Framework for coordination of heterogeneous UAV's

Ján Zelenka[1][0000-0001-5852-0429] and Tomáš Kasanický[1][0000-0002-1931-6537]

[1] Institute of Informatics, Slovak Academy of Sciences, Dúbravská cesta 9, Bratislava 84507, Slovakia
jan.zelenka@savba.sk, tomas.kasanicky@savba.sk

Abstract. A natural development in the field of drone-based environmental monitoring is to enhance the method's effectiveness through parallelization. Swarm robotic drones are increasingly a focus of research in this area. Several works address this issue[4]. But these are homogeneous closed systems. This work aims to integrate multiple devices into a single functional solution. It was created as a result of the international Silvanus H2020 project. This work presents a comprehensive and tested framework for coordinating drones, with support for loading *.klm files. It also introduces the integration of no-fly zones to protect selected environments during an intervention. All of this has been tested in several real-world environments across different countries.

Keywords: multitrajectory, swarm robotic, drone

1 Introduction and problem description

In modern firefighting, information is the most critical resource. The efficacy of suppression efforts is functionally limited by the "situational awareness gap"—the delay between fire ignition and its detection, and the subsequent lag in tracking its volumetric growth. While traditional aerial surveillance relies on single sorties of piloted aircraft or satellites with intermittent pass-over times, the deployment of Unmanned Aerial Vehicle (UAV) swarms offers a paradigm shift toward continuous, distributed, and real-time situational awareness. The most vital asset in contemporary firefighting is information. The success of suppression activities is intrinsically hampered by the "situational awareness gap"—the delay between a fire's ignition, its eventual detection, and the subsequent lag in monitoring its volumetric expansion. Traditional aerial monitoring relies on single-piloted aircraft sorties or on satellites with intermittent coverage. However, utilizing swarms of Unmanned Aerial Vehicles (UAVs) introduces a transformative approach, enabling continuous, distributed, and real-time situational awareness.

This work is a continuation of previous research [4, 5]. It was created as a result of the international Silvanus H2020 project [1]. The work presents a comprehensive, validated framework for drone coordination that supports loading *.kml files. Furthermore, it incorporates no-fly zones to safeguard specific environments during operational interventions. While the system has been successfully tested in diverse real-world settings internationally, rare edge cases revealed area-partitioning errors caused by the constraints of these no-fly zones

2 Route planning for spatial mapping

2.1 Algorithm description and comparison of solutions

The user selects the polygon where drone intervention is required, the no-fly zone (if any), and the number and type of drones. Based on the selected coverage, a clustering method is used to divide the drones into individual sectors. In the given sectors, optimal drone paths are determined based on the weather, including avoiding the no-fly zone. These generated flight paths are saved in KLM format and uploaded to individual drones via the platform (in more detail described in [4]).

To ensure even drone coverage of the area of interest, it is essential to divide it evenly while accounting for the no-fly zone. There are several techniques for finding evenly spaced sets. Poisson Disc Sampling, Sobol or Halton Sequences, Stratified (Jittered) Grid, but the most "perfect" mathematical evenness is Lloyd's algorithm. We decided to test two approaches: the first is k-means, the other is the Self-organizing map (SOM), also known as the Kohonen map. Both algorithms can produce evenly spaced points within a polygon, and their results are based on the Voronoi diagram, like Lloyd's algorithm. When a user generates a flight area with defined no-fly zones and specifies the number of drones to operate in the area, the polygon is divided into sub-areas for each drone using the aforementioned algorithms. The number of drones is therefore equal to the number of clusters that K-means and SOM search for in a given polygon. This ensures uniformity of area for each drone. The clustering method divides the area of interest into sub-areas for individual drones. This equal distribution is ensured by using the K-nearest neighbor (KNN) algorithm. The parameter number of clusters is set equal to the number of drones. The result of this setting is an evenly divided area

A feature of the K-means algorithm is that it can produce inconsistent results when run multiple times. There is another issue: if the no-fly zone is on the border of the monitored area and does not have a specific shape (e.g., a V-shaped space, sharp edges, or an inner polygon), algorithms based on the K-nearest neighbors clustering method work fine. But if the space contained these areas within the monitored area, there could be cases where the flight paths were complicated and were just due to clustering methods. Figure 1 represents the difference between the KNN (left column) and the SOM algorithm (right column). It can be seeing problematic areas of the KNN method for different areas, e.g. classifying yellow points (left column first row) within a narrow part of the monitored area, turquoise points (left column second row) in the no-fly zone area or pink points (left column third row) within the no-fly zone. According to the KNN method, they are close to each other, but when generating the flight trajectory, there is a complication with avoiding the no-fly zone.

