Autonomous Drone for Dynamic Smoke Plume Tracking*

Srijan Kumar Pal, Shashank Sharma, Nikil Krishnakumar and Jiarong Hong

***Abstract*— This paper presents a novel autonomous drone-based smoke plume tracking system capable of navigating and tracking plumes in highly unsteady atmospheric conditions. The system integrates advanced hardware and software, along with a comprehensive simulation environment, to ensure robust performance in controlled and real-world settings. Equipped with a quadrotor platform, high-resolution imaging systems, and an advanced onboard computing unit, the drone performs precise maneuvers while accurately detecting and tracking dynamic smoke plumes under fluctuating conditions. Our software implements a two-phase flight operation: descending into the smoke plume upon detection and continuously monitoring the smoke's movement during in-plume tracking. Leveraging PID control and a Proximal Policy Optimization Deep Reinforcement Learning (DRL) controller enables adaptation to plume dynamics. Simulations using Unreal Engine evaluate performance under various smoke-wind scenarios, from steady flow to complex, unsteady fluctuations, showing that while the PID controller performs adequately in simpler scenarios, the DRL-based controller excels in more challenging environments. Field tests corroborate these findings. This system opens new possibilities for drone-based monitoring in areas like wildfire management and air quality assessment. The successful integration of DRL for real-time decision-making advances autonomous drone control for dynamic environments.**

I. Introduction

The atmospheric transport of particulate matter (PM) is an interdisciplinary field with profound implications for environmental science, climate modeling, and public health [1-3]. Examples of such transport include the dispersion of smoke plumes from forest fires, the distribution of volcanic ash during eruptions, and the movements of sand, dust, or snow migration by wind [4-7]. Understanding the dynamics of these particles is vital for predicting their environmental and health impacts, including effects on climate change, ecosystem dynamics, and respiratory health issues.

These particle transports usually span a wide range of scales, from the kilometer-scale movement of flows to the micrometer-scale size of particles [8]. Particle morphology and composition significantly influence their dispersion, yet existing field data and measurement tools are insufficient for accurately tracking these properties, which is critical for effective modeling [9]. Current techniques such as lidar and satellite imaging effectively capture large-scale particle movements but lack the resolution to provide detailed particle characteristics [8, 10], whereas in situ PM sensors, which estimate particle size distribution based on light scattering or aerodynamic properties, often rely on various assumptions, leading to uncertainties, especially for irregular particles like volcanic ash [11-15].

In response to these challenges, Bristow et al. introduced an innovative autonomous drone system equipped with a Digital inline holography (DIH) sensor for mapping particle distribution within a smoke plume [16]. In their approach, the drone initially flies above the plume, capturing top-down images using a machine vision camera. These images are analyzed in real-time using optical flow techniques to extract plume flow information, which is then used to guide the drone's navigation within the plume. Their system successfully navigated through smoke and monitored changes in particle properties during controlled experiments. However, a significant limitation of their method was its exclusive reliance on top-down imaging for flow analysis, lacking intelligent navigation once the drone entered the plume. This absence of feedback control within the smoke hindered the drone's ability to adapt to shifts in wind direction, leading to inconsistent tracking in real-world scenarios characterized by rapidly changing wind patterns and turbulent environments. Therefore, here we aim to overcome these limitations by developing an advanced computer vision-based control system. This system enables the drone to dynamically adjust its trajectory and effectively respond to directional changes in the plume caused by shifting wind conditions, thereby enhancing its performance in real atmospheric environments.

To date, there appears to be a paucity of research specifically focusing on the use of drones for tracking atmospheric particle transport like smoke plume dispersion. Relevant studies in this field have primarily concentrated on employing drones to track more predictable static or dynamic objects such as vehicles, people, or other drones [17, 18]. The methods typically include tracking the motion of the target object within the camera frame [18-21] or actively following the target using the drone. In these scenarios, object detection is achieved using traditional image processing [19, 22] or deep learning approaches [23-25]. The subsequent tracking maneuvers are then executed using non-linear Proportional Integral–Derivative (PID) controllers [18] often coupled with Kalman filtering [20] to address uncertainties.

However, these approaches are primarily suited for well-defined and predictable objects, and they fall short when applied to the complex nature of atmospheric flows like smoke. Atmospheric particle transport, such as smoke plumes

*Research Funding Support from National Science Foundation Grant No: NSF-MRI-2018658.

Srijan Kumar Pal, Minnesota Robotics Institute, Minneapolis, MN 55455 USA and St. Anthony Falls Laboratory, Minneapolis, MN 55414 USA (phone: 763-923-4386; e-mail: pal00036@umn.edu).

Shashank Sharma, Minnesota Robotics Institute, Minneapolis, MN 55455 USA and St. Anthony Falls Laboratory, Minneapolis, MN 55414 USA (phone: 763-327-0419; e-mail: sharm964@umn.edu).

Nikil Krishnakumar, Minnesota Robotics Institute, Minneapolis, MN 55455 USA and St. Anthony Falls Laboratory, Minneapolis, MN 55414 USA (phone: 763-485-3376; e-mail: krish375@umn.edu).

Jiarong Hong, Mechanical Engineering and Minnesota Robotics Institute, Minneapolis, MN 55455, and St. Anthony Falls Laboratory, Minneapolis, MN 55414 USA (phone: 612-626-4562; e-mail: jhong@umn.edu).

