# A Multiple UAV Path-Planning Approach to Small Object Counting with Aerial Images


Miles Krusniak
*Yale University*
New Haven, Connecticut
miles.krusniak@yale.edu

Alec James
*University of California, Berkeley*
Berkeley, California
alehero@berkeley.edu

Angel Flores
*University of Puget Sound*
Tacoma, Washington
aherreraflores@pugetsound.edu

Yi Shang
*University of Missouri*
Columbia, Missouri
shangy@missouri.edu



***Abstract*—UAV and object detection technology can automate object counting tasks, but low-altitude images, providing better inferences, require greater energy expenditure. Cooperative UAV teams can overcome this shortcoming, particularly when object density information is predetermined. We consider an aerial object-counting problem for a conservation use case: multiple quadrotor UAVs operating in a continuous region with a known probability distribution of objects (waterfowl). We build upon single-drone algorithms, generating density-based subregions covered by UAVs with unique roles. This approach scales with mission parameters, resulting in an F1 score improvement of more than 0.2 under various conditions compared to naive multiple-drone approaches.**



***Index Terms*—Unmanned aerial vehicle, small object detection, aerial image, UAV path planning**


## I. Introduction

The expansion of unmanned aerial vehicles (UAVs or drones) into the consumer market has allowed many difficult aerial imaging tasks to be automated and simplified, including applications such as search and rescue, remote sensing, and wildlife monitoring [1]. One such task has been counting wildlife such as waterfowl to aid in conservation efforts, providing crucial information to issue hunting licenses, define protected areas, and monitor populations. Currently, this task is typically done using a manned fixed-wing aircraft flying over refuge areas, with a human passenger estimating the number of waterfowl below on the spot. We intend to automate this process, covering the area with a team of quadrotor UAVs and using an object detection algorithm to find and count waterfowl based on the high-quality images produced. This requires an algorithm which dynamically creates a path for each UAV in a team, covering as much area as possible while taking into account objects detected during the flyover. Since altitude has a strong effect on image resolution and therefore object detection (as indicated in Figure 1), such paths may revisit objects at a lower altitude as needed [2].

This specific type of UAV path planning problem presents difficulties and opportunities not represented in classical coverage path planning or trajectory planning problems. Existing studies with ground robots (and those which treat UAVs as analogous to ground robots) account for neither the greater freedom of movement of an aerial environment nor the possible change in field of view available to UAVs by ascending or descending. However, these approaches can provide a helpful basis for an initial, predetermined plan before any objects have been detected.

In typical multiple-agent coverage path planning, every $k$-fold increase in team size can at most improve coverage efficiency by a factor of $k$, assuming homogeneous agents and a simple area. In this detection quality regulated UAV object detection problem, however, it may be desirable to take multiple images of the same object from different altitudes, meaning that strategic placement of UAVs can improve coverage even further. For instance, UAVs flying at a high altitude can provide a detailed map of the objects below to a low-altitude UAV, avoiding unnecessary ascents and descents and splitting the problem into high-altitude coverage path planning and low altitude point-by-point visits.

The extent to which larger UAV team sizes can leverage this effect depends on the exact distribution of objects. Fortunately, in cases such as waterfowl counting, historical records and congregation patterns can inform algorithms of the relative importance of subregions of the area, which merit more detailed imaging (though it may still be important to cover the entire area). In fact, as demonstrated in Figure 2, the preferred approach changes fundamentally based on object density - at a low density, it is effective for UAVs to descend immediately upon detecting an object to gain more detailed imagery, while at a high density, it is easier to allocate the detailed imaging task to a lower-altitude drone (represented by the green path in Figure 2).

Fig. 1. Decoy waterfowl at various altitudes. As the number of pixels per waterfowl decreases with altitude, detection accuracy does so as well.


This research was supported in part by NSF Grant CNS-1950873 and a grant from the Missouri Department of Conservation.