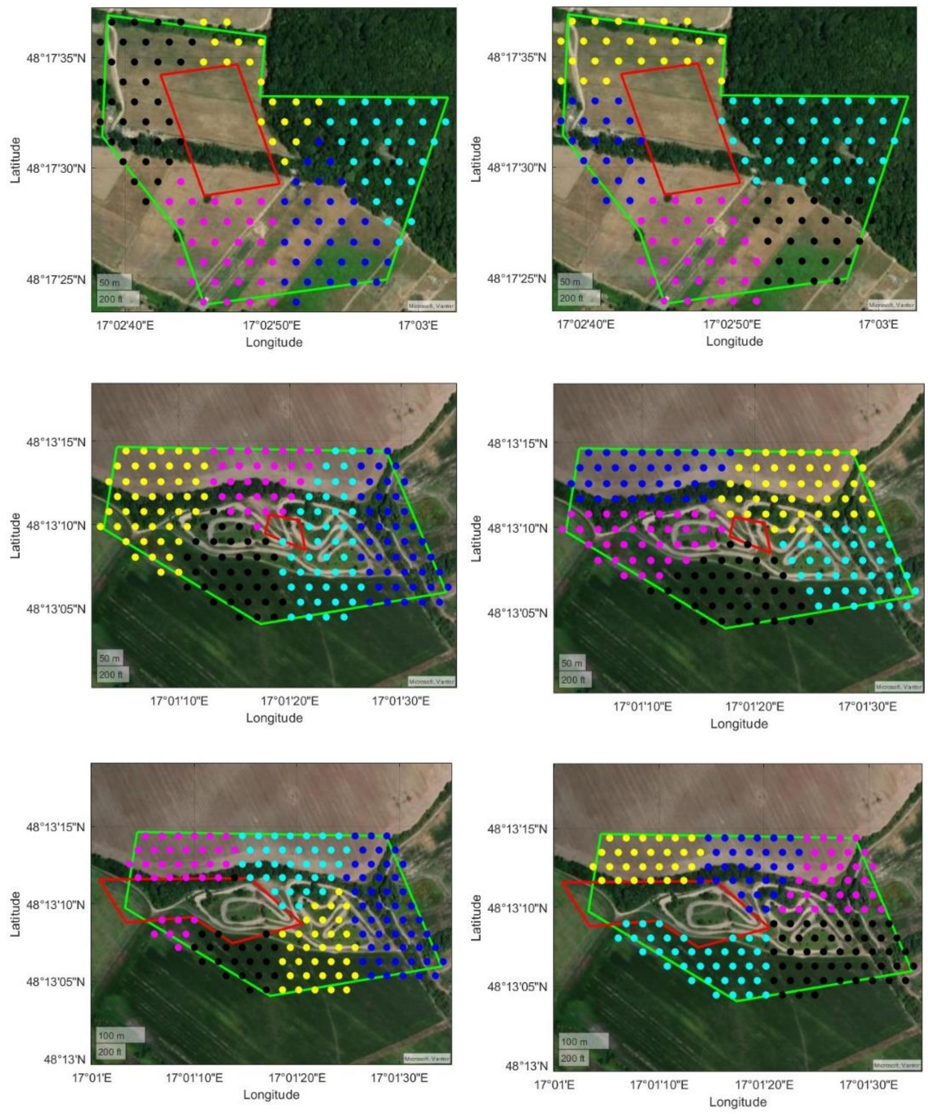

Fig. 1. Comparison of the clustering method, K- nearest neighborhood left column, Self-organizing map right column.