Code: https://github.umn.edu/HongFlowFieldImagingLab/Autonomous-Drone-for-Dynamic-Smoke-Plume-Tracking.git

or dust clouds, is fluid and dynamic, differing significantly from the more predictable objects typically tracked by drones [17, 18], [22]. This fluidity requires algorithms that can adapt to continuously changing shapes, densities, and movements. Additionally, the environmental conditions where these flows occur, such as varying wind speed, direction, and turbulence, add complexity that current tracking systems, optimized for controlled environments [18, 26] struggle to handle. The current detection algorithms used on drones often produce bounding boxes that deviate from the actual centroid of the dynamic plume, leading to inaccurate tracking. Moreover, higher inference times cause delays in drone responses, resulting in slow or non-reactive behavior during rapid shifts in smoke movement. These challenges underscore the need for real-time adaptive tracking solutions capable of managing the unpredictability of atmospheric flows.

Recently, advanced deep reinforcement learning (DRL)-based drone navigation has been explored to enhance adaptability and robustness in dynamic and unpredictable environments [27]. These methods include vision and depth-based localization and navigation, which are primarily applied to object avoidance, tracking, and drone racing applications [26, 28-30]. Despite the potential of DRL-based techniques, there has been no prior research focused specifically on using these methods to track and follow atmospheric flows, such as smoke plumes. Adapting DRL-based drone navigation to atmospheric flow tracking presents unique challenges, particularly because existing methods designed for object tracking do not adequately address the complexities of atmospheric particle transport.

To address these gaps, our study proposes a novel approach to integrating active deep learning, computer vision, and advanced control strategies when the drone gets inside the smoke plume. This enables the drone to adjust its trajectory within the plume, targeting more concentrated areas despite dynamic changes in wind conditions. This approach seeks to enhance the robustness of autonomous drone systems in tracking realistic and constantly deforming atmospheric flows in real-time, based on their changing characteristics and the surrounding environmental conditions. By integrating real-time environmental data into tracking, our system seeks to achieve a level of adaptability and precision that is currently lacking in existing drone-based tracking technologies.

The structure of this paper is as follows: Section II describes the proposed drone system in detail. In Section III and IV, we demonstrate the effectiveness of our approach through simulation and real-world field deployments. Finally, we summarize our findings and discuss their implications.

II. Methodology

A. Overview

Our autonomous drone-based smoke tracking system operates on a quadrotor platform equipped with a machine vision camera and an edge computing device for real-time processing, supporting tasks such as YOLO-based [31] smoke detection and control algorithms using PID/DRL controllers. The system operates in two key phases: the descending phase and the in-plume tracking phase. In the descending phase, the drone begins by positioning itself above the smoke plume, using the YOLO-based smoke detection algorithm to identify the initial presence of smoke, and then initiates its descent. A PID controller governs the drone's trajectory, ensuring stable

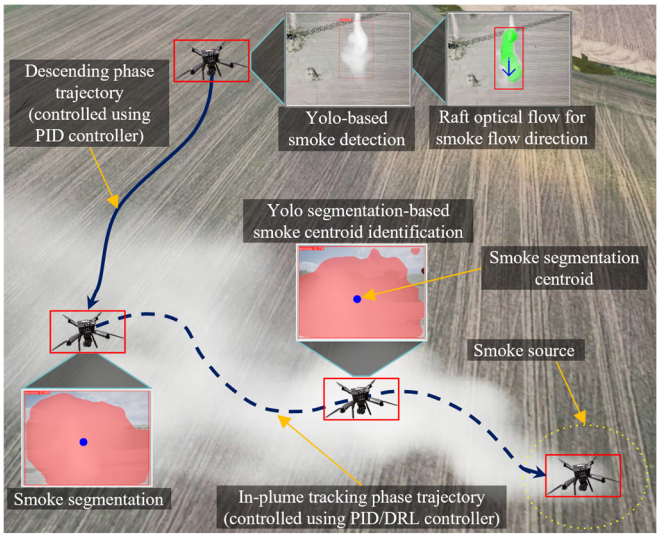

Figure 1. Autonomous drone-based smoke tracking system working principle.

movement towards the smoke plume dispersion region, while optical flow aids in aligning the drone with the direction of smoke flow. Upon reaching the plume, the system transitions into the in-plume tracking phase, triggered by detecting the smoke segmentation centroid. During this phase, the camera reorients to maintain a continuous view of the smoke, using YOLO segmentation to track the smoke's centroid in the camera's field of view. The drone's trajectory is continuously adjusted by a combination of a PID controller and a DRL controller based on the Proximal Policy Optimization (PPO) algorithm, ensuring that the drone remains aligned with the smoke centroid even as wind conditions shift and the plume changes direction. This allows the drone to continuously track and follow the densest regions of the smoke plume under dynamic environmental conditions.

B. Hardware

The drone hardware consists of a quadrotor body, high-resolution imaging systems, and an onboard computing unit, as illustrated in Figure 2. The system builds on our previous design detailed in [16], incorporating upgrades to the imaging system and the onboard edge computing device to enhance real-time autonomy. Specifically, the core computational unit has been upgraded to the Nvidia Jetson Orin Nano, delivering up to 40 TOPS—nearly double the performance of the previous Jetson Xavier NX. It operates on Linux (Jetpack 5.1.3, Ubuntu 20.04 LTS, Linux Kernel 5.10) with ROS Noetic for efficient communication, boots from an NVMe SSD, and utilizes TensorRT to accelerate deep learning inference. To handle the high memory demands of DRL tasks,

