfires. Working with a simulation called AMASE, SwarmSense gives the drones a state that defines their specific behavior, and the drones solve the problem through collaboration and data sharing. In the initial search module, the Voronoi diagram was modified to suit the SwarmSense, so the area was divided to perform that the drones search efficiently when the scenarios started. It causes that the drones speed up finding firezones. Next, the firezone scanning module allows the drones to find their position without touching the firezone and perform their mission quickly. Also, the terrain following module improved the survival ability of the drones by applying the stair algorithm to prevent the drones from hitting the terrain altitude. The Firezone Mapping Module maximizes the score at AMASE by preventing the drones from entering the firezone and maximizing the area covered by the drones.

6. FUTURE WORK

There are a few points to improve and to adapt Reinforcement Learning (RL) algorithm. First, in terms of coordination between drones, all drones share each other all collected information of the entire disaster area to narrow down the area that needs to search. Also, all drones know others location and state, so can assign the next zone to go like firezone, smokezone or the areas that need to search by itself when the number of drones or the number of zones is changed. For example, if a drone detects a new firezone, all drones decide where to go based on current zones location and other drones' location. For now, to focus on firezone mapping precision, drones didn't charge own battery. But in reality, to maintain the number of available drones in mission is important, so batteries should be charged by checking progress of mission, other drone's battery state. Next step is solving this problem using Reinforcement Learning (RL) algorithm. These days, RL has been used for finding new solution which is what people can't find. Also, Deep Learning can be used in finding a function that people are hard to find. This work has many problems that we can't define all rules. We introduced many modules that solve each problem we defined; we also can think about RL approach to solve the problems. For tracking fire zone, there are many possible ways to decide drone's angle of incidence and direction. We are impossible to make this formula. And wind also affects drone's movement and position of each zone (fire and smoke). Drones have to change their direction to go as the wind blows. In the environment that has wind information as state and drone's direction, speed as action, well trained RL agent can solve this problem. RL can also solve coordination issue. When two drones are tracking fire zone, they can find optimal tracking solution, communicating with each other. States will be drone's status, being assumed all communications are possible. We can also do end to end training approaches to find new solutions. But that is the case, we will have to use elaborated calculated reward function.

7. REFERENCES

[1] H.J. Min and N. Papanikolopoulos, "The Multi-robot Coverage Problem for Optimal Coordinated Search with an Unknown Number of Robots", *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9-13, 2011

[2] I. Maza, A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms", *In: Alami R., Chatila R., Asama H. (eds) Distributed Autonomous Robotic Systems 6. Springer*, 2007

[3] S. Waharte, N. Trigoni, "Supporting Search and Rescue Operations with UAVs", *International Conference on Emerging Security Technologies*, Canterbury, UK, 6-7 Sept. 2010

[4] K. Sundar and S. Rathinam, "Algorithms for Routing an Unmanned Aerial Vehicle in the presence of Refueling Depots", *IEEE Transactions on Automation Science and Engineering*, Vol 11, No. 01, pp. 287-294, Oct 2013

[5] I. Strumberger, M. Sarac, D. Markovic and N. Bacanin, "Moth Search Algorithm for Drone Placement Problem", *International Journal of Computers*, Vol 3, pp 75-80, 2018

[6] T.M. Cabreira, L.B. Brisolara, P.R. Ferreira Jr., "Survey on Coverage Path Planning with Unmanned Aerial Vehicles", *Drones*, Vol 3, No. 01, 4, 2019

[7] J. Fisher, "Visualizing the Connection Among Convex Hull, Voronoi Diagram and Delaunay Triangulation", 2004

[8] V. Ramasubramaniana, K.K. Paliwalb, "Voronoi Projection-Based Fast Nearest-Neighbor Search Algorithms: Box-Search and Mapping Table-Based Search Techniques", *Digital Signal Processing*, Vol 7, No. 04, pp. 260-277, Oct 1997

[9] M.V. Srinivasan, S. Thurrowgood, D. Socco, "An Optical System for Guidance of Terrain Following in UAVs", *IEEE International Conference on Video and Signal Based Surveillance*, pp. 51, Nov 2006

[10] M. Rahim, S. M. Malaek, "Intelligent Operation Using Terrain Following Flight in Unmanned Aerial Vehicles", *IEEE Aerospace Conference*, Big Sky, MT, US, 18 June 2007

[11] R. Samar and A. Rehman, "Autonomous terrain-following for unmanned air vehicles" *Mechatronics*, Vol. 21, No. 05, pp. 844-860, Aug 2011