Energy-Efficient Collaborative Target Tracking in UAV Swarms via Enhanced Voronoi Partitioning

Anubhav Elhence, Tejas Sriganesh, Vinay Chamola, *Senior Member, IEEE*

***Abstract*—Collaborative drone tracking using swarms of unmanned aerial vehicles (UAVs) presents significant advantages in surveillance and monitoring tasks but poses challenges in energy efficiency and coordination. This paper introduces an Enhanced Voronoi Method for optimizing energy consumption and extending mission duration in multi-UAV target tracking operations. By partitioning the operational area into Voronoi cells and implementing a predictive edge-crossing algorithm, the method strategically assigns drones to active, passive, or grounded states based on the target's location and movement predictions. Real-time object detection is achieved using the YOLO model, ensuring accurate and efficient tracking. Simulation results demonstrate that the proposed method significantly outperforms baseline and traditional Voronoi methods, achieving up to a 374% increase in tracking time and a substantial reduction in average battery consumption per drone. The Enhanced Voronoi Method effectively enhances energy efficiency and scalability in collaborative drone tracking, offering a robust solution for applications requiring prolonged and reliable UAV operations.**

***Index Terms*—Unmanned Aerial Vehicles (UAVs), Swarm Intelligence, Resource Optimization, Energy Efficiency**

I. Introduction

The rapid advancement of unmanned aerial vehicles (UAVs), commonly known as drones, has significantly impacted various domains such as surveillance, environmental monitoring, search and rescue, and mapping [1], [2]. Drones offer real-time data acquisition and access to hard-to-reach areas, making them invaluable tools in both civilian and military applications. However, relying on a single drone often limits operational coverage and efficiency due to constraints like battery life, sensor range, and processing capabilities.

To overcome these limitations, collaborative drone swarms have been introduced, where multiple drones work in unison to achieve a common objective [3], [4]. Collaborative drone tracking involves coordinating a swarm of drones to monitor and track moving targets over expansive areas. This approach enhances coverage, provides redundancy, and improves reliability compared to single-drone operations. Despite these advantages, several challenges persist, including efficient area partitioning, seamless coordination among drones, energy management, and real-time data processing.

Traditional methods often involve all drones actively pursuing the target, leading to redundant efforts and rapid battery depletion [5]. Coordinating drone movements to maintain continuous tracking without overlaps or gaps remains a complex problem. Efficiently managing the drones' energy consumption while ensuring seamless target tracking is critical for the success of prolonged missions.

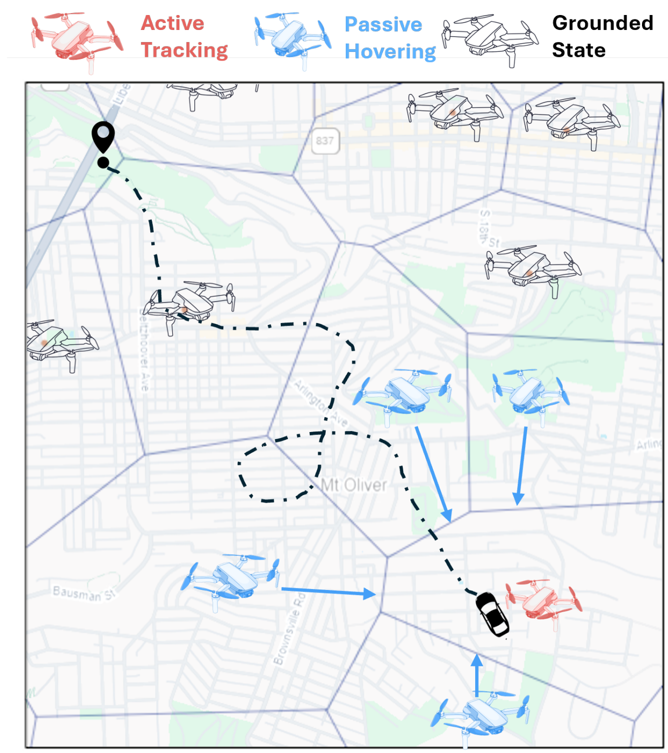

Fig. 1: Illustration of our Proposed Method for Collaborative Drone Tracking.

Voronoi diagrams have been widely utilized in UAV applications to optimize area partitioning and improve coverage efficiency [6], [7]. By partitioning the operational area based on the proximity to each drone, Voronoi diagrams ensure efficient spatial allocation, minimizing overlap and redundancy. However, static Voronoi partitioning may not be sufficient for dynamic environments where targets move unpredictably across cell boundaries [8].

To address these challenges, we propose an Enhanced Voronoi Method for collaborative drone tracking. Our method partitions the operational area into Voronoi cells based on the initial positions of the drones. Each drone is assigned to a specific cell and is responsible for monitoring and tracking the target within that region. As the target moves across cell boundaries, drones coordinate to ensure seamless handover of

Anubhav Elhence, Tejas Sriganesh and Vinay Chamola are with the Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani Campus, India 333031 (e-mail: elhenceanubhav@gmail.com, f20211415@pilani.bits-pilani.ac.in, vinay.chamola@pilani.bits-pilani.ac.in).

Digital Object Identifier: XXXXXXXXXXXX

tracking responsibilities. Drones in neighboring cells hover passively near the boundaries, anticipating potential handovers, while drones in non-adjacent cells remain grounded or in low-energy standby mode to conserve energy.

Figure 1 illustrates our proposed method. The operational area is divided into Voronoi cells, each associated with a specific drone. The drone in the cell where the target is currently located engages in active tracking (red drone), while neighboring drones (blue drones) hover passively near the cell boundaries, ready to take over if the target enters their region. Drones in non-adjacent cells remain grounded, conserving energy until needed.

The key advantages of our method include energy efficiency through strategic drone deployment, seamless coordination for continuous tracking, and optimized resource utilization by engaging only necessary drones in active operations. Our approach effectively balances energy consumption and operational effectiveness, significantly extending the tracking duration compared to traditional methods.

Our major contributions are as follows:

- We develop an Enhanced Voronoi Method for collaborative drone tracking that partitions the operational area into Voronoi cells, assigning specific regions to individual drones for efficient coverage.
- We introduce a predictive edge-crossing algorithm that enables drones to anticipate target movements across cell boundaries, facilitating seamless handovers and continuous tracking.
- We implement an energy-efficient strategy by keeping non-critical drones grounded or in low-energy standby mode, significantly reducing unnecessary energy consumption and extending mission duration.
- We validate our proposed method through simulations, demonstrating significant improvements in tracking time and energy efficiency compared to baseline methods.

The remainder of this paper is organized as follows. Section II reviews related works on Voronoi partitioning and collaborative object tracking, Section III details our proposed methodology, including the Enhanced Voronoi Method and the predictive edge-crossing algorithm. Section IV presents the implementation and simulation settings and Section V discusses the results and the performance improvements achieved. Finally, Section VI concludes the paper.

II. Related Works

The coordination of multiple unmanned aerial vehicles (UAVs) for tasks such as surveillance, mapping, target tracking, and network security has been a significant area of research in recent years. Efficient area coverage, path planning, collision avoidance, energy management, and security are critical challenges that have been addressed through various methodologies, including the use of Voronoi diagrams, cooperative algorithms, optimization techniques, and game-theoretic approaches.

