



End-to-end Precision Agriculture UAV-Based Functionalities Tailored to Field Characteristics

Emmanuel K. Raptis^{1,2} · Marios Krestenitis² · Konstantinos Egglezos³ · Orfeas Kypris³ · Konstantinos Ioannidis² · Lefteris Doitsidis⁴ · Athanasios Ch. Kapoutsis² · Stefanos Vrochidis² · Ioannis Kompatsiaris² · Elias B. Kosmatopoulos^{1,2}

Received: 24 June 2022 / Accepted: 24 October 2022 / Published online: 27 January 2023
© The Author(s) 2023

Abstract

This paper presents a novel, low-cost, user-friendly Precision Agriculture platform that attempts to alleviate the drawbacks of limited battery life by carefully designing missions tailored to each field's specific, time-changing characteristics. The proposed system is capable of designing coverage missions for any type of UAV, integrating field characteristics into the resulting trajectory, such as irregular field shape and obstacles. The collected images are automatically processed to create detailed orthomosaics of the field and extract the corresponding vegetation indices. A novel mechanism is then introduced that automatically extracts possible problematic areas of the field and subsequently designs a follow-up UAV mission to acquire extra information on these regions. The toolchain is finished by using a deep learning module that was made just for finding weeds in the close-examination flight. For the development of such a deep-learning module, a new weed dataset from the UAV's perspective, which is publicly available for download, was collected and annotated. All the above functionalities are enclosed in an open-source, end-to-end platform, named Cognitive Operations of micro Flying vehicles (CoFly). The effectiveness of the proposed system was tested and validated with extensive experimentation in agricultural fields with cotton in Larissa, Greece during two different crop sessions.

Keywords Precision agriculture · UAVs · Coverage · Remote sensing · Site-specific inspection · Convolutional neural networks · Weed detection

1 Introduction

The course of human history can be characterized as a continuous struggle for food, wealth distribution, and prosperity. The industrial revolution reformed this process by setting the basis for mass and automated production. The technological advancements in the sector of agriculture reflect this constant human labor to guarantee adequate food

resources. Nowadays, increasing the global productivity and quality of food resources remains a challenging field for humanity.

From the scope of economics, the primary sector embraces a wide part of it, especially in the case of less heavily industrialized countries, where agriculture remains a prime activity that leads to income and export revenue. Increasing food production is primarily a prerequisite for human well-being but also economic enhancement. Considering that apart from food resources, agriculture is responsible also for the production of fabric, wood, paper, etc., increasing the overall productivity rate remains a crucial issue from an economic perspective also. However, while the production of all these viable goods is becoming more and more overwhelming due to the continuous population growth [41], the already finite arable land is further being limited by extended farming that deteriorates soil quality and increases the regions unsuitable for cultivation [51]. According to the Food and Agriculture Organization of United Nations (FAO), it is estimated that

✉ Emmanuel K. Raptis
eraptis@ee.duth.gr

¹ Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, 67100, Greece

² Information Technologies Institute, The Centre for Research & Technology, Hellas, Thessaloniki, 57001, Greece

³ iKnowHow SA, Athens, 11526, Greece

⁴ School of Production Engineering and Management, Technical University of Crete, Chania, 73100, Greece

annually around 20%-40% of any agricultural region is lost due to pests and diseases [21]. This cultivation regional loss along with the constant need for increased productivity leads to excessive use of pesticides that clearly threatens human health and pollutes the environment.

The worldwide prevalence of the aforementioned issues emerged a call for immediate action and improvement in the agricultural sector [26], aiming to develop more effective vegetative treatment strategies for enhancing the quality and quantity of crop yields. Towards this direction, utilizing remote sensing technology holds the promise of a feasible solution to the aforementioned challenges. Processing of sensory data based on artificial intelligence and visual analysis can provide propitious, so-called, Precision Agriculture (PA) techniques for monitoring food, fabric, wood, etc. production. These PA techniques focus on building systems for automation of important farming operations such as monitoring the crop's growth and performance, assessing its fertilizer rate and water level, etc. To achieve that, data acquisition from the agricultural fields can provide information to the farmers regarding crop diseases, pest infestations, water stress, and other relevant issues that affect crop productivity. By further processing the acquired sensory data, PA systems aim to correlate production techniques with early crop management decisions to reduce agrochemical use and manage natural resources more efficiently while boosting overall productivity.

However, particular challenges need to be met to allow the uptake of innovative technologies in smart farming to reach its promise [43]. The main challenge when combining innovative technologies in precision agriculture is the standardizing technology and the data synchronization of the involved assets. Operating standards across different technologies may differ as many commercial or non-commercial services adopted in the agricultural sector, have been developed based on a completely different conceptual approach and are intended to operate individually and only for a specific task (e.g., mission planning, image processing, etc.). This lack of interoperability causes monotonous and time-consuming work for the operators since they have to manually feed the output data from one subsystem to another. Another challenging task is the data management plan. Even a small farm has thousands of data from a crop session where information must be gathered to estimate and assess its performance. Attempting to process and understand these data, without a field service management system to store them, whether that is daily, monthly, or seasonally, can be overwhelming for both agriculturalists and farmers. Last but not least, the lack of scalability of many robot-related services, which are designed only for scanning operations, are sure to end up with limited energy autonomy while operating across hundreds of acres [9]. For

precision agriculture to take hold, any technological tool that is expected to assist farmers must be scalable to the size of their needs, expectation, and farming operations. Conceptually, any agricultural software which is integrating robotics and computer vision techniques must have a common holistic view of agricultural processes enhancing crop yield management decisions by assisting even non-expert users in proper manipulations.

In this direction, utilizing intelligence devices such as Unmanned Aerial Vehicles (UAVs) have expanded the capabilities of PA techniques by enabling flexibility, wieldy and adjustable to the case requirements solutions. Based on that, integrated end-to-end platforms have been developed to automate the procedures of UAV navigation, data collection, and processing for knowledge extraction and decision support. Thanks to such platforms, common-commercial UAVs have started to bring the agriculture production systems into economic and environmental benefits, making productivity in the primary sector more sustainable. In the following subsection, an overview of the important and recent related works regarding end-to-end platforms in the field of agriculture is presented.

1.1 Related Work

Recent technological developments, a scale-up in sensor and UAV production with a simultaneous decrease in cost, have given rise to a booming startup industry. While in the beginning applications were limited to military use, soon enough a large number of industrial and civilian use cases emerged, one of them being precision agriculture.

According to a recent report [11], the agriculture drone market will be worth \$5.19 Billion by 2025, growing at an impressive Compound Annual Growth Rate (CAGR) of 31.1% from 2019. An overview and comparison among the most significant commercial agricultural products in terms of their farming technological features can be seen in the upper section of Table 1. Pix4D [35], DroneDeploy [16], and Sentra FieldAgent [5] are commercial drone mapping software which incorporate digital farming technologies, emerging at precision agriculture and farming development. They can produce UAV flight paths, capture aerial photographs, visualize health indices, as well as provide a timeline view of previous field scans for continuous monitoring purposes. Agisoft Metashape [3] is advertised as a product for creating 3D maps using photogrammetry, including crop yield and plant health maps. Botlink [12] is a UAV-based agricultural software with an integrated flight planning framework that can provide high-definition 2-D and 3-D outputs as well as vegetation indices. While Blue River technology [47] is not a UAV technology company, but rather a provider of smart farm machines to manage crops at a plant level, their product is mention-worthy, as it uses computer vision and deep learning techniques to

Table 1 Competitiveness matrix listing commercially available products and research works, including this work (CoFly), comparing them with respect to different features

Application name	Flight planning	Health indices	Timeline view	Deep learning for weed detection	No-fly zones	Site-specific mission
Pix4D [35]	✓	✓	✓	✗	✗	✗
Drone Deploy [16]	✓	✓	✓	✗	✗	✗
Sentera FieldAgent [5]	✓	✓	✓	✗	✗	✗
Agisoft [3]	✗	✓	✗	✗	✗	✗
Botlink [12]	✓	✓	✗	✗	✗	✗
Blue River Technology [47]	✗	✗	✗	✓	✗	✗
Christiansen et al. [15]	✓	✗	✗	✗	✗	✗
Primicerio et al. [37]	✓	✓	✗	✗	✗	✗
Uto et al. [48]	✓	✓	✗	✗	✗	✗
Vasudevan et al. [49]	✓	✓	✗	✗	✗	✗
Zheng et al. [53]	✓	✓	✗	✗	✗	✗
Karatzinis et al. [29]	✓	✓	✗	✗	✓	✗
CoFly	✓	✓	✓	✓	✓	✓

monitor each plant in the field individually, a feature which integrated UAV products are currently lacking.

Research-wise, a large number of UAV-based applications have been developed, for a variety of agricultural practices, such as crop health and yield monitoring (second half of Table 1), making up the majority of agricultural applications due to the ease of use, portability, and economic viability compared to available commercial products, while others incorporate crop spraying techniques [19], [20], which are rather limited owing to the lower payloads that most autonomous UAVs can carry. According to a recent review paper [39], most of these developed systems use commercial software for 3D photogrammetry, visualization, and health index calculation (Pix4dMapper [35], Agisoft Metashape [3]). In terms of contribution, these works mostly focus on improving specific workflows; for example, estimating the overall plant volume and the soil surface for specific crop parcels [15]; monitoring and recording the reflectance of the vegetation canopy [37]; estimation of chlorophyll levels in rice paddies by evaluating vegetation indices derived by hyperspectral data [48]; inspecting periodically the status of crops while obtaining multiple images of them and calculating various indices, as well as pH level and acidity [49]; evaluating the capabilities of a hyperspectral camera in identifying the nitrogen status in rice crops [53]; monitoring and mapping vineyards as well as identification of discontinuous crop rows based on image stitching and health index computation [29].

From a motion planning view, a great majority of bespoke systems use ROS framework [38], while others use Ardupilot Mission Planner software [8], Pix4DCapture

[35] or any other software compatible with the drone. While the aforementioned UAV flight planning software is broadly used and open-source, they do not offer capabilities that fit each field's characteristics such as no-fly zones and automatic UAV-based weed detection. A deeper look into available open-source UAV flight controllers and flight simulators [17], reveals that there are ample options for flight control and mission planning [4, 6, 30] however, none of these extend their capability beyond that to offer data post-processing capabilities.

Overall, several research works include remote sensing capabilities for a variety of UAV-based applications in precision agriculture. Despite their effectiveness and usefulness, the main drawback of these systems is the fact that are intended for specific tasks and not for a holistic view of agricultural processes. For example, some of them focus more on flight planning and health index calculation without having an integrated system including a friendly user interface that eases and automate the utilization of the UAVs [29, 37, 48, 49]. Some others utilize separate commercial software for UAV planning in terms of data acquisition and crop health index computation, making their customization in agriculture applications difficult as they require some expertise in order to be consolidated and manageable as an overall system [15, 53]. Besides, utilizing commercial products might be non-affordable as some of them establish a price according to daily or monthly use. Finally, almost none of the aforementioned commercial or research works incorporate no-fly zones inside the area of interest but support only generic coverage missions which are universally applied to all crop fields (no site-specific missions).

1.2 Contributions

In this work, we propose a holistic approach focusing in a high-level, user-friendly, and centralized platform developed to perform precision agriculture practices based on UAV functionalities, optimized for real-life use and specifically tailored to each field's characteristics. The proposed platform aims at enhancing the software-dependent system abilities in various robot-related research areas while the hardware requirements are kept at the most significant so that the cost could be kept at low levels.

In a nutshell, the overall workflow of the proposed platform is illustrated in Fig. 1 and consists of the following steps.

1. *UAV flight planning*: Initially, a Region of Interest (ROI) with a possible no-fly zone along with the UAV's flight planning parameters is defined. Given this user-defined information, the CoFly framework extracts an automated mission for the drone mapping project where it can be monitored live through the smartphone.
2. *Computation and display of health indices*: Each frame per second an aerial image taken by the UAV is uploaded to CoFly software for processing. When the mission ends, all the collected visual data are merged and provide an orthomosaic as an accurate map of the ROI. Given the orthomosaic, several Vegetation Indices (VIs) maps for assessing crop health are available to display.

3. *Site-specific mission*: CoFly software classifies the available VI maps and estimates the center of the problematic areas which are considered essential for further processing and marked as Points of Interest. On top of that, an automated site-specific mission with low altitude focused on site-specific treatment and weed detection is designed to provide a more thorough inspection of these problematic locations.

Although several UAV platforms are available and capable of providing high-end, informative representations of the field, all of them are severely restricted by the hardware (mainly battery capabilities) of the deployed UAV, leading to limited energy autonomy and operational time. CoFly system aims to alleviate this drawback by utilizing an extra automatic way of creating also precision UAV flights considering the extracted field-specific knowledge (site-specific mission). In other words, the idea of this added feature is to avoid an over-detailed scanning of the whole field, but rather to combine a quick overview with cognition capabilities to accommodate solutions and alleviate the drawbacks of the UAV's limited battery life while operating across hundreds of acres improving, thus, the operating time and energy efficiency of the monitoring process. More specifically, the novelty of this paper lies in a holistic framework for precision agriculture integrating the following unique characteristics and custom software solutions:

- Low-cost deployment by using off-the-shelf drones,

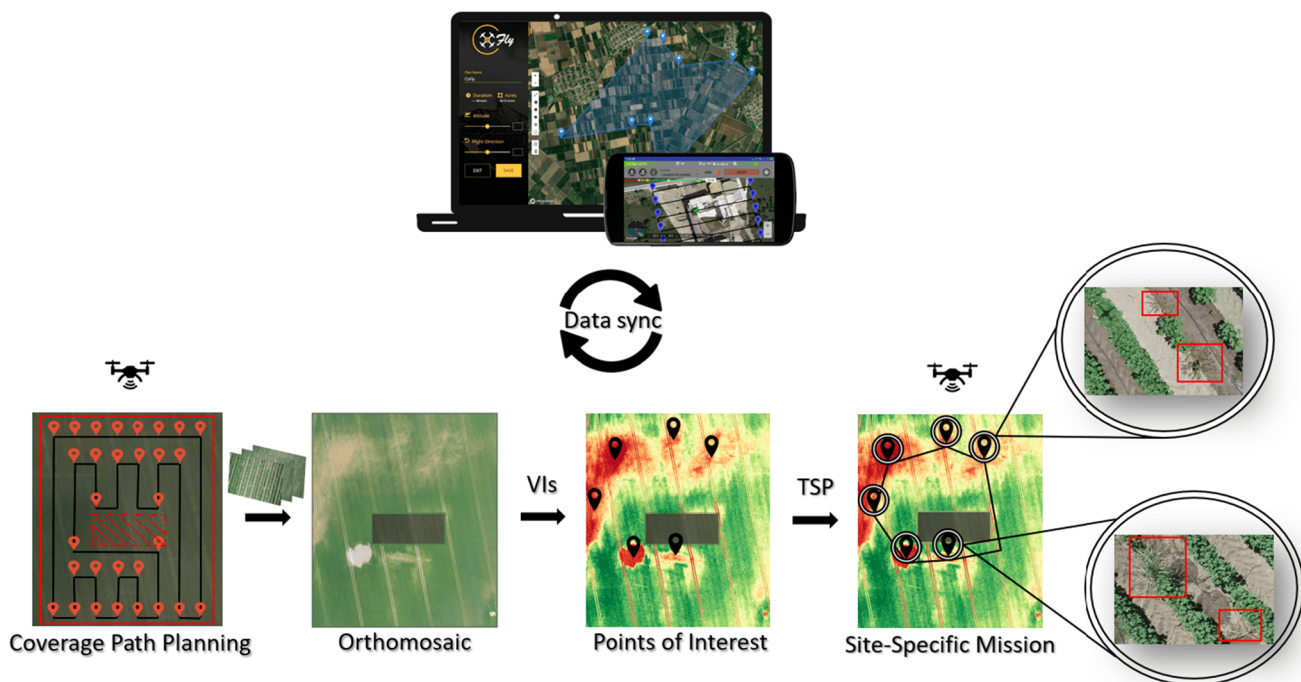


Fig. 1 CoFly's high-level architecture

- Easy-to-use and compatible with any commercial UAV,
- Mobility and data synchronization across devices,
- UAV built-in flight planning for automated crop monitoring,
- Integrated image processing functionalities for ortho-mosaic extraction and vegetation health estimation,
- UAV built-in site-specific mission based on the extracted knowledge and combined with incorporated deep learning model for weed detection,
- No-fly zones/Obstacle avoidance,
- Timeline view for storing information according to the year's harvest.