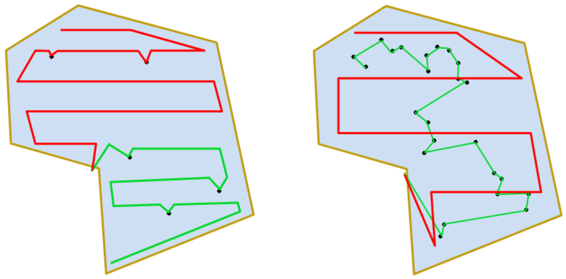

Fig. 2. Intuitive two-drone approaches for low (left) and high (right) object density.

Our mission is to operate a team of drones through a polygon-defined area, using predetermined density information to collect high-quality imagery which produces accurate object detection results. We assume the team of UAVs consists of effectively identical quadcopters equipped with cameras of fixed aspect ratio. Using the intuitive distinction between high- and low-density regions shown above, an approach can scale well with both area size and object quantity and achieve better detection than either approach alone or naive methods.

## II. Related Work

### A. Classical Coverage Path Planning (CPP)

With a single drone and no altitude effects (on tool size or detection confidence), the UAV routing problem described here is equivalent to a standard coverage path planning (CPP) problem, which has been studied extensively in the literature. The general logistics of CPP are as follows: Given an area of interest and a robot with an attached tool (often a sensor, or, in this case, a camera), determine a path that allows the robot to cover the area's entire environment with the tool, and avoid obstacles if possible [3]. Fortunately, as we are using aerial robots at a fairly high altitude, we need not consider physical obstacles.

There are several various CPP setups that have been proposed, including methods to extend CPP to multiple drones and various ways to decompose complex areas into simpler ones. We assume the input area can be described as a polygon, which may be concave. Many CPP algorithms use a decomposition method which simplifies the traversal of a concave area by dividing it into subregions. [3] divides these methods into "exact" and "approximate" cellular decompositions. Approximate cellular decompositions can better account for abnormalities such as deep concavities and missing spaces within the area by constructing an inexact grid representation of the area, at the cost of unnecessary coverage around the edges where the grid may not line up with the area.

One particularly interesting CPP algorithm, based on an approximate cellular decomposition, is Spanning Tree Coverage (STC) [4]. In this approach, an area is divided into cells dilated by a factor of two with respect a robot's tool size. Then, a minimum spanning tree (MST) is calculated, where each cell is a node and each edge is a possible branch. A circumnavigation of this MST will cause the entire area to be covered, and the robot will return to its original position. Because STC can be expanded to multiple drones and always returns to the start point, it is especially promising for large areas which a team of drones may be expected to handle; as such we incorporate STC as a component of our UAV counting approach below.

Classical CPP does not typically account for multiple robots, though some approaches like the one above can be extended easily to this case [4]–[6]. While it is trivial to split a long, single-drone coverage path into equal segments to be followed by other agents, this may interfere with start location limitations.

### B. Energy Efficient Path Planning

To further optimize path planning strategies, it is important for UAVs to fly at an optimal speed during the counting process. This optimal speed varies based on path length and current speed [7], and UAVs travelling in short, interrupted paths may have decreased air time compared to long-distance UAVs. Within our simulations, optimal speed is calculated automatically based on the work from [7] and does take into account the benefits of long-distance flight.

Furthermore, it is also important to consider that once the mission starts, it is preferable for a UAV to be in motion if possible. A UAV which is motionless is still spending energy hovering while not contributing to traversal, and compared to the energy cost of simply staying in the air, motion is relatively inexpensive [8]. To avoid this, hovering UAVs awaiting assignments from other drones in the team can follow along another UAV's path, under the assumption that incoming assignments will be within that UAV's field of view and therefore nearby.

### C. Multiple UAV Path Planning

There is also a significant body of work which considers task management among UAV teams in a three-dimensional environment. Mazdin and Rinner [9] suggest a framework for UAVs imaging three-dimensional objects in which a UAV can take one of four essential jobs, which they name "Explore," "Cover," "Auction," and "Observe." Drones shift through these roles to detect objects, calculate observation points, assign those observation points to other drones, and visit those observation points respectively. As our object-counting problem can involve stages analogous to "Explore" and "Observe," in which objects are first detected and then visited at a low altitude, this framework is a useful basis for an approach. In our use case it is especially important to tailor the Explore and Auction roles, as waterfowl are much smaller than the objects considered by [9] and may be clustered such that multiple waterfowl can be covered with a single auctioned point.