Voronoi diagrams have been widely utilized in UAV applications to optimize area partitioning and improve coverage efficiency. Andreou et al. [6] proposed an algorithmic technique using Voronoi diagrams with circles to optimize network coverage in intelligent transportation systems, demonstrating a simple time complexity and effectiveness in 6G+ infrastructures. Similarly, Huang et al. [9] utilized Voronoi diagrams in combination with threat circles to construct feasible paths for UAVs, enhancing obstacle avoidance and path optimization in dynamic environments.

In the realm of multi-UAV systems, Voronoi-based methods have been employed to enhance formation control and collision avoidance. Hu et al. [3] developed a formation control algorithm based on Voronoi partitioning, allowing UAVs to switch destinations when they reach local equilibrium, validated through simulations and experiments. Dong et al. [7] introduced a fast and communication-efficient multi-UAV exploration method using a hierarchical approach based on dynamic topological graphs and Voronoi partitions, effectively addressing exploration in large environments. Gui et al. [10] proposed a decentralized multi-UAV cooperative exploration using dynamic centroid-based area partitioning, demonstrating the effectiveness of area partitioning methods in enhancing exploration efficiency.

The use of Voronoi diagrams extends to UAV patrolling and monitoring applications. Giuseppi et al. [1] employed dynamic Voronoi tessellations on satellite data for UAV patrolling in wildfire monitoring, demonstrating the applicability of Voronoi-based methods in disaster management scenarios. Pan et al. [11] proposed a multi-UAV relay deployment algorithm based on Voronoi diagram division, improving the balance of UAV relay load and lowering energy requirements for relay communication.

Path planning is another critical aspect where Voronoi diagrams have been leveraged. Chen and Chen [8] researched UAV dynamic path planning algorithms based on Voronoi diagrams, utilizing improved Dijkstra algorithms for real-time tracking of moving targets in dynamic environments. Yang et al. [12] combined ant colony optimization algorithms with Voronoi graphs to enhance convergence speed and prevent local optima in UAV route planning. Zhang et al. [13] conducted a quantitative evaluation of Voronoi graph search algorithms in UAV path planning, analyzing the impact of threat proportion and number of threats on planning results.

In agricultural applications, Voronoi diagrams have been used for analyzing UAV images. Ren et al. [2] developed a method that effectively extracts seedling quality metrics from UAV images of maize, performing better in representing seedling uniformity than traditional methods.

Cooperative exploration and coverage using multi-UAV systems have also been extensively studied. Huang et al. [4] proposed a Voronoi-based algorithm enabling cooperative UAVs to search for targets in unknown environments, effectively preventing collisions and reducing the search solution space. Chen et al. [14] designed a distributed coverage algorithm for multi-UAV systems using average Voronoi partitioning, ensuring identical areas of coverage for each UAV in a convex region. Renzaglia et al. [15] suggested using partial initial knowledge of the environment to find suitable starting configurations for agents in local optimization, utilizing constrained centroidal Voronoi tessellations for coverage of partially known 3D surfaces. Wang and Wang [16] incorporated attention mechanisms into multi-UAV area coverage track planning, enhancing feature fusion capability and generating optimal paths.

Energy efficiency and resource optimization are critical in multi-UAV operations. Lu et al. [17] researched 3D deployment of heterogeneous UAV base stations based on Voronoi diagram division, improving user channel capacity and reducing average path loss. Song et al. [18] proposed a method based on bounded Voronoi diagrams and watershed segmentation algorithms for mosaicking UAV orthoimages, offering a feasible solution for large-scale UAV photomosaic generation.

Security and routing in UAV swarms have been addressed using game-theoretic and optimization approaches. Bansal et al. [5] proposed a pricing Stackelberg game for Security as a Service (SECaaS) in UAV swarms, formulating behavioral utilities and creating optimal price strategies to maximize profit. In another work, Bansal et al. [19] introduced SHOTS, a scalable secure authentication-attestation protocol using optimal trajectories in UAV swarms, leveraging Physical Unclonable Functions (PUFs) to guarantee physical security and establish trust in a lightweight manner.

Multi-objective routing in UAV networks has been explored by Mahajan et al. [20], who proposed a Multi-Objective Markov Decision Process (MDP)-based routing in UAV networks for search-based operations. The method uses Q-learning in an MDP framework and compares routing paths using metrics like REMEN, PD ratio, and ED, demonstrating effectiveness in optimizing routing decisions.

Our work differs from these studies by focusing on an enhanced Voronoi-based method for collaborative drone tracking that emphasizes energy efficiency through strategic drone deployment. While previous works have utilized Voronoi diagrams for area partitioning and path planning, our approach integrates a predictive edge-crossing algorithm and implements an energy-efficient strategy by keeping non-critical drones grounded, significantly extending mission duration. Additionally, our method leverages real-time object detection and tracking using optimized algorithms, such as the YOLO model [21], which has shown superior performance in accuracy and processing speed compared to other models like SSD and R-CNN.

In summary, while prior research has laid the groundwork for using Voronoi diagrams in UAV applications for area partitioning, path planning, cooperative exploration, and security, our contribution lies in the integration of these concepts with energy-efficient strategies and predictive algorithms to enhance collaborative drone tracking performance

III. Methodology

In this section, we present our Enhanced Voronoi Method for collaborative drone tracking, which aims to optimize energy efficiency and maintain continuous target tracking in drone swarms. The methodology comprises four main components: Voronoi-based partitioning for efficient area coverage, target tracking using optimized object detection, an edge-crossing prediction algorithm for seamless handover between drones, and energy-efficient drone deployment strategies. The integration of these components enhances tracking duration, accuracy, and resource utilization.

A. Voronoi-Based Partitioning

Voronoi diagrams are fundamental tools in computational geometry for partitioning space based on proximity to a set of generator points [22]. In our context, each drone acts as a generator point, and the operational area is partitioned into Voronoi cells. This ensures that every location within the area is assigned to the nearest drone, minimizing overlap and redundancy, and optimizing coverage efficiency.

Figure 2 illustrates the step-by-step Voronoi partitioning process for drone swarms. The diagrams show how Voronoi cells are generated and updated as drones are added, ensuring each drone is responsible for a specific, non-overlapping region.

1) Mathematical Formulation: Let $\mathcal{D} = D_1, D_2, \ldots, D_n$ be the set of n drones, where each drone D_i has a position $\mathbf{p}_i \in \mathbb{R}^2$. The Voronoi cell V_i associated with drone D_i is defined as:

$$V_i = \mathbf{x} \in \mathbb{R}^2 \mid |\mathbf{x} - \mathbf{p}_i| \leq |\mathbf{x} - \mathbf{p}_j|, \ \forall j \neq i \ , \tag{1}$$

where $|\cdot|$ denotes the Euclidean norm. Each drone D_i is responsible for monitoring its Voronoi cell V_i.

This static Voronoi partitioning ensures:

- **Efficient Coverage**: The entire area is covered without overlaps or redundancy.
- **Simplified Coordination**: Fixed boundaries eliminate the need for dynamic recalculations, reducing computational overhead.
- **Scalability**: Easily extendable to larger areas by increasing the number of drones and their initial positions.

