JOURNAL OF
EMERGING INVESTIGATORS

Article

AeroPurify: Autonomous air filtration UAV using real-time 3-D Monte Carlo gradient search

Varini Kadakia[1], Kuan-Chuen Wu[2]
[1] Archbishop Mitty High School, San Jose, California
[2] ANI.ML Health Research, Toronto, Ontario, Canada

SUMMARY

Outdoor air pollution is the fourth-leading cause of global mortality, claiming 4.2 million lives annually and impacting 99% of the world. Accessible and efficient solutions for pollution mitigation are lacking, despite the widespread impacts of outdoor air pollution. We present an autonomous drone air filtration system prototype, designed to detect and mitigate outdoor air pollution through utilizing a novel autonomous navigation algorithm and a custom-built data processing and transmission system (DPTS). The DPTS detects and transmits real-time particulate matter data to the navigation algorithm and has a Fibonacci Spiral-based filtration attachment that allows for air purification. Currently, one method for path finding is called gradient ascent (GA); however, this algorithm, when simulated, was time-consuming, visiting extraneous waypoints. In this paper, we propose an alternative to the GA algorithm called the gradient ascent ML particle filter (GA/MLPF) algorithm, which assists the drone in its autonomous traversal of the pollution field to find the source of the pollution. Based on the Bayesian state estimation particle filter, we hypothesized that the GA/MLPF algorithm would outperform the conventional GA algorithm by creating a time-efficient path and reducing the number of waypoints. Our results showed that the GA/MLPF algorithm did outperform the conventional GA algorithm: it took an average of 70% less time and reduced the number of waypoints by at least 28%. The GA/MLPF algorithm developed in this project is an innovative approach to tackling outdoor air pollution, and the algorithm's mobility and effectiveness allow it to be used in many diverse environments.

INTRODUCTION

Particulate matter (PM) consisting of harmful, microscopic particles emitted from vehicle exhaust, industrial emissions, wildfires, and other sources poses significant health risks to human populations worldwide (1). PM particles come in a wide variety of sizes, with those less than 2.5 microns classified as PM2.5 and those between 2.5-10 microns classified as PM10. Inhaling toxic PM can lead to illnesses including cancer, respiratory and cardiovascular diseases, and neurological problems (2). Since these particles (PM2.5 and PM10) are nearly invisible to the naked eye, it is difficult to avoid them, and in high concentrations or with long-term exposure, the harmful effects of these particles can become heightened, potentially causing premature death (3-6). Cleaning pollution in real-time is essential not only for safeguarding human health, but also for mitigating environmental degradation and ensuring long-term sustainability. Due to the ubiquity of PM and their detrimental effects, effective pollution control measures are imperative.

Current solutions to mitigate outdoor air pollution, such as electrostatic precipitators (ESP) and baghouses, have limitations that hinder their widespread adoption and effectiveness. ESP-based systems, while capable of collecting pollution particles through electrical currents, generate ground-level ozone which may pose safety concerns and may aggravate certain health conditions (7). Additionally, the stationary nature of such systems limits their coverage area, requiring multiple units in order to achieve a safe outdoor environment. Moreover, baghouses, which utilize fans to pass polluted air through several fabric filters, also have several limitations. They need constant maintenance, are a safety hazard due to their highly combustible nature, and are stationary, facing the same problems as ESP-based systems (8). Industrial air cleaning devices, including the baghouses and the ESPs, tend to be expensive and might not be available for consumers to purchase (9).

Therefore, there remains a critical need for an effective and mobile outdoor air purification system that can address pollution at its source. In response to this need, we developed an autonomous drone system equipped with onboard pollutant sensors, guided by a machine learning (ML) navigation model. By leveraging artificial intelligence (AI)-driven navigation and real-time pollutant detection, the system will seek to locate sources of pollution in three-dimensions and clean the air, thereby reducing the concentration of PM in the air and enabling subsequent mitigation actions.

Our primary goal was to design, build, and test an autonomous drone system that uses an accurate navigation algorithm to actively locate and filter the source of outdoor air pollution in real time rather than passively filtering the air in the vicinity of the user. Within this project, we developed a novel gradient ascent ML particle filter (GA/MLPF) algorithm, using an ML particle filter. We hypothesized that this newly developed algorithm would outperform the conventional gradient ascent (GA) algorithm by creating a time-efficient path and reducing the number of points visited. By integrating the on-board pollutant sensor with AI-driven navigation models, the system aimed to provide an accessible and mobile solution for reducing pollution at its source.

When comparing the conventional GA algorithm to our newly developed GA/MLPF algorithm, our results showed that the GA/MLPF algorithm outperformed the conventional GA algorithm, taking an average of 70% less time and

https://doi.org/10.59720/24-246

reducing the number of waypoints by at least 28%. Through testing and validation, we sought to demonstrate the feasibility and efficacy of the proposed drone-based approach in improving air quality, protecting human health, and advancing environmental sustainability. Ultimately, our work contributes to the development of scalable and accessible pollution control measures that can address the pressing challenges posed by outdoor air pollution on a global scale.

RESULTS

Data Transmission and Accuracy Test

We evaluated the data processing and transmission system (DPTS) for its ability to transmit PM2.5 concentration readings, and assed the accuracy of the Plantower PM2.5 sensor used to collect those concentration values **(Figure 1).** In order to test the DPTS's transmission ability, the distance between the Arduinos used the ground station and in the DPTS was slowly increased. Despite the growing increases in distance, the DPTS was found to be able to accurately transmit the real-time pollution throughout the whole system (from the air to the ground).

Next, we looked at the accuracy of the Plantower PM2.5 Sensor compared to a market-quality air quality meter using statistical analysis. We conducted three independent runs. During each run, both the Plantower PM2.5 sensor and the air quality meter were placed side by side and a match was lit and blown out to simulate air pollution. The measurements were videorecorded and later ten data points were extracted per run to form the data used for the statistical analysis.

