

Article

A Framework for Multiple Ground Target Finding and Inspection Using a Multirotor UAS[†]

Ajmal Hinas^{1,*} , Roshan Ragel², Jonathan Roberts¹ and Felipe Gonzalez¹

¹ Robotics and Autonomous Systems, Queensland University of Technology (QUT), Brisbane City QLD 4000, Australia; Jonathan.roberts@qut.edu.au (J.R.); felipe.gonzalez@qut.edu.au (F.G.)

² Department of Computer Engineering, University of Peradeniya (UOP), Peradeniya 20400, Sri Lanka; roshanr@pdn.ac.lk

* Correspondence: ajmal.hinas@hdr.qut.edu.au or ajmalhinas@gmail.com; Tel.: +61-0470534175

[†] This paper is an expanded version of “Multiple Ground Target Finding and Action Using UAVs” published in Proceedings of 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019, and “A Framework for Vision-Based Multiple Target Finding and Action Using Multirotor UAVs” published in Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS 2018), Dallas, TX, USA, 12–15 June 2018.

Received: 26 November 2019; Accepted: 31 December 2019; Published: 3 January 2020



Abstract: Small unmanned aerial systems (UASs) now have advanced waypoint-based navigation capabilities, which enable them to collect surveillance, wildlife ecology and air quality data in new ways. The ability to remotely sense and find a set of targets and descend and hover close to each target for an action is desirable in many applications, including inspection, search and rescue and spot spraying in agriculture. This paper proposes a robust framework for vision-based ground target finding and action using the high-level decision-making approach of Observe, Orient, Decide and Act (OODA). The proposed framework was implemented as a modular software system using the robotic operating system (ROS). The framework can be effectively deployed in different applications where single or multiple target detection and action is needed. The accuracy and precision of camera-based target position estimation from a low-cost UAS is not adequate for the task due to errors and uncertainties in low-cost sensors, sensor drift and target detection errors. External disturbances such as wind also pose further challenges. The implemented framework was tested using two different test cases. Overall, the results show that the proposed framework is robust to localization and target detection errors and able to perform the task.

Keywords: unmanned aerial vehicle; unmanned aerial system; vision-based navigation; search and rescue; vision and action; OODA; inspection; target detection; remote sensing

1. Introduction

Small, unmanned aerial systems (UASs) now have advanced waypoint-based navigation capabilities, which enable them to collect surveillance [1], wildlife ecology [2] and air quality data [3] in new ways. One capability that is not available in most off-the-shelf UASs is the ability for a higher level of onboard decision-making for vision and action.

The ability to remotely sense and find a set of targets, and descend and hover close to each target is desirable in many applications, including spot spraying in agriculture (such as pesticide spraying for brown planthopper (BPH) [4] and spear thistle weed [5]), inspection and search and rescue operations in a larger area with limited human resources, or in difficult terrain, such as searching for survivors or an aviation black box.

Figure 1 is an illustrative example of a search scenario for survivors. The task can be divided into multiple steps. Initially, the UAS searches for ground targets at search height h_s (1). If target/s are

found, the UAS picks a target, based on a criterion such as first detected target and changes its original path and moves toward the target and descends (2). The UAS then hovers above the target closely ($h_h = 0.5\text{--}2.5$) to inspect the target (3). After inspection, the UAS climbs slightly (<2 m) and reaches height h_c and moves laterally to the next target (4). After reaching the next target, the UAS descends (5) and inspects that target (3). The UAS repeats steps 3, 4 and 5 with each target.