Authors in [10] provide an important decision making strategy to find the next movement to be taken among a drone team. We consider both the current energy level of a UAV and the proximity of other agents around each UAV to determine a path. Furthermore, we also use the current position and path of each drone to decide which UAVs are suited for new assignments.

As far as area coverage is concerned, some authors consider multi-UAV coverage as similar to classical CPP, while others have proposed fundamentally different approaches. [11], for instance, generates multiple partial approximate cellular decompositions with different cell sizes to account for a team of non-identical UAVs, while [12] proposes a similar method based on an exact cellular decomposition - mirroring the existing dichotomy between CPP approaches. On the other hand, [13] suggests an interesting online approach which is easily adapted to multiple drones, in which an initial image is improved upon by recursively repeating the process on subareas which contain objects of interest. Unfortunately, areas large enough to warrant multi-drone coverage cannot fit into a single image at a reasonable altitude. For this reason, any multiple-UAV system with an object counting mission adheres to some constraints of two-dimensional, single-drone path planning.

Our method, below, is related to these existing multi-UAV approaches, but better accounts for a varied distribution of objects. This method is inspired primarily by Mazdin and Rinner [9]'s approach of assigning distinct jobs to UAVs, and Santamaria [11]'s approach in decomposing the region into different-sized cells to account for individual UAV requirements.

## III. Approach

### A. Problem Formulation

**Existing coverage formulation**: This work uses a variation of the dynamic-height coverage path problem discussed in [2]. In that formulation, we know the following about the region $R$ (considered to be a 2 dimensional subspace of $\mathbb{R}^3$ at height 0) and the objects it contains:

$\rho(j,h)$, the detection confidence at altitude $h$ of object $j$;
$\eta(j)$, an indicator for the ground truth of object $j$; and
$\tau(p \in \mathbb{R}^3)$, the objects within the field of view from point $p$.

Of those, $\tau(p)$ and $\rho(j,h)$ are available to a function $\delta(\{p_0..p_{i-1}\},\{\tau(p_0)..\tau(p_{i-1})\})$, which produces the next point $p_i$ to which a drone should travel based on points it has already traveled to and the objects detected at those points.

**Expanded formulation**: There are two additions to this formulation required to fit it to a density-aware, multiple-drone use case. First, we assume that there is a function $\zeta(q \in \mathbb{R}^3)$ which is a probability density function over the region $R$ such that $\zeta(q \notin R) = 0$ and $\int_{q \in R} \zeta(q) = 1$. In a conservation use case, $\zeta(q)$ can be determined from historical records and environmental knowledge of where waterfowl congregate.

Second, we must account for the inclusion of multiple drones in the team. We assume all $n$ drones have the same capabilities and energy capacity. For a drone indexed $i$ in the team, let $\delta_i(P,T)$ output the next point in the path of that drone, where $P$ is the set of past points for all drones and and $T$ is the set of objects detected at each of those points; that is, $T = \{\tau(p) : \forall p \in P\}$. We do assume that all drones have access to the complete state of every other drone no matter their relative distance (as the wireless range of modern drones exceeds the width of any area which can be practically covered.) Note that the relationship among the $\delta_i$ may be complex, as not every point in $R$ requires the same amount of travel time and energy. Furthermore, detected objects found by other drones may affect $T$ before drone $i$ has finished reaching its next point, requiring an abrupt change of plan. The exact behavior is dependent on a motion model of the drones, and though difficult to analyze directly may be reflected through simulation.

To determine the effectiveness of our algorithm, we have used F1 score as a metric (treating the problem as primarily a question of object detection). This metric is easily extracted from the formulation with the indicators $\eta(j)$ (indicating ground truth) and $\rho(j,k(j)) > t$ (indicating detected truth) for an object $j$, the lowest altitude at which it was detected $k(j)$, and a confidence threshold $t$.

### B. UAV Team Division

The formulation above allows UAVs to more wisely divide and cover an area by splitting the task not only according to the amount of area which must be covered, but also the density of objects in that area.