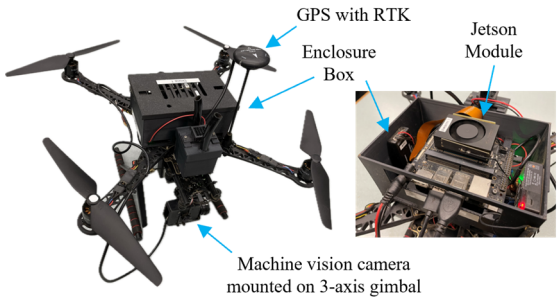

Figure 2. Autonomous drone-based smoke tracking system hardware.

an additional 8GB of swap space has been added to the existing 8GB of RAM. The flight controller has been upgraded to the Pixhawk 6C (from Pixhawk 4), which interfaces with the Jetson through MAVROS for MAVLink communication and integrates RTK technology with GPS to achieve centimeter-level positioning accuracy. To minimize latency and maintain high resolution with minimal distortion, the machine vision system has been enhanced with a 12-megapixel ArduCam, replacing the previous GoPro camera. Mounted on a 3-axis gimbal for both top-down and in-plume views, the camera operates at a reduced sensor size of 640 x 480 pixels at 30 frames per second, which reduces unnecessary computational load and improves response times. Together, these upgrades significantly enhance real-time image processing and autonomous control, improving the drone's multi-phase and multi-modal smoke tracking capabilities.

C. Algorithm and Software Architecture

As depicted in Figure 3, the framework of the autonomous drone operation algorithm is divided into two main phases: the descending phase and the in-plume tracking phase, each comprising several steps. Specifically, the descending phase consists of the following steps:

1) Hovering and Smoke Detection Setup: The drone hovers above the smoke plume in forward scan mode, with the state set to 'GUIDED' to enable autonomous control commands from the Jetson module. The gimbal is pitched downward to capture a top-down view.

2) Smoke Detection: A detection node processes the top-down images using a custom-trained YOLOv8m model to detect the smoke plume, outputting bounding boxes with coordinates and dimensions.

3) Optical Flow Analysis: Once smoke is detected, the drone halts all movement, and RAFT Optical Flow [32] analysis begins. The optical flow node processes the sequential bounding boxes to compute the directional vector of the smoke flow until the mean direction is established.

4) Yaw Alignment and Descent: The drone yaws to align with against the smoke flow and initiates its descent. A PID controller maintains the smoke bounding box in the upper half of the frame, ensuring the descent stays within the smoke dispersion area. The descent continues until the drone is fully immersed in the smoke.

Once the drone reaches the target altitude, it switches to the in-plume tracking phase, adjusting the gimbal for a forward view from within the plume toward the smoke source. The in-plume tracking phase proceeds using the following steps:

1) Smoke Segmentation: At this stage, the detection node is disabled, and a smoke segmentation node takes over. The node uses a YOLOv8m-seg model, trained to identify and segment denser regions within the smoke plume. It validates segments that exceed a set threshold and calculates the centroid of each valid segment by averaging the x and y coordinates, representing the densest smoke region. This data, including segmentation area, centroid location, and mask, is used as inputs for drone trajectory control.

2) Trajectory Control: The drone's movement is managed by either a PID or DRL controller. The PID controller calculates the positional error between the smoke centroid and the center of the camera frame, generating velocity commands to adjust the drone's position along the horizontal and vertical axes for smooth tracking. In contrast, the DRL controller processes the

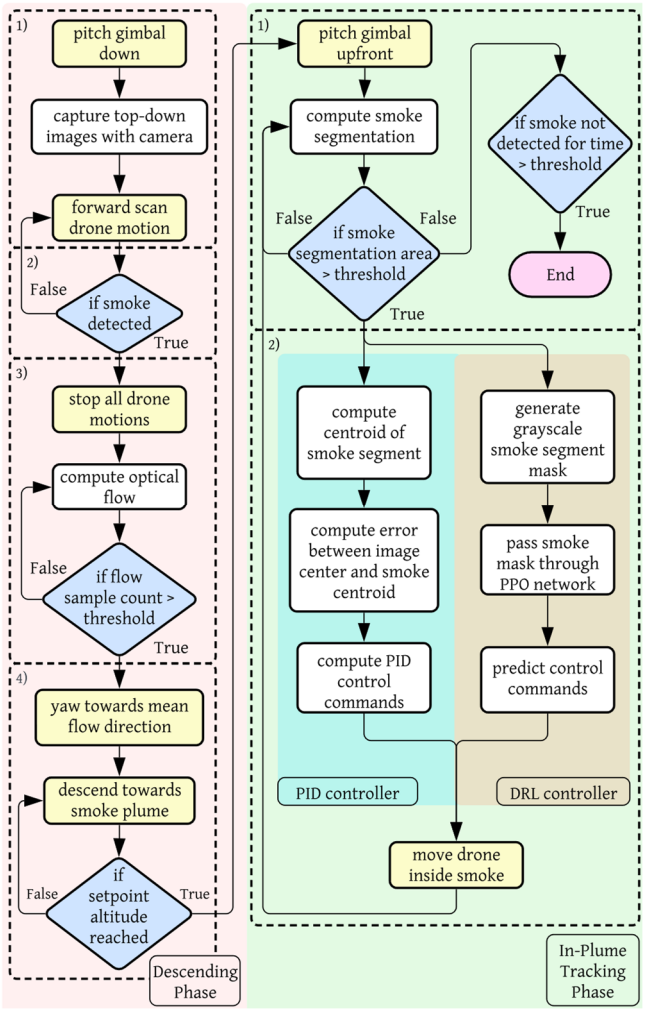

Figure 3. The framework of the autonomous drone operation algorithm.

binary smoke segmentation mask through an actor-critic policy network, selecting actions from a discrete set to maneuver the drone in a manner similar to the PID controller. While the PID controller provides faster, more reliable responses when well-tuned, the DRL controller excels in dynamic or unpredictable environments. Currently, the system allows manual switching between these controllers via operator commands, but future versions will incorporate automatic switching based on real-time smoke conditions to optimize performance. Further details on DRL customization are discussed in the following section.