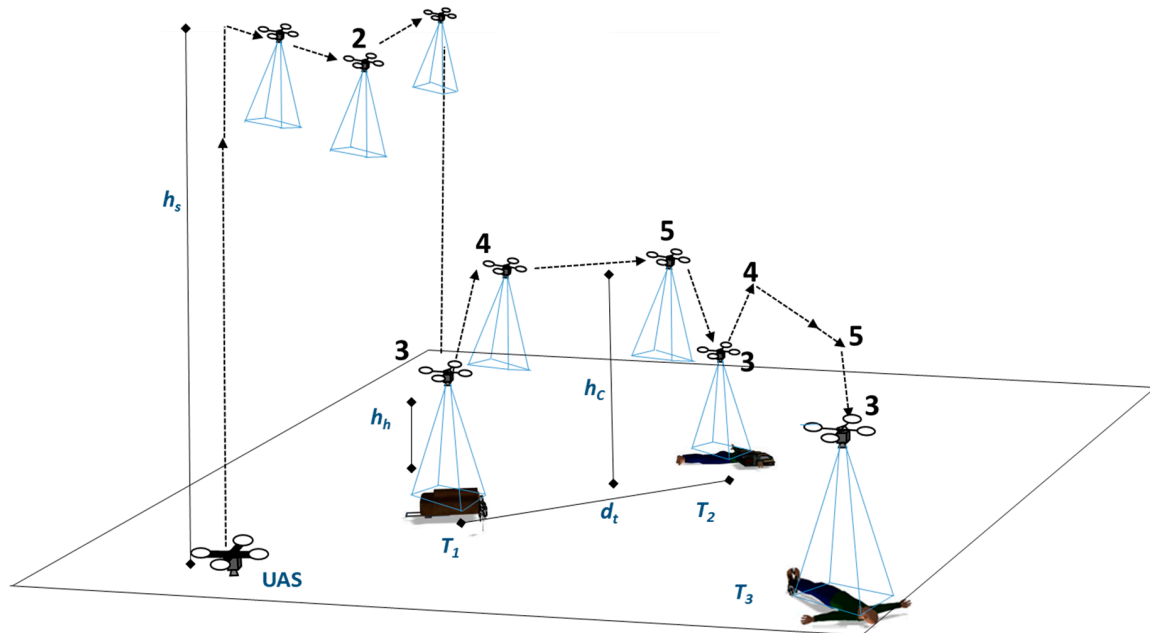


Figure 1. Illustrative search for survivors scenario with multiple objects scattered on the ground. The numbers 1–5 indicate the steps performed in the mission. Steps 3–5 are repeated for each target.

Very accurate target detection is needed to perform this task. However, target detection algorithms have a certain degree of uncertainty in real outdoor environments. Typical single frequency GPS receivers have an accuracy of 5–6 m [6,7]. Another challenge is that a greater localization accuracy needed to act on targets. In this paper, our aim is to develop and demonstrate a framework for autonomous vision-based ground target finding and action using a multirotor UAS that is robust to the (i) errors and uncertainties in the localization of the UAS; (ii) errors and uncertainties in localization of the target; (iii) target detection errors and; (iv) the effects of external disturbances such as wind.

The framework uses the high-level decision-making approach of observe, orient, decide and act (OODA) [8–10] and a series of techniques to perform the necessary navigation and control and target tracking. The proposed framework was implemented as a modular software system and demonstrated in two different test cases.

The contributions of the paper are summarized as follows: (1) An onboard framework for finding and acting on multiple ground targets which is robust to errors in localization of the UAS, localization of the targets and target detection, and external disturbances; (2) A computationally efficient voting-based scheme to avoid false detections in vision-assisted navigation or target tracking using UASs.

2. Related Work

The development of the multiple target detection and action framework is closely related to previous work on vision-based landing, target detection, target tracking, infrastructure inspection and OODA.

The vision-based landing task of UASs is a well-explored area of research. In this task, the UAS guides and descends towards a detected landing pad using vision. Mostly, a single, known target is used to detect the landing area such as a landing pad, and the accuracy of a few meters from the landing

pad is usually acceptable. Specialized patterns, such as “H”, are common to improve detection [4–6] or control the UAS [11]. Techniques such as Image-Based Visual Servoing (IBVS) and the relative distance-based approaches [7] are common. However, these techniques have limitations for multiple targets when the targets have unknown shapes and sizes or are similar.

Zarudzki et al. [12] explored multiple target tracking using an IBVS scheme and different colors for the targets to separate them. However, the IBVS schemes are limited when all or a subset of the targets are visually similar.

Many researchers have explored the tracking of a moving target [8–11]. For example, Greatwood et al. [13] used the relative location of an unmanned ground vehicle (UGV) to track it. However, relative location-based approaches are limited for multiple similar targets. In another study, Vanegas et al. [14] demonstrated finding a UGV and tracking it in GPS-denied and cluttered environments using a Partially Observable Markov Decision Process (POMDP) formulation and known obstacle position to correct the UAS pose. However, such knowledge is not available in our work. Tracking of a user-selected object was also demonstrated by Cheng et al. [14]. The authors used a UAS fixed with a gimbaled camera and a Kernelized Correlation Filter (KCF)-based visual tracking scheme to track the selected object. All of these works are limited to a single known target.