Intuitively, the preferred strategy for covering an area with a high confidence is differs based on the density of objects within the area. As mentioned, at a low density, a single drone may immediately descend upon detecting objects because most images will contain no objects and therefore descent is rarely required. However, a high density requires frequent descent, incurring a significant energy cost which cannot be planned for and may result in failure to traverse the entire area. While this is a significant hurdle for a single drone, with multiple drones it is possible to strategically divide the workload to avoid this effect. The object-counting approach given here differs from others found in the literature by capitalizing on this fact.

Figure 3 demonstrates the distinction between direct-descent and layered strategies by showing simulation results for two-drone systems implementing each strategy. Both strategies were tested with twenty trials against a one-kilometer-wide

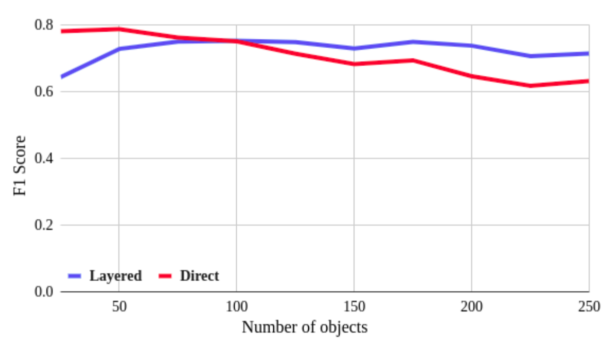


Fig. 3. F1 score comparison for two two-drone approaches. In the "layered" approach one drone scans at a high altitude and another visits detected objects at a low altitude; in the "direct" approach each drone descends itself upon object detection.

area containing a variable number of equally distributed objects. According to these results, the object density at which the preferred strategy changes is around one hundred objects in the area (about 127 per square kilometer - relatively few in the waterfowl counting case).

To take advantage of this, we have used a framework similar to the one presented in [9], in which every drone may take one of three roles. The subregions covered by each role are determined by density. Continuous regions in which the object density is greater than a threshold are calculated, forming a set of high density areas such that each is a polygon. Next, each drone is assigned to one of the following roles:

**"Traverse"**: Drones with the "Traverse" role cover high density areas at a medium altitude, ensuring that all objects in those areas are found, albeit with some false positives.

**"Explore"**: Drones assigned to the "Explore" role cover low density areas, detecting outliers while conserving energy by flying at a high altitude.

**"Observe"**: "Observe" drones remain at a very low altitude, visiting nearby objects as directed by "Traverse" drones to achieve high confidence on as many as possible.

Roles are assigned using parameters $p$ and $q$, such that $p\%$ of all drones are "Traverse" or "Observe", and $q\%$ of those are "Observe". Figure 4 visually demonstrates each of these roles. Below, we discuss each in greater detail.

### C. Low Density Exploration

At a low density where there are relatively few objects, exploration resembles a height-variant but otherwise traditional coverage path planning problem. While we assume that it is important to cover the entire area in case of unexpected clusters (and to provide imagery which can be analyzed by a human if desired), most images will not contain objects, and any energy spent diverging from the path to inspect objects at a lower altitude will be small compared to the energy cost of the path as a whole. Due to the large size of any area warranting coverage by multiple drones, it is of greater importance here to end exploration near the start point.

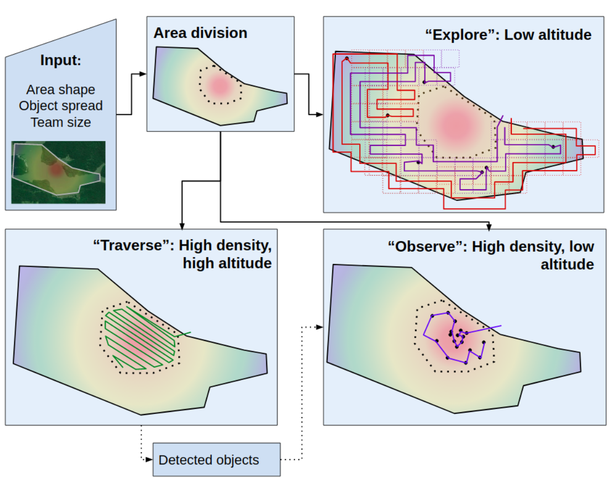


Fig. 4. Flowchart of the algorithm.