As the target moves between cells, tracking responsibilities are seamlessly handed over to the appropriate drone in the corresponding cell. This approach minimizes redundant energy consumption and optimizes resource utilization.

B. Target Tracking Using Optimized Object Detection

Accurate and real-time target tracking is crucial for effective surveillance and monitoring. We employ the YOLO (You Only Look Once) object detection model [21] for efficient target identification and localization. As demonstrated in our performance comparison (see Table I), YOLOv3 outperforms other models such as SSD, R-CNN, and ResNet in terms of mean Average Precision (mAP), Recall, F1-score, and Accuracy.

The high accuracy and real-time performance of YOLOv5 facilitate the processing of high-resolution video data from drones, enabling timely detection and tracking of the target within each drone's Voronoi cell.

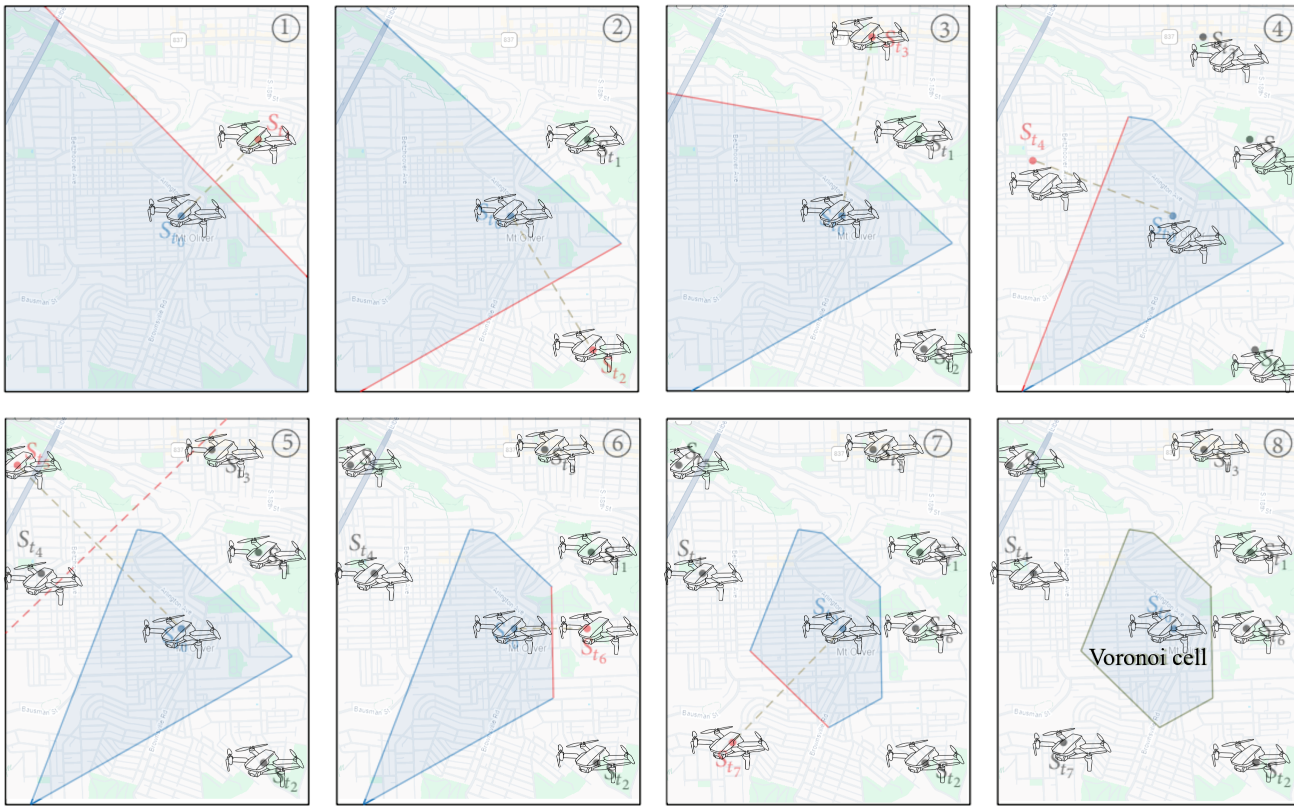

Fig. 2: Step-by-step Voronoi partitioning process for drone coverage. The diagrams show how Voronoi cells are generated and updated as drones are added, ensuring each drone is responsible for a specific, non-overlapping region.

TABLE I: Performance Comparison of Object Detection Models

Model	mAP	Recall	F1-Score	Accuracy
SSD	0.847	0.668	0.874	92.66%
R-CNN	0.913	0.725	0.924	93.54%
ResNet	0.924	0.875	0.875	95.61%
YOLOv3	**0.979**	**0.935**	**0.956**	**98.78%**

1) Kalman Filtering for Target State Estimation: To maintain the target centered in the drone's field of view and to predict its future position, we apply a Kalman filter **[?]** for estimating the target's state. The Kalman filter effectively handles noisy measurements and provides optimal estimates of the target's position and velocity.

Assume the target has a state vector $\mathbf{s}(t) = [x(t), y(t), \dot{x}(t), \dot{y}(t)]^\top$, where $(x(t), y(t))$ is the position, and $(\dot{x}(t), \dot{y}(t))$ is the velocity at time t. The motion model is:

$$\mathbf{s}(t+1) = \mathbf{A}\mathbf{s}(t) + \mathbf{w}(t), \tag{2}$$

where $\mathbf{A}$ is the state transition matrix, and $\mathbf{w}(t)$ represents process noise.

Each drone uses the Kalman filter for predicting the target's movement and adjusting its orientation accordingly. This results in efficient and accurate tracking, enabling seamless handovers between drones when the target crosses into neighboring cells.

C. Edge-Crossing Prediction Algorithm

Predicting when the target will cross from one Voronoi cell to another is crucial for seamless tracking and coordination among drones. We develop an Edge Crossing Prediction Algorithm that anticipates boundary crossings based on the target's estimated trajectory.

1) Algorithm Overview: The algorithm aims to predict the likely crossing point of a moving target as it approaches the boundary of its current cell and estimate the future cell it may enter. By passively tracking the target's position and velocity, the algorithm calculates the closest perpendicular distance to each edge in the current cell.

Algorithm 1 outlines the steps involved in predicting edge crossings and repositioning drones accordingly.

2) Mathematical Formulation: Let the target's velocity vector be $\mathbf{v}(t)$. To determine if the target is moving towards edge E_j, we compute the angle between the target's velocity vector and the normal vector of the edge. The edge normal vector $\mathbf{n}_j$ is given by rotating $\mathbf{AB}_j$ by 90 degrees.

If the dot product $\mathbf{v}(t) \cdot \mathbf{n}_j < 0$, the target is moving towards the edge. This, combined with the distance d_j, allows us to predict potential edge crossings.