D. DRL Drone Control Algorithm

Our study employs Proximal Policy Optimization (PPO), a widely used DRL algorithm implemented using the Stable Baselines3 library and OpenAI Gym environments, to address the complex task of smoke tracking. PPO is known for its stability, robustness in high-dimensional and stochastic environments, making it suitable for the unpredictable nature of smoke plume dynamics. It operates using an actor-critic framework, a type of policy gradient method, where the actor proposes action probabilities, and the critic evaluates their expected value. To ensure stable learning, PPO employs a clipped objective function that prevents large, destabilizing policy updates, crucial for controlling drones in dynamic environments. PPO has been widely applied in various drone

control tasks, such as autonomous maneuvering [33], drone tracking [34], and path planning [35]. However, adapting this approach to the unique challenges of smoke plume tracking required several adaptations, as detailed below.

1) Custom CNN Architecture: The core of our DRL system is a custom Convolutional Neural Network (CNN) integrated into the PPO framework to handle the unique visual environment of smoke plumes. The CNN handles single-channel binary segmentation masks of size 320 x 320 pixels and consists of an input layer for single-channel binary images, three convolutional layers with increasing filter depths (32, 64, 128) using strides to reduce spatial dimensions, ReLU activation functions for non-linearity, a flattening layer to convert 2D feature maps into a 1D feature vector, and a fully connected layer that outputs a fixed-size feature vector to the policy and value networks. This architecture is optimized for efficient feature extraction and low latency, enabling real-time operation on the resource-constrained Jetson platform.

2) Discrete Action Space: The DRL predicts within a discrete action space with seven movement options: no movement, up, down, left, right, hard left, and hard right. Each action corresponds to predefined velocity adjustments along the drone's horizontal and vertical axes, allowing precise trajectory control. The range of actions, from subtle adjustments (left, right) to more aggressive maneuvers (hard left, hard right), offers the flexibility needed to navigate effectively inside smoke.

3) Custom reward function: The custom reward function incentivizes the drone to track denser smoke regions by rewarding movements that align with the smoke segmentation centroid's position in the image frame. For example, if the centroid is on the left and the DRL controller predicts a left movement, a positive reward is given. If the centroid is far left, a "hard left" action earns a full reward, while a simple "left" receives a partial reward. Incorrect or suboptimal movements result in negative or reduced rewards. Additionally, the drone is rewarded for maintaining its position when the centroid is centered, promoting stability during steady plume conditions. This reward shaping encourages exploration while reinforcing behaviors that lead to successful tracking.

4) Training Configuration: The PPO-based DRL architecture was trained with specific hyperparameters to enhance learning efficiency and adaptability. We used a learning rate of 3×10^{-4}, a discount factor γ =0.99, 2,048 steps per update, a batch size of 256, and 10 epochs per update, training the model over 1 million timesteps. The smoke scenarios in Unreal Engine alternated between steady linear flows and fluctuating directions with varying oscillation frequencies, providing comprehensive training data that mimics real-world smoke conditions.

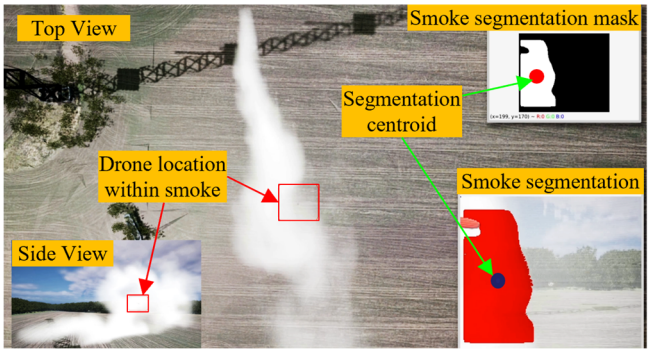

Figure 4. Simulation of the deployment of autonomous drone-based smoke tracking system at Eolos Field Station, Rosemount, Minnesota, USA.

5) Inference and Simulation-to-real-world (Sim2Real) Transfer: During inference, the trained model applies the learned policy to control the drone's movements in both simulated and real-world smoke environments. The segmentation model, trained on both real and simulated smoke, generates binary smoke masks, which the DRL controller processes to predict actions. These actions are then mapped directly to drone control commands. Since the segmentation model handles the differences between simulated and real smoke, the DRL controller functions seamlessly, ensuring smooth policy transfer and effective navigation across both environments.