The proposed platform aims at providing the operator/farmer with a cost-effective and end-to-end integrated system to accomplish autonomous tasks that were used to be completed manually (e.g. crop growth monitoring). Besides, it will play a supportive role in decision-making objectives (e.g. possible weed infestations) enhancing crop yield management decisions by assisting non-expert users in proper manipulations. A comparison of the main features, of all available platforms, as long as the advantages of the CoFly approach are presented in Table 1.

To the best of our knowledge, CoFly is the first to integrate (i) mission planning supporting no-fly-zones and obstacles, (ii) plant health index calculation-visualization, (iii) weed detection using deep learning, with (iv) a user-friendly and intuitive graphical user interface. All of the aforementioned services are build upon modules of enhanced and properly modified well-established methodologies, providing unique characteristics toward a fully functional end-to-end field service management system.

Our work is an open source alternative, directly comparable to commercially available platforms, aiming to maximize the usage of PA by non experienced in UAV flight end-users. At the same time, we offer the research community a tool to develop and test custom solutions for specific agricultural applications. The overall CoFly ecosystem is open-source and publicly available¹ to the community.

1.3 Paper Structure

The rest of the paper is organized as follows. Section 2 presents the translation of a general coverage motion planning framework to an optimization problem and describes the developed algorithm for tackling such a problem. In this section, a simulated evaluation is performed in different scenarios to adequately analyze the

performance of the developed algorithm. Section 3 presents the proposed methodologies for extracting vegetative knowledge regarding the crop's health status along with a pixel-wise pipeline for detecting possible problematic locations. Section 4 describes the proposed motion planning algorithm for dealing with the site-specific inspection and presents the weed detection module by using deep learning methods. Section 5 presents an overview of the graphical user interface and displays the overall workflow for the system's integration. Section 6 includes the real-life experiments along with the visual results and finally, in Section 7, concluding remarks are given.

2 UAV Coverage Path Planning

2.1 Problem Formulation

For the purpose of covering a continuous area, it is essential to deploy a method that provides safe and efficient paths while ensuring complete coverage of the agricultural field. To efficiently cope with challenges faced in real-world scenarios, one should analyze all the aspects that govern Coverage Path Planning (CPP) problems [24]. With the term CPP, we refer to the task of determining a path that passes through all the points of an area while avoiding any No-Fly Zones (NFZs)/Obstacles possible defined within the operational area.

In this paper, the proposed platform deals with the problem of designing a path given:

A. Field-based info:

- a *user-defined polygon* (A) which contains a series of points that define the agricultural field to be scanned,
- a *user-defined sub-polygon* (\mathcal{O}), if needed, representing the NFZs/Obstacles within A .

B. UAV-based info:

- the *scanning density*, in meters, representing the distance between two adjacent flight path lines,
- the *flight direction*, in degrees, defining the direction of the path within A ,
- the *flight altitude* of the UAV for the mission,
- the *speed* of the UAV for the mission.

The objective is to compile all this user-defined information and extract a set of coordinates, representing a safe and efficient path, so as to provide a complete coverage of the A . Note that the defined problem does not assume any type of UAV, but rather adapts to each UAV's capabilities by tuning the UAV-based info (as defined previously) accordingly.

¹<https://github.com/CoFly-Project>

2.1.1 Field Approximation on Grid

To begin with, let us assume that the agricultural field that needs to be covered has an arbitrary shape and is constrained within a two-dimensional polygon A defined by a set of coordinates (x, y) . As a first step, A must be represented as a grid. For the grid approximation, A is decomposed into a finite set of equal cells which do not overlap while filling the entire area of A . The number of these cells represents the required spatial resolution of the UAV's collected images for monitoring crop growth. It should be noted that in the proposed platform, the spatial level resolution can be specified through the values of the *flight altitude* and *scanning density*.

The grid representation of A is described as follows:

$$\mathcal{U} := \{x, y : x \in [1, rows], y \in [1, cols]\} \quad (1)$$

where *rows*, *cols* enumerate the rows and columns after the grid discretization of the terrain to be covered. Apparently, the size of the grid \mathcal{U} is given by $n = rows \times cols$.

During cellular decomposition, we also assume that there are n_o obstacles at the size of a cell placed in several known positions. The set of these obstacles are described as follows:

$$\mathcal{O} := \{(x, y) \in \mathcal{U} : (x, y) \text{ is occupied}\} \quad (2)$$

where each cell that corresponds to any obstructed area on the actual field is marked as occupied. Evidently, the number of cells that need to be covered are reduced to $l = n - n_o$.

Therefore, the set of unoccupied cells that should be covered is described as follows:

$$\mathcal{L} = \mathcal{U} \setminus \mathcal{O} \quad (3)$$

2.1.2 Nodes Placement

Having defined the operational area \mathcal{U} , we proceed by defining the allowed UAV movements, from each grid cell within \mathcal{L} , to fully cover an agricultural field A . To achieve that, an undirected weighted graph $G = (V, E)$ is introduced. Each vertex $V \in G$ is placed in the center of each grid cell of set \mathcal{L} and represented as a node. The edges among these nodes denote the allowed movements of the UAV when navigating inside \mathcal{L} . Generally, each node is connected with four adjacent nodes following the Von Neumann neighborhood [36]. Nodes that are placed on the boundaries of the operational area or are adjacent to an obstacle have less edges.

Definition 1 Two nodes $u_{(x_i, y_i)}, v_{(x_j, y_j)} \in G$ are considered to be *adjacent* if:

$$\|(x_i, y_i) - (x_j, y_j)\| = s_d \quad (4)$$

where s_d defines the value of *scanning density* in meters.

Definition 2 A *valid path* of length m is considered any sequence of nodes

$$\mathcal{X} = \{l_1, l_2, \dots, l_m\} \quad (5)$$

where $\exists E(l_i, l_{i+1}) \forall i \in [1, l_m - 1]$.

In other words, the set of $(V, E) \in G$ within \mathcal{L} provides us with information corresponding to a sequence of adjacent nodes allocated in the center of the unoccupied cells. Consequently, if a series of nodes follows Def. 2 could constitute a *valid path* that passes through all the points of \mathcal{U} while avoiding obstacles. Hence, G to provide a *valid path* specifically tailored to each field's characteristics is also directly related to the characteristic hyperparameter *flight direction*. Modifying the flight direction feature intends to optimize the node's placement inside \mathcal{U} to create a more favorable -in terms of coverage- representation of G .

The key idea is the rotation of each node within the grid \mathcal{U} by an angle ϑ in a counter-clockwise direction around a center-point. Given the desired angle ϑ determined by the *flight direction* value, the rotation of each node is defined by the following equations:

$$x_{new} = c_x + (x - c_x) \cdot \cosh(\vartheta) - (y - c_y) \cdot \sinh(\vartheta) \quad (6)$$

$$y_{new} = c_y + (x - c_x) \cdot \sinh(\vartheta) + (y - c_y) \cdot \cosh(\vartheta) \quad (7)$$

where the following constraints are hold:

- A is defined in a two-dimensional space,
- the center-point c_x, c_y is defined as the center of A ,
- $(x, y) \forall V \in G$ are in the form of Cartesian plane,
- ϑ , which controls the rotation of G over \mathcal{U} , is in range $[0, 360^\circ]$.

Modifying the flight's direction, there might appear more favorable configurations of G inside \mathcal{U} , where the number of nodes within \mathcal{L} is greater. Such configurations are more likely to provide paths that achieve a higher percentage of coverage for the given A . It must be emphasized that the x_{new}, y_{new} variables are rotating with respect to the user's desired *scanning density*.

2.1.3 Optimization Problem

Given the mathematical description presented above, the problem of finding a minimum size of a collection of edges (UAV movements) that cover all the vertices of the graph can be translated into the minimization of the covering path \mathcal{X} , so as,

$$\begin{aligned} &\text{minimize} \quad m \\ &\text{subject to} \quad E \subseteq \{l_i, l_{i+1}\}, i = 1, \dots, l_m - 1 \\ &\quad \mathcal{X} \supseteq \mathcal{L} \end{aligned} \quad (8)$$

where

- m is a positive integer which denotes the cardinality of set \mathcal{X} ,
- $l \in \mathcal{L}$ is the number of unoccupied cells,
- E is a set of edges-movements,
- \mathcal{X} is a *valid path*.

2.2 Navigation Algorithm

Having illustrated all the prime features that constitute a completely safe and efficient path, we proceed to present a navigation algorithm in steps, as sketched in Fig. 2, for solving the optimization problem of (8). The navigation system that we propose in this paper, utilizes the Spanning Tree Algorithm [23] which deploys an off-line coverage path planning algorithm as the UAV covers any arbitrary shape, while knowing in advance any related information about the location of the NFZs/Obstacles. Conceptually, this algorithm along with the equations of Section 2.1 tackles the aforementioned challenges and achieves the overall mission objectives within five distinct steps.

Algorithm 1² outlines in pseudo-code the key steps of the proposed navigation algorithm.

Step 1. As a first step, in lines 1 and 2, the user's input data A and \mathcal{O} are transformed from WGS84-EPGS:4326 into WGS84-EPGS:3857 [44]. For ease of understanding, let us analyze the usage of this transformation.

To start with, EPSG standard codes are used as Spatial Reference System Identifiers for map datasets. EPSG:4326 projects the map dataset in a coordinate system on the surface of a sphere while EPSG:3857 projects the map dataset in a coordinate system on the surface of a square. The reason why this transformation is needed is because A and \mathcal{O} are initially defined in a set of latitude/longitude pairs while *scanning density* is defined in meters. In latitude/longitude (EPSG:4326), distances are not measured in meters, but rather in decimal degrees. To efficiently deal with these challenges, a Web-Pseudo Mercator projection (EPSG:3857) was used to project these latitude/longitude pairs in a square shape which supports a metric system in meters. This suitable projection was used for dealing with the above aforementioned mathematical descriptions of Section 2.1, extract the path points, and project them back to latitude/longitude pairs.

Having transformed the input coordinates A , \mathcal{O} to a Cartesian Plane shape, in lines 3-10, the entire area of A is grouped into large square-shaped cells, each of which is either entirely blocked or entirely unblocked, as shown in Fig. 2(a). It must be noted,

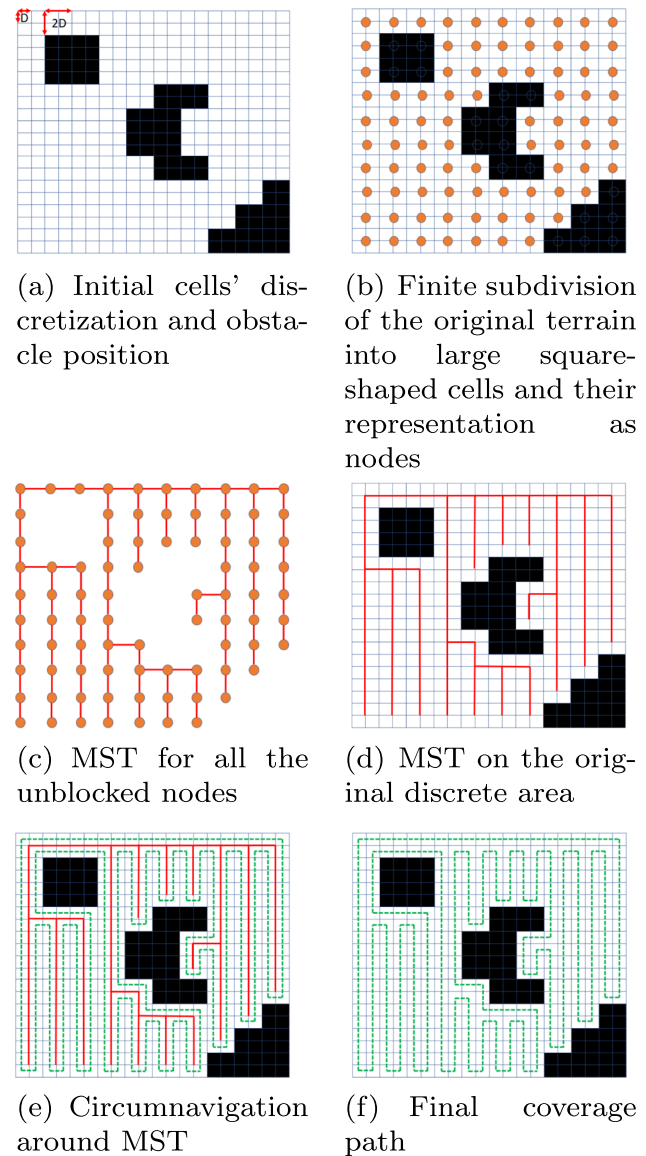


Fig. 2 Spanning-Tree Coverage algorithm main steps and terminology

as the only algorithm's requirement, that the areas which include stationary obstacles within the grid cannot be smaller than a large square-shaped cell.