Vision-based or assisted navigation of UASs is also a well-studied area of research [15–17]. Stefas et al. [18] studied the navigation of a UAS inside an apple orchard. Missions following water channels [19] and roads [20] have also been explored. Other inspection tasks using UASs are power line inspection [21,22] and railway track inspection [23]. In these studies, a common approach is to follow the continuous image feature lines present in the observed structures. However, this research investigates the problem of finding and acting on multiple discrete targets.

In other works for railway semaphore inspection [24] and pole inspection [25], visual control techniques have been explored to fly the UAS around a pole-like structure. However, finding and approaching the target is not considered in these studies.

Vision-based ground target finding and action systems have been studied for a single target [5,26,27]. Multiple ground target finding and action capability have also been demonstrated [28,29]. However, those works did not consider the uncertainties in target detection.

Most previous studies used a control system approach to control the UAS. Another possible approach is a cognitive model-based approach of OODA [30,31]. For example, Karim and Heinze [32] implemented and compared two distinct autonomous UAS controllers. They found OODA is more intuitive and extensible than the control system approach and allows asynchronous processing of information between stages. The work of Karim and Heinze was limited to a simple mission of choosing an alternative waypoint. Priorities in developing a framework include design intuitiveness, extendability, and scalabilities. In our framework, the OODA approach is used to control the UAS.

UASs have been explored for search and rescue (SAR) missions in other related work [33–35]. Hoai and Phuong [36] studied anomaly color detection algorithms on UAS images for SAR work. Niedzielski et al. [37] conducted SAR field experiments to guide ground searchers in finding a lost person. In these studies, the focus was limited to identifying the targets from UAS images, and the computation was primarily performed at an offboard computation infrastructure. Some researchers have also documented that the communication delay between the UAS and the offboard computation infrastructure is a severe limitation. However, our focus is to develop an autonomous system that can perform onboard detection in real time and closely approach the detected target to perform a close inspection for more informed decision-making.

The assessment of autonomy level is an important aspect to compare and evaluate the performance of the system presented in this paper. Several models have been proposed to assess the autonomy level of the systems, including autonomy level evaluation [38] formulated by the National Aeronautics and Space Administration (NASA), the Autonomous Control Levels (ACL) method [39] proposed by the U.S. military, and the Autonomy Levels for Unmanned Systems (ALFUS) framework [40] by the National Institute of Standards and Technology (NIST)—each model has its drawbacks [41].

The autonomy level evaluation formulated by NASA is used in this research as it is simple and easy to use for comparison of unmanned systems platforms, regardless of their operating environment.

In summary, the task of vision-based multiple ground target finding and action involves solving problems in detection, tracking, navigation, and action on a set of discrete ground targets where visual uniqueness of the targets cannot always be guaranteed, and targets may also move out of the camera's field of view (FOV).

3. Framework

The main contribution of this paper is a navigation framework. This framework was developed and implemented as a modular software system. The current implementation of the framework has six modules: main module, image capture module, a target detection module, mapping module, autopilot driver module and external sensor module. Figure 2 describes the software system architecture.

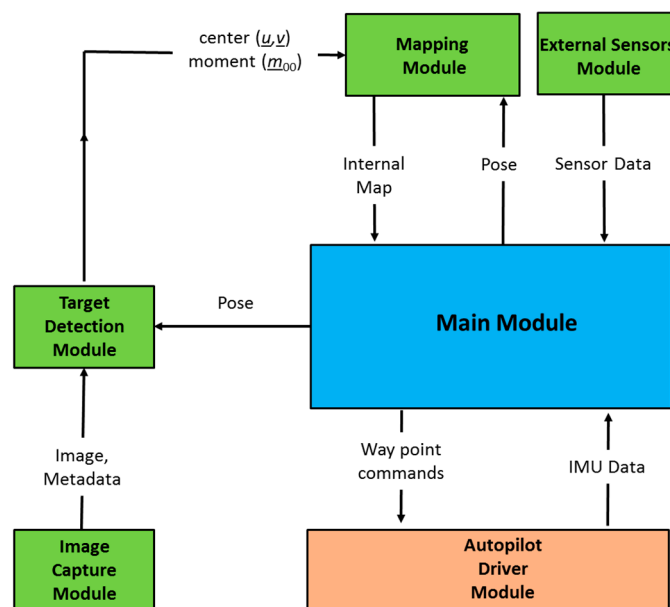


Figure 2. Software system architecture.

The image capture module (Section 3.1), target detection module (Section 3.2), mapping module (Section 3.3), and the main module (Section 3.5) were developed in C++, and Mavros [42] is used as the autopilot driver module. It is available as a ROS [43] package. The main module performs the orient and decide functions of the OODA loop. The following subsections describe each module.