With these considerations in mind, this algorithm is based on work done in [4], using elements of wavefront transform methods ( [14]) to expand it to be compatible with multiple drones. [4] demonstrates that a discretized area's spanning tree can be circumnavigated to create a complete coverage path throughout a region, simultaneously causing an agent following it to return to its starting position, saving energy in return flights. This utilizes an approximate cellular decomposition according to the hierarchy presented in [3], appropriate as the low-density area may have holes and irregularities where high-density areas have been removed.

We discretize $R$ into rectangular, "major" cells through a grid overlay, each major cell of area $4a$ and of aspect ratio $l * w$, with major cells lying within a high-density polygon disconsidered. These major cells can then be subdivided into "minor" cells of size $a$ with the same aspect ratio. To split the cells among all drones with the "Explore" role, each cell in this discretized area is scored based on its distance from the edge of the entire traversal area. Starting at some initial point, each drone selects the single cell which fulfills the following criteria:

- It is orthogonally adjacent to a cell already in that drone's tree (or if no cells fulfill this property, it is as close as possible);
- no other drone's tree contains this cell; and
- it has the lowest score among such cells.

This process repeats until no cells remain, ensuring that all drones' spanning trees have the same number of cells, trace the edge when possible, and contain a point near the start. Paths are generated by circumnavigating these trees.

When an "Explore" drone encounters an object, it will descend to a low-altitude point above the detected object and take a detailed image independently, then return on its normal path. It does *not* delegate this task to another drone.

### D. High Density Traversal

In areas of high density, it is important to achieve a high-quality initial traversal in order to ensure that most objects are detected with any confidence initially (as a decrease in resolution not only decreases detection confidence but also decreases detection likelihood.) It is also less feasible to visit detected objects mid-traversal due to the increased quantity.

Since these subregions are polygonal, they are traversed effectively with an exact cellular decomposition. As these subregions tend to be relatively simple but may be narrow (e.g., along edges of the area), we have selected a simple plow motion in the direction of greatest width of the subregion. A corresponding path is calculated for each high density area, and every "Traverse" drone is assigned one or more such subregions to cover.

If there are more "Traverse" drones than high-density areas, the extraneous drones are given the "Observe" role.

### E. Single Point Observation

As indicated above, drones traversing high-density areas (with the "Traverse" role) should not visit detected objects themselves but instead delegate the task to drones which are already nearby and at a low altitude. In our algorithm, this is the task of drones with the “Observe” role.

Visiting each individual object that has been detected essentially forms a traveling salesman problem which can be solved using any TSP solver. Since the path must be recomputed every time the drone receives information about a new object, a quick heuristic solver is preferred in simulation. The results shown here therefore use a simple greedy solver, but a more nuanced solver will improve the path: for instance, [15] describes a genetic algorithm approach to the multiple-traveling-salesmen problem which not only creates an effective path over objects already assigned to a drone but also reassigns objects as necessary.

As drones with the ”Observe” role do not have a predetermined path, it is possible that they will be required to idle due to a lack of detected points. In this case, an ”Observe” drone follows the path of the nearest ”Traverse” drone, staying nearby (at a lower altitude) until objects are detected.

## IV. Analysis

### A. Simulation

Our analysis of this approach uses the lightweight simulator for detection confidence regulated path planning problems presented in [2], with modifications to allow for arbitrary object distribution and the execution of multiple simultaneous UAVs. Since choreographing multiple UAV actions (or, more mathematically, selecting the next $\delta_i$ with which to build a list of visited points) requires significant precision, the modifications include a simple UAV motion model built out to follow the empirical energy data reported in [7]. UAVs are allowed to interrupt established routes of other team members as if by broadcasting new data, and final calculations are made based on all images gathered by the team.

All simulation results presented here are accurate to the Iris Quadcopter, a simple, consumer-grade UAV which is apt for use in small swarms due to its versatility and relatively low cost. Within the simulator, the quadcopter is capable of taking images while in motion, which prevents unnecessary slowing and conserves energy. All drones within a team start at the same point near the edge of the area. The region itself is assumed to be flat with no aerial obstacles, and must be a continuous region which can be represented by a polygon.