III. Simulation Assessment

A. *Simulation Environment*

Developing an autonomous drone system for smoke plume tracking presents several challenges, including the need for large, open testing areas, unpredictable drone behavior that can lead to costly crashes, and weather conditions like wind and rain that limit testing opportunities. To overcome these challenges, we developed a simulation environment using Unreal Engine 5.1.1 to enable rapid algorithm testing and refinement under controlled and realistic smoke-wind scenarios before field deployment. The simulation replicates real-world conditions at our primary field-testing site, the Eolos Field Station in Rosemount, Minnesota, USA. This includes simulating terrains, vegetation, and infrastructure to create an environment that closely mirrors the actual testing site. For smoke generation, we utilize Unreal Engine's Niagara Plugin, which includes Niagara Fluids, Chaos Niagara, and the Niagara Custom Data Interface, to produce realistic smoke dynamics. The simulation environment allows full control over wind speed and direction through a custom blueprint in the event graph, which calculates time-dependent wind vectors. This setup supports both variable and constant wind flows, allowing flexibility in test conditions. The drone's flight control and machine vision are simulated using the PX4 Software-in-the-Loop (SITL) integrated with AirSim and running on Windows Subsystem for Linux (WSL2). AirSim captures sensor data and camera images, which are published to ROS topics for analysis. The WSL2 virtual Linux environment emulates Jetson hardware, enabling real-time computer vision and deep learning computations. The ROS framework in WSL2, combined with MAVROS, ensures seamless MAVLink communication with PX4 SITL for autonomous drone control within the simulation. This integration allows accurate simulation of drone flight and its interaction with the smoke plume. Overall, this simulation environment provides a robust platform for algorithm testing, from drone flight to smoke detection and segmentation, under a wide range of realistic smoke-wind scenarios. The flexibility and reproducibility of the Unreal Engine smoke simulation allow us to thoroughly evaluate the drone's performance in diverse test conditions before field deployment.

B. *Performance Assessment using Simulation*

The smoke tracking performance of the drone, using both PID and DRL-based control algorithms, was evaluated under different smoke generation and wind conditions in the

simulation. Four test scenarios, with increasing levels of tracking difficulty, were designed to assess the algorithms:

1) Steady Smoke Flow (S): The smoke plume is steady under constant streamwise wind (V_y) of 4.5 m/s with no fluctuations in direction or speed. This represents a stable and predictable condition, ideal for baseline performance evaluation.

2) Unsteady Smoke Flow with Low-Frequency Horizontal Fluctuation (UL): The smoke experiences a mild low-frequency fluctuating crosswind which is superimposed on top of the primary wind of $V_y = 4.5$ m/s. The crosswind is specified as $V_x = 1.35\sin(0.02\pi t)$ m/s with an amplitude of 1.35 m/s and a frequency of 0.01 Hz.

3) Unsteady Smoke Flow with High-Frequency Horizontal Fluctuation (UH): The smoke experiences a stronger high-frequency crosswind superimposed on top of the primary wind of $V_y = 4.5$ m/s. The crosswind is specified as $V_x = 1.95\sin(0.04\pi t)$ m/s with an amplitude of 1.95 m/s and 0.02 Hz frequency. Under such conditions, the smoke changes direction more rapidly, increasing the tracking challenge.

4) Unsteady Smoke Flow with 3D Fluctuation (U3D): Both horizontal and vertical wind fluctuations are introduced on top of the primary wind of $V_y = 4.5$ m/s to further increase the tracking challenges. The horizontal crosswind is specified as $V_x = 1.95\sin(0.04\pi t)$ (amplitude 1.95 m/s, frequency 0.02 Hz), and vertical wind $V_z = 0.3\sin(0.02\pi t)$ (amplitude 0.3 m/s, frequency 0.01 Hz). Under such conditions, the plume fluctuates in a complex 3D pattern.

To accurately assess the drone's smoke tracking performance, we positioned an observer drone equipped with a camera at a fixed altitude above the smoke plume, providing a top-down view of the tracking drone as shown in Figure 5a. As the tracking drone enters the smoke plume, visual methods like bounding boxes become ineffective. Instead, we mapped the tracker drone's 3D GPS coordinates to 2D-pixel coordinates within the observer drone's camera image. By using the observer drone's GPS location, camera focal length, and applying the haversine formula, we computed the image's dimensions in the world frame and used an affine transformation matrix to map each pixel to GPS coordinates. The inverse transformation allowed us to accurately track the drone's location in the camera image without relying solely on image-based techniques.

As shown in Figure 5b, custom metrics are introduced to compare PID and DRL controllers based on the tracking drone's position relative to the smoke plume contour to evaluate how close the drone stays within the plume center during the entire tracking duration. The plume contour is generated through thresholding and extracting the largest contour, with the red circle representing the drone's location and the green spline depicting the smoke plume's mean line (skeleton). In total the following five metrics are used for performance evaluation including normalized average distance of the drone from the mean line $\tilde{\mu}_m = mean(d_m) / L_{ref}$, normalized maximum distance of the drone from the mean line $\tilde{d}_{m,max} = (d_m) / L_{ref}$, normalized average distance when outside the smoke plume $\tilde{\mu}_c = mean(d_c) / L_{ref}$, normalized maximum distance when outside the smoke plume $\tilde{d}_{c,max} = (d_c) / L_{ref}$, percentage of time inside the smoke plume $\tilde{t}_R$, where L_{ref} is the total smoke tracking length, d_m is the drone's distance from the smoke mean line, and d_c is the distance from the smoke contour.

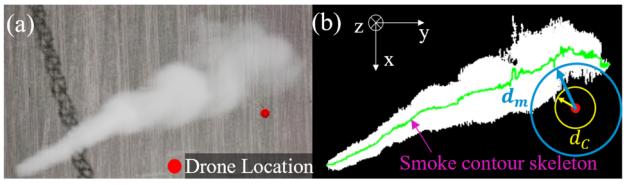

Figure 5. Illustration of (a) top-view images and (b) the metrics used to evaluate the performance of our drone-based smoke tracking system.