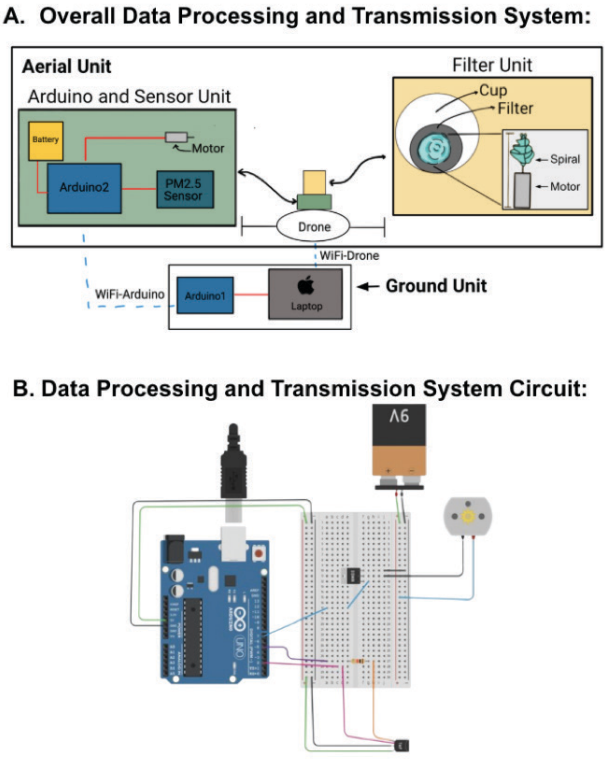

Figure 1: Data processing and transmission system layout. Setup of the data processing and transmission system (DPTS). **A)** The filter unit (yellow box) and the Arduino and sensor unit (green box). Please note the motor depicted within the green box is the same as the one used within the yellow box to power the filter unit. **B)** A more in-depth view of the circuit used within the DPTS. Specifically, it shows the interconnections between the following components of the Aerial Unit: Arduino2 board, PM2.5 sensor (black icon at bottom right), filter motor, 9V battery, and N-type metal–oxide–semiconductor (NMOS).

Based on the results of histograms and quartile-quartile plots, the data collected was found to be non-normal (**Figure 2**). Given small, non-normal dataset, we performed the Wilcoxon signed-rank test to test for a significant statistical difference between the readings from the Plantower PM2.5 Sensor and market-quality air quality meter. We found in Runs 1 and 3, the p-values (Run 1 p_value: 0.0645, Run 3 p_value: 0.0840) were greater than 0.05, suggesting there could be no statistical difference in the distribution of readings from the PM2.5 sensor and the air quality sensor. However, in Run 2, the p-value (p_value: 0.0039) was less than 0.05 which means that the two sensors' distributions are significantly different. Overall, the results of the Wilcoxon signed-rank test suggest variability in conclusions across different runs.

Comparing the GA and GA/MLPF Algorithms

The GA algorithm starts at the drone's current point and traverses in all three directions (+*X,* +*Y,* +*Z*). At each of these new points, the GA algorithm will take a concentration reading and determine how much moving in each direction increased the pollution concentration readings. The drone will then move in the direction that caused the largest increase. The drone will continue following this algorithm until there are no more increases in concentration readings. The GA/MLPF algorithm leverages the machine learning particle filter to create a probability density function that is used to determine which points in a random set of 500 points are most likely to be source points. As the drone traverses the pollution field, more data is collected and used with the probability density function to reduce the number of points that are likely to be a source point for 500 to 200. Once at around 200, the drone will go to the mean of the points and this final point is the predicted source location.

We checked the accuracies of the conventional GA and the GA/MLPF algorithms through running four separate subtests simulations in MATLAB for each algorithm. Run #1 and Run #2 show the effect of changes in the drone's starting location, whereas Run #3 and Run #4 show the effect of changes in the pollution source location. In this test, both algorithms' generated paths and the discrepancies between their estimates of the location of the simulated pollution source and the true location were compared and expressed as relative improvement (in %) of the GA/MLPF algorithm with respect to the GA algorithm. Even though none of the algorithms had reached the source point exactly, they were within a maximum of approximately 1.6 units of simulation distance (**Table 1**). This offset is not small and implies that further work needs to be done to improve accuracy, but it does display how the algorithms were able to navigate the drone in the right directions. Each algorithm was able to generate paths from the drone's starting position to the simulated pollution source with the offsets (**Table 1**). Overall, neither algorithm performed better, with the GA/MLPF algorithm working the best in the first two subtests and the GA algorithm working the best in the last two subtests.