Step 2. In lines 11-21, after the discretization of the field, every unoccupied large square-shaped cell is translated into a node and for each pair of nodes, a number (the weight) is assigned to each edge. These nodes-edges, regulated by the flight direction feature, form the graph $G = (V, E)$ which defines the allowed movements of the UAV (Fig. 2(b)).

Step 3. As a next step (line 23), a graph minimization methodology by using Kruskal's algorithm [32] is applied in the previously formed graph G as illustrated in Fig. 2(c). The resulting minimum

²<https://github.com/CoFly-Project/Waypoint-Trajectory-Planning>

Algorithm 1 Trajectory planner.

Input: $A = \{Lat_A, Lon_A\}$, $\mathcal{O} = \{Lat_{\mathcal{O}}, Lon_{\mathcal{O}}\}$, *scanning density*, *flight direction*
Output: $\mathcal{X} = \{Lat_{\mathcal{X}}, Lon_{\mathcal{X}}\}$ ▷ Solution of (8)

```

1:  $\{x_A, y_A\} \leftarrow$  Project  $\{Lat_A, Lon_A\}$  from EPSG:4326-to-3857
2:  $\{x_{\mathcal{O}}, y_{\mathcal{O}}\} \leftarrow$  Project  $\{Lat_{\mathcal{O}}, Lon_{\mathcal{O}}\}$  from EPSG:4326-to-3857
3:  $x \leftarrow \min(x_A)$ 
4: while  $x \leq \max(x_A)$  do ▷ Iterate over 2D area
5:    $y \leftarrow \min(y_A)$ 
6:   while  $y \leq \max(y_A)$  do
7:      $y \leftarrow y + \text{scanning density}$ 
8:      $\mathcal{U} \leftarrow (x, y)$  ▷ Grid representation of  $A$  (1)
9:      $x \leftarrow x + \text{scanning density}$ 
10:  $\mathcal{L} \leftarrow$  Discard every cell in  $\mathcal{U}$  according to (2)
11:  $V \leftarrow \{(x, y) \mid (x, y) \in \mathcal{L}\}$ 
12: for each node  $\in V$  do
13:   if Def. 1 is True then
14:      $E \leftarrow \{(u, v, \text{weight} = 0)\}$ 
15:   else
16:      $E \leftarrow \{(u, v, \text{weight} = 1)\}$ 
17:  $G \leftarrow (V, E)$  ▷ Undirected weighted graph
18: if flight direction  $\neq 0$  then ▷ Node rotation
19:   for each  $x, y \in V$  do
20:      $x_{new} \leftarrow (6)$ 
21:      $y_{new} \leftarrow (7)$ 
22:  $\mathcal{U} \leftarrow$  Subdivide every cell  $\in \mathcal{L}$  in four identical subcells
23: MST  $\leftarrow$  Construct a minimal spanning tree for  $G$  by using any minimum-spanning-tree
    algorithm,  $MST \in G$ 
24: A starting cell  $S \in \mathcal{L}$  as the initial position of UAV
25:  $\{x_{sub}, y_{sub}\} \leftarrow$  Traverse MST in a clock-wise direction through subcells of line 22,
    starting from cell  $S$ . Halt when  $S$  is encountered again.
26:  $\{Lat_{\mathcal{X}}, Lon_{\mathcal{X}}\} \leftarrow$  Project  $\{x_{sub}, y_{sub}\}$  from EPSG:3857-to-4326
27: Smooth the trajectory in turns with corner radius

```

spanning tree (MST) contains the minimum number of edges among all nodes (allowed movements) so that G remains fully connected. Figure 2(d) graphically depicts the resulting MST above the original discrete area of Fig. 2(b).

Step 4. The UAV, then, to cover the area of interest circumnavigates the MST [28] in a clockwise direction along the subdivided subcells of line 22, as illustrated in Fig. 2(e). The circumnavigation of the spanning tree (line 25) generates a simple closed path $\mathcal{X} = \{(Lat_1, Lon_1), (Lat_2, Lon_2), \dots, (Lat_m, Lon_m)\}$, where m denotes the number of waypoints, (Fig. 2(f)).

Step 5. Once Algorithm 1 determines a path \mathcal{X} , in line 27, a smoothing hyperparameter called *corner radius* is introduced and applied in turns. Figure 3 illustrates the path smoothing process with the black line indicating the path \mathcal{X} generated in Step 4 while the blue line indicates an *optimal* \mathcal{X} in terms of covering time- solution after the applied smoothing process.

Having defined the navigation algorithm, let us provide some necessary preliminaries about the flight hyperparameters (*flight altitude*, *scanning density*, *flight direction*, *corner radius*, *speed*) and how they affect the overall quality

of aerial mapping. An important analysis of those hyperparameters is analyzed below:

Flight altitude. Low flight altitudes yield the highest resolution and best precision in mapping with more matched features per image. In contrast, flying at a higher altitude covers more space and allows the capture of common unique features in multiple images which can be useful when mapping areas with homogeneous imagery. Consequently, *flight altitude* determines the

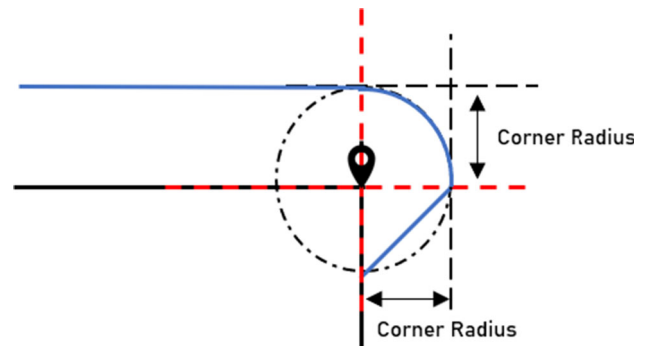
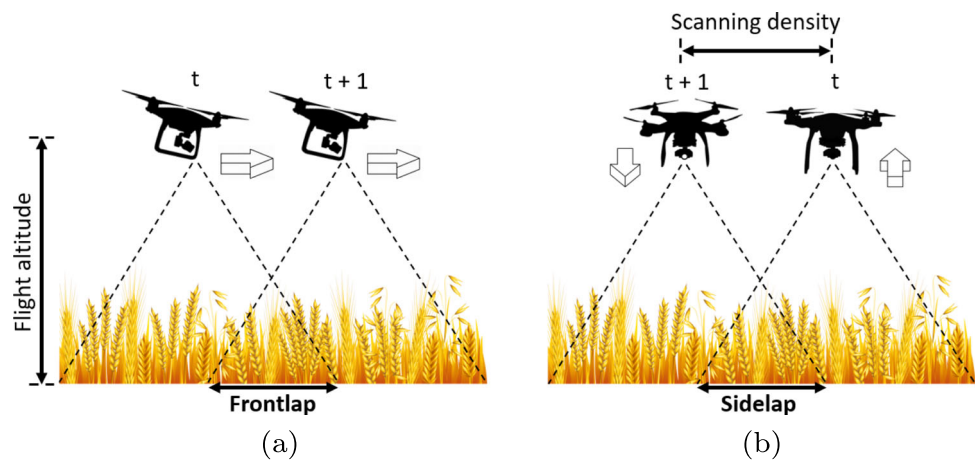


Fig. 3 Path smoothing process regulated by the hyperparameter *corner radius*

Fig. 4 Overlap settings regulated by the hyperparameters *flight altitude* and *scanning density*. Subfigure (a) illustrates the overlap between two sequential images on the same flight path line. Subfigure (b) illustrates the overlap between two sequential images in adjacent parallel flight path lines



overlap between two corresponding images in the same flight path line as sketched in Fig. 4(a) (*frontlap*).

Scanning density. In the proposed CPP method, the distance between two sequential trajectories is determined by the value of *scanning density*. As a result, this distance can control the overlap between images from two adjacent flight path lines as sketched in Fig. 4(b) (*sidelap*).

Flight direction. Using the flight direction feature allows you to change the direction that your drone flies and ensure that the extracted path pattern is tailored to the field. Figure 5 illustrates an example where the flight direction feature is applied to a specific *complex-shaped* polygon.

Corner radius. Speed deviation and 90° degree turn on the spot maximize the flight time of the mission and can lead in many cases to inefficient missions. Controlling indirectly the *corner radius* hyperparameter minimizes the flight time by allowing smooth turns where the UAV, during a turning point, does not make a 90° degree turn on the spot but rather moves along the perimeter

of a circle with a radius equal to the value of the *corner radius*. Note that the value of the *corner radius* should fluctuate at low levels e.g. 0.5-1m to minimize the aircraft's deceleration at the turning points and at the same time avoid deviating from the resulted coverage path \mathcal{X} .

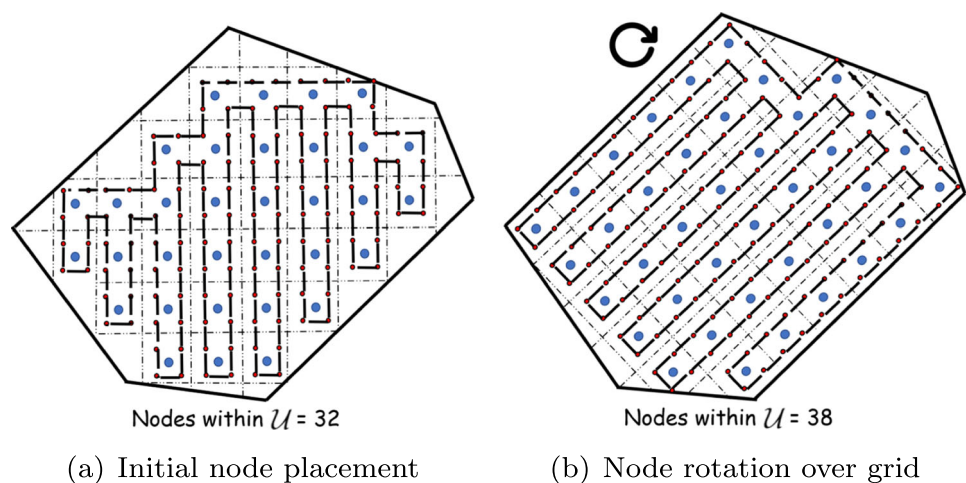
Speed. Last but not least, *speed* controls the speed at which the UAV flies during a mission. The UAV's flight speed varies according to the flight planning parameters and image overlap settings.

Adjusting the aforementioned hyperparameters to the crop yield can lead to an optimized setup by generating a path pattern tailored to each field's characteristics and optimize the chance of generating high-resolution maps.

2.3 Simulated Evaluation

To assess the performance of the proposed navigation algorithm to the quantity of coverage and the overall quality

Fig. 5 Flight path direction regulated by the *flight direction* hyperparameter based on node setup



of paths, two different metrics are considered. The first metric *Percentage of Coverage (PoC)* is for the accuracy in terms of coverage and the second metric *Estimated Flight Time (EFT)* is for the efficiency in terms of coverage time.

The metric *PoC* intends to represent the percentage of coverage directly matched to a path \mathcal{X} within A . Given the problem definition, *PoC* is defined as follows:

$$PoC = \frac{n_A}{n_{A_{\max}}} \times 100\% \quad (9)$$

where n_A denotes the number of nodes placed inside a polygon A which will be used for the MST construction and $n_{A_{\max}}$ denotes the theoretically maximum number of nodes that could fit inside a given polygon A . Thus given A and *scanning density*, $n_{A_{\max}}$ is formulated as follows:

$$n_{A_{\max}} = \frac{A \text{ (m}^2\text{)}}{\text{scanning density}^2 \text{ (m}^2\text{)}} \quad (10)$$

where A is defined in m^2 and *scanning density* is defined in *meters* in the x-axis and the y-axis, respectively (Algorithm 1, lines 7 & 9).

The second metric *EFT* intends to represent the estimation time, in minutes, for the UAV to follow a path. As to covering time, the *EFT* is determined by the simulation time.

RotorS UAV [22] simulator framework was utilized to test and evaluate the proposed navigation algorithm. RotorS is an open-source, simulator for unmanned aerial vehicles based on the Robot Operating System (ROS) and the Gazebo 3D robotics simulator. All the experiments were carried out in the waypoint-navigator package [34] of the aforementioned simulator, which is a high-level waypoint-following package that simulates a trajectory through a list of waypoints formatted in WGS84-EPSC:4326. For the representation and visualization of the trajectories we used the RViz (Robot Visualization) program [45], an auxiliary tool for visualizing our waypoint data in a real domain. RotorS was hosted in a common laptop (Intel Core i7-8750H CPU) with Ubuntu 16.04 LTS and a combination of v7.7/Kinetic as a version of Gazebo and ROS package, respectively.

For the evaluation of the proposed method, a set of 6 generated polygons is used. This set includes *complex-shaped* and *non complex-shaped* polygons, with or without obstacles. The simulations were carried out with the default UAV model of RotorS equipped with GPS and IMU sensors. Moreover, it is assumed that the UAV is initially located at (Lat_1, Lon_1) of each simulated trajectory \mathcal{X} . For all sets, the proposed method was executed with fixed *flight altitude* and *speed* of 100 m and 3 m/s, respectively. Additional information associated to each scenario is presented in Table 2. It should be noted that each scenario was designed in such a way to demonstrate the performance of Algorithm 1 and test the effect of the *flight direction* and *corner radius* hyperparameters on the resulting path in terms of coverage percentage and time. The rest of the hyperparameters were not evaluated in RotorS as they do not directly affect the path pattern but rather the image overlap settings.

For each scenario, Algorithm 1 was employed to calculate the respective UAV path to solve optimization problem (8) aiming at maximizing the *PoC* while minimizing the *EFT*. The visualization of the produced trajectories \mathcal{X} for each scenario along with their corresponding performance is presented in Fig. 6 and the last two columns of Table 2.