3) Illustration of Tracking Conditions: Figure 3 illustrates different target tracking conditions in a Voronoi-partitioned environment. The figure demonstrates various scenarios when a target transitions between Voronoi cells, emphasizing the

Algorithm 1 Edge Crossing Prediction Algorithm

Data: $\mathbf{s}(t)$: Target's state at time t, $\mathbf{p}_i$: Drone's position, V_i: Voronoi cell of drone D_i, $\mathbf{v}$: Target velocity, $\mathbf{E}$: Edges of Voronoi cell, $\mathbf{T}$: Target trajectory model

```
// To predict edge crossing and reposition drone:
```

for *each drone* D_i **do**
- **Identify the edges of the current Voronoi cell:** $E_i = \{E_1, E_2, ..., E_k\}$
- **Predict target's future position:** $\hat{\mathbf{s}}_i(t+\Delta t) = \mathbf{s}(t) + \mathbf{v}\Delta t$
- **for** *each edge* $E_j \in E_i$ **do**
  - **Compute the closest point on edge:**
  - Calculate the vector $\mathbf{AB}_j$ of edge E_j: $\mathbf{AB}_j = B_j - A_j$
  - Compute the projection factor t_j: $t_j = \frac{(\mathbf{P}-A_j)\cdot\mathbf{AB}_j}{|\mathbf{AB}_j|^2}$
  - Clamp t_j to [0, 1] to ensure the closest point lies on the segment:
  - $t_j = \max(0, \min(1, t_j))$
  - **Compute the closest point** Q_j **on edge** E_j**:**
  - $Q_j = A_j + t_j \cdot \mathbf{AB}_j$
- **end**
- **Predict the target's crossing point for each edge:**
- **for** *each edge* $E_j \in E_i$ **do**
  - **If the target is moving towards edge:**
  - Compute the perpendicular distance to edge: $d_j = |\mathbf{P}(t) - Q_j|$
  - **Reposition drone to predicted crossing point:**
  - Move drone D_i to Q_j as the closest point for tracking
- **end**

end

seamless handover of responsibility among drones to maintain continuous tracking.

a) Condition 1: Target Moving Inside a Single Cell: The target is located within a single Voronoi cell, and only the drone assigned to that cell is actively tracking the target.

b) Condition 2: Target Approaching a Boundary: The target is near the boundary of its current Voronoi cell. The drone predicts the target's trajectory and alerts the neighboring drone to prepare for a potential handover.

c) Condition 3: Target Crossing a Boundary: The target crosses into a neighboring Voronoi cell. The current drone hands over tracking responsibilities to the neighboring drone, which takes over active tracking.

d) Condition 4: Target Near Multiple Boundaries: The target moves close to the intersection of multiple Voronoi boundaries. Multiple neighboring drones are alerted, but only the drone that the target is most likely to enter (based on trajectory prediction) prepares for handover.

D. Energy-Efficient Drone Deployment

To optimize energy consumption and extend mission duration, we implement an energy-efficient strategy by dynamically adjusting the states of the drones:

- **Active Tracking**: Only the drone in the Voronoi cell where the target is currently located engages in active tracking.
- **Passive Hovering**: Drones in neighboring Voronoi cells hover passively near the edges, ready to take over tracking if the target crosses a boundary.
- **Grounded State**: Drones in non-adjacent Voronoi cells remain grounded or in low-energy standby mode, conserving battery life.

By minimizing the number of drones in the air at any given time and strategically activating drones based on the target's predicted movements, we significantly reduce unnecessary energy consumption. This approach aligns with energy management strategies discussed in [19] and demonstrates improved tracking duration as shown in our results.

E. Communication and Coordination Protocol

Efficient communication among drones is essential for coordination and seamless handovers. We utilize a lightweight communication protocol with the following features:

- **Event-Driven Communication**: Drones communicate only when necessary, such as when the target is approaching a boundary or when a handover is required.
- **Broadcast Messages**: Drones broadcast alerts to neighboring drones when a potential edge crossing is predicted.
- **Acknowledgment and Synchronization**: Receiving drones send acknowledgments and synchronize their tracking parameters to ensure a smooth transition.

The communication protocol is designed to be energy-efficient and scalable, minimizing bandwidth usage and avoiding unnecessary transmissions, similar to approaches in [20].

Our Enhanced Voronoi Method integrates static Voronoi partitioning, optimized object detection using YOLO, an edge-crossing prediction algorithm, and energy-efficient drone deployment strategies to enhance collaborative drone tracking. By addressing challenges related to area coverage, seamless coordination, energy management, and real-time processing, our approach improves tracking accuracy and extends mission duration.

In the next section, we present the results of our simulations, demonstrating the effectiveness of the proposed methodology in achieving significant performance gains over baseline methods.

IV. Implementation

In this section, we detail the practical implementation of our Enhanced Voronoi Method for collaborative drone tracking. We describe the simulation environment, the modeling of drones and the target, the setup of simulation parameters, and the realization of the algorithms presented in the methodology. We also discuss the calculation methods for the results, including mathematical formulations where necessary.

Our implementation aims to validate the proposed method through simulations that replicate realistic operational scenarios. The simulations are designed to evaluate the performance of our method in terms of tracking time, energy efficiency, and scalability.