The following part presents the pseudocode of the proposed algorithm. Wind direction, area of interest, no-fly zone, and parameters of each drone must be present as inputs to the algorithm. In the first step, the algorithm evenly divides the area of interest into polygons. The number of polygons is equal to the number of drones. This is done using described clustering algorithm, where the inputs are the number of clusters (number of drones) and the polygon (area of interest), with a no-flight zone. The clustering results divide the area of interest into evenly spaced partial subspaces, each corresponding to a drone's flight zone. The next step is to generate the drone's flight path. For each drone, select one of the subspaces. The trajectory mesh is based on wind direction (forecast or measured), so drones fly upwind and downwind while mapping the area. The flight height and the number of pictures the drone takes are calculated based on the drone's specifications (resolution, camera specifications, flight duration). Then, the root from the starting point to the appropriate polygons is generated for each drone. The results of the calculation, including the trajectory from the start to the polygon and the mesh trajectory within the polygon, are saved as KLM files. These files are sent to drone operators via platform described in section 2.2.

Algorithm: Multi-UAV Area Coverage and Trajectory Planning

Let the following be the defined Inputs:

- N: total number of drones
- $\mathcal{D} = \{d_1, d_2, \dots, d_N\}$: set of drone parameters, where each d_i contains specs (res_i, cam_i, dur_i)
- $\vec{w}$: wind direction vector
- $\mathcal{A}$: the geometric area of interest
- $\mathcal{Z}$: the set of no-flight zones
- $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$: set of drone starting coordinates

Let the Output be:

- $\mathcal{K} = \{k_1, k_2, \dots, k_N\}$: set of exported trajectory files (KML)

Procedure:

Step 1: Spatial Partitioning (Clustering)

1. Define the flyable operational space $\mathcal{O}$ by subtracting the no-flight zones from the total area:

$$\mathcal{O} = \mathcal{A} \setminus \mathcal{Z}$$

2. Partition the operational space $\mathcal{O}$ into N evenly spaced, mutually exclusive subsets (polygons) using a spatial clustering function C:

$$\mathcal{P} = \{p_1, p_2, \dots, p_N\} \leftarrow C(\mathcal{O}, N)$$

$$Ensure\ that \bigcup_{i=1}^{N} p_i = \mathcal{O} and p_i \cap p_j = \emptyset for i \neq j.$$

Step 2: Trajectory Generation

3. **For each** $i \in \{1, 2, \dots, N\}$ **do:**

4. **Compute Flight Parameters:** Let f be the function that derives flight altitude (h_i) and photo count (c_i) from drone hardware capabilities d_i:

$$(h_i, c_i) \leftarrow f(d_i)$$

5. **Generate Mesh Trajectory:** Let g be the sweep-line generation function that aligns parallel tracks parallel/anti-parallel to the wind vector $\vec{w}$ within polygon p_i:

$$M_i \leftarrow g(p_i, \vec{w}, h_i, c_i)$$

6. **Calculate Transit Route:** Let *r* be the routing function from the start point s_i to the initial entry coordinate of the mesh $M_{i,start}$ without way through the no-fly zone:

$$R_i \leftarrow r(s_i, M_{i,start})$$

7. **Merge Trajectories:** Combine the transit route and the coverage mesh into a unified path T_i:

$$T_i \leftarrow R_i \cup M_i$$

8. **File Export:** Save the discrete points of T_i encoded into a file format k_i:

$$k_i \leftarrow ExportToKML(T_i)$$

9. **End For**

Step 3: Output Delivery

10. Return the final set of files $\mathcal{K}$ to the operator terminal.

After several experiments, we decided to use the SOM algorithm, which consistently produced useful results. For clusters of measurement points generated by clustering algorithms, it is essential to perform flight-trajectory planning that explicitly accounts for current meteorological conditions, operational limits, and the UAV's maneuvering characteristics. Such a trajectory minimizes operational risks, enhances mission robustness, and enables efficient autonomous traversal of the individual clusters in full compliance with safety and regulatory requirements.

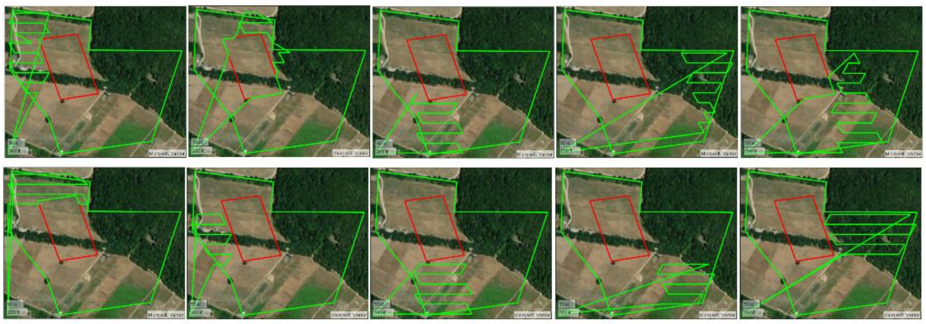

Fig. 2. Comparison of the path generation by the K- nearest neighborhood (first line) and Self-organizing map (second line) for 5 UAV.