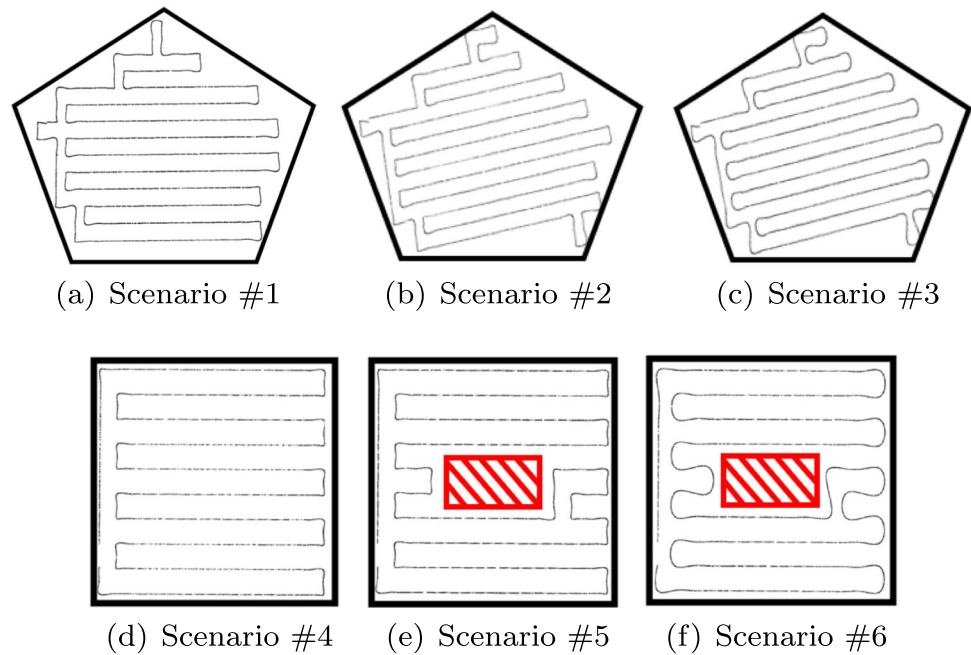
The results of the evaluation procedure are reported and analyzed in the following paragraphs.

In Scenario #1, 31 nodes with *flight direction* = 0, are selected for the MST construction as important regions of interest for the UAV. In Scenario #2, by only modifying the flight's direction, the navigation algorithm finds a more profitable -in terms of coverage- configuration of \mathcal{U} inside A with 32 nodes selected as important regions of interest. As presented in the last column of Table 2, the resulting path \mathcal{X} of Scenario #2 in comparison to Scenario #1 is more *optimal* in terms of coverage. However, due to the increasing number of waypoints, the flight time of Scenario #2 is greater than Scenario #1. Hence, Scenario #3 is considered to demonstrate the effectiveness of smoothing the trajectory in turns. As presented in the *EFT* column of Table 2, by tuning the *corner radius*, the overall flight time of the resulting path \mathcal{X} of Scenario #3 is decreasing in contrast to Scenario #2, providing thus the exact percentage of coverage. While Scenarios #1,#2 and #3 are

Table 2 Evaluation metrics based on scenarios description

Scenario #	Area (acres)	NFZs/Obstacles (acres)	Scanning density	Flight direction	Corner radius (m)	EFT [22]	PoC
1	7.39	0	30	0	0	8:31.394	93.31%
2	7.39	0	30	20	0	8:94.510	96.32%
3	7.39	0	30	20	0.5	8:44.197	96.32%
4	10.65	0	35	0	0	10:88.281	99.48%
5	10.65	0.65	35	0	0	10:11.192	96.96%
6	10.65	0.65	35	0	0.5	9:53.855	96.96%

Fig. 6 Path visualizations in RViz. Effect of the hyperparameters *flight direction* & *corner radius* on the resulting path according to each scenario description



demonstrated in *complex-shaped polygon*, in Scenario #4 when in *non-complex polygon* even without optimizing the nodes placement within \mathcal{U} (*flight direction* = 0), Algorithm 1 achieves a high percentage of coverage. In addition, the effectiveness of Algorithm 1 is also demonstrated in Scenarios #5 and #6. In both Scenarios, 27 nodes were selected for the MST construction while 5 nodes were considered to contain NFZs/Obstacles. For both scenarios the resulting path either with smoothing or not achieves an average *PoC* above 96%.

Overall, the developed algorithm designs the UAV paths, in such a way as to achieve high *PoC* both in complex and non-complex fields while keeping the flight time as minimum as possible, enabling the prolonged utilization of the UAV in the agricultural application.

3 Location Estimates for Crop Health Monitoring

To maximize the system's usability and incorporate novel and cognitive functionalities, the framework involves sequentially executed processes that eventually improve the crop monitoring process. After the execution of the generic coverage mission presented in Section 2, a set of RGB georeferenced images covering the whole field has been collected. This set comprises the base of the subsequent processing procedures aiming at i) extracting knowledge regarding the crop's health status and ii) providing identified locations over the field that correspond to the centers of detected problematic areas. The following sub-sections analyze the integrated procedures.

3.1 Image Stitching Process

The aim of the current module is to merge the collected visual data and provide an orthomosaic as an accurate map of the examined area. Towards this direction an appropriate tool for image stitching has been deployed which is based on the following process:

1. Firstly, the telemetry from the metadata of each image is extracted in order to speed up and improve the accuracy of the stitching process.
2. Then, a sparse point cloud is generated with the Open Structure from Motion (OpenSfM) [2] library.
3. The generated point cloud is transformed to triangle meshes via the Poisson Surface Reconstruction [10] method.
4. The triangle meshes are textured over by selecting appropriate patches from the acquired images generating the georeferenced orthoimage.

The above process can be implemented with the OpenDroneMap (ODM) [1] toolkit. In Fig. 7 the generated orthophoto of an examined area is demonstrated. The outcome of the aforementioned stitching process constitutes a comprehensive map-quality representation of the field and can be further processed as a whole, to extract qualitative features, such as an estimation of the vegetation health, and provide the end-user an overview of the crop status.

3.2 Vegetation Indices (VIs)

To enhance the capacities of the proposed framework toward a cognitive precision agriculture system, the previously

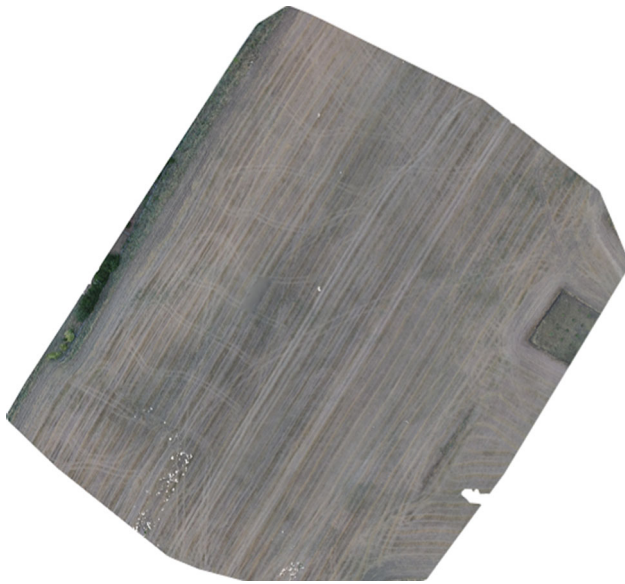


Fig. 7 Extracted orthomosaic map of an examined area

defined stitched map of the under-study field is further processed to estimate the vegetation's health. Considered as the main objective and motivation of the proposed system, crop health estimation prerequisites the utilization of such metrics. These estimates, widely known as Vegetation Indices (VIs), are strictly dependent on the available sensor that is equipped on the deployed UAVs. Relevant sensors involve operations under either the visual or a wider light spectrum which eventually is used to identify defects and problematic crop areas.

The use of a multi-spectral camera would provide accurate and more meaningful data for estimating plant health via the calculation of some pertinent VIs. Despite their widest exploitation and their increased performance in precision agriculture applications, deploying a multi-spectral sensor will significantly increase the total cost while technical restrictions might be inserted such as a considerably lower acquisition rate. On the other hand, employing off-the-shelf UAVs utilizing RGB cameras consists of a low-cost and easy-to-deploy solution. To this

end, VIs focused on the visual spectrum were incorporated as the usage of a visual camera is more flexible and affordable. Therefore, multiple VIs were adopted and developed to cover the majority of aspects that are related to crop health. The employed RGB-based VIs are presented in Table 3.

Each one of the four selected VIs represents the actual reflectance of the field's vegetation in different color bands and thus, it can reflect different measures of crop health. More specifically, the Visual Atmospheric Resistance Index (*VARI*) [25], originally designed for satellite imagery, considers the blue light scattering as the blue spectrum travels in smaller wavelengths. On the contrary, Green Leaf Index (*GLI*) [33] was proposed to measure wheat cover and all main color bands are considered in a weighted approach. Similarly, Normalized Green-Red Difference Index (*NGRDI*) [27] is obtained by calculating the reflectance of the green and the red zone of the electromagnetic spectrum. A similar approach is adopted for the Normalized Green-Blue Difference Index (*NGBDI*) [52] nonetheless, the estimation relies on the reflectance of the green and the blue zone. These VIs were selected as an established choice capable to reflect and represent the under-study agriculture-related issues.

The estimated VI maps are defined within a limited range and displayed to the operator by utilizing a red-green color-map by using the appropriate scale, as demonstrated in Figs. 8(a)-(d). Since each calculated VI reflects different aspects of the visual light that is incorporated, the corresponding visualization for every VI is unique as different aspects of plant health are assumed.

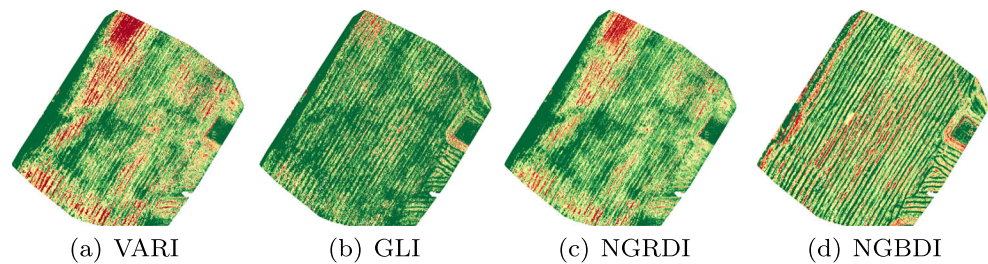
3.3 Location Extraction of Problematic Areas

The extracted VIs image representations can provide the end-user, with a concrete overview of the crop's health. Yet, to automate the procedure, individual field regions that present poor conditions, in terms of vegetation health, have to be defined. This module aims to detect these problematic areas and feed their location to the site-specific mission module for a targeted, on-the-spot, inspection.

Table 3 Vegetation indices (VIs)

Vegetation index	Name	Formula	References
VARI	Visible atmospheric Resistant index	$\frac{G-R}{G+R-B}$	[25]
GLI	Green leaf index	$\frac{2 \times G - R - B}{2 \times G + R + B}$	[33]
NGRDI	Normalized green red Difference index	$\frac{G-R}{G+R}$	[27]
NGBDI	Normalized green blue Difference index	$\frac{G-B}{G+B}$	[52]

Fig. 8 Image representations of the four calculated VIs. Lower index values are colored with red tones while higher index values correspond to green tones



The significance of VIs is not only their ability to visualize the health status of the crop but also the quantification of this status by assigning a single value to each pixel of the processed image, usually, in the range $[-1, 1]$. Based on that, a pixel-wise processing pipeline has been developed to detect these regions where the corresponding index value is low and thus can be considered problematic. The deployed module consists of the following processing steps:

1. Outlier detection
2. Binary thresholding
3. Connected components detection
4. Centers of mass calculation
5. Centers aggregation

Although the typical range of the presented VIs is $[-1, 1]$, in real-world cases the actual range can differ significantly and values close to the typical extrema can be considered outliers. Towards this direction, initial filtering of the index array outliers takes place. At first, the histogram of the index array is calculated. Assuming that k_1 is the first histogram's bin corresponding to the interval $[-1, -0.9]$, in case the frequency of this bin is smaller than 5% of N , where N is the number of the pixels corresponding to the entire field region, then the values belonging on the specific bin range are considered outliers and -0.9 is the actual minimum (notated as \min) of the processed index array. The process is repeated for the next bin until the outlier criterion is not satisfied. A similar approach is applied to estimate the maximum value (\max).

Next, binary thresholding is applied to separate the problematic areas from the healthy ones. As problematic areas are considered those with pixel values in the range $[\min, t]$, where t is the binary threshold equal to $t = \min + 0.25(\max - \min)$. Although it is established that t is equal to the quarter-half of the VI's actual range, one could fine-tune this value manually to achieve better, crop-oriented performance. In Fig. 9(a) the corresponding binary image acquired by thresholding the VARI index of Fig. 8(a) is depicted. Afterward, a connected components analysis is employed to detect connected components on the binary image that correspond to a single problematic area. For

every detected area, the center of mass is calculated leading to a set of points, as presented in Fig. 9(b).

These center points are considered the destination points where the site-specific mission has to take place, to provide a more thorough inspection regarding the low health conditions of the specific area. Yet, a large number of waypoints is usually prohibitive, since the flight time of a conventional drone is not adequate to operate in all these individual areas. Furthermore, there are cases where neighboring problematic areas can be considered as parts of a wider area that can be examined with a single pass of the UAV asset. To overcome this issue, the k-Means clustering algorithm is employed, aiming to group the center points of smaller individual problematic regions and lead to a more manageable number of destination points. Focusing on an autonomous operation, the number of clusters k is selected automatically in the range $[2, 10]$ based on the maximization of the Calinski-Harabasz score [13]. Furthermore, to acquire more meaningful results, a weighted clustering approach is selected, where the cluster centroids are calculated with respect to the size of each area. Figure 9(c) illustrates the outcome of the k-Means clustering. The acquired cluster centers are forwarded to the site-specific missions module, as the points of interest where a further on-spot examination will be conducted. Moreover, the ability to fine-tune or exclude destination points is provided to the user, to create a more tailored site-specific mission that meets his requirements.

As mentioned in Section 3.2, the employment of RGB-based indices aims to compose a low-cost and flexible framework. However, note that the presented approach for the identification of the problematic areas can be applied in the case of multispectral indices without any modifications and thus, it can be considered a “plug-and-play” solution.