Table 1 summarizes the performance metrics for both PID and DRL controllers under the different test conditions. In steady smoke flow (S), both PID and DRL controllers perform similarly, effectively tracking the plume with comparable $\tilde{\mu}_m$ and $\tilde{t}_R$, indicating that both controllers can track the steady smoke plume effectively. Under low-frequency, low-amplitude unsteadiness (**UL**), the DRL controller shows a marked improvement in tracking the mean line of the plume. The $\tilde{\mu}_m$ drops significantly from 4.0% (PID) to 1.8% (DRL), highlighting its ability to maintain closer alignment with the plume. The variability remains comparable, indicating that DRL's performance gain is not due to increased instability but rather improved trajectory correction. In higher-frequency unsteady flow (**UH**) with stronger crosswinds, the DRL outperforms PID across all metrics, notably improving in $\tilde{d}_{c,max}$, where the DRL controller reduces the deviation to 12.3%, compared to 18.6% for PID and significantly extending the $\tilde{t}_R$ achieving 85% compared to 69% for PID. Finally, under 3D unsteadiness (**U3D**), the DRL controller exhibits significant improvements across all metrics showing a much closer alignment with the smoke, with $\tilde{\mu}_m$ significantly reduced compared to PID, and it achieves a greater $\tilde{t}_R$. The DRL controller demonstrates superior adaptability in this highly dynamic and complex smoke environment, maintaining better overall performance.

		$\tilde{\mu}_m$(%)	$\tilde{d}_{m,max}$(%)	$\tilde{\mu}_c$(%)	$\tilde{d}_{c,max}$(%)	$\tilde{t}_R$ (%)
S	PID	1.6±0.2	7.5±2.1	1.4±0.6	2.9±0.7	95.1±2.5
	DRL	1.4±0.2	7.0±2.3	1.8±1.2	4.0±3.3	94.1±1.4
UL	PID	4.0±0.4	11.9±2.2	2.5±1.3	6.6±3.1	87.1±2.2
	DRL	1.8±0.5	10.1±2.3	2.5±1.1	8.4±3.3	86.9±1.8
UH	PID	7.2±1.5	28.1±10.9	5.3±3.8	18.6±7.4	69.4±4.7
	DRL	5.4±1.1	26.0±8.4	4.6±3.1	12.3±6.5	85.0±6.4
U3D	PID	4.1±0.6	15.5±6.5	4.5±2.6	12.8±5.9	79.5±2.3
	DRL	1.9±0.5	10.6±4.4	1.5±2.1	4.3±4.3	95.0±4.1

Table 1. Performance Metrics evaluated using the simulation and mean and standard deviation are calculated based on five tests under each condition for both PID and DRL control separately.

IV. Field Demonstration

Field testing was conducted at the Eolos Field Station at Rosemount, Minnesota, USA, an open agricultural field ideal for controlled experiments. Non-harmful smoke plumes were generated using a high-density smoke generator that combined food-grade chemicals (glycerin, propylene glycol, and artificial smoke fluids) in proportions, producing the plume shown in Figure 6. Deployments were performed under S-SE winds, with speeds of 4.9-6.7 m/s (11–15 mph) and gusts up to 8.8m/s, and temperatures around 25°C, ensuring optimal visibility and safe drone operations. Before each experiment, all scripts for smoke tracking were initialized,

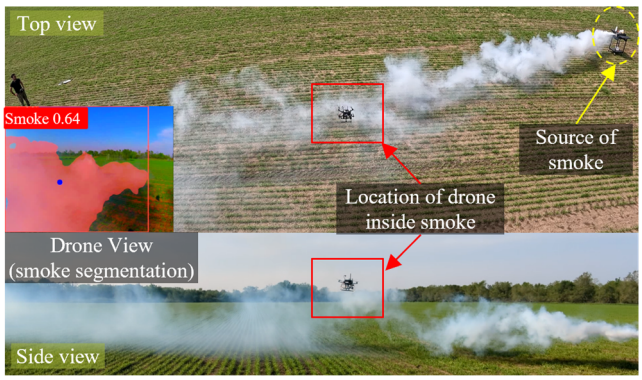

Figure 6. Field demonstration of autonomous drone-based smoke tracking algorithm at Eolos Field Station, Rosemount, Minnesota, USA.

including image capture, smoke segmentation, and the generation of drone control commands (either PID- or DRL-based). The drone, set to 'GUIDED' mode, responded to autonomous control commands only when the smoke plume was established. The test began with the drone hovering near the plume's dispersing region, after which it autonomously tracked the smoke by adjusting its trajectory in response to wind direction changes. Experiments were conducted using both PID and DRL control algorithms in separate tests. In both cases, the drone consistently maintained its track as the plume shifted, dynamically adjusting its trajectory to follow the plume and eventually reach the smoke source. As shown in Table 2, the field performance metrics indicate that the DRL controller performs better than the PID controller in terms of staying inside the smoke plume for a longer period, achieving a $\tilde{t}_R$ of 72.7% compared to 71.2% for PID. However, the PID controller maintained closer proximity to the smoke plume core, with $\tilde{\mu}_m$=6.4 compared to $\tilde{\mu}_m$=8.1 for the DRL controller. The metrics $\tilde{d}_{c,max}$ and $\tilde{\mu}_c$ also suggest that PID remained slightly closer to the smoke plume than DRL. However, unlike in the simulation, it was not possible to maintain identical smoke conditions during the PID and DRL experiments in the field, meaning the results may not quantitatively mirror the trends observed in simulation.