3.1. Image Capture Module

The image capture module captures images from the attached camera sensor and sends the captured images and image metadata, such as timestamp and resolution, to the target detection module.

3.2. Target Detection Module

The target detection module (Figure 3) detects targets and extracts interesting features of the target, such as the center and size, and passes the information to the mapping module. In a ground target finding and action mission, the UAS flies through different sets of altitudes depending on which step (Figure 1) the UAS performs. A static target detection algorithm may not perform well for all altitudes. Therefore, the target detection module receives the UAS pose information from the main module and adjusts the detection method according to the altitude of the UAS for more accurate detection.

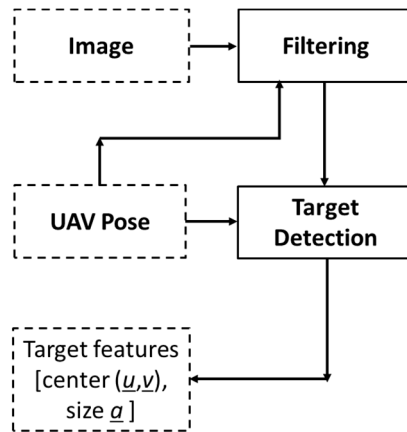


Figure 3. Target detection module.

The target detection module performs two different functions: filtering and target detection. The filtering function rejects unsuitable images from the processing pipeline and helps to reduce some processing cycles of the onboard computer. The target detection error increases with the rotation rate of the UAS. Therefore, the criterion given in Equation (1) is used to discard the images.

$$\exists n \in \mathbb{Z}^+, \quad |\phi_n - \phi_{n-1}| > \alpha \vee |\theta_n - \theta_{n-1}| > \alpha \vee |\psi_n - \psi_{n-1}| > \alpha \quad (1)$$

Here, ϕ_n , θ_n and ψ_n are the roll, pitch and yaw angles of the UAS in the n th frame, and α is the threshold value determined experimentally. UAS images might also be susceptible to motion blur depending on the camera and UAS speed. Therefore, the filtering function can be implemented to include automatic blur detection algorithms described in [44] to filter those images, which is beneficial when more complex target detection algorithms are used. However, our current implementation did not use these methods, considering the processing overhead.

3.3. Mapping Module

The mapping module creates an internal map of all detected targets. The map helps to track the targets when they go outside the FOV. The map is also used to compensate for the effects of localization errors due to GPS noise and drift. Figure 4 shows an image (Figure 4a) and the corresponding internal map (Figure 4b) in a graphical illustration. In Figure 4a, the fourth target T_4 is outside the FOV.

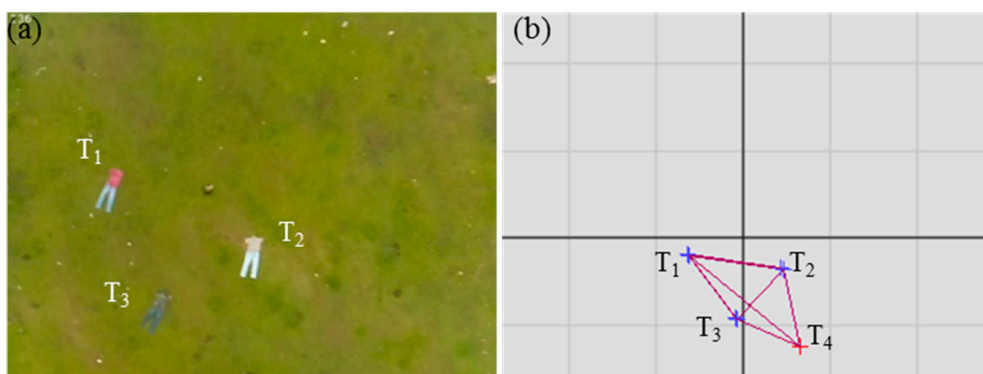


Figure 4. Image with targets (a) and corresponding internal map in graphical illustration (b).

The mapping module creates an internal map through the following steps:

3.3.1. Estimate Target Position

The estimate target position step estimates the 3D position of the target in the inertial frame from the outputs of the target detection module. Our previous paper [27] describes the detailed steps of the method.

3.3.2. Track Targets

The track targets step matches the target between each image frame. The estimated position of each target may vary throughout the mission due to inaccuracies and sensor drift. However, the variation between adjacent frames is small; therefore, targets are matched using the Euclidean distance between the estimated target positions in adjacent frames.