Object detection within the simulator is based on confidence results from Faster R-CNN model trained on a relatively small set of images taken across altitudes. As a result, algorithm effectiveness within the simulator is limited by this model: even when all objects are detected at ten meters of altitude (the minimum permitted in the simulator), F1 score is rarely greater than 0.85, reflecting the inherent difficulties of small object detection that merit this detailed imaging algorithm.

### B. Comparisons

To determine the improvement of this method over naive multi-drone methods, we collected simulation results for four different approaches: a naive, fixed-height approach in which a boustrophedon coverage path is split equally across drones (”Naive Plows”), a dynamic-height approach similar to that in [2] but with an overview path split equally across drones (”Height Variant Plows”), the approach described above (”Hotspotter”), and a hypothetical, superoptimal approach where a single drone has existing knowledge of every object and visits each one by one (”Oracle”). Since most realistic single-drone approaches cannot even finish traversal on the large areas which may be covered by multi-UAV teams, the ”oracle” method is used as a stand-in.

Each test was conducted with twenty trials on randomly generated hexagonal areas. Algorithms were evaluated by F1 score over various area sizes ($w$), object quantities ($j$), object densities ($d$), and drone team sizes ($n$). Figures 5, 6, 7, and 8 show these results, respectively.

The density-aware approach presented above (”Hotspotter”)

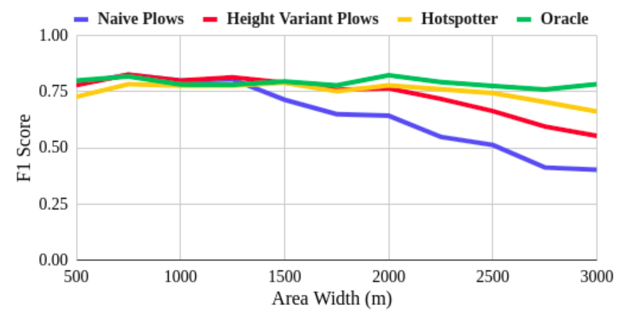


Fig. 5. Comparison across area width (j=20, d=100, n=4)


Fig. 6. Comparison across team size (n=4, d=100, w=3000)

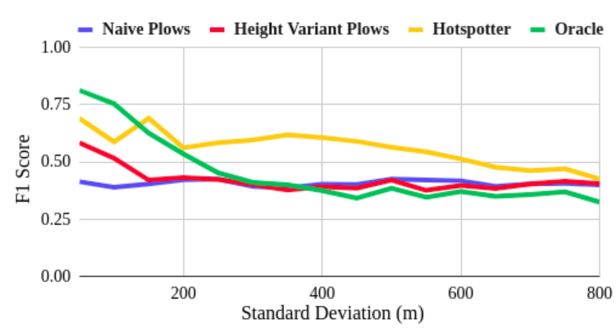


Fig. 7. Comparison across object density (j=100, w=3000, n=4)

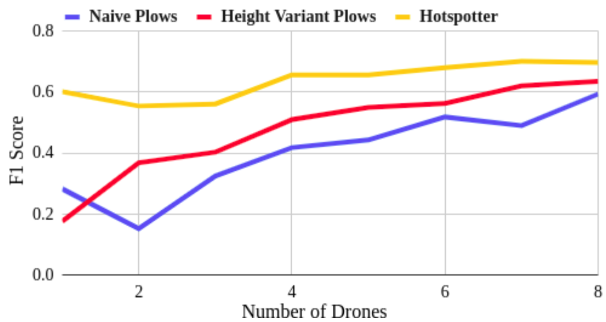


Fig. 8. Comparison across number of objects (j=100, d=100, w=3000)

scales well with all four of these parameters, with behavior based on different object density highlighting the most notable discrepancy between this approach and multi-drone approaches which do not account for existing knowledge of object density. Per Figure 7, there is a significant "sweet spot" of object spread in which most detectable objects (95%, assuming a distribution close to Gaussian) lie within 10% or less of the area. With a less strongly clustered object spread, the "Hotspotter" and "Height Variant Plow" methods begin to converge.