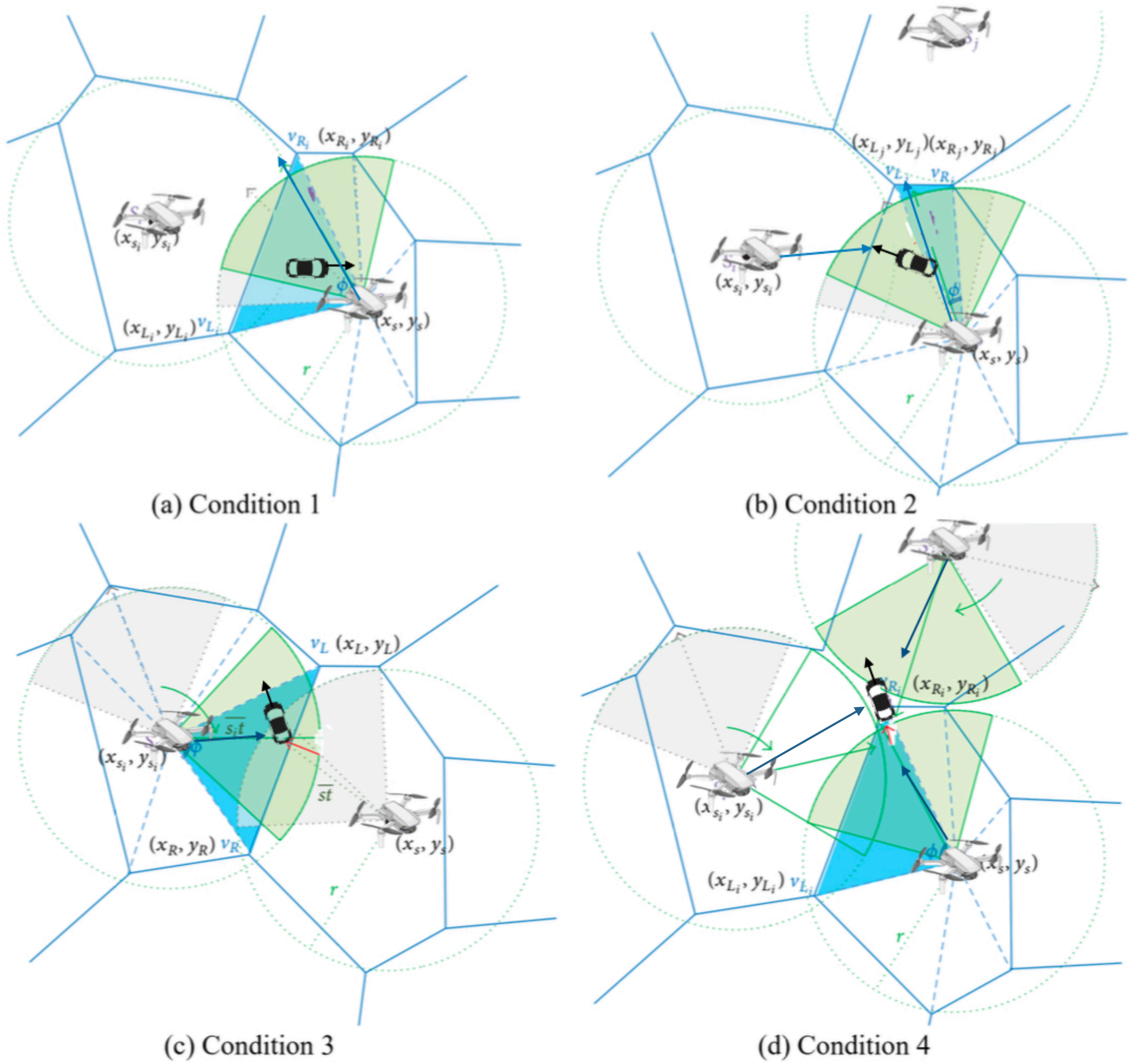

Fig. 3: Illustration of target tracking conditions in a Voronoi-partitioned environment for drone swarms. The diagrams show how drones coordinate tracking as the target moves across Voronoi boundaries.

We utilized Python and relevant libraries to develop a custom simulation environment, ensuring flexibility and reproducibility. The simulation provides a controlled setting to test the proposed method under various conditions and to collect data for analysis.

A. Simulation Environment

The key components of the simulation environment include:

- **Operational Area**: A two-dimensional square space representing the surveillance region, partitioned into Voronoi cells based on the initial positions of the drones.
- **Drones**: Modeled as agents with properties such as position, velocity, energy levels, and operational states (active, passive, grounded).
- **Target**: A moving object with a predefined or random trajectory, representing the entity to be tracked.

The simulation environment updates the positions of the target and drones at discrete time steps, allowing for the dynamic modeling of interactions and state transitions.

B. Drone and Target Modeling

Each drone is represented by its position $\mathbf{p}_i(t)$ at time t, energy level $E_i(t)$, and operational state $S_i(t) \in$ active, passive, grounded. The drones have limited battery capacities and consume energy based on their activities:

$$E_i(t+1) = E_i(t) - (e_{\text{active}} \cdot a_i(t) + e_{\text{passive}} \cdot p_i(t)) \cdot \Delta t, \quad (3)$$

where $a_i(t)$ and $p_i(t)$ are indicator functions representing whether drone i is in active or passive state at time t, and e_{active}, e_{passive} are the corresponding energy consumption rates. Grounded drones consume negligible energy and are considered to have $e_{\text{grounded}} = 0$.

The target is modeled with a state vector $\mathbf{s}(t) = [x(t), y(t), \dot{x}(t), \dot{y}(t)]^\top$, moving within the operational area. The target's trajectory can be predefined (e.g., following a specific path) or randomized to simulate unpredictable movement.

C. Implementation of the Enhanced Voronoi Method

We compute the Voronoi diagram of the operational area based on the initial positions of the drones using the `scipy.spatial.Voronoi` function from the SciPy library [23]. Each drone is assigned its corresponding Voronoi cell, which remains static throughout the simulation. This partitioning ensures efficient coverage and simplifies coordination among drones. Each drone utilizes the YOLO object detection algorithm to detect the target within its Voronoi cell. For the implementation, we used the pre-trained YOLOv3 model provided by the `Darknet` framework [21], interfaced via the `PyTorch` library for Python.

The drones apply a Kalman filter to estimate the target's state. The prediction and update equations are implemented as per the standard Kalman filter algorithm, using NumPy arrays for matrix computations. The Edge Crossing Prediction Algorithm is implemented as described in Algorithm 1. For each drone, the algorithm calculates the distance to the edges of its Voronoi cell and predicts potential boundary crossings based on the target's estimated trajectory. To compute the projections and distances, we use vectorized operations in NumPy for efficiency. The clamping of the projection factor t_j is handled using the `numpy.clip` function.

The drones transition between active, passive, and grounded states based on the target's location and predicted movements. The state transitions are implemented using conditional statements that evaluate the target's proximity to the drone's Voronoi cell and the results from the edge-crossing prediction. Table II summarizes the key parameters used in the simulations.

TABLE II: Simulation Parameters

Parameter	Value
Operational area size (L)	1000 meters
Number of drones (n)	10
Drone maximum speed (v_{max})	10 m/s
Drone sensing range (r_{sense})	200 meters
Drone communication range (r_{comm})	300 meters
Target speed (v_{target})	5 m/s
Time step (Δt)	1 second
Total simulation time (T_{total})	1000 seconds
Battery capacity per drone (E_{max})	1000 units
Energy consumption/s (active pursuit) (e_{move})	2 unit/sec
Energy consumption/s (passive hovering) (e_{move})	1 unit/sec

D. Calculation of Results

The total tracking time before battery depletion is calculated by simulating the drones' operations until all drones have exhausted their batteries. We record the time at which the last drone depletes its battery, representing the maximum tracking duration achievable under the given method and drone count.

The tracking time is directly obtained from the simulation by keeping track of the simulation time variable t until the condition $\forall i, E_i(t) \leq 0$ is met.

The average battery consumption per drone is calculated over specified simulation durations (e.g., 100 seconds and 250 seconds). For each drone, we compute the cumulative energy consumed up to the specified time and calculate the average across all drones:

$$\text{Average Battery Consumption} = \frac{1}{n}\sum_{i=1}^{n}\left(E_{\text{max}} - E_i(T_{\text{sim}})\right), \tag{4}$$

where $E_i(T_{\text{sim}})$ is the remaining energy of drone i at simulation time T_{sim}.

For analyzing battery depletion patterns, we plot the energy levels $E_i(t)$ of each drone over time. This provides insights into the energy consumption behavior under different methods and drone counts. The staggered battery depletion patterns observed in the graphs indicate the efficiency of workload distribution among the drones. We implemented the main simulation loop and the key algorithms using Python functions and classes. The pseudocode for the main simulation loop is provided in Algorithm 2.

E. Visualization and Analysis

We used Matplotlib to visualize the operational area, drone positions, Voronoi partitions, and target trajectory. For the results, we plotted graphs of tracking time versus drone count, battery consumption over simulation durations, and battery depletion patterns for varying drone counts. These visualizations facilitate the comparison of our method with baseline and traditional Voronoi methods.

The implementation utilized the following software and libraries:

- **Python 3.8**: Programming language for the simulation.
- **NumPy**: For numerical computations and array operations.
- **SciPy**: For computing Voronoi diagrams [23].
- **Matplotlib**: For plotting and visualization.
- **PyTorch**: For implementing the YOLO object detection model.