	$\tilde{\mu}_m$(%)	$\tilde{d}_{m,max}$(%)	$\tilde{\mu}_c$(%)	$\tilde{d}_{c,max}$(%)	$\tilde{t}_R$(%)
PID	6.4	21.0	3.6	10.0	71.2
DRL	8.1	27.8	4.1	12.2	72.7

Table 2. Performance Metrics evaluated in field deployment

V. Conclusion And Discussion

This paper presents a novel autonomous drone-based smoke plume tracking system designed to effectively navigate and track plumes even in highly unsteady and challenging atmospheric conditions. Our system integrates both hardware and software advancements, along with a comprehensive simulation environment to ensure robust performance in both controlled and real-world settings. From a hardware perspective, the drone is equipped with a quadrotor platform featuring high-resolution imaging systems and an advanced onboard computing unit. Incorporating the Jetson Orin Nano with machine vision system for real-time processing and the Pixhawk flight controller enables the drone to perform precise maneuvers while tracking dynamic smoke plumes. Our software architecture implements a two-phase flight operation: the descending phase, where the drone detects and descends into the smoke plume, and the in-plume tracking phase, where it continuously monitors and follows the smoke's movement. This process leverages a combination of PID control and a DRL controller, enabling the system to adapt to changes in plume dynamics and maintain effective tracking. We developed a detailed simulation environment using Unreal Engine to evaluate the drone's performance under various smoke-wind scenarios. The simulation was designed to replicate real-world conditions, ranging from steady smoke flow to complex, 3D unsteady smoke fluctuations. Four test cases were used to assess the system's capabilities, and the results demonstrated that while the PID controller performed adequately in simpler scenarios, the DRL-based controller excelled in more challenging environments with high-frequency fluctuations. Field tests corroborated the effectiveness of our approach, yielding results consistent with those observed in the simulation.

Our work extends our previous study [16] by advancing the ability to autonomously track smoke plumes in realistic, unsteady atmospheric environments. By integrating deep learning for smoke detection and segmentation with DRL-based control, we provide a significant improvement in tracking deformable and unpredictable targets, unlike previous drone tracking systems that focused on static or predictable objects. This system has broad potential applications beyond smoke tracking. It opens new possibilities for drone-based monitoring and analysis in areas such as wildfire monitoring, air quality assessment, environmental hazard tracking, and particle transport studies. The system's ability to track dynamic, irregular plumes could be particularly useful for emergency response operations and studying atmospheric phenomena like fog, pollution clouds, and volcanic ash plumes. Additionally, the successful integration of DRL for real-time decision-making in complex environments represents a major step forward in autonomous drone control and could be applied to other scenarios involving fluid dynamics or airborne particle tracking.

Despite promising results in both simulations and field tests, several limitations remain. The smoke detection and segmentation model, trained on specific datasets, may not generalize well to all types of smoke or atmospheric conditions, especially in extreme environments. Furthermore, field testing has so far been conducted in relatively controlled settings, and more rigorous testing in harsher and more unpredictable conditions will be necessary to validate the system's robustness. Future work will focus on improving the generalization of the smoke detection and segmentation models, enhancing the DRL controller to handle more extreme environmental changes, and expanding the system's capabilities for fully autonomous operation with minimal manual intervention. We will also explore extending this approach to other atmospheric and environmental applications, such as tracking fog, pollution, or ash plumes.

Acknowledgment

This work was supported by the NSF Major Research Instrumentation program (NSF-MRI-2018658), which provided crucial funding for the development and deployment of the autonomous drone system for smoke plume tracking. We would also like to extend our gratitude to Mayan Iyer, Sujeendra Ramesh, Sri Ganesh Lalit Aditya Divakarla, and Rammesh A. Saravanan for their invaluable assistance during the initial phase of the work. We also acknowledge the use of ChatGPT for assisting in refining content in this manuscript.

REFERENCES

[1] Kumar, P., Ketzel, M., Vardoulakis, S., Pirjola, L., & Britter, R. (2011). Dynamics and dispersion modeling of nanoparticles from road traffic in the urban atmospheric environment—A review. Journal of Aerosol Science, 42(9), 580-603.

[2] Kok, J. F., Parteli, E. J. R., Michaels, T. I., & Bou Karam, D. (2012). The physics of wind-blown sand and dust. Reports on Progress in Physics, 75(10), 106901.

[3] Evangeliou, N., Grythe, H., Klimont, Z., Heyes, C., Eckhardt, S., & Stohl, A. (2020). Atmospheric transport is a major pathway of microplastics to remote regions. Nature Communications, 11(1), 1-11.

[4] Mott, R., Schirmer, M., Bavay, M., Grünewald, T., & Lehning, M. (2010). Understanding snow-transport processes shaping the mountain snow cover. The Cryosphere, 4(4), 545-559.

[5] Jaffe, D. A., O'Neill, S. M., Larkin, N. K., Holder, A. L., Peterson, D. L., Halofsky, J. E., & Rappold, A. G. (2020). Wildfire and prescribed burning impacts on air quality in the United States. Journal of the Air & Waste Management Association, 70(6), 583-615.

[6] Butwin, M. K., von Löwis, S., Pfeffer, M. A., & Thorsteinsson, T. (2019). The effects of volcanic eruptions on the frequency of particulate matter suspension events in Iceland. Journal of Aerosol Science, 128, 99-113.

[7] Dentoni, V., Grosso, B., Pinna, F., Lai, A., & Bouarour, O. (2022). Emission of fine dust from open storage of industrial materials exposed to wind erosion. Atmosphere, 13(2), 320.

[8] Sokolik, I. N., Soja, A. J., DeMott, P. J., & Winker, D. (2019). Progress and challenges in quantifying wildfire smoke emissions, their properties, transport, and atmospheric impacts. Journal of Geophysical Research: Atmospheres, 124(23), 13005-13025.