https://doi.org/10.59720/24-246

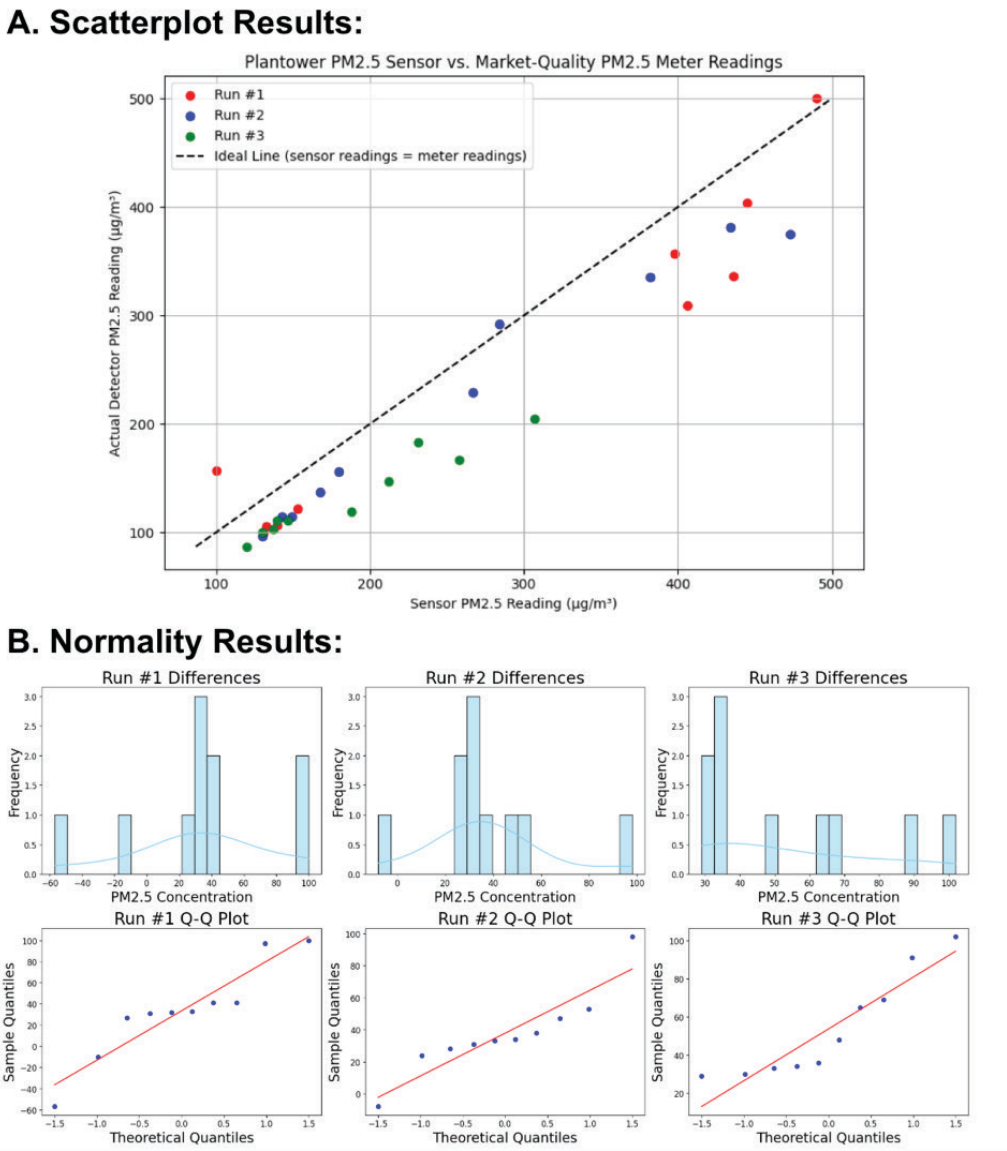

Figure 2: Results for data processing and transmission test. The data processing and transmission test verifies the accuracy of the primary PM2.5 sensor readings against the secondary air quality meter readings and transmission over WiFi. There was a total of three runs for this test. Each was for 1 minute, the concentration readings were collected continuously while the two sensors' displays were videorecorded, and the selected timepoints' data within each of the three runs were obtained from the video. **A)** The data points collected form each run with a dotted line, representing the ideal scenario when the sensor and the meter have the same reading. **B)** Each of the runs' distribution represented by the histograms are not normal. Moreover, the quantile-quantile (Q-Q) plots support this conclusion. In each of the Q-Q plots, because the plotted blue points deviate drastically from the red line, it indicates the distributions can be skewed and there are potential outliers, making the distribution not normal. Due to the un-normal nature of the data, the Wilcoxon Signed-Rank Test was conducted to determine a significant statistical difference.

Comparison of GA and GA/MLPF Algorithms in Real-Time Field Test

The third test consisted of four real-time field drone flight tests utilizing a simulated pollution plume in MATLAB. Run #1 and Run #3 show the effect of changes in the drone's starting location, whereas Run #2 and Run #4 show the effect of changes in the pollution source location. From this test, we found the GA/MLPF algorithm showed reductions in run-time of up to 70% and visited up to 92% fewer points than the GA algorithm alone, underscoring a dramatic improvement in efficiency and supporting our research hypothesis (**Tables 2 and 3, Figure 4**).

DISCUSSION

Outdoor air pollution is a huge issue affecting 99% of the world's population and is the fourth-leading cause of death worldwide, claiming 4.2 million lives annually. Current outdoor air purifiers are stationary, only cleaning air within a certain radius, and are ground-mounted, therefore ignoring the main source of pollution and causing the problem to be constant. In this paper, we developed a drone-mounted sensor network that uses the GA/MLPF algorithm and the DPTS to provide an effective and affordable alternative to outdoor air pollution cleaning. By using the GA/MLPF algorithm to target the source of pollution, this product can reduce pollution by stopping

https://doi.org/10.59720/24-246

Start Point	Actual Source Point	Radius From Source Point	Final Approximated Source Point Location		Distance From Source point		Reached the Source Point Exactly	
			GA	*GA/MLPF*	*GA*	*GA/MLPF*	*GA*	*GA/MLPF*
(0, 0, 0)	(2, 2, 3)	2 units	(2.5961, 1.7537, 2.7502)	(2.3256, 1.9734, 2.9583)	0.692 units	0.329 units	No	No
(0, 0, 0)	(5, 6, 7)	2 units	(5.5992, 5.75, 6.7508)	(5.4069, 5.8395, 7.0342)	0.695 units	0.439 units	No	No
(2, 2, 2)	(0, 3, 3)	2 units	(0.66924, 2.7498, 2.781)	(0.47235, 1.9332, 1.8951)	0.747 units	1.607 units	No	No
(3, 4, 3)	(0, 3, 3)	2 units	(0.59999, 2.75, 2.75)	(0.84756, 3.9675, 3.1409)	0.696 units	1.294 units	No	No

Table 1: Comparison of gradient ascent (GA) and gradient ascent machine learning particle filter (GA/MLPF) algorithms. Comparison of how close the GA and GA/MLPF algorithms' predictions of the pollution source location are relative to the actual location. The maximum distance between the algorithm's prediction and the actual location of the source is 1.6 units. Neither algorithm outperformed the other in terms of accuracy, since the GA/MLPF was better than the GA algorithm for the first two subtests but the GA algorithm was better than the GA/MLPF algorithm for the last two subsets. In all of these subtests, none of the algorithms were able to exactly pinpoint the source point.