The resulting mission trajectories, obtained after applying all operational, environmental, and safety constraints, are illustrated in the following Fig. 2 (case from first line of the Fig. 1). These trajectories represent the final optimized flight paths that fully comply with UAV performance limitations, weather-related considerations, and enforced non-fly-zone restrictions. The visualization provides a comprehensive overview of how the planned route adheres to all imposed requirements while maintaining safe and efficient navigation across the clustered target areas.

2.2 Drone flight routes computing process in SILVANUS platform [1]

As we mentioned, this algorithm was developed for the Silvanus project. Following the chapter, describe the integration of the algorithm in the Silvanus project. The fire command center (FCC) can ingest data collected during drone flights into the SILVANUS Cloud [1, 2]. During the first period, a process was developed to request the computation of drone flight routes (using the described algorithms) via an EmerPoll system [3] deployment at the FCC [2, 6]. As input, the operator enters the system a number of available drones and their features (camera resolution, lens focal distance, number of pixels on the camera chip), wind directions, an area of interest (a polygon), and a required mapping resolution or approved flight height.
As the output, the optimal flight route plans are calculated using different available services for a specified number of drones. The individual routes need to be subsequently loaded to individual drones (resp. to drone pilots). The related work was also published in [4, 5]. The process of computing drone flight routes via coordination algorithm service is described in Figure 3.

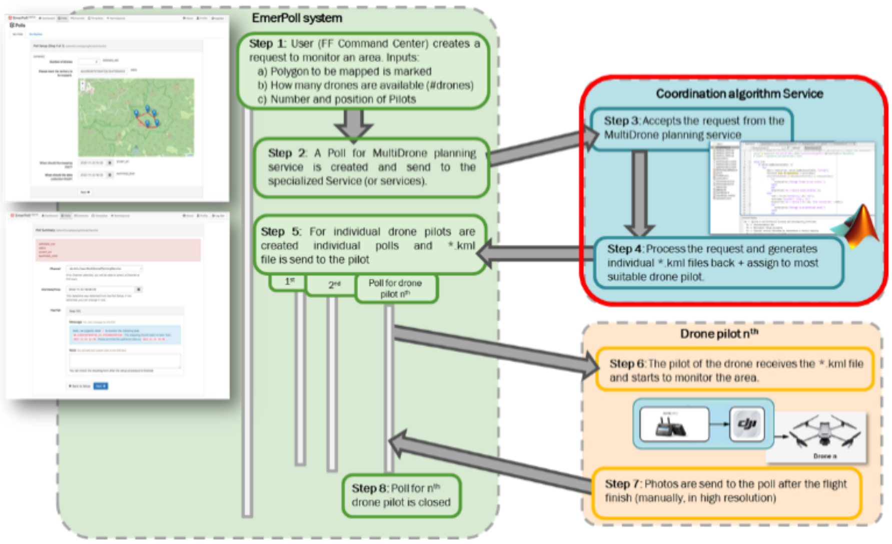

Fig. 3. Process of computing drone flight routes at the EM EmerPoll system [3].

A distinctive strength of this approach is its ability to collect and aggregate images from a wide range of drones—across different models and types—in a consistent and unified manner. Images are initially stored in the EmerPoll image repository. Selected images can then be ingested into the Silvanus dashboard platform (Fig. 4) and forwarded to other system components for further processing, such as fire detection (as described in the previous section). A new dedicated map layer is also being developed and integrated into the Dashboard to visualize drone flight paths alongside the images captured during the missions. Individual photos from the flight path are displayed in the right part, where they can be viewed sequentially with the relevant additional information.