[9] Lähde, A., Gudmundsdottir, S. S., Joutsensaari, J., Tapper, U., Ruusunen, J., Ihalainen, M., & Gizurarson, S. (2013). In vitro evaluation of pulmonary deposition of airborne volcanic ash. Atmospheric Environment, 70, 18-27.

[10] Wandinger, U. (2005). Introduction to lidar. In C. Weitkamp (Ed.), Lidar (Vol. 102, pp. 1-18). Springer.

[11] Johnson, T. J., Irwin, M., Symonds, J. P. R., Olfert, J. S., & Boies, A. M. (2018). Measuring aerosol size distributions with the aerodynamic aerosol classifier. Aerosol Science and Technology, 52(6), 655-665.

[12] Hagan, D. H., & Kroll, J. H. (2020). Assessing the accuracy of low-cost optical particle sensors using a physics-based approach. Atmospheric Measurement Techniques, 13, 6343-6355.

[13] Madokoro, H., Kiguchi, O., Nagayoshi, T., Chiba, T., Inoue, M., Chiyonobu, S., Nix, S., Woo, H., & Sato, K. (2021). Development of drone-mounted multiple sensing systems with advanced mobility for in situ atmospheric measurement: A case study focusing on PM2.5 local distribution. Sensors, 21(14), 4881.

[14] Kristiansen, N. I., Prata, A. J., Stohl, A., & Carn, S. A. (2015). Stratospheric volcanic ash emissions from the 13 February 2014 Kelut eruption. Geophysical Research Letters, 42, 588-596.

[15] Grimm, H., & Eatough, D. J. (2009). Aerosol measurement: The use of optical light scattering for the determination of particulate size distribution, and particulate mass, including the semi-volatile fraction. Journal of the Air & Waste Management Association, 59(1), 101-107.

[16] Bristow, N. R., Pardoe, N., & Hong, J. (2023). Atmospheric aerosol diagnostics with UAV-based holographic imaging and computer vision. IEEE Robotics and Automation Letters, 8(9), 5616-5623.

[17] Barták, R., & Vykovský, A. (2015). Any object tracking and following by a flying drone. In 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI) (pp. 35-41). IEEE.

[18] Maalouf, A., et al. (2024). Follow anything: Open-set detection, tracking, and following in real-time. IEEE Robotics and Automation Letters, 9(4), 3283-3290.

[19] Teuliere, C., Eck, L., & Marchand, E. (2011). Chasing a moving target from a flying UAV. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, USA.

[20] Barisic, A., Car, M., & Bogdan, S. (2019). Vision-based system for real-time detection and following of UAV. In 2019 Workshop on Research, Education, and Development of Unmanned Aerial Systems (RED UAS) (pp. 156-159). Cranfield, UK.

[21] Hansen J. G., de Figueiredo R. P. (2024). Active Object Detection and Tracking Using Gimbal Mechanisms for Autonomous Drone Applications. Drones, 8(2), 55.

[22] Cesetti, A., Frontoni, E., Mancini, A., Zingaretti, P., & Longhi, S. (2009). Vision-based autonomous navigation and landing of an unmanned aerial vehicle using natural landmarks. In Mediterranean Conference on Control and Automation (pp. 910-915).

[23] Kanellakis, C., & Nikolakopoulos, G. (2017). Survey on computer vision for UAVs: Current developments and trends. Journal of Intelligent & Robotic Systems, 87(1-2), 141-168.

[24] Ramachandran, A., & Sangaiah, A. K. (2021). A review on object detection in unmanned aerial vehicle surveillance. International Journal of Cognitive Computing in Engineering, 2, 215-228.

[25] Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning-based object detection models. Digital Signal Processing, 126, 103514.

[26] Kaufmann, E., Bauersfeld, L., Loquercio, A., & Scaramuzza, D. (2023). Champion-level drone racing using deep reinforcement learning. Nature, 620, 982-987.

[27] Aburaya, Anas & Selamat, Hazlina & Muslim, Mohd. (2024). Review of vision-based reinforcement learning for drone navigation. International Journal of Intelligent Robotics and Applications. 1-19.

[28] Ma, B., et al. (2023). Deep reinforcement learning of UAV tracking control under wind disturbances environments. IEEE Transactions on Instrumentation and Measurement, 72, 1-13.

[29] Ma, B., et al. (2023). Target tracking control of UAV through deep reinforcement learning. IEEE Transactions on Intelligent Transportation Systems, 24 (6), 5983-6000.

[30] Zhou, B., Wang, W., Liu, Z., & Wang, J. (2019). Vision-based Navigation of UAV with Continuous Action Space Using Deep Reinforcement Learning. In 2019 Chinese Control And Decision Conference (CCDC) (pp. 5030-5035).

[31] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics

[32] Teed, Z., & Deng, J. (2020). RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), Computer Vision – ECCV 2020 - 16th European Conference, 2020, Proceedings (pp. 402-419). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 12347 LNCS). Springer Science and Business Media Deutschland GmbH.

[33] Cano Lopes, G., Ferreira, M., da Silva Simões, A., & Luna Colombini, E. (2018). Intelligent Control of a Quadrotor with Proximal Policy Optimization Reinforcement Learning. In 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE) (pp. 503-508). João Pessoa, Brazil: IEEE.

[34] Tan, Z., & Karaköse, M. (2023). A new approach for drone tracking with drone using Proximal Policy Optimization based distributed deep reinforcement learning. SoftwareX, 23, 101497.

[35] Qi, C., Wu, C., Lei, L., Li, X., & Cong, P. (2022). UAV path planning based on the improved PPO algorithm. In 2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE) (pp. 193-199). Qingdao, China