Target $T_{i,n}$ and target $T_{j,n-1}$ are the same target if:

$$\exists n, i, j \in \mathbb{Z}^+, \sqrt{(\hat{x}_{i,n} - \hat{x}_{j,n-1})^2 + (\hat{y}_{i,n} - \hat{y}_{j,n-1})^2} < d \quad (2)$$

where $T_{i,n}$ is the i th detected target from the n th frame, $(\hat{x}_{i,n}, \hat{y}_{i,n})$ is the position estimation of the i th detected target from the n th image frame, and d is the gating distance.

The filtering stage keeps d within a reasonable limit.

3.3.3. Build Internal Map of Adjacent Targets

When the UAS descends towards a selected target for an action, other targets go out of the camera FOV. However, the target selected for the action can be visible most of the time. An internal map of the targets is built using weighted graph representation to use this property. This map also tracks the locations of the targets already registered, but is undetected in specific frames (false negatives).

The target $T_{i,m}$ and the target $T_{j,n}$ are adjacent if:

$$\exists m, n, i, j \in \mathbb{Z}^+, m = n \wedge i \neq j \quad (3)$$

Figure 5 shows an example of an internal map using four targets. Nodes T_1 – T_4 represent the targets, and the weights d_1 – d_6 represent the distance between them.

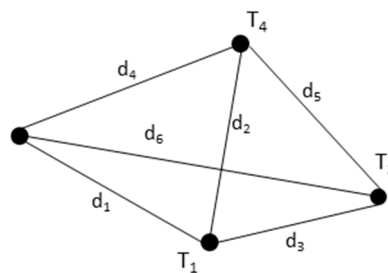


Figure 5. An example of the internal map using four targets.

3.3.4. Remove Duplicates

Due to the errors in estimating the target's position, duplicate nodes (targets) are introduced in the internal mapping step. This step removes these duplicate nodes and their vertices from the map. The node T_i and its vertices are removed if:

$$\exists n, i, j \in \mathbb{Z}^+, \sqrt{(\hat{x}_{i,m} - \hat{x}_{j,n})^2 + (\hat{y}_{i,m} - \hat{y}_{j,n})^2} < d \quad (4)$$

3.3.5. Update Position

Due to detection errors in the target detection module, an already registered target may not be detected in adjacent frames, or the target may go outside the camera FOV when the UAS is flying at a lower altitude. In this case, the positions of the previously registered but not visible (not detected/out of FOV) targets are updated in this step by considering one of the visible targets as a base target. The updated location of the other target is given by:

$$\exists n, i, j, k \in \mathbb{Z}^+, i \neq k$$

$$\hat{x}_{i,n} = \hat{x}_{k,n} + D\hat{x}_{k,j} \quad (5)$$

$$\hat{y}_{i,n} = \hat{y}_{k,n} + D\hat{y}_{k,j} \quad (6)$$

Here the k th target is a base target. $D\hat{x}_{k,j}$, $D\hat{y}_{k,j}$ are the distance between the k th and i th targets along the x and y axis, respectively. Here the distances are estimated using the j th frame.

3.3.6. Remove False Targets

Very accurate target detection is needed for reliable navigation of the UAS because false positives and false negatives may change the UAS's path and make the entire operation inefficient. All target detection algorithms have a certain degree of uncertainty unless the target is very simple. The degree of uncertainty may increase with the distance between the target and the UAS. In an outdoor environment, other factors such as fog, time of day, and lighting changes may also increase this uncertainty.

It is found that false positives are usually not persistent across all images. A weighted voting scheme is used to reject false positives. In this scheme, each target T_i holds votes V_i . If the target T_i is detected in a frame, it casts w_d number of votes:

$$\exists n, i, j \in \mathbb{Z}^+, V_i \leftarrow V_i + w_d \quad (7)$$

where w_d can be a simple constant value or modeled as a complex function of other parameters such as distance, weather, time of day and illumination conditions, which determine the certainty of detection. For example, detection at a lower altitude may be assigned more votes than detection from a higher altitude. The optimal function of w_d is specific to each particular application.

If the target T_i is not detected in a frame but its location is inside the FOV, it casts $-w_n$ number of votes (Equation (8)). The value of w_n can also be determined similar to w_d :

$$\exists i, w_n \in \mathbb{Z}^+, V_i \leftarrow V_i - w_n \quad (8)$$

The target T_i is removed if:

$$\exists i \in \mathbb{Z}^+, V_i < G \quad (9)$$

The value of the G is experimentally determined.

3.4. External Sensors Module