To ensure reproducibility, we set random seeds where applicable and documented all simulation parameters. We conducted multiple runs for each scenario to account for stochastic variations in the target's movement and to obtain average performance metrics.

Parameter tuning was performed to optimize the energy consumption rates and threshold distances for edge-crossing predictions. We adjusted the parameters based on initial test runs to achieve a balance between tracking performance and energy efficiency.

Algorithm 2 Main Simulation Loop

Data: $\mathbf{x}_{\text{target}}(t)$: Target position at time t, $\mathbf{v}_{\text{target}}$: Target velocity, n: Number of drones, $\mathbf{p}_i(t)$: Drone position, $\hat{\mathbf{s}}_i(t)$: Drone state estimate, $\mathbf{z}_i(t)$: Observation from drone D_i, $\mathbf{E}_i(t)$: Battery level of drone D_i

`// For each time step` t:

for *each time step* t **do**

  Target Movement:

  Update target position: $\mathbf{x}_{\text{target}}(t) = \mathbf{x}_{\text{target}}(t-1) + \mathbf{v}_{\text{target}} \cdot \Delta t$

  for *each drone* D_i **do**

    Repositioning:

    Update drone position: $\mathbf{p}_i(t+1)$ to stay within its assigned cell or at the closest point on the edge if near the boundary.

    Observation:

    Acquire observation: $\mathbf{z}_i(t)$ if the target is within the sensing range of D_i.

    State Estimation:

    Update the state estimate: $\hat{\mathbf{s}}_i(t)$ using the Kalman filter based on the acquired observation $\mathbf{z}_i(t)$.

    Edge Prediction:

    Predict when the target will cross the boundary of the drone's cell by calculating the closest point on the edge and estimating the crossing time.

    Communication:

    Exchange information with neighboring drones regarding target position, state estimates, and predictions of edge crossings.

    Data Fusion:

    Fuse estimates from neighboring drones using Covariance Intersection to improve the accuracy of the target's state.

    Battery Update:

    Update drone's battery level: $\mathbf{E}_i(t)$ based on energy consumption (e.g., movement, communication).

    if $\mathbf{E}_i(t) < \mathbf{E}_{critical}$ **then**

      Emergency Landing:

      Initiate emergency landing protocol.

    end

  end

  Logging:

  Log positions, state estimates, and energy levels for all drones and the target.

end

The mathematical models used in the implementation, such as the Kalman filter equations and energy consumption calculations, were validated against standard formulations. We verified the correctness of the algorithms by testing them in isolated scenarios before integrating them into the main simulation.

Our implementation focuses on a simulated environment, and certain real-world factors such as communication delays, physical drone dynamics, and environmental conditions are simplified. Future work includes extending the implementation to account for these factors and to validate the method using real drones in field experiments.

V. Results and Discussion

To evaluate the performance of our proposed Enhanced Voronoi Method, we conducted extensive simulations comparing it with the Baseline Method and the traditional Voronoi Method. The simulations assessed key performance metrics, including total tracking time before battery depletion, average battery consumption per drone, and battery depletion patterns over time. The results demonstrate the effectiveness of our method in enhancing tracking duration, energy efficiency, and scalability in collaborative drone tracking.

A. Total Tracking Time Before Battery Depletion

Figure 4 presents the total tracking time before battery depletion for the three methods across different numbers of drones (6, 8, and 10). The Baseline Method, where all drones are continuously in active pursuit of the target, exhibits the shortest tracking time of 500 seconds regardless of the number of drones. This is attributed to the high energy consumption resulting from all drones operating in the energy-intensive active mode simultaneously, leading to rapid battery depletion.

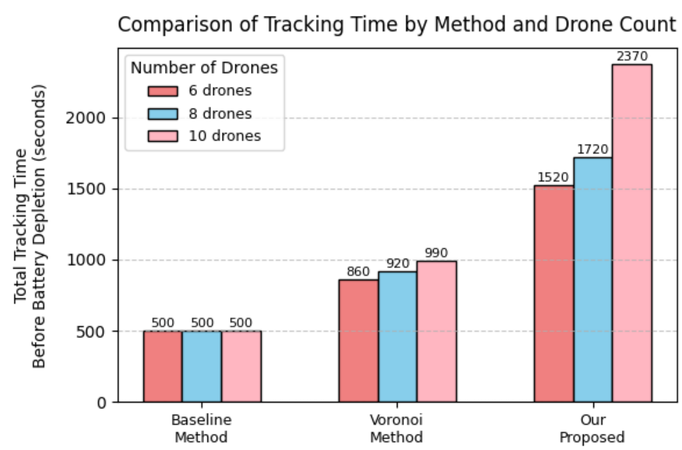

Fig. 4: Comparison of total tracking time before battery depletion across different methods and drone counts.

The Voronoi Method shows an improvement in tracking time, with durations increasing from 860 seconds with 6 drones to 990 seconds with 10 drones. In this method, only the drone within the target's Voronoi cell actively pursues the target, while neighboring drones hover passively near cell boundaries. This reduces the number of drones consuming energy at the higher active rate, thereby extending the tracking time compared to the Baseline Method.

Our proposed Enhanced Voronoi Method achieves the longest tracking times, significantly outperforming the other methods. With 6 drones, the tracking time extends to 1520 seconds, and with 10 drones, it reaches 2370 seconds. This substantial improvement is due to the strategic grounding of non-neighboring drones, which conserves energy by minimizing unnecessary passive hovering. Only the drone in the target's cell is active, and neighboring drones are in passive hover mode, ready to take over if the target moves into their cell. This efficient energy management allows the drones to maintain operations for a much longer period before battery depletion.

These results demonstrate that the Enhanced Voronoi Method effectively extends the mission duration by optimizing drone deployment and energy consumption. As the number of drones increases, the tracking time improves, highlighting the scalability of the method and its suitability for large-scale operations.

B. Average Battery Consumption per Drone

Figure 5 compares the average battery consumption per drone across the three methods over simulation durations of 100 seconds and 250 seconds. The Baseline Method consistently exhibits the highest battery consumption due to all drones being in active pursuit mode. Over 100 seconds, drones consume an average of 200 units of energy, which increases to 500 units over 250 seconds, representing a linear relationship between time and energy consumption.

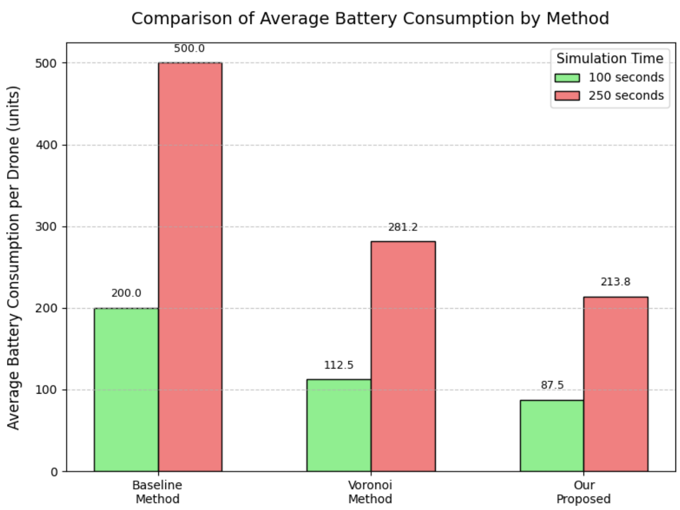

Fig. 5: Comparison of average battery consumption per drone across methods over simulation durations of 100 seconds and 250 seconds.