Drone Start and Pollution Source	Time Taken to Run Algorithm		Number of Computations		Points Visited by Drone		% Improvement in Time	% Improvement in Computation	% Improvement in Points Visited
	GA	*GA/MLPF*	*GA*	*GA/MLPF*	*GA*	*GA/MLPF*	*GA/MLPF from GA*	*GA/MLPF from GA*	*GA/MLPF from GA*
Start: (0, 5, 5) Source: (0, 3, 3)	~ 12 mins	~ 7 mins	~ 162	~512	84	3	42%	-216%	96%
Start: (0, 0, 0) Source: (2, 4, 2)	~ 43 mins	~ 13 mins	~ 342	~ 575	66	18	70%	-68%	73%
Start: (1, 4, 4) Source: (0, 3, 3)	~ 4 mins	~ 7 mins	~87	~545	36	26	-75%	-526%	28%
Start: (0, 0, 0) Source: (0, 3, 3)	~ 13 mins	~ 6 mins	~ 165	~ 521	98	17	53%	-216%	83%

Table 2: Gradient ascent (GA) and gradient ascent machine learning particle filter (GA/MLPF) real-time test percent comparison. During the third test, two variables were tested: a different pollution source point (shown in the table as source) and a different drone starting position (shown in the table as start). Each row describes the situation with a start and source point.

the spread from the source. Additionally, when compared to the conventional GA algorithm, our results supported our research hypothesis that the GA/MLPF algorithm was able to reduce the time and number of points visited by the drone as it finds the source of pollution.

While the GA algorithm performed better than the GA/MLPF algorithm in terms of final distance from the actual pollution source in certain cases, additional factors should be taken into consideration such as computation speed (smaller run time, smaller number of computations, and smaller number of points visited), where the GA/MLPF algorithm has an advantage over the GA algorithm. The accuracy of GA/MLPF in terms of final distance from pollution source can be achieved by allowing more run time and by fine tuning its modeling parameters in additional simulation trials.

In Run 3 of both tests testing the navigation algorithm (comparing the GA and GA/MLPF algorithms, comparing the GA and GA/MLPF algorithms in real-time field test), there were large discrepancies in the distance traveled and the ending point of the algorithms in comparison to the true source point (**Tables 1–3**). However, each of the algorithms move the drone system in the right directions, seeing that the drone's path heads towards the relative direction of the source point, though there is still a big distance gap (**Tables 1–3**). There is more work to be done to improve the algorithms' accuracy and ensure the algorithms can reach within the vicinity of the source point. The discrepancy for the GA/MLPF algorithm can be explained due to the randomness of the selection. Within the first round of GA/MLPF, the algorithm randomly chooses 500 points that it believes could be the source point. In this test case, the points picked could have affected the outcome of the algorithm, causing it to be far off from the true point. Supporting the research hypothesis, through the inclusion of the machine learning particle filter within the GA/MLPF algorithm allowed for a time-efficient algorithm that visits many less waypoints at the cost of more computations. At every time step, using the newly collected information of the gradient field, the GA/MLPF algorithm re-evaluates the potential source points by adjusting its probability density function. This re-evaluation uses more

https://doi.org/10.59720/24-246

Drone Start and Pollution Source	Final Position		Reached the Source Point Exactly	
	GA	*GA/MLPF*	*GA*	*GA/MLPF*
Start: (0, 5, 5) Source: (0, 3, 3)	(2.6, 3.75, 1.75)	(2.4063, 3.9654, 1.9638)	No	No
Start: (0, 0, 0) Source: (2, 4, 2)	(0.5999, 2.75, 2.7501)	(1.3451, 2.9578, 2.434)	No	No
Start: (1, 4, 4) Source: (0, 3, 3)	(0.67731, 2.7506, 2.8721)	(0.40158, 2.8729, 3.0306)	No	No
Start: (0, 0, 0) Source: (0, 3, 3)	(0.59936, 2.75, 2.7506)	(0.39211, 2.8951, 3.0367)	No	No

Table 3: Gradient ascent (GA) and gradient ascent machine learning particle filter (GA/MLPF) real-time test final position comparison. During the third test, two variables were tested: a different pollution source point (shown in the table as source) and a different drone starting position (shown in the table as start). Neither of the algorithms had reached the exact source location.

Figure 3: Materials and assembly of the filter unit. To make the filter unit inexpensive, these materials were found around the house and used for initial testing. The spiral, motor, and cup are assembled to create the cleaning function of the data processing and transmission system. This part of the system will be triggered by a signal sent between the Arduinos and has been proven effective in cleaning polluted air (17).

computations than the number of computations used in each step of the GA algorithm. However, these extra computations allow for the GA/MLPF to visit a lot less points than the GA algorithm, allowing the GA/MLPF, in many cases, to reach the source pollution faster than the conventional GA algorithm, allowing it to prevent the spread of the pollution and clean the air faster. The faster cleaning and reduced spread of pollution achieved through the GA/MLPF algorithm makes the sacrifice of computation reasonable. Despite improving in almost every aspect, different starting points cause GA/MLPF to take more time to find the source and to have a less accurate prediction of the actual location of the source, but with further tuning and more training this problem could be mitigated. Moreover, when looking at the paths produced, the GA and the GA/MLPF have very different paths: GA takes a longer, less obvious path and GA/MLPF takes a shorter, more obvious path. Paths produced could be smoother with additional optimizations.

One of the problems encountered was that the filter unit system was too heavy for the drone to carry. Given that the drone is about the size of one's hand and the battery only allows it to fly for 13 minutes, adding a filter unit that carries a microprocessor, motor, sensor, and 3-D printed part was too much for such a small drone to carry. One solution to this problem is to create a printed circuit board (PCB) for the filter unit to make it lightweight. On the DPTS, we currently use an ArduinoR4 microcontroller to power and control the system. When using the ArduinoR4, more than half of the features were left unused. Therefore, it would make the most sense to develop a PCB that only contains the features needed for this project. Moreover, due to the battery dropping rapidly, when testing only a few minutes of good flight time was allocated. To solve this issue, MATLAB's simulation feature was used to get a good idea of what the drone's path may look like and to fix any errors found in the program before deploying it onto the drone.