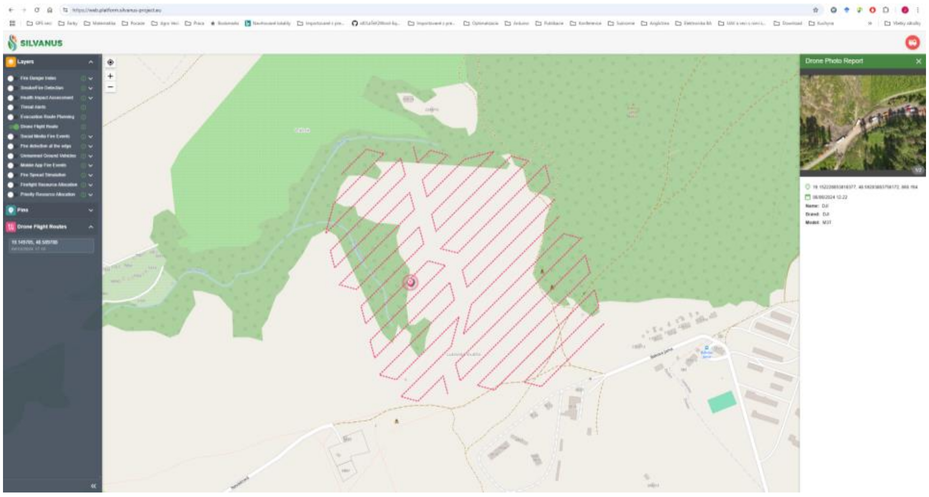

Fig. 2. Results of drone path generation for three drones in Slovak pilot with the Silvanus dashboard visualization for three UAVs. Visualization of the images taken by drones in the dashboard [6].

3 Conclusion

The presented work offers a comprehensive framework for coordinating a group of drones from different manufacturers, with support for the KLM format. The presented algorithm is robust and computationally inexpensive. It allows defining no-fly zones, but on the other hand, optimizes their bypassing and navigation. In the future, it may be possible to integrate ground robots into mission planning and expand integration during the intervention. The algorithm showed its capability in real real-world scenario in 6 countries during 8 fire drills.

This work was supported by the project SILVANUS-SK (Grant No. 09I01-03-V04-00107), funded through the Recovery and Resilience Plan Mechanism, VEGA project No. 2/0135/23 and bilateral project Cyber-Physical System for environmental monitoring and data analysis (BAS-SAS-2024-01).

References

1. SILVANUS project homepage, Integrated Technological and Information Platform for Wildfire Management, https://silvanus-project.eu/ , last accessed 2026/2/11.
2. SILVANUS dashboard, https://silvanus-project.eu/user-products/silvanus-dashboard/ , last accessed 2026/2/11.
3. Balogh, Z.; Gatial, E.; Dolatabadi, S.H.; Dlugolinský, Š.; Saltarella, M.; Scipioni, M.P.; Grunwald, D.; Przybyszewski, M.; Majlingova, A.; Majlingova, Y.; et al. Communication Protocol for using Nontraditional Information Sources between First Responders and Citizens during Wildfires. In Proceedings of the 20th International ISCRAM Conference, Omaha, NE, USA, 28–31 May 2023; pp. 152–165.
4. Zelenka, J, Kasanický, T., Gatila, E., Balogh, Z., Majlingová, A., Brodrechtová, Y., Kalinovská, S., Rehák, R., Semen, Y., Boussu, G., Coordination of drones swarm for wildfires monitoring. In Proceedings of the International ISCRAM Conference. - ISCRAM, 2023, vol. 20, pp. 144-151. ISSN 2411-3387. https://doi.org/10.59297/MUJT3755 (20th Global Information Systems for Crisis Response and Management Conference.
5. Zelenka, J, Kasanický, T., Gatila, E., Balogh, Z., Rehák, R., Koordinácia roju bezpilotných prostriedkov pre monitorovanie lesného požiaru pre podmienky lokality SK a CZ. In Požární ochrana 2023 : Recenzovaný sborník abstraktů XXXII. ročníku mezinárodní konference. - Ostrava : Sdružení požárního a bezpečnostního inženýrství, z.s., 2023, p. 93-96. ISBN 978-80-7385-267-2. (Požární ochrana 2023 : XXXII. ročník mezinárodní konference.)
6. Grunwald D., Wawrzyniak M., Przybyszewski M., Gatial E., Balogh Z., Hassankhani-Dolatabadi S. SILVANUS Dashboard for Wildfire Management, (2024), In Proceedings of the International ISCRAM Conference. - ISCRAM, 2024, 13 pp. ISSN 2411-3387. Available at: https://ojs.iscram.org/index.php/Proceedings/article/view/109 (21st International ISCRAM Conference.) .