Area Coverage Strategy using Multiple UAVs

Akash Kumar Singh[1] and Dharambir Poddar[2]

Abstract— **Modern flying robots, aka drones, are amazing machines. Their application potential is huge and still growing. No wonder so many researchers are turning their attention to this technology and coming up with new exciting usage scenarios. Applications such as environmental monitoring, search & rescue, precision agriculture, and surveillance may benefit from this usage of UAVs with onboard sensors for spatial coverage. These images are usually post-processed for the extraction of desired information, such as digital terrain maps and vegetation indexes.**

In this context, efficient UAV path planning algorithms are of great importance, since the operation time, costs and the quality of the information extracted from the images are directly related to the quality of such planning. So, the aim of this project is to develop and demonstrate an algorithm which given a configuration space, and the number of drones can find a motion plan to cover the whole area in the minimum amount of time and distance covered by the drones.

I. LITERATURE SURVEY

The main contributions of [1] are twofold. First, it provides initial sound discussion and results concerning the construction of the tree as a crucial base for any efficient coverage algorithm. Second, it describes a polynomial-time tree construction algorithm that, as shown in extensive simulations, dramatically improves the coverage time even when used as a basis for a simple, inefficient, coverage algorithm. In this method, Gabriely and Rimon assume that the robot is equipped with a square-shaped shaped tool of size D, hence the area was divided into N cells of size D placed on a grid. The grid was then coarsened such that each new cell is of size 2D X 2D, and a spanning tree was built according to this new grid. After such a tree was built, the robot follows the tree around, creating a hamiltonian cycle visiting all cells of the original grid. Furthermore to tackle obstacles remove the smallest cell from the acquired grid.

Coverage can be sped up with multiple robots but solving several versions of multi-robot coverage problems with minimal cover times is NP-hard. In [2] the author provides some motivation for designing polynomial-time constant-factor approximation algorithms. Then describe multi-robot forest coverage (MFC), a new polynomial-time multi-robot coverage algorithm based on an algorithm for finding a tree cover with trees of balanced weights. The theoretical results show that the cover times of MFC in weighted and unweighted terrain are at most about a factor of 16 larger than minimal. The simulation results show that the cover times of MFC are close to minimal in all tested scenarios and smaller than the cover times of an alternative multi-robot coverage algorithm.

One of the main contributions of [3] is that the number of UAVs used to cover the area is automatically selected by solving the optimization problem. The number of UAVs is influenced by the vehicles' maximum flight time and by the setup time, which is the time needed to prepare and launch a UAV. The strategy for solving the problem presented in the previous section is divided into two parts. The first part, we decompose the area to be covered as a set of sweeping rows, using 'optimal line-sweep-based decompositions for coverage algorithms' these rows form the edges of a graph that is used in the second part of the method, which is based on a vehicle routing problem solution.In that case this paper use 2,3,4 UAVs as sample and optimize the time of covering area by optimization algorithm by iterating numbers of times. Its also take a consideration of launch time, set up time and flying time.

In [4] the author formulates the problem of area coverage as a very familiar graph problem called Multiple Travelling Salesman Problem (MTSP), which aims to visit all the given cities by multiple salesmen along minimized paths. The MTSP is a non-deterministic polynomial-time hard (NP-hard) problem, and requires a lot of computation time to obtain optimal solution. The author divides the whole area as a non overlapping subset of unit cells. These unit cells denote the areas that an UAV can access and becomes the nodes of the graph. The proposed algorithm consists of two parts. First, collision-free subareas are generated. By doing this, it is assured that collision between UAVs as well as collision between UAVs and the obstacles do not take place during the mission. In the second part, the subareas are updated to find the partitioning that minimizes the longest UAV path. By doing this, each UAV can perform nearly equivalent amounts of mission.