In the comparison of the GA and GA/MLPF algorithms in real-time test, due to the drone's weight and battery limits, the filtration unit was not able to be used and was instead replaced by a simulated polluted environment in which the navigation algorithm could extract concentration readings through calling a function. Although the filtration unit was not used in the real-time test due to the drone's weight and battery limits, it is a necessary part of the system. The PM2.5 concentration data collected through the filtration unit is crucial in determining the next location of the drone. A better drone having a higher payload capacity and longer battery life would be able to carry the filtration unit during field tests.

Moreover, the inconsistency with the Plantower PM2.5 sensor used in DPTS was another problem. With inconsistent readings, the pollution field could be misinterpreted by the navigation algorithm, causing the system to be less

https://doi.org/10.59720/24-246

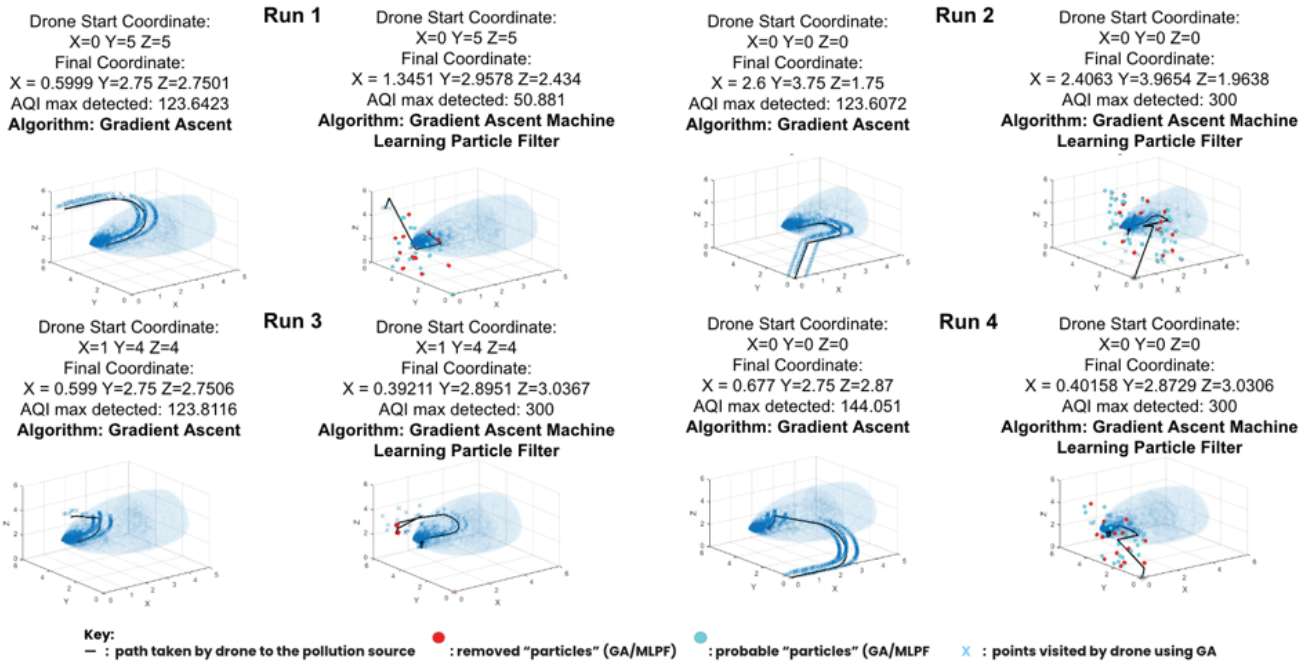

Figure 4: Drone trajectory comparison for GA and GA/MLPF algorithm real-time test. The paths depicted in this figure were developed in a MATLAB simulation of the GA and GA/MLPF algorithms. Two variables were altered: the pollution source location and the drone's starting location. Run #1 and Run #2 show the effect of changes in the drone's starting location, whereas Run #3 and Run #4 show the effect of changes in the pollution source location. Each case plots the drone's path in black and shows the drone's starting coordinates, the pollution source's coordinates, the number of points visited, the time taken to traverse the path, and which algorithm generated the path.

effective. A potential cause of this outcome is that the air in the testing region and the sensor were not properly cleaned prior to collecting readings, which would mean that there were already existing PM2.5 particles that could cause the readings from the Plantower sensor to be consistently higher than that from the air quality meter. Another cause could stem from the internal design of the sensor itself. The current systems reported in this paper for PM2.5 detection, including the air quality meter and the Plantower PM2.5 Sensor, are low-cost systems that utilize laser-based detection systems and detect particles utilizing aspects of light scattering. Light scattering, especially in low-cost sensors, prevents these sensors from accurately separating particles by size. This makes the concentration readings from these low-cost sensors, especially for PM2.5 and PM10, to have a lot of variation with little correlation between data points. More research regarding other methods of particle detection, such as infrared detection, as well as how to accurately recalibrate these low-cost sensors to produce more accurate readings (11).

In terms of future research, adding more drones to the system would allow more space to be covered and reduce the time needed to clean an area. Moreover, in addition to additional drones, adding more advanced sensors to enable communication between drones would ensure that the space is cleaned and covered efficiently and that there are no safety concerns when adding more drones. For future improvements, more random points or more constraints regarding where the random points are picked from can be added to ensure that there is a wide spread of points, allowing the algorithm to have the best chance of accurately locating the source point. Overall, the GA/MLPF algorithm developed in this project is an innovative approach to tackling outdoor air pollution, and the algorithm's mobility and effectiveness allow it to be used in many diverse environments.

MATERIALS AND METHODS

Data Processing and Transmission System (Setup and Attachment)

The Data Processing and Transmission System (DPTS) consisted of two main parts, the aerial unit and the ground unit (**Figure 1**). The Aerial unit utilized a pre-made and pre-approved DJI Tello Educational Drone with built-in Wi-Fi, an Arduino R4 UNO with Wi-Fi, a PM2.5 Plantower PMS5003, and a filter unit (**Figure 3**). The filter unit was made up of a Mini Micro Coreless Motor, a 3-D printed Fibonacci Spiral, a paper/plastic cup, and a PM2.5 filter (16). The filter unit is a motor connected to a Fibonacci spiral that sucks air into a PM2.5 filter paper attached to the back of a cup (**Figure 3**). To customize the filter's size, shape, and material and ensure a lightweight and efficient filter, a 3-D printed spiral was used.