4 Site-Specific Mission

Towards assessing the aforementioned problematic locations, a site-specific mission focused on the detected points of interest is designed. The main objective of this mission is to provide a more thorough inspection of these individual

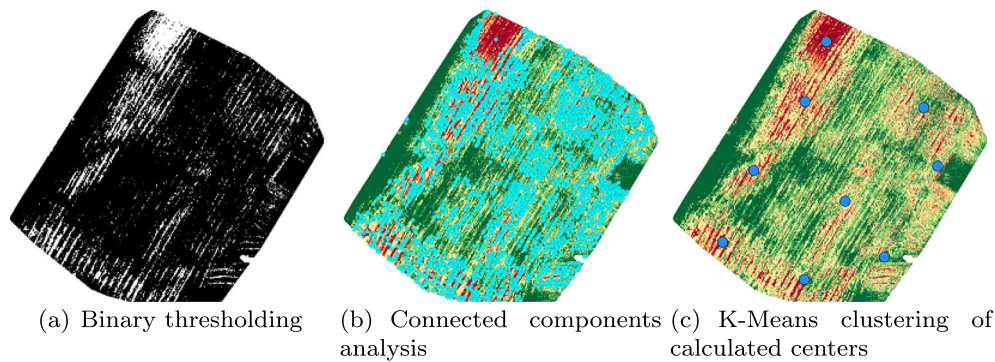


Fig. 9 Demonstration of the processing pipeline for the problematic areas identification. Initially the index array is (a) binary thresholded, then (b) connected component analysis is conducted to detect the

center of mass of every problematic regions, finally (c) K-Means clustering is employed to estimate the centers of wider problematic areas

spots of the examined area that seem to be problematic in terms of plant health and provide valuable information to the end-user. To efficiently monitor these individual spots, visual footage from lower altitudes and different camera angles is captured aiming to visually detect crop health deterioration sources, such as weed plants, to enable site-specific treatment.

4.1 Hot-Point Mission

From the motion planning perspective, visiting each of the individual points of interest can be deduced to the problem of finding the shortest path that visits all nodes (Fig. 9(c)). For ease of understanding, let us assume that the crop health indicators (VIs) of section III extract k points of interest $\{1, 2, \dots, k\}$ each of one is expressed as $\chi_i = \{Lat, Lon\}$, where $i \in \{1, 2, \dots, k\}$. Overall, the set of points of interest is defined by the following equation:

$$\chi := [\chi_1, \chi_2, \dots, \chi_k] \quad (11)$$

At this point, the path planning problem is formulated into the *optimal* round trip that will guarantee the shortest possible route while passing through all points, only once. For individual coverage, we applied the Travelling Salesman algorithm (TSP) [7], which is renowned for finding the shortest path within an undirected graph. The pseudo-code for monitoring each χ_i is given in Algorithm 2³.

Given the takeoff/landing point and the overall set χ , in lines 1-6, we treat the defined points as an undirected stationary graph representing each point within the examined area as a vertex. The distance between each pair of vertices is defined by the weight of their corresponding edge.

In line 7, by solving the TSP algorithm an *optimal* path

that visits every vertex is defined. In lines 9-10, every point - vertex of the resulted *optimal* path is labeled as *Hot-Point*. As a next step (lines 11-15), we generate a set of m additional points with an angle $\vartheta \in [0^\circ, 360^\circ]$ and a constant radius r around a specified *Hot-Point*. Note that, the number of points m that enclose a *Hot-Point* depends on an angle φ (line 13) which determines the angular distance between any two consecutive points on the perimeter of the circle. The smaller the value of φ , the greater the number of points on the perimeter of the circle. Figure 10(a) illustrates an indicative example of a circular flight that encircles a *Hot-Point*.

Based on Algorithm 2, the UAV will perform a fixed radius cycle around the central points of interest (*Hot-Points*) as it follows the resulted path of the TSP algorithm as sketched in Fig. 10(b). It should be highlighted that the resulting path starts and ends at a specific vertex (line 1) which denotes the takeoff/landing waypoint after having encountered all the other vertices of the graph, exactly once. Moreover and if needed, during the execution of the mission, the user can also use the physical remote control to modify the speed, heading, and altitude of the aircraft.

4.1.1 A Hot-Point Approximation Study on Flight Time for UAVs

This study aims to estimate the UAV's mission time and analyze its association with the number of points of interest. For the evaluation study, 10 χ sets are used. Every set includes a different number of distinct points of interest. For each set, 100 simulation experiments, with a random-shaped distribution of points of interest, were performed within an area of 10.77 acres (Fig. 10(b)). All sets were executed with a fixed radius r of 5 m and an angle $\varphi = 45^\circ$. The simulations were carried out with a fixed altitude and speed of 10 m and 3 m/s, respectively. The overall performance of Algorithm 2 in terms of covering time regarding each set

³<https://github.com/CoFly-Project/Waypoint-Trajectory-Planning>

Algorithm 2 Hot-point mission.

Input: takeoff/landing $\{Lat, Lon\}, \chi, r, \varphi$
Output: An *optimal* circular path around points of interest

```

1:  $V[0] \leftarrow$  takeoff/landing waypoint ▷ Starting/ending
2: for each  $i \in \{1, 2, \dots, k\}$  do
3:    $V \leftarrow \{(Lat, Lon) \mid (Lat, Lon) \in \chi_i\}$  ▷ Create fully connected graph
4: for each  $i \in \{1, 2, \dots, k\}$  do
5:   for each  $j \in \{1, 2, \dots, k\}$  do
6:      $E \leftarrow \{(i, j, \|\chi_i - \chi_j\|.meters)\}$  ▷ Distance between points of interest
7:  $TSP \leftarrow TSPSolver(V, E)$  ▷ TSP solution [7]
8:  $count \leftarrow 0$  ▷ Enumerate Hot-Points
9: for each  $\chi \in TSPSolver$  do
10:   $Hot-Point[count] \leftarrow \{(Lat, Lon) \mid (Lat, Lon) \in \chi\}$ 
11:   $CenterPoint \leftarrow Hot-Point[count]$  ▷ Define center
12:   $count \leftarrow count + 1$ 
13:  for  $(\vartheta \in 0^\circ, 360^\circ, step = \varphi)$  do ▷  $\varphi \in (0^\circ, 360^\circ)$ 
14:     $Hot-Point[count] \leftarrow$  Generate points uniformly within a circular region with the
      analogous  $\vartheta$  and a given radius  $r$  around  $CenterPoint$ .
15:     $count \leftarrow count + 1$ 
16:    Halt when circle is formed.
17:    Halt when takeoff/landing waypoint is encountered again.

```

of χ is presented in Fig. 11.

According to Fig. 11, each of the sets χ indicates that the estimation coverage time varies and depends on the geographical location (latitude, longitude) of each of the points of interest within the field. Although, the key outcome of this study is that the developed algorithm, for each of the 100 simulations, achieves a persistent coverage of 3–30 points of interest within an estimation coverage time close to the average flight time of most common commercial UAVs, i.e. 25–30 min. Such a feature is of paramount importance, as it supports trajectories for covering a large number of individual spots while concurrently respecting the average energy efficiency of a conventional UAV. Note that, the *Estimated Flight Time* on the y-axis could be achievable only if the user does not intervene during the execution of the Hot-Point mission. However, when

users use the remote controller to adjust the flight status of the UAV, covering time is increasing or decreasing proportionally to the duration of each action. As a result, the Hot-Point mission as a feasible approach for site-specific inspection can be applied to precision agriculture practices for automated visual footage focusing on low-cost smart farming by utilizing common commercial UAVs.

4.2 High Level Semantic Information Extraction

During the Hot-Point mission, a closer inspection by the end-user of the problematic field areas is enabled through the captured visual footage. Yet, artificial intelligence can be employed to automate this process and compose the basis of a decision support system. Towards this direction, a weed detection module is deployed to analyze the visual

Fig. 10 An *optimal* circular path around points of interest

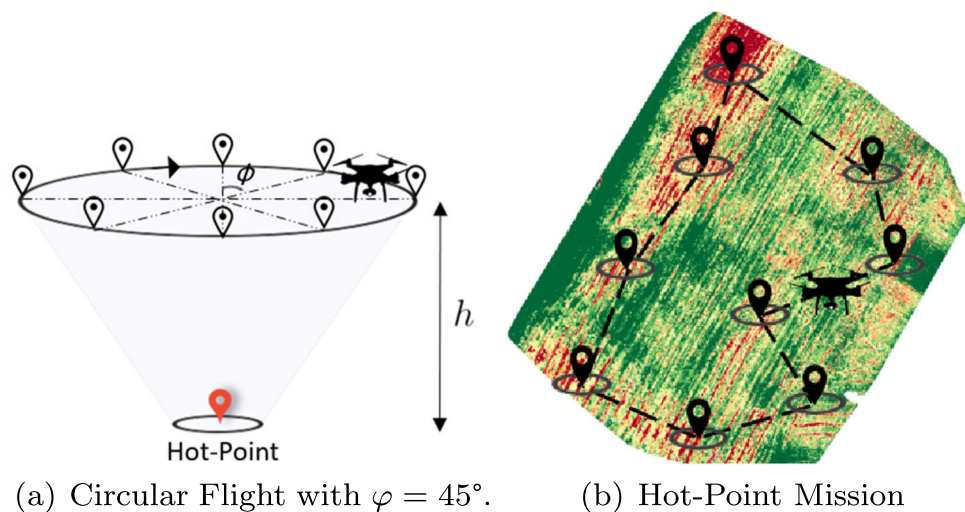
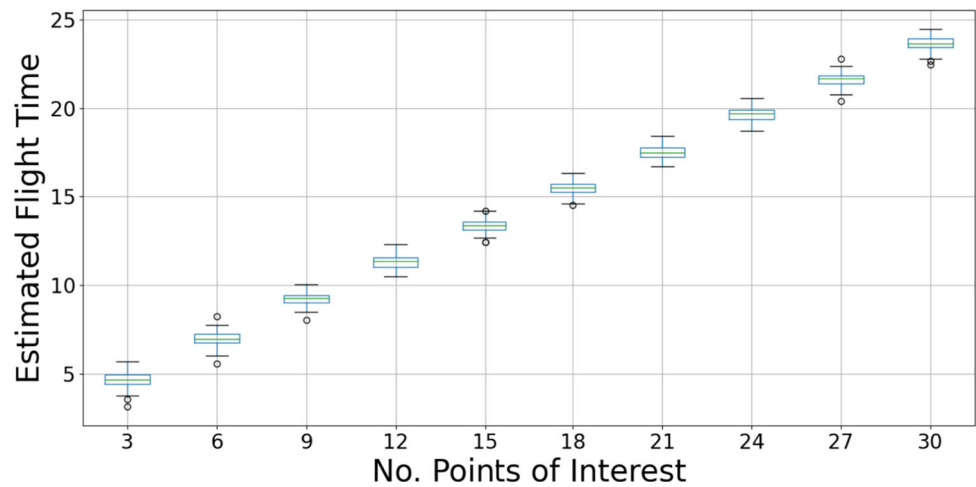


Fig. 11 Hot-Point Experimental Data. A box plot visualization of the data distribution for a given set χ concerning the estimated flight time to serve a Hot-point mission



data collected during the Hot-Point mission and provide information regarding the detected weeds to the end-user.

Accurate weed detection is a crucial, yet challenging, task of precision agriculture in order to maintain the health of the overall crop and minimize the damage through selective treatment. Convolutional Neural Networks (CNNs) have reported remarkable results in a wide variety of computer vision tasks and especially in the fields of object detection [46] and semantic segmentation [42]. The proposed framework employs these techniques for the task of weed detection. In specific, the particular module is based on DeepLabv3+ [14], a robust deep-learning model for image semantic segmentation possessing state-of-the-art results on PASCAL VOC 2012 dataset [18]. Contrary to more naive approaches of image classification where the input image is classified as a weed or not, the semantic segmentation approach enables the detection of multiple instances and types of weeds by applying a pixel-wise annotation on the input and thus, provides accurate information regarding the location of the detected weeds. The developed module is capable of semantically segmenting weed instances depicted on input RGB images captured during the site-specific mission and thus, provides critical high-level information regarding the location of detected weeds to the end-user in order to schedule counteractions on the spot.

4.2.1 Dataset

It is important to note that weed detection is a quite challenging task due to the nature of the problem since the captured weed instances present a wide variety in terms of form, size, and shape. Moreover, the development of a meaningful dataset that can enclose the majority of different occasions is not an easy task, since it requires an extreme

amount of man-hours and the corresponding workforce, as well as the collection of data over a wide period of time. Yet, to train and evaluate the deployed model, a custom dataset was designed for the task of weed semantic segmentation, as an initial approach to create the required challenging dataset. Towards this direction, a set of 1280×720 RGB images was collected during real-life experiments over a cotton field. The acquired images depict crop rows of cotton plants where different types of weeds interfere among the planting lines. The annotation of the collected data was conducted by field experts utilizing the LabelMe annotation tool [50]. The annotation process led to a set of 201 labeled images where depicted weed clusters are labeled as “weed” while the rest of the image is annotated as “background”. The developed dataset [31] is publicly available⁴ to the research community. To further demonstrate the challenging nature of the specific task, Table 4 presents the number of pixels for each class, among the number of depicted weed instances contained in the built dataset. As expected, the developed dataset is imbalanced due to the complex nature of the problem.

The developed dataset was split into a training (80%) and a testing (20%) set. The splitting process was repeated 3 times aiming to create subsets with different class distributions in order to conduct a more thorough evaluation of the employed detector.

4.2.2 Train & Evaluate

The deployed DeepLabv3+ instance was pre-trained on PASCAL VOC 2012 dataset and further trained on the previously described, custom dataset for the weed semantic

⁴<https://github.com/CoFly-Project/CoFly-WeedDB>

Table 4 Number of pixels & instances per class contained in the custom-built dataset

Class	# Pixels $\times 10^6$	# Instances
Background	175	—
Weed	9	383

segmentation task. The model was trained for 500 epochs with a batch size of 12 on a GeForce RTX 2060 Super GPU. As an optimization algorithm, Adam solver with a learning rate equal to 10^{-3} was selected. To tackle the imbalance issue among the dataset classes, focal loss was employed. Moreover, data augmentation techniques were utilized to enhance the generalization ability of the model. In specific, every input image of the training set was horizontally/vertically flipped and randomly resized on a scale between 0.5 to 1.5 times of the initial size with a chance of 50%. Finally, a patch-sized 320×320 pixels were randomly cropped from the original image before being fed to the model, to further augment the training data.

The developed weed detector is trained and evaluated on the three different dataset splits, while its efficiency is measured in terms of Intersection-over-Union (IoU), a standard accuracy measure for semantic segmentation tasks. In Table 5 the IoU for each class among the overall mean-IoU (mIoU) is reported. Results imply that the deployed model is capable of correctly classifying the samples of background class with significantly high accuracy for all splits. Regarding the weed class, the accuracy is lower mostly due to the reduced number of samples in the dataset. Yet, one can consider the reported model performance satisfying by taking into consideration that for the weed detection task it is adequate to detect a wide region where weed clusters are present, compared to other applications of semantic segmentation, such as autonomous driving, where a crisp prediction of the detected object shape is required.