In contrast, the Voronoi Method reduces average battery consumption to 112.5 units over 100 seconds and 281.2 units over 250 seconds. This reduction is due to only one drone being in active pursuit at any given time, while neighboring drones consume less energy in passive hovering mode. However, the energy consumed by passive drones still contributes to overall consumption.

The Enhanced Voronoi Method achieves the lowest average battery consumption, with 87.5 units over 100 seconds and 213.8 units over 250 seconds. By keeping non-neighboring drones grounded, the method minimizes the number of drones consuming energy, leading to significant energy savings. The results indicate that the Enhanced Voronoi Method is approximately 57% more energy-efficient than the Baseline Method and 24% more efficient than the Voronoi Method over 250 seconds.

These findings underscore the effectiveness of the Enhanced Voronoi Method in conserving energy and highlight its potential for extending mission durations in practical applications where energy resources are limited.

C. Battery Depletion Patterns Over Time

To further analyze energy efficiency and workload distribution, we examined the battery depletion patterns over time for different drone counts under the Enhanced Voronoi Method. Figures 6, 7, and 8 depict the battery levels of drones over time for 6, 8, and 10 drones, respectively.

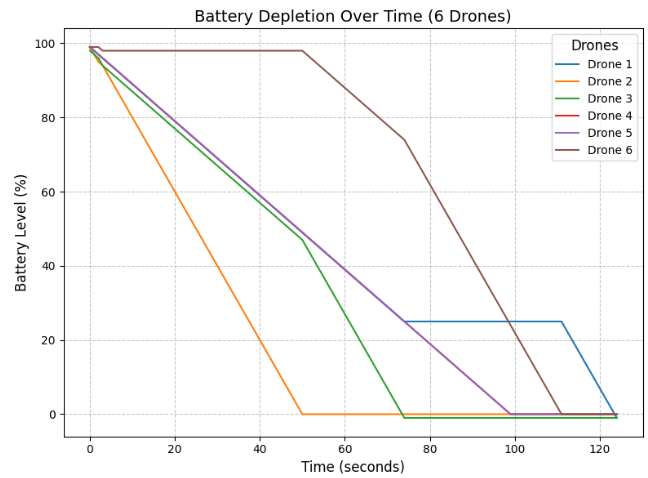

Fig. 6: Battery depletion patterns over time for 6 drones using the Enhanced Voronoi Method.

In the case of 6 drones (Figure 6), the battery levels decrease relatively quickly, with complete depletion occurring around 1520 seconds. The energy burden is higher on each drone due to the smaller number of drones sharing the workload. Some drones operate continuously in active pursuit mode, while others alternate between passive hovering and preparing for edge-crossing predictions. The limited number of drones results in a higher workload per drone, leading to faster battery depletion.

With 8 drones (Figure 7), the battery depletion is more staggered, indicating better workload distribution. The operational time extends to approximately 1720 seconds, as the drones alternate between active, passive, and grounded states more effectively. The increased number of drones allows for more efficient sharing of tracking responsibilities, reducing the energy consumption per drone.

For 10 drones (Figure 8), the battery depletion patterns show the most efficient energy usage, with drones maintaining higher battery levels for longer periods. The total operational time reaches 2370 seconds, demonstrating the scalability of the method. The increased number of drones allows for a more effective distribution of workload, as drones spend more time in the grounded state when not immediately required for tracking. This reduces the overall energy consumption and extends the mission duration significantly.

These patterns highlight the Enhanced Voronoi Method's ability to distribute the workload among a larger number of

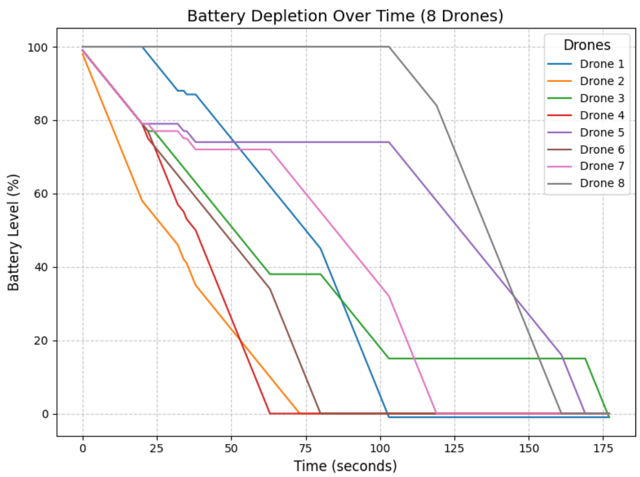

Fig. 7: Battery depletion patterns over time for 8 drones using the Enhanced Voronoi Method.

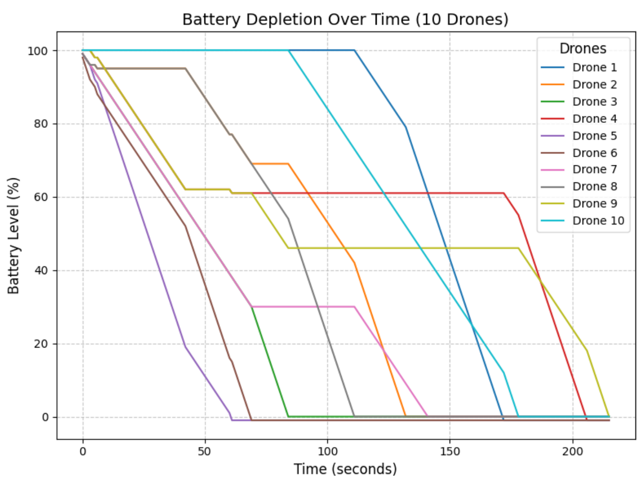

Fig. 8: Battery depletion patterns over time for 10 drones using the Enhanced Voronoi Method.

drones, reducing energy consumption per drone and enhancing the overall efficiency of the drone swarm.

D. Comparison of Performance Metrics

Table III summarizes the maximum tracking time and average battery consumption across the methods for different drone counts. The data corroborate the observations from the figures and reinforce the advantages of the Enhanced Voronoi Method.

The table clearly shows that the Enhanced Voronoi Method consistently outperforms the other methods in terms of maximum tracking time and energy efficiency. The improvements are more pronounced as the number of drones increases, highlighting the method's scalability and effectiveness in larger drone swarms.

E. Discussion

The results demonstrate that our Enhanced Voronoi Method significantly improves collaborative drone tracking by optimizing energy usage through strategic drone deployment. By keeping non-critical drones grounded and leveraging predictive algorithms for seamless handovers, the method extends mission duration and enhances energy efficiency without compromising tracking accuracy.

The method effectively addresses challenges related to energy management and coordination in multi-UAV systems, which are critical for prolonged operations in surveillance, environmental monitoring, and search-and-rescue missions. The scalability of the method makes it suitable for various applications requiring extensive area coverage and continuous target tracking.