A complete circuit was built using these parts and attached to the drone according to the following guidelines (**Figure 1**). The PM2.5 sensor was connected to Arduino2 to get power and was allowed 30 seconds for stabilization after waking from sleep mode. The sensor has continuous, and polling modes based on the RCX pin voltage. The filter motor was connected to a 9V battery using a N-type metal–oxide–semiconductor (NMOS) to ground it. The NMOS gate was connected to pin 7 on the Arduino to complete the circuit and allow for the activation of the motor using the 9V battery.

The ground unit consisted of an Arduino R4 UNO with

https://doi.org/10.59720/24-246

Wi-Fi connected to a laptop. The navigation algorithm was run on the laptop. The ground Arduino Uno was connected to the Arduino in the aerial unit through WiFi-Arduino and the computer is connected to the drone through WiFi-Drone. Within the DPTS, the two Arduino R4 UNOs with Wi-Fi sent commands and data through the WiFi-Arduino channel. This is how the algorithm, described further in the navigation algorithm section, gained real-time pollution concentration data. Additionally, through the WiFi-Arduino channel, signals were sent to trigger the filter unit to begin running and cleaning the polluted air if the air pollution concentration passed a certain threshold. The DPTS' Ariel Unit was mounted on the Tello Drone utilizing a four-wall cardboard structure, attached to the landing gear (**Figure 3**).

Navigation Algorithm

Within the creation of the navigation algorithm, clear conditions were assumed (no wind or other weather conditions), the pollution source was assumed to be a point, multiple pollution sources were not within the scope of this project, and the effects of gravity on the plume field distribution were ignored. For simulation and consistency purposes, the following Gaussian plume model was used for the spatial distribution of the pollution (12, 13):

$$C(x,y,z) = \frac{Q}{2\pi u \sigma_y \sigma_z} e^{\frac{-(y-y_0)^2}{2\sigma_y^2}} \left(e^{\frac{(z-z_0-H)^2}{2\sigma_z^2}} + e^{\frac{-(z-z_0+H)^2}{2\sigma_z^2}} \right) \quad (1)$$

where the source of pollution is taken to be (x_0, y_0, z_0) with

$$a_y = 0.5, b_y = 0.92 \quad (2)$$

$$\sigma_y = a_y \cdot |x - x_0|^{b_y} \quad (3)$$

$$a_z = 0.5, b_z = 0.87 \quad (4)$$

$$\sigma_z = a_z \cdot |x - x_0|^{b_z} \quad (5)$$

which, when substituted into equation (**Equation 1**), shows the dependence on *x* as:

$$C(x,y,z) = \frac{Q}{2\pi u \cdot \left(a_y \cdot |x-x_0|^{b_y}\right)\left(a_z \cdot |x-x_0|^{b_z}\right)} \cdot e^{\frac{-(y-y_0)^2}{2\left(a_y \cdot |x-x_0|^{b_y}\right)^2}} \cdot \left(e^{\frac{(z-z_0-H)^2}{2\left(a_z \cdot |x-x_0|^{b_z}\right)^2}} + e^{\frac{-(z-z_0+H)^2}{2\cdot\left(a_z \cdot |x-x_0|^{b_z}\right)^2}} \right) \quad (6)$$

in which C = pollutant concentration, Q = source emission rate [g/s], u = wind speed [m/s] = 1 (factor was not changed), y = crosswind distance from stack of point of interest [m], z = vertical height of point of interest [m], H = effective stack height [m] = 0, and x = x coordinate of point of interest [m].

The creation of the navigation algorithm used two algorithms: the well-known gradient ascent (GA) algorithm was used as a control and for comparison, and a newly developed gradient ascent with a machine learning particle filter (GA/MLPF) algorithm. The GA algorithm commanded the drone to traverse in the three dimensions (in the x-direction, the y-direction, and the z-direction) at each step to decide on the direction to move (14). The traversal started from (0,0,0) and moved 0.1 meters in each direction. At each of these locations, Arduino2 took the PM readings and sent them to Arduino1, which in turn sent them to the algorithm. Using the readings from each of these directions, the algorithm took a weighted average of all the values to determine the next point to go to. This information was sent on the WiFi-Drone channel from the computer. Following this method, the drone followed the trend of the pollution gradient.

The drone continued to follow this algorithm until it found the pollution source. The drone knew it had found the pollution source since when it moved in the *x*, *y*, and *z* directions the change was negative, indicating that it had found the highest pollution concentration. The maximum concentration was a setting parameter for these algorithms and was used to control how close the drone goes to the source of pollution. The GA/MLPF algorithm uses a mix of the machine learning particle filter with the gradient ascent algorithm, to help the drone autonomously navigate the air pollution source (15).

The novel idea is to use state estimation algorithms like Bayesian particle filters to estimate the source of pollution from the data that was gathered by the GA algorithm and based on PM2.5 sensor readings. The main idea of the Bayesian particle filter is to determine the location of an object within a certain location using a specific input source. The particle filter starts off with random points scattered around the area, and over time with more data collected, the filter begins narrowing down and clustering at certain points it believes the object is at, until it ultimately locates the object.

This project employed a novel use of the machine learning particle filter. In a normal setting, when the particle filter is utilized, it is within a known region, however in this case, the gradient environment was unknown and only pollution concentration readings were used to navigate through the plume field in order to try to find the pollution source. Using the particle filter method, the algorithm started by randomly labeling 500 points the source of pollution and saved the state of the pollution source as (*x, y, z*) coordinates, indicating that the pollution source could have been at any of these points, but as the program continued, the number of possible pollution source points got smaller. These pollution source candidate points were located in the plume field which spanned a 5 m x 5 m x 5 m cubic region.

Following the algorithm, the drone first 5 rounds of the GA algorithm to get an understanding of the unknown field. In these rounds, during each new point, actual air quality data was collected, and the list of candidate source points was re-evaluated. At this current point, the algorithm cycled through estimated "source points", sending the coordinates of each estimated source point and the current point to a pollution concentration modeling function.