5 System Integration

5.1 Graphical User Interface

All of the aforementioned services are packed inside an end-to-end precision agriculture platform called CoFly. The CoFly software is designed with the user workflow in mind to provide him with an interactive environment based on precision agriculture applications. The primary goal is to make scanning, imaging, and parametric analysis of crops easy and accessible via the user interface⁵ (UI) (Fig. 12). These functionalities are adequate for most simple

⁵<https://github.com/CoFly-Project/cofly-gui>

Table 5 Model accuracy in terms of IoU (%)

	Background	Weed	mIoU
Split 1	95.44	44.93	70.18
Split 2	95.83	43.49	69.66
Split 3	96.62	43.96	70.29

missions that require surveillance of a new crop, analysis of photometric indices, and detection of intrusive plant species. To finalize the CoFly software as an integrated end-to-end platform, the connection between the autonomous navigation system, the remote sensing control system, and the graphical user interface is essential to ensure stability in such a large-scale application. The communication of data among different modules is implemented by a Restful API Service [40] which allows the different sub-systems to connect and interact with each other by retrieving the necessary data for their execution. Data transmission between sub-systems is done via .json files which contain all the necessary information needed to run any operation of the overall system. Figure 13 illustrates the overall CoFly workflow along with the communications among its modules. It must be emphasized that each and every sub-system of Fig. 13 is integrated with the CoFly UI.

A video demonstration illustrating the sequence of events among all the aforementioned services is attached to the following link⁶

5.2 CoFly-to-UAV Bidirectional Communication System

For attaining a drone mapping project defined by the GUI, the coverage and inspection tasks are executed by forwarding the objectives of the corresponding mission to the UAV. As an intermediate communication layer between the CoFly software and the UAV, a smartphone that runs a mobile application is connected to the UAV's onboard controller. Towards this direction, in this paper, to accommodate a common commercial UAV, a custom android app adaptor has been developed with the use of the DJI API, as a data transceiver between Algorithm 1 & 2 with the UAV (as shown in Fig. 14). This application is responsible for forwarding the path to the UAV as well as gathering information (telemetry, photographs, statuses, and mission logs) and transmitting them to the backend of the platform.

Note that the aforementioned android application has been set up to provide i) a high level of autonomy for the missions and ii) a dynamic visualization of the flight data.

⁶https://www.youtube.com/watch?v=C0hdCu-ZRQk&ab_channel=CoFlyProject

Fig. 12 CoFly GUI, showing the available flight settings menu bar on the left, and the map with the polygon drawn by the user



Fig. 13 An overview of the CoFly system workflow illustrating the information exchanged among its modules

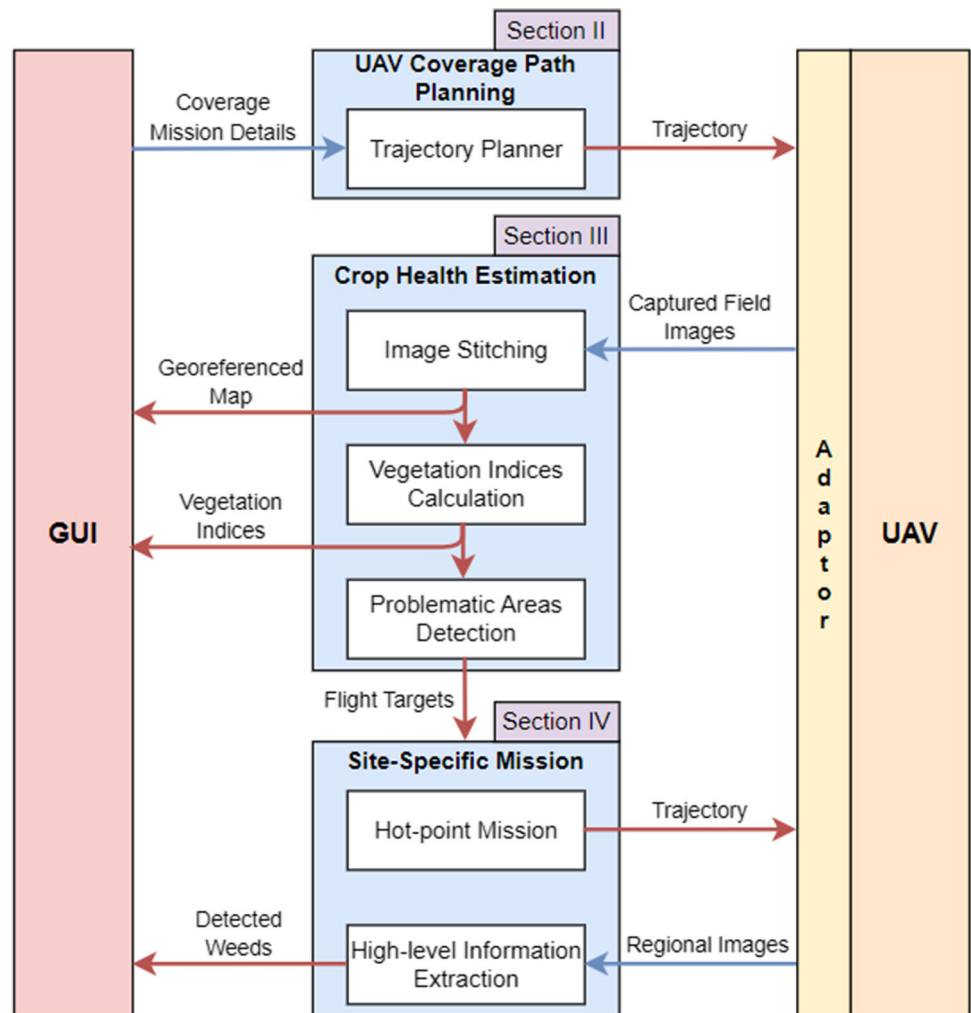
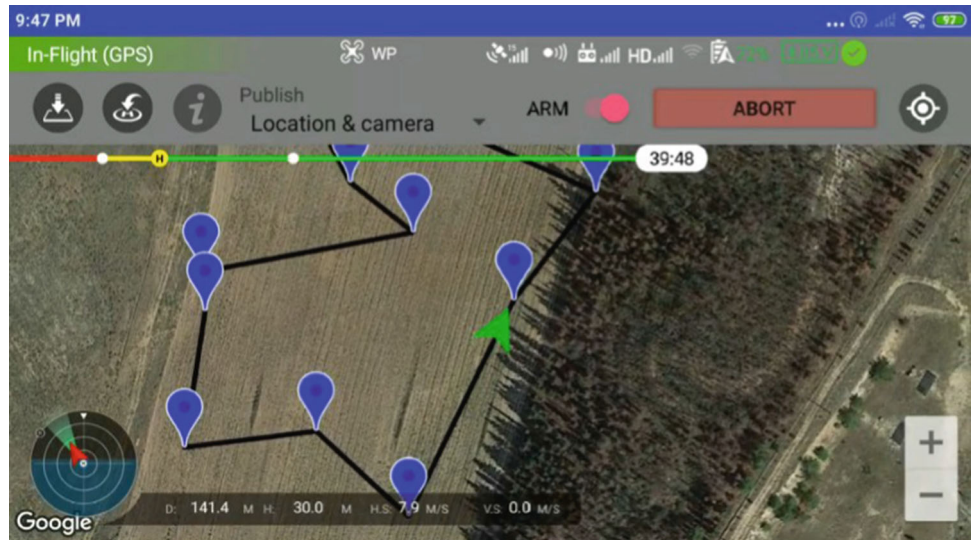


Fig. 14 Custom-developed android application adaptor



Based on this logic, the interaction between the user and the CoFly software can be done entirely from the GUI, with no need to directly control the UAV with the controller or the smartphone application. Therefore, as the mobile application also works as an on-the-field pilot by allowing to monitor live the mission, the user can take control and directly operate the UAV, if needed. Note that according to the type of the UAV and its functional requirements, a different plugin as an intermediate communication layer might be required (see Adaptor in Fig. 13).

6 Real-life Experiments

This section presents two real-life experiments, where the proposed CoFly application was used to monitor, acquire data, and assess the vegetation health of real agricultural fields. The experiments were conducted in the fertile land of the Thessalian plain and more specifically in a rural area planted with cotton located in the city of Larissa, Greece. The functionalities and technologies of the CoFly software, as well as the produced results, were evaluated by the experienced agronomists of the Bios Agrosystems S.A.⁷, who specialize in crop production, soil control, and soil management. This section aims to evaluate the robustness and efficiency of the CoFly software in real-world precision agriculture operations.

6.1 Experimental Setup

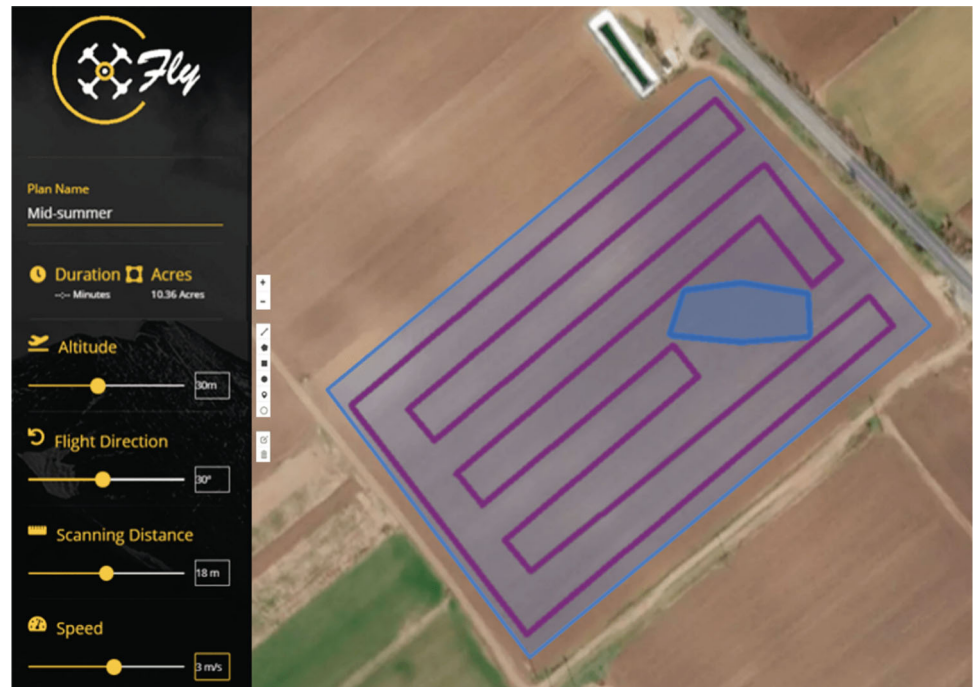
For the real-life experiments, a commercial UAV (DJI Phantom 4 Pro) equipped with an RGB camera was used for image acquisition. The UAV flight planning for

monitoring and inspection was done by the developed algorithm 1 and 2, respectively. For mission control, an android smartphone device (Xiaomi Mi Max 2) was used to run the android app adaptor, described in Section 5.2. Finally, a portable 4G router (tp-link M7350) was used as an intermediate communication layer between the smartphone and the CoFly software. To evaluate the efficiency of the proposed framework, two scenarios were considered. The first scenario was performed in July 2019 during the first stages of plant growth, while the second scenario was performed in September 2019 during the ripening period, when the cotton fibers had appeared. The main objectives of these two experiments were to (i) identify the optimum flight envelope to acquire the best data and provide an orthomosaic as an accurate map of the field, (ii) extract plant health information (VIs) that could give indications on the year's harvest and cotton's growth stage (timeline view) and (iii) detect the existence of weeds in a specific site of crops. In the next subsections, each experiment is analyzed and distinguished according to the flight date and cotton's developmental phase.

6.2 Mid-Summer

As to the first scenario, the cotton field during its early stage along with the calculated path is depicted in Fig. 15. To demonstrate the practical enforcement of the proposed navigation algorithm (Algorithm 1) in terms of avoiding NFZs/Obstacles, we considered an area within the original field as restricted, meaning that the UAV is not allowed to fly above. For coverage mission (see Table 6), the selected *flight altitude* and *scanning density* resulted in a spatial level resolution of 90%. In addition, the operational speed was selected to be 3 m/s, which is low enough to maintain the quality of the collected images, and the gimbal pitch was

⁷<http://www.bios-agrosystems.gr/en-US/>

Fig. 15 Calculated coverage path for mid-summer mission**Table 6** Mission details corresponding to Algorithm 1 & 2 for monitoring and inspection tasks

Mission	Coverage						Inspection	
Crop season	Flight altitude (<i>m</i>)	Scanning density (<i>m</i>)	Flight direction ($^{\circ}$)	Speed (<i>m/s</i>)	Corner radius (<i>m</i>)	Gimbal pitch ($^{\circ}$)	<i>r</i> (<i>m</i>)	φ ($^{\circ}$)
Mid-summer	30	18	30	3	0.5	-90	5	45
Early-autumn	50	15	30	3	0.5	-90	–	–