[5] presents a coordinated multi-UAV strategy in two scenarios. In the first scenario, symmetric placement of UAVs is assumed at a common optimal altitude and transmit power. In the second scenario, asymmetric deployment of UAVs with different altitudes and transmit powers are assumed. Then, the coverage area performance is investigated as a function of the separation distance between UAVs that are deployed in a certain geographical area to satisfy a target signal-to-interference-plus-noise ratio (SINR) at the cell boundary. The numerical results unveil that the SINR threshold, the separation distance, and the number of UAVs and their formations should be carefully selected to achieve the maximum coverage area inside and to reduce the unnecessary

[1]Akash Kumar Singh, Department of Electrical Engineering, Indian Institute of Technology, Kanpur

[2]Dharambir Poddar, Department of Aerospace Engineering, Indian Institute of Technology,Kanpur,

expansion outside the target area. Thus, this paper provides important design guidelines for the deployment of multiple UAVs in the presence of co-channel interference. The effects of co-channel interference between multiple UAVs in the placement optimization problem,this paper proposes a coordinated multi-UAV framework to study the coverage area performance in presence of co-channel interference. Specifically, multiple UAVs are deployed at the predefined coordinates in a two-dimensional (2D) Cartesian plane by exploiting hexagonal layout. These coordinates are specified for a minimum UAV separation distance to avoid collision in a given target area and utilize SNR measures to find the optimal altitude of UAVs. After the initial deployment of UAVs at the specific coordinates and the optimal altitude, this paper characterizes the impact of the UAV separation distance on the coverage area optimization in the presence of co-channel interference with the help of signal-to-interference-plus-noise ratio (SINR) metrics.

II. SOLUTION

We went through all of the above papers and read different approaches that they had in solving the area coverage problem. Some have modeled it as a graph while some just propose doing a sweeping motion to cover the whole space. But we have decided to try out an approach which has never been done before. We are going to use Linear Temporal Logic to describe the problem of area coverage using multiple drone. We will explain what is it and we will give an example how we are going to describe the problem in case of 4 UAVs. The reason of choosing LTL over others was that it is more generisable. It works flawlessly with 4-10 number of drones and obstacle free environment. It is easy to understand. It is describe the problem this way.

Linear Temporal Logic is different from propositional logic in the sense that it contains a time element. In propositional we can't describe the requirements that should be met in sequences, but using LTL we can do it. That is what makes it convenient to describe the mission of a robot in LTL. Using LTL to describe the coverage and then using SMT solver, like Z3-solver, to later solve the problem.

Linear Temporal Logic (LTL) formulas are constructed from a set of observations, Boolean operators, and temporal operators. We use the standard notation for the boolean operators (i.e., T (true), ¬ (negation), ∧ (conjunction)) and the graphical notation for the temporal operators (e.g., O ("next"), U ("until")). The operator is a unary prefix operator and is followed by a single LTL formula, while U is a binary infix operator. Formally, we define the syntax of LTL formulas as follows:

$$\Phi = T|o|\Phi 1 \wedge \Phi 2|\neg\Phi|O\Phi|\Phi 1 U \Phi 2 \tag{1}$$

where o ϵ O is an observation and Φ, Φ1 and Φ2 are LTL formulas.

Example: Let us assume that a 2D workspace is divided into small rectangular blocks using a grid. The size of the workspace is 5 X 5. L, R, U, D that can take the robot from its current block location to the left, right, upper and lower block respectively. The cost of incurring a movement are as follows:

L, R, U, D - 1 unit

Diagonal Movements - 1.5 unit

Stay - 0.5 unit

There are 4 robots with an initial position. The robots have to explore the whole workspace and reach their final destinations. LTL formula that captures the requirement stated above are as follows:

	X	X		
	X		X	
R3	R4			
R1	R2	X	X	

E - empty space, X - obstacle, (R1, R2, R3, R4) - robots' initial location

Let X[r][t] and Y[r][t] store x and y coordinates of robot r at time step t. Let P[r][t] be the motion primitive taken by robot r at time step t. C[r][t] stores the cost incurred by robot r upto time step t. P[r][t] can take integer values as defined below:

P = 0 (same location), P = 1 (up), P = 2 (down), P = 3 (left), P = 4 (right), P = 5 (up left), P = 6 (up right), P = 7 (down left), P = 8 (down right)

So we will be using X[r][t], Y[r][t], P[r][t], C[r][t] to establish temporal logic relation for collision avoidance, obstacle avoidance:

Simultaneous occupancy:

$$(X[ri][t], Y[ri][t])! = (X[rj][t], Y[rj][t])(i! = j) \tag{2}$$

Head on collison:

$$(X[ri][t], Y[ri][t])! = (X[rj][t+1], Y[rj][t+1]) \tag{3}$$

$$(X[rj][t], Y[rj][t])! = (X[ri][t+1], Y[ri][t+1]) \tag{4}$$

Obstacle avoidance:

$$AND((X[r][t], Y[r][t])! = (Xobs, Yobs) \tag{5}$$

(Xobs, Yobs) ϵ {(2, 0), (3, 0), (1, 2), (3, 2), (1, 4), (2, 4)}

So, all the above 3 conditions are combined together using conjunction as (2) ∧ ((3) ∨ (4)) ∧ (5), such that the result of the whole expression should true in all the conditions. This is then fed to the z3-solver (an SMT solver), which produces one of many possible solution to the problem.

Upon getting this solution we can translate it into 3D coordinates and feed it to a local planner to follow. We plan to Gazebo along with PX4 SITL (check simulation section more details). We are going to code a local planner which can take a series of waypoint and generate a trajectory which we can then feed to the PX4 controller in gazebo. Then we can any image stitching algorithms whose implementations are easily available on the internet.

III. Simulation Setup Details

Gazebo is a powerful 3D simulation environment for autonomous robots that is particularly suitable for testing object-avoidance and computer vision. It does not have the rendering capabilities like Airsim but has all the necessary tools to at least check the dynamics of the problem. When we are using Gazebo we need to simulate the dynamics of a quadcopter and its aerodynamics and that is a different problem in itself. So we will be using a tool that already takes care of that and allows us to just focus on the high-level planning problems. PX4 Gazebo SITL is an integration between the PX4 controller and the Gazebo simulator. PX4 is a low-level controller that is responsible for achieving a particular attitude passed on by the guidance and navigation system. Gazebo SITL has a plugin that simulates the PX4 firmware on the real hardware. When we want to use ROS with the current setup then MAVROS becomes very crucial. MAVROS is a ROS package that acts as a link between PX4 firmware which runs as a gazebo and the ROS node. Using MAVROS one can send target position, speed, acceleration, etc. One can also change the modes of the controller using various services. We will be developing some landscapes for testing out the algorithm that we will develop.

IV. Timeline

- **20th Sep - 3rd Oct:** Writing the script to produce the LTL conditions.
- **4th Oct - 10th Oct:** Setup the Gazebo-PX4 SITL.
- **Mid Sem Evaluation** (Review and catch up on the backlogs)
- **18th Oct - 24th Oct:** Make a simplistic local planner that can execute a given motion plan from the high level planner.
- **25th Oct - 7th Nov:** Make a proper local planner with trajectory generator given a sequence of waypoints from the high level planner.
- **8th Nov - 21st Nov:** To have a working demo in gazebo
- **22nd Nov - 28th Nov:** Documentation and finishing all the backlogs.
- **End Sem Evaluation** (Final Report submit)

References

[1] N. Agmon, N. Hazon and G. A. Kaminka, "Constructing spanning trees for efficient multi-robot coverage," Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., Orlando, FL, 2006, pp. 1698-1703, doi: 10.1109/ROBOT.2006.1641951.

[2] X. Zheng, S. Koenig, D. Kempe and S. Jain, "Multirobot Forest Coverage for Weighted and Unweighted Terrain," in IEEE Transactions on Robotics, vol. 26, no. 6, pp. 1018-1031, Dec. 2010, doi: 10.1109/TRO.2010.2072271.

[3] Avellar, G.S.C.; Pereira, G.A.S.; Pimenta, L.C.A.; Iscold, P. Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time. Sensors 2015, 15, 27783-27803.

[4] Ann, Sungjun Kim, Youdan Ahn, Jaemyung. (2015). Area Allocation Algorithm for Multiple UAVs Area Coverage Based on Clustering and Graph Method. IFAC-PapersOnLine. 48. 204-209. 10.1016/j.ifacol.2015.08.084.

[5] A. A. Khuwaja, G. Zheng, Y. Chen and W. Feng, "Optimum Deployment of Multiple UAVs for Coverage Area Maximization in the Presence of Co-Channel Interference," in IEEE Access, vol. 7, pp. 85203-85212, 2019, doi: 10.1109/ACCESS.2019.2924720.