Because many pollution plumes can be modeled as Gaussian plumes, the Gaussian field equation was used to estimate pollution concentration at a point when given a source point (16). From the current point and the candidate source point, the PM concentration at the current point was calculated assuming that the source point was the real source point used within the experiment. The large set of approximately 500 possible source points was reduced by applying a threshold-based posterior likelihood elimination strategy, and if a candidate source point got eliminated, then it was considered as an ordinary sample point within the pollution field. If the absolute value of the difference between the Gaussian plume model concentration and the PM sensor

https://doi.org/10.59720/24-246

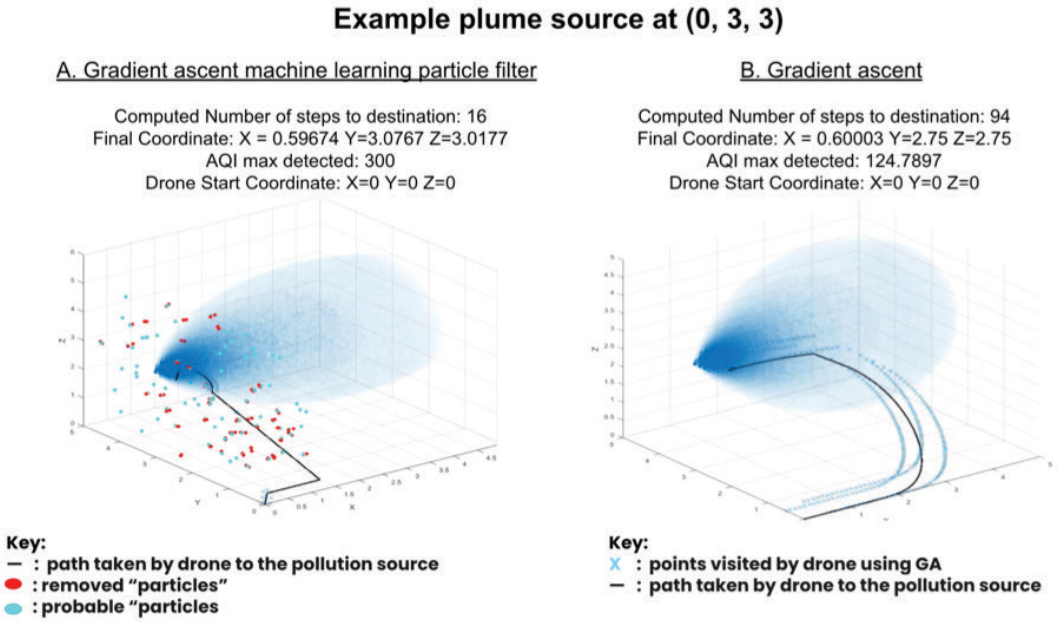

Figure 5: Drone Trajectories for the GA/MLPF and GA algorithms. The two graphs created in this figure were made in MATLAB. This figure depicts two example paths, traced in black, that conform to the conditions of Test 2 with the drone's starting position at (0, 0, 0) and the pollution source's location at (0, 3, 3).

reading at the current point was greater than 2 parts per million (ppm), then the candidate source point was eliminated. The threshold value of 2 ppm quickly reduced the number of candidate source points in the simulation studies through ensuring only the candidate source points that were included within the set that could be a source point are the ones with the closest approximated concentration readings.

As the drone moved in the pollution field, it acquired more and more PM concentration readings. Using this information, the MLPF identified candidate source points by assigning each particle a sampling weight based on a sequential Monte Carlo posterior probability density function. Rather than removing candidate source points whose probabilities are below a certain threshold value, the Monte Carlo iterative resampling process causes candidates having lower sampling weights to get removed from consideration. When the number of candidate source points was less than 200, the drone was sent to the median of the 3-D coordinates of the remaining candidate source points. From this location, the GA was used to find the actual location of the pollution source. Using MATLAB, paths from both algorithms (GA alone and GA/MLPF) were graphed, and all the possible source points are colored either red if they are excluded from, or light blue if they are included in, the final iteration's median coordinate calculation (**Figure 5**).

Data Transmission and Accuracy Test

The data transmission and accuracy test were conducted to verify reliable communication between the ground and aerial units and to assess the accuracy of PM2.5 concentration measurements collected by the Plantower Sensor on the aerial unit.

The ground and aerial units communicate via Arduinos connected through a WiFi-based Arduino channel. To ensure the reliability of this communication link, sample PM2.5 data was transmitted and then verified for accuracy upon receipt. This test was conducted outdoors in the operational field, with varying distances between the ground and aerial units, to ensure that Arduino-based communication remains robust under real-world conditions. To ensure the PM2.5 sensor was transmitting accurate data, air quality was also changed by adding smoke that changed the PM2.5 concentration. The proposed smoke source was a burnt matchstick placed in front of the sensor.

Values from the PM2.5 sensor were analyzed and compared against an air quality sensor to ensure correctness. Per run analyses were conducted to account for the rapid fluctuations of pollution within the short 3-minute span of the test. We used histograms and quartile-quartile plots to display the normality of the data. Additionally, we used a Wilcoxon signed-rank test with a significance level of 0.05 to test for a significant statistical difference between the readings from the Plantower PM2.5 Sensor and the market-quality air quality meter.

Comparing the GA and GA/MLPF Algorithms

When running the pollution search algorithms during testing, clear operating conditions (such as no wind, no rain, etc) and having the drone placed within the pollution field were assumed. We compared the GA and GA/MLPF algorithms to determine the functionality of the algorithms before they were deployed for testing on the drone. Due to safety and battery constraints, the four paths developed by the GA and GA/MLPF algorithms were simulated within MATLAB and checked for accuracy in finding the source (**Table 3**). Source code can be accessed here: https://github.com/varinik/AeroPurifySimulationMATLABCode/tree/main.