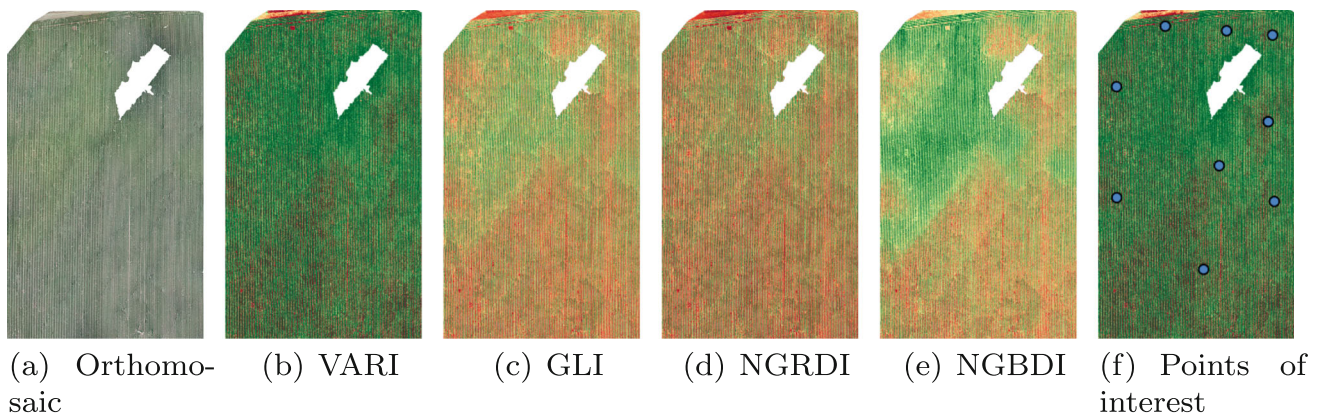
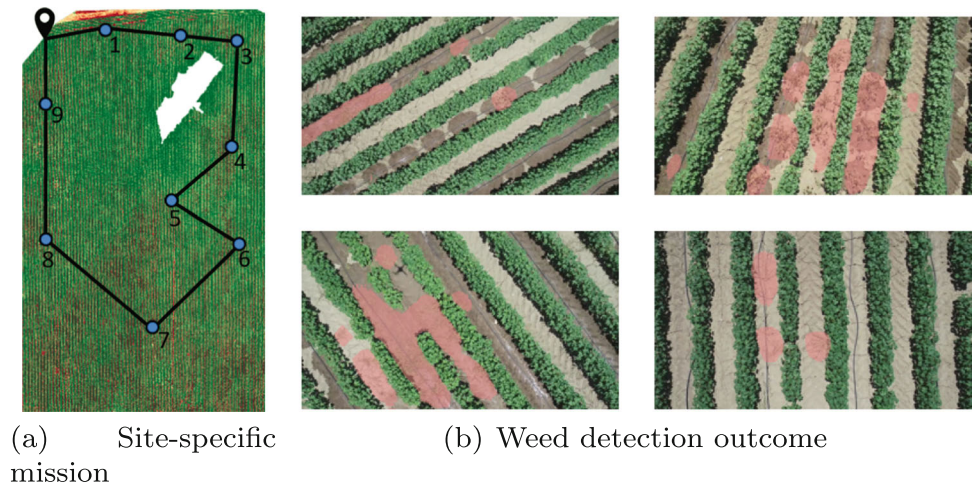
**Fig. 16** Results of the knowledge extraction module from the experimental evaluation conducted on mid-summer

Fig. 17 Designed (a) site-specific mission and the corresponding (b) weed detection module output over the acquired visual footage. In (a) the take-off/landing area is notated with the black marker while in (b) the detected weeds are highlighted with light red color



selected to be -90° (meaning that the camera is always in the “top-view” mode), an angle that calibrates images for 2D map outputs. Figure 15 graphically illustrates the resulted path as calculated from Algorithm 1. A direct outcome is that the path-planning algorithm managed to efficiently alter the UAV path to meet the field characteristics. More specifically, path (i) completely covers the area of interest, (ii) does not contain bootstrapping (path crosses), and (iii) carefully avoids the NFZs.

The acquired images during the conducted mission are forwarded to the knowledge extraction module for further processing. Initially, the orthomosaic is calculated and presented in Fig. 16(a). It can be noted that the collected data are sufficient to create a high-resolution map of the examined agricultural area, implying the robust operation

of the developed path planning module. In Fig. 16(b)–(e) the image representations of the calculated VIs are presented. The calculated VIs can provide a crisp overview of the crop’s health, indicating areas with relatively poor vegetation conditions. Finally, in Fig. 16(f) the result of the points of interest detection module is illustrated for the VARI index. One can see that the implemented module can efficiently detect problematic, in terms of vegetation health, field regions where the site-specific mission should be deployed.

Given the geolocation of the depicted points of interest, Algorithm 2 was deployed to provide a site-specific inspection as illustrated in Fig. 17(a). Additional information corresponding to the Hot-Point mission for the inspection task can be found in Table 6. In Fig. 17(b) qualitative results

Fig. 18 Calculated coverage path for early-autumn mission

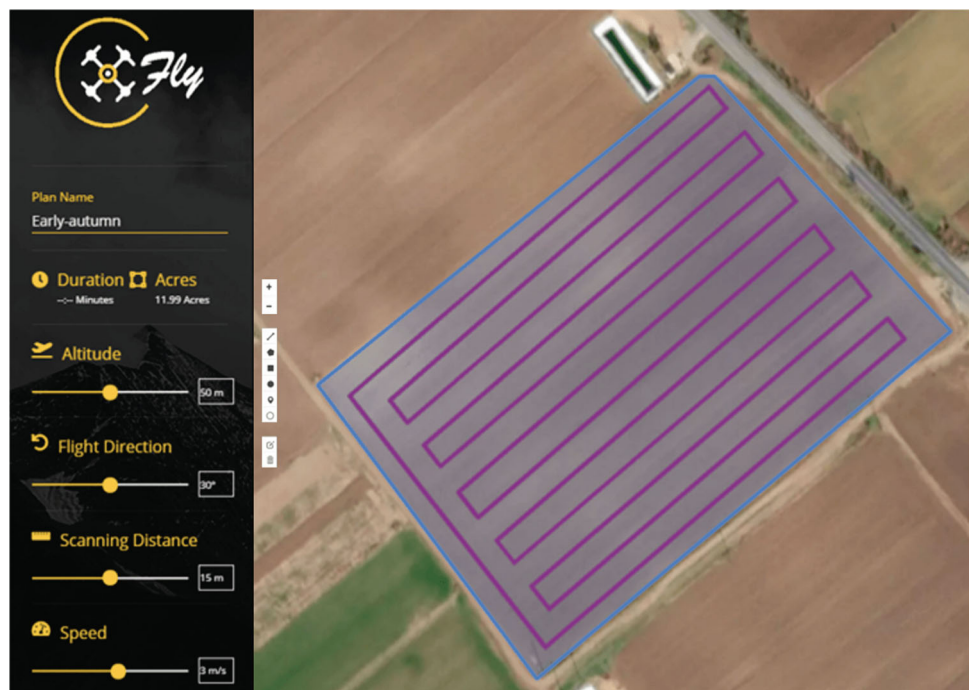
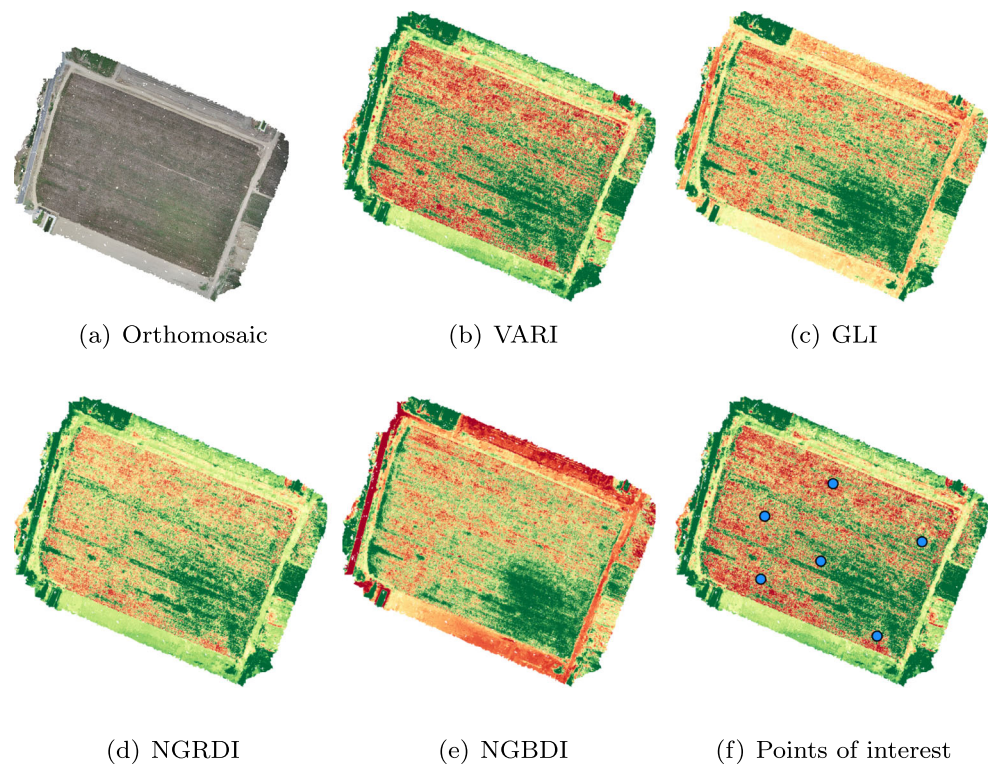


Fig. 19 Results of the knowledge extraction module from the experimental evaluation conducted on early autumn



from the site-specific mission in the aforementioned areas are presented. It is noted that the deployed weed detector can sufficiently detect regions where weed instances are depicted. The advantage of a semantic segmentation approach is also illustrated in the results since accurate spatial information regarding the existence of weeds in the field can be provided to the end-user through the selected approach.

6.3 Early-Autumn

As to the second scenario in the early autumn, the cotton field during its boll and fiber maturation along with the calculated path is depicted in Fig. 18. Note that the selected aspects of navigation for the coverage task were adjusted to achieve a spatial level resolution of 90%, while the UAV's altitude was set to be 50m (see Table 6).

In Fig. 19(a) the extracted orthomosaic of the examined area is presented. Similarly to the previous case, the designed path enables the collection of sufficient data to provide a detailed overview of the field. Figure 19(b)–(e) illustrate the calculated VIs, pointing out regions where the vegetation conditions are poor. Finally, in Fig. 19(f) the key outcome of the problematic areas detection module is demonstrated for the VARI index. It should be noted that the specific experiment was conducted in early autumn, a period in which cotton is harvested and the cotton plants are fully

developed. Thus, any interfering weeds are covered by the foliage of the cotton plants and cannot be visually detected by an above inspection. Furthermore, the treatment of weeds can be considered an off-season activity during this time period, since it is more crucial to conduct it at the early stage of the crop's life cycle when the dominance of crop plants over the weeds is crucial. Towards this direction, the site-specific mission for weed detection was not executed in this case.

7 Conclusions

UAV systems innovation is valuable in agriculture. Following leading-edge technologies, in this paper, an end-to-end and low-cost precision agriculture platform based on a fully functional autonomous UAV system has been proposed. In contrast to the majority of the already existing agriculture platforms, which use cloud-based approaches for processing agricultural data and establishing prices according to the information processing, the proposed platform has been designed to provide offline and real-time at the field's edge, stand counts, and assessments. Towards that purpose, coverage, and inspection missions for any type of UAV are developed to provide plant health maps, weed measurements, and detection by owning an integrated deep learning module.

All of these capabilities are packed inside an end-to-end platform functioning on mobile devices including high-level user commands and proper data representations that ease the utilization of UAVs in precision agriculture applications.

The functionality of the proposed system was tested and validated with extensive experimentation in agricultural fields with cotton in Larissa, Greece during two different crop sessions i) in mid-summer and ii) in early autumn. In both experiments, CoFly preserved all the desired features of common agricultural software, while at the same time, it achieved fully autonomous inspection tasks empowering the UAV robots also as a tool for site-specific precision farming. This feature is of paramount importance in agribusiness, as it can be utilized in information procurement, examination, and consistently observing fields by the farming community to acquire contemporary field management skills. CoFly software with a low budget provides beneficial and high accuracy insights to turn into the critical segments of the farming industry.

As future directions, we are interested in evolving this research by integrating multi-UAVs for monitoring and inspection tasks. In such an approach, the proposed agricultural software should be enriched with a multi-robot decision-making scheme to autonomously navigate each robot to perform individual coverage subtasks according to the overall set of the user's objectives. This work may examine more complicated scenarios where UAVs will also perform pesticide spraying treatments to have a robust system tailored to a variety of agricultural applications.

Acknowledgments We gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs used for this research.

Author Contributions E. K. Raptis contributed to the material preparation, UAV data collection, analysis, conceptualization, writing - original draft, visualization, and software. M. Krestenitis contributed to the material preparation, conceptualization, data curation, writing - review & editing, visualization, and software. Graphical user interface and integration were performed by K. Egglezos and O. Kypris. L. Doitsidis and A. Ch. Kapoutsis did the supervision, project administration, writing - review & editing. K. Ioannidis, E. B. Kosmatopoulos, S. Vrochidis, and I. Kompatsiaris performed the supervision, resources, and funding acquisition. All authors read and approved the final manuscript.

Funding Open access funding provided by HEAL-Link Greece. This research has been financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (T1EDK-00636).

Data Availability The overall precision agriculture platform along with the corresponding modules and the dataset generated during the real-life experiments are open-source and available in the <https://github.com/CoFly-Project> repository.