Furthermore, integrating the YOLO object detection model enhances real-time target tracking capabilities, ensuring accurate and timely responses to target movements [**?**]. The combination of advanced object detection with efficient coordination strategies positions the Enhanced Voronoi Method as a robust solution for collaborative drone tracking.

These findings align with previous research emphasizing the importance of energy efficiency and coordination in multi-UAV systems [19], [20]. Our method contributes to this field by providing a practical approach that significantly extends operational time and optimizes resource utilization.

While the simulation results are promising, certain limitations exist. The simulations assume ideal conditions, neglecting potential real-world factors such as communication delays, environmental obstacles, varying weather conditions, and dynamic changes in the operational area. Future work includes incorporating these factors into simulations and conducting field experiments to validate the method under real-world conditions.

Additionally, exploring adaptive Voronoi partitioning in dynamic environments and integrating machine learning techniques for improved predictive capabilities could further enhance the method's performance. Investigating the impact of varying drone speeds, communication ranges, and energy consumption rates would provide deeper insights into optimizing multi-UAV systems.

VI. Conclusion

We have presented an Enhanced Voronoi Method for collaborative drone tracking that significantly improves energy efficiency and extends mission duration in multi-UAV systems. By partitioning the operational area into Voronoi cells and strategically managing drone deployment based on the target's location and predicted movements, our method minimizes unnecessary energy consumption by grounding non-critical drones and optimizing active and passive roles among the swarm. Simulation results demonstrate that our approach outperforms traditional methods, achieving up to a 374% increase in tracking time compared to the baseline and a substantial reduction in average battery consumption per drone. The integration of real-time object detection using the YOLOv5 model further enhances tracking accuracy and responsiveness. Our method effectively addresses key challenges in multi-UAV coordination and energy management, offering a scalable and practical solution for prolonged surveillance and monitoring applications.

TABLE III: Comparison of Maximum Tracking Time and Average Battery Consumption Across Methods

Method	**Maximum Tracking Time (seconds)**			**Average Battery Consumption (units)**	
	6 Drones	**8 Drones**	**10 Drones**	**100 seconds**	**250 seconds**
Baseline Method	500	500	500	200.0	500.0
Voronoi Method	860	920	990	112.5	281.2
Enhanced Voronoi Method	1520	1720	2370	87.5	213.8

REFERENCES

[1] A. Giuseppi, R. Germanà, F. Fiorini, F. D. Priscoli, and A. Pietrabissa, "Uav patrolling for wildfire monitoring by a dynamic voronoi tessellation on satellite data," *Drones*, 2021.

[2] L. Ren, C. Li, G. Yang, D. Zhao, C. Zhang, B. Xu, H. Feng, Z. Chen, Z. Lin, and H. Yang, "The detection of maize seedling quality from uav images based on deep learning and voronoi diagram algorithms," *Remote. Sens.*, vol. 16, p. 3548, 2024.

[3] J. wen Hu, M. Wang, C. Zhao, Q. Pan, and C. Du, "Formation control and collision avoidance for multi-uav systems based on voronoi partition," *Science China Technological Sciences*, vol. 63, pp. 65 – 72, 2019.

[4] C. Huang, B. Du, and M. Chen, "Multi-uav cooperative online searching based on voronoi diagrams," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, pp. 3038–3049, 2024.

[5] G. Bansal, V. Chamola, B. Sikdar, and F. Yu, "Uav secaas: Game-theoretic formulation for security as a service in uav swarms," *IEEE Systems Journal*, vol. 16, pp. 6209–6218, 2022.

[6] A. Andreou, C. Mavromoustakis, J. M. Batalla, E. Markakis, and G. Mastorakis, "Uav-assisted rsus for v2x connectivity using voronoi diagrams in 6g+ infrastructures," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 15 855–15 865, 2023.

[7] Q. Dong, H. Xi, S. Zhang, Q. Bi, T. Li, Z. Wang, and X. Zhang, "Fast and communication-efficient multi-uav exploration via voronoi partition on dynamic topological graph," *ArXiv*, vol. abs/2408.05808, 2024.

[8] X. Chen and X. Chen, "The uav dynamic path planning algorithm research based on voronoi diagram," in *Chinese Control and Decision Conference*, 2014, pp. 1069–1071.

[9] Z. Huang, Z. Zhang, J. Han, C. Huang, and H. Zhang, "Research on uav path planning based on voronoi map construction and improved ant colony algorithm," in *International Conference on Electronic Information Technology*, vol. 12719, 2023, pp. 1 271 948 – 1 271 948–8.

[10] J. Gui, T. Yu, B. Deng, X. Zhu, and W. Yao, "Decentralized multi-uav cooperative exploration using dynamic centroid-based area partition," *Drones*, 2023.

[11] W. Pan, N. Lv, K. Chen, Y. Pan, X. Hong, and X. Zhang, "Multi-uav relay deployment algorithm based on voronoi diagram division," in *International Conference on Emerging Technologies*, 2023, pp. 624–628.

[12] F. Yang, X. Peng, L. Zhao, and B. Cui, "Route planning of four rotor uav based on voronoi graph and ant colony algorithm," 2017.

[13] C. Zhang, H. Liu, and Y. Tang, "Quantitative evaluation of voronoi graph search algorithm in uav path planning," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, 2018, pp. 563–567.

[14] S. Chen, C. Li, and S. Zhuo, "A distributed coverage algorithm for multi-uav with average voronoi partition," in *International Conference on Control, Automation and Systems*, 2017, pp. 1083–1086.

[15] A. Renzaglia, J. Dibangoye, V. L. Doze, and O. Simonin, "Multi-uav visual coverage of partially known 3d surfaces: Voronoi-based initialization to improve local optimizers," *ArXiv*, vol. abs/1901.10272, 2019.

[16] J. Wang and R. Wang, "Multi-uav area coverage track planning based on the voronoi graph and attention mechanism," *Applied Sciences*, 2024.

[17] Y. Lu, Z. Mi, Y. Zhou, D. Wu, and H. Wang, "Research on the 3d deployment of heterogeneous user uav base stations based on voronoi diagram division," in *2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2024, pp. 240–246.

[18] M. Song, Z. Ji, S. Huang, and J. Fu, "Mosaicking uav orthoimages using bounded voronoi diagrams and watersheds," *International Journal of Remote Sensing*, vol. 39, pp. 4960 – 4979, 2018.

[19] G. Bansal, . Naren, V. Chamola, and B. Sikdar, "Shots: Scalable secure authentication-attestation protocol using optimal trajectory in uav swarms," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 5827–5836, 2022.

[20] P. Mahajan, B. Palanisamy, A. Kumar, I. G. S. S. C. S. Member, I. V. C. S. Member, and I. M. K. S. Member, "Multi-objective mdp-based routing in uav networks for search-based operations," *IEEE Transactions on Vehicular Technology*, vol. 73, pp. 13 777–13 789, 2024.

[21] G. N. Leela, U. Varun, M. Uma, P. Srinath, and A. S. S. Sharan, "Deep learning based object tracking and detection for autonomous drones using yolov3," in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 2024, pp. 323–328.

[22] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed. Chichester, UK: Wiley, 2000.

[23] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.