Each of the four tests had an outlined started place for the drone and the source point of the pollution plume. In MATLAB, the simulated environment consisted of a model Gaussian

https://doi.org/10.59720/24-246

Plume centered at the pre-set source point (**Equation 6**). In the simulated environment, did not model the drone as a physical object but did keep track of the drone's current location based on the outputs of the algorithms. The simulated environment focused on capturing the logic of the two algorithms. At the start of the simulation, the current location of the drone was set to the pre-determined starting location. For the algorithms to get concentration readings from the field, a function that utilizes the Gaussian Plume equation (**Equation 6**) was used to convert values represented by the pre-determined source point (x_0, y_0, z_0) and the current location (x, y, z) to a pollution concentration value. The simulation was run and the amount of time it took the algorithm being tested to reach the source point, the number of points the algorithm visited, and the number of computations performed by the algorithm were measured. An accepted radius of two units of simulation distance (modeled using a spherical source rather than a point source) was used as since in a real-world scenario pollution is likely to be the same concentration within a radius from the pollution source, assuming clear conditions. Therefore, within this test, having the drone be within that radius would be accepted as reaching the target.

Comparison of GA and GA/MLPF Algorithms in Real-Time Field Test

The comparison of GA and GA/MLPF algorithms in real-time field test is a real-world test involving the physical drone system. This test used a simulated environment to mimic a real-life pollution environment (as outlined in the comparing the GA and GA/MLPF algorithms test), and the calculated steps were deployed and run on a drone. In the simulated environment, data such how many points the path consisted of, how long it took for the drone to reach the pollution source using the path, and how many computational steps were needed, were collected and compared between the two algorithms. As in real-life scenarios, the drone may not always originate in the vicinity of the pollution source, and the pollution gradient may not always originate from a fixed starting point, hence two variables were changed: the drone's starting location and the pollution source's location. With these changes, the data elements cited above were collected and compared between the two algorithms, yielding the experimental results and scientific interpretations previously discussed. For this test, six paths were produced (**Figure 4**).

Received: June 20, 2024
Accepted: February 2, 2025
Published: September 1, 2025

REFERENCES

1. "What Is Particle Pollution?" EPA, Environmental Protection Agency, 20 June 2024, https://www.epa.gov/pm-pollution. Accessed 5 Oct. 2024
2. Staff, MND. "Hold Your Breath: Mexico City Air Contains 172 Toxic Compounds." Mexico News Daily, https://mexiconewsdaily.com/news/hold-your-breath-mexico-city-air-contains-172-toxic-compounds/. Accessed 8 Aug. 2024.
3. "Air Pollutants." *Centers for Disease Control and Prevention*, https://www.cdc.gov/air-quality/pollutants/index.html. Accessed 20 Nov. 2023.
4. "Protecting Yourself from Wildfire Smoke." *California Air Resources Board*, https://ww2.arb.ca.gov/smokereadyca. Accessed 21 Nov. 2023.
5. "What is Soot?" *Restoration Local*, https://www.restorationlocal.com/what-is-soot-and-is-it-dangerous. Accessed 21 Nov. 2023.
6. Murray, Virginia, *et al.* Hazard Information Profiles: Supplement to UNDRR-ISC Hazard Definition & Classification Review: Technical Report. United Nations Office for Disaster Risk Reduction; International Science Council, 2021. https://www.undrr.org/publication/hazard-definition-and-classification-review-technical-report.
7. Hazardous Ozone-Generating Air Purifiers, California Air Resource Board, https://ww2.arb.ca.gov/our-work/programs/air-cleaners-ozone-products/hazardous-ozone-generating-air-purifiers. Accessed 11 Nov. 2024.
8. "Baghouses and Baghouse Filters Selection Guide: Types, Features, Applications | GlobalSpec." GlobalSpec, GlobalSpec, https://www.globalspec.com/learnmore/manufacturing_process_equipment/air_quality/baghouses. Accessed 11 Nov. 2024.
9. Donnelly, Nicole. "How Much Does a Dust Collector Cost?" U.S. Air Filtration, U.S. Air Filtration, 7 Sept. 2023, https://www.usairfiltration.com/how-much-does-a-dust-collector-cost/?srsltid=AfmBOooFe9w2TNe0ouKqFE2NJboXCtOFhtqR0s0NaligoBF0nmMMH08U.
10. Kim, Seung Hyeok, *et al.* "Personal Exposure to Airborne Particulate Matter in Various Microenvironments and Its Relationship with Lung Function in Elderly Adults." *Environmental Health and Toxicology,* vol. 38, 2023, Article e2023005, https://pmc.ncbi.nlm.nih.gov/articles/PMC10018765/.
11. Bhandarkar, Upendra V., and D. V. Shet. "Numerical and Experimental Investigation of a Filter-Based Air Purification Device." *Journal of Aerosol Science,* vol. 162, 2022, Article 105961, https://doi.org/10.1016/j.jaerosci.2021.105833.
12. T. C. Henderson and E. Grant, "Gradient calculation in sensor networks," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2004, vol.2, pp. 1792-1795, https://doi.org/10.1109/IROS.2004.1389656.
13. Cooper, David, and F.C. Alley. "Air Pollution Control." Waveland Press, 2002.
14. "Gradient Descent: An Introduction to One of Machine Learning's Most Popular Algorithms." *Built In*, https://www.builtin.com/data-science/gradient-descent. Accessed 16 Nov. 2023.
15. Carpin, M, *et al.* "UAVs using Bayesian Optimization to Locate WiFi Devices." *ArXiv*, 2015, https://doi.org/10.48550/arXiv.1510.03592.
16. Micallef, A., and C. Micallef. "The Gaussian Plume Model Equation for Atmospheric Dispersion Corrected for Multiple Reflections at Parallel Boundaries: A Mathematical Rewriting of the Model and Some Numerical Testing." *Sci*, vol. 6, no. 3, 2024, pp. 48, https://doi.org/10.3390/sci6030048.
17. Kadakia, V., and T.A. Nguyen. "Drone-Based System to Reduce Air Pollution Using Inexpensive Materials." *Journal of Student Research*, vol. 12, no. 4, Nov. 2023, https://doi.org/10.47611/jsrhs.v12i4.5432.

JOURNAL OF
EMERGING INVESTIGATORS

https://doi.org/10.59720/24-246