Note also that there are a few situations in which this approach is actually even more effective than the "oracle" approach. For instance, as demonstrated in Figure 6, the F1 score of the "oracle" approach is inversely related to the number of objects due to the requirement of visiting every one, whereas in the "hotspotter" approach presented here, a drone in the "Traverse" role will cover the entire high density area regardless of how many objects it actually contains. (A drone in the "Observe" role will face the same struggle to reach every point as the "oracle," but even if the observe drone fails to visit all objects delegated to it due to energy constraints, it is guaranteed that at least some image on those objects already exists.) The "oracle" also fails dramatically for the same reasons with a wide distribution of objects.

Figure 8 indicates the improvement made upon adding additional drones to various teams. Note that while there seems to be only a relatively small increase in the effectiveness of any algorithm as the team size increases, the area size remains the same as the team grows, and at a certain team size the area is already being covered effectively.

## V. Conclusion

As seen from the figures above, our analysis shows that this multiple-drone, density-aware approach to the UAV object counting problem shows significant improvements to naive multiple-drone approaches, collecting higher-quality object imagery (increasing detection F1 score by approximately 0.2 under a variety of conditions) and scaling well with drone team size, area size, and object quantity.

Many improvements can be made to this type of object detection approach. For instance, our low-altitude "Observe" drones travel in a traveling-salesman type fashion to each object that our "Traverse" drones detect, which is inefficient in cases of very dense clusters - it may be more effective to group nearby objects to be imaged together, avoiding extraneous short-distance travel between objects. This approach also displays shortcomings of the CPP algorithms on which it is based. For instance, because low-density coverage uses a grid-based approximate decomposition, a "staircasing" motion can result when area edges do not align with the grid. An improved or different CPP basis algorithm may alleviate this issue. Finally, future research may involve more flexibility between certain drone roles. Perhaps there are conditions under which "Observe" drones should convert into "Traverse" or "Explore" drones, and vice versa, allowing for the team to be more dynamic and adaptive.

## References


[1] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, "Flying ad-hoc networks (fanets): A review of communication architectures, and routing protocols," in *2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*, 2017, pp. 1–9.

[2] M. Krusniak, K. Leppanen, Z. Tang, F. Gao, Y. Wang, and Y. Shang, "A detection confidence-regulated path planning (dcrpp) algorithm for improved small object counting in aerial images," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2020, pp. 1–6.

[3] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr., "Survey on coverage path planning with unmanned aerial vehicles," 2019.

[4] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 77–98, 2001.

[5] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[6] N. Hazon and G. A. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 735–741.

[7] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of uavs," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*. IEEE, 2015, pp. 111–117.

[8] T. Dietrich, S. Krug, and A. Zimmermann, "An empirical study on generic multicopter energy consumption profiles," in *2017 Annual IEEE International Systems Conference (SysCon)*, 2017, pp. 1–6.

[9] P. Mazdin and B. Rinner, "Coordination of mobile agents for simultaneous coverage," in *PRIMA 2019: Principles and Practice of Multi-Agent Systems*, M. Baldoni, M. Dastani, B. Liao, Y. Sakurai, and R. Zalila Wenkstern, Eds. Cham: Springer International Publishing, 2019, pp. 170–185.

[10] M. Messous, S. Senouci, and H. Sedjelmaci, "Network connectivity and area coverage for uav fleet mobility model with energy constraint," in *2016 IEEE Wireless Communications and Networking Conference*, 2016, pp. 1–6.

[11] E. Santamaria, F. Segor, and I. Tchouchenkov, "Rapid aerial mapping with multiple heterogeneous unmanned vehicles." in *ISCRAM*. Citeseer, 2013.

[12] I. Maza and A. Ollero, "Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6*, R. Alami, R. Chatila, and H. Asama, Eds. Tokyo: Springer Japan, 2007, pp. 221–230.

[13] S. A. Sadat, J. Wawerla, and R. T. Vaughan, "Recursive non-uniform coverage of unknown terrains for uavs," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1742–1747.

[14] A. Zelinsky, R. A. Jarvis, J. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proceedings of international conference on advanced robotics*, vol. 13, 1993, pp. 533–538.

[15] O. K. Sahingoz, "Flyable path planning for a multi-uav system with genetic algorithms and bezier curves," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 41–48.