Declarations

Conflict of Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. OpenDroneMap. <https://github.com/OpenDroneMap/ODM>
2. Opensfm. <https://github.com/mapillary/OpenSfm>
3. Agisoft: Agisoft Metashape. <https://www.agisoft.com> (2020) [Online; accessed 22 October-2020]
4. Aiello, G., Valavanis, K.P., Rizzo, A.: Fixed-wing uav energy efficient 3d path planning in cluttered environments. *J. Intell. Robot. Syst.* **105**(3), 1–13 (2022)
5. Analytics, S.F.: Agriculture Mapping Software. <https://sentera.com/> [Online; accessed 22 October-2020]
6. Apostolidis, S.D., Kapoutsis, P.C., Kapoutsis, A.C., Kosmatopoulos, E.B.: Cooperative multi-uav coverage mission planning platform for remote sensing applications. *Auton. Robot.* **46**(2), 373–400 (2022)
7. Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The traveling salesman problem: a computational study. Princeton University Press (2006)
8. ArduPilot: Ardupilot Mission Planner. <http://ardupilot.org/planner> (2020) [Online; accessed 22 October-2020]
9. Basiri, A., Mariani, V., Silano, G., Aatif, M., Iannelli, L., Glielmo, L.: A survey on the application of path-planning algorithms for multi-rotor uavs in precision agriculture. *J. Navig.* **75**(2), 364–383 (2022)
10. Bolitho, M., Kazhdan, M., Burns, R., Hoppe, H.: Parallel poisson surface reconstruction. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Wang, J.X., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) *Advances in Visual Computing*, pp. 678–689. Springer, Berlin (2009)
11. Bombe, K.: Agriculture drone market worth \$5.19 billion by 2025, growing at a cagr of 31.1% from 2019- global market opportunity analysis and industry forecasts by meticulous research®. Meticulous Market Research Pvt. Ltd., GLOBE NEWSWIRE (2020) [Online; accessed 09 June-2020]
12. Botlink: Automated Drone Flight Software. <https://botlink.com/> (2020) [Online; accessed 22 October-2020]
13. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. *Commun. Stat. - Theory Methods* **3**(1), 1–27 (1974)
14. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818 (2018)

15. Christiansen, M.P., Laursen, M.S., Jørgensen, R.N., Skovsen, S., Gislum, R.: Designing and testing a uav mapping system for agricultural field surveying. *Sensors* **17**(12), 2703 (2017)
16. Deploy, D.: Drone Mapping Software. <https://www.dronedeploy.com/> (2020) [Online; accessed 22 October-2020]
17. Ebeid, E., Skriver, M., Terkildsen, K.H., Jensen, K., Schultz, U.P.: A survey of Open-Source UAV flight controllers and flight simulators. *Microprocess. Microsyst.* **61**(May), 11–20 (2018). <https://doi.org/10.1016/j.micpro.2018.05.002>
18. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. *Int. J. Comput. Vis.* **111**(1), 98–136 (2015)
19. Faical, B.S., Freitas, H., Gomes, P.H., Mano, L.Y., Pessin, G., de Carvalho, A.C., Krishnamachari, B., Ueyama, J.: An adaptive approach for uav-based pesticide spraying in dynamic environments. *Comput. Electron. Agric.* **138**, 210–223 (2017)
20. Faical, B.S., Pessin, G., Geraldo Filho, P., Carvalho, A.C., Furquim, G., Ueyama, J.: Fine-tuning of Uav control rules for spraying pesticides on crop fields. In: 2014 IEEE 26Th International Conference on Tools with Artificial Intelligence, pp. 527–533. IEEE (2014)
21. FAO: Building a common vision for sustainable food and agriculture: Principles and approaches (2014)
22. Furrer, F., Burri, M., Achtelik, M., Siegwart, R.: Rotors—A modular gazebo Mav simulator framework. In: *Robot Operating System (ROS)*, pp. 595–625. Springer (2016)
23. Gabriely, Y., Rimon, E.: Spanning-tree based coverage of continuous areas by a mobile robot. *Ann. Math. Artif.* **31**(1–4), 77–98 (2001)
24. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. *Rob. Auton. Syst.* **61**(12), 1258–1276 (2013)
25. Gitelson, A., Stark, R., Grits, U., Rundquist, D., Kaufman, Y., Derry, D.: Vegetation and soil lines in visible spectral space: a concept and technique for remote estimation of vegetation fraction. *Int. J. Remote Sens.* **23**(13), 2537–2562 (2002)
26. Holden, N.M., White, E.P., Lange, M.C., Oldfield, T.L.: Review of the sustainability of food systems and transition using the internet of food. *npj Sci. Food* **2**(1), 1–7 (2018)
27. Hunt, E.R., Cavigelli, M., Daughtry, C.S., McMurtrey, J.E., Walthall, C.L.: Evaluation of digital photography from model aircraft for remote sensing of crop biomass and nitrogen status. *Precis. Agric.* **6**(4), 359–378 (2005)
28. Kapoutsis, A.C., Chatzichristofis, S.A., Kosmatopoulos, E.B.: Darp: divide areas algorithm for optimal multi-robot coverage path planning. *J. Intell. Robot. Syst.* **86**(3–4), 663–680 (2017)
29. Karatzinis, G.D., Apostolidis, S.D., Kapoutsis, A.C., Panagiotopoulou, L., Boutalis, Y.S., Kosmatopoulos, E.B.: Towards an integrated low-cost agricultural monitoring system with unmanned aircraft system. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1131–1138. IEEE (2020)
30. Koutras, D.I., Kapoutsis, A.C., Kosmatopoulos, E.B.: Autonomous and cooperative design of the monitor positions for a team of uavs to maximize the quantity and quality of detected objects. *IEEE Robot. Autom. Lett.* **5**(3), 4986–4993 (2020)
31. Krestenitis, M., Raptis, E.K., Kapoutsis, A.C., Ioannidis, K., Kosmatopoulos, E.B., Vrochidis, S., Kompatsiaris, I.: Coffy-weeddb: A uav image dataset for weed detection and species identification. *Data in Brief*, pp. 108575 (2022)
32. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.* **7**(1), 48–50 (1956)
33. Louhaichi, M., Borman, M.M., Johnson, D.E.: Spatially located platform and aerial photography for documentation of grazing impacts on wheat. *Geocarto Int.* **16**(1), 65–70 (2001)
34. Popović, M., Galceran, E., Khanna, R., Sa, I.: High-level waypoint-following for micro aerial vehicles (UAVs). https://github.com/ethz-asl/waypoint_navigator [Online; accessed 06 August-2018]
35. Pix4D: Professional photogrammetry and drone mapping software. <https://www.pix4d.com/> (2020) [Online; accessed 22 October-2020]
36. Popovici, A., Popovici, D.: Cellular Automata in Image Processing. In: *Fifteenth International Symposium on Mathematical Theory of Networks and Systems*, vol. 1, pp. 1–6. Citeseer (2002)
37. Primicerio, J., Di Gennaro, S.F., Fiorillo, E., Genesio, L., Lugato, E., Matese, A., Vaccari, F.P.: A flexible unmanned aerial vehicle for precision agriculture. *Precis. Agric.* **13**(4), 517–523 (2012)
38. Quigley, M., Faust, J., Foote, T., Leibs, J.: Ros: an open-source robot operating system
39. Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., Moscholios, I.: A compilation of UAV applications for precision agriculture. *Comput. Netw.* **172**, 107148 (2020). <https://doi.org/10.1016/j.comnet.2020.107148>
40. Richardson, L., Amundsen, M., Ruby, S.: RESTful Web APIs. O'Reilly Media, Inc (2013)
41. Rööös, E., Bajželj, B., Smith, P., Patel, M., Little, D., Garnett, T.: Greedy or needy? land use and climate impacts of food in 2050 under different livestock futures. *Glob. Environ. Change* **47**, 1–12 (2017)
42. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520 (2018)
43. Singh, G., Yogi, K.K.: Internet of things-based devices/robots in agriculture 4.0. In: *Sustainable Communication Networks and Application*, pp. 87–102. Springer (2022)
44. Surveying, O., Committee, P., et al: Epsg geodetic parameter dataset. <http://www.epsg.org/> (2005) [Online; accessed 12 August-2006]
45. Takaya, K., Asai, T., Kroumov, V., Smarandache, F.: Simulation environment for mobile robots testing using ros and gazebo. In: 2016 20Th International Conference on System Theory, Control and Computing (ICSTCC), pp. 96–101. IEEE (2016)
46. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10781–10790 (2020)
47. Technology, B.R.: Blue River Technology. <https://www.bluerivertechnology.com/> (2020) [Online; accessed 22 October-2020]
48. Uto, K., Seki, H., Saito, G., Kosugi, Y.: Development of Uav-mounted miniature hyperspectral sensor system for agricultural monitoring. In: 2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS, pp. 4415–4418. IEEE (2013)
49. Vasudevan, A., Kumar, D.A., Bhuvanewari, N.: Precision farming using unmanned aerial and ground vehicles. In: 2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), pp. 146–150. IEEE (2016)
50. Wada, K.: labelme: Image Polygonal Annotation with Python. <https://github.com/wkentaro/labelme> (2016) [Online; accessed 08 April-2018]
51. Wu, X.D., Guo, J.L., Han, M., Chen, G.: An overview of arable land use for the world economy: from source to sink via the global supply chain. *Land Use Policy* **76**, 201–214 (2018)

52. Yang, D.: Gobi vegetation recognition based on low-altitude photogrammetry images of Uav. In: IOP Conference Series: Earth and Environmental Science, vol. 186, pp. 012053. IOP Publishing (2018)
53. Zheng, H., Zhou, X., Cheng, T., Yao, X., Tian, Y., Cao, W., Zhu, Y.: Evaluation of a Uav-based hyperspectral frame camera for monitoring the leaf nitrogen concentration in rice. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 7350–7353. IEEE (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Emmanuel K. Raptis is a Ph.D. student at the Department of Electrical and Computer Engineering (E.E.C.E) of the Democritus University of Thrace. In 2018, he received his Diploma in Electrical Engineering and completed his M.Sc. in Automatic Control Systems and Robotics Laboratory in 2020, both at Democritus University of Thrace. Since 2018, he has been working as a Research Associate at the Information Technologies Institute of the Centre for Research and Technology Hellas (CERTH). During these years, he is being participated in several H2020 and National research projects. His research interests focus on the field of Robotics, Automatic Control, Multi-Agents and Autonomous systems, Artificial Intelligence, and Reinforcement Learning.

Marios Krestenitis received the Diploma degree in Electrical and Computer Engineering in 2015 and the M.Sc. in Informatics in 2019, both from the Aristotle University of Thessaloniki, Greece. Currently, he is a Research Assistant at the Information Technologies Institute (ITI) of the Centre for Research and Technology Hellas (CERTH). His research interests include machine learning, computer vision, digital signal processing, pattern recognition and aerial imagery. He has been involved in various EU and National funded projects.

Konstantinos Egglezos is a Robotics & Automation Software engineer with more than 6 years of experience working alongside the executive team of IKH. Konstantinos specializes in develop software for autonomous systems & graphical interfaces. He has a Bachelor of Science in Aegean University from the Department of Product and Systems Design Engineering and also Master of Science in Mechatronics, Robotics, and Automation Engineering on National Technical University of Athens.

Orfeas Kypris received the B.Eng. degree in Electrical and Electronic Engineering, and the M.Sc. degree in Magnetics from Cardiff University, Cardiff, U.K., in 2009 and 2010, respectively, and the Ph.D. degree in Electrical Engineering from the Department of Electrical and Computer Engineering, Iowa State University, USA, in 2015. From 2015 to 2017, he was a Post-Doctoral Researcher with the Department of Computer Science, University of Oxford, where he developed sensors for indoor localization and structural health monitoring using low-frequency magnetic fields. His research interests include nondestructive evaluation, applied electromagnetism, and robotics. He is currently a software engineer specializing in machine learning applications in the industry.

Dr. Konstantinos Ioannidis received the bachelor's and Ph.D. degrees from the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece, in 2006 and 2013, respectively. Currently, he is a Postdoctoral Research Fellow at Centre for Research and Technology Hellas (CERTH), Information Technology Institute (ITI). His research interests include mainly the areas of digital twins, path planning and collective behaviour in swarm robotics as well as various computer vision techniques mainly focused on robotic applications such as object detection, SLAM, 3d reconstruction, activity recognition, aerial imaging, image fusion, activity recognition, weed detection etc. Mr. Ioannidis has participated in more than 15 European and National research projects, as a deputy coordinator, scientific and technical manager etc., and has been a member of various organization teams of conferences, workshops etc. He has been an author of more than 40 scientific journals, conferences, workshops and book chapters. He has served as a reviewer in several international Journals and as a Technical Program Committee in various peer reviewed Conferences/Workshops.

Dr. Lefteris Doitsidis is an Assistant Professor at the School of Production Engineering & Management of the Technical University of Crete. Prior to his appointment, he was a faculty member (Associate & Assistant Professor) at the Department of Electronics, Hellenic Mediterranean University (former TEI of Crete). He was also a visiting scholar at the Department of Computer Science and Engineering, University of South Florida, FL, U.S.A. He is the author of more than 45 publications, in international journals, conference proceedings and book chapters. His research interests lie in the areas of multirobot teams, design of novel control systems for robotic applications, autonomous operation and navigation of unmanned robotic vehicles and computational intelligence. He has been involved as a senior researcher into numerous research projects including funded by European and National funds.

Athanasios Ch. Kapoutsis received the Diploma and Ph.D. degrees from the Department of Electrical and Computer Engineering (DUTH), Xanthi, Greece, in 2012 and 2017, respectively. He is currently a postdoctoral researcher with the Information Technologies Institute (CERTH), Thessaloniki, Greece. During the past years, he has been involved in several EU FP7, H2020 and National funded Research and Development projects. His research is mainly focused on robotics, reinforcement learning, multi-agent systems, machine intelligence, adaptive control, and pattern recognition.

Dr. Stefanos Vrochidis received the Diploma degree in Electrical Engineering from Aristotle University of Thessaloniki, Greece, the MSc degree in Radio Frequency Communication Systems from University of Southampton and the PhD degree in Electronic Engineering from Queen Mary University of London. Currently, he is a Senior Researcher (Grade C) with the Information Technologies Institute of the Centre for Research and Technology Hellas (ITI-CERTH) and the Head of the Multimodal Data Fusion and Analytics (M4D) Group. His research interests include multimedia understanding and retrieval, multimodal fusion, computer vision, multimodal analytics, artificial intelligence, as well as media & arts, environmental and security applications. Dr. Vrochidis has participated in more than 80 European and National projects and has been member of the organization team of several conferences and workshops. He has edited 3 books and authored more than 300 related scientific journal, conference and book chapter publications. He has served as a reviewer in several international Journals and as Technical program committee in well reputed conferences and workshops.

Dr. Ioannis (Yiannis) Kompatsiaris is the Director of the Information Technologies Institute, Research Director at CErTH-ITI and the Head of Multimedia Knowledge and Social Media Analytics Laboratory. His research interests include AI/Machine Learning for multimedia analysis, Semantics (multimedia ontologies and reasoning), Social Media and Big Data Analytics, Multimodal and Sensors Data Analysis, Human Computer Interfaces, e- Health, Arts and Cultural, Media/Journalism, Environmental and Security applications. He is the co-author of 178 papers in refereed journals, 63 book chapters, 8 patents and 560 papers in international conferences. Since 2001, Dr. Kompatsiaris has participated in 88 National and European research programs, in 31 of which he has been the Project Coordinator. He has also been the PI in 15 contracts from the industry.

Elias B. Kosmatopoulos received the Diploma, M.Sc. and Ph.D. degrees from the Technical University of Crete, Greece, in 1990, 1992, and 1995, respectively. He is currently a Professor with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece and a collaborative Professor with the Information Technologies Institute, Centre for Research & Technology, Hellas, Greece. Previously, he was a faculty member of the Department of Production Engineering and Management, Technical University of Crete (TUC), Greece, a Research Assistant Professor with the Department of Electrical Engineering-Systems, University of Southern California (USC) and a Postdoctoral Fellow with the Department of Electrical & Computer Engineering, University of Victoria, B.C., Canada. Dr. Kosmatopoulos' research interests are in the areas of learning systems, adaptive optimization and control and their applications to Internet of Things; energy efficient buildings and smart grids; intelligent transportation systems; and robotic swarms. He is the author of over 55 journal papers. Prof. Elias Kosmatopoulos has been leading many research projects funded by the European Union and the private sector with a total budget of about 17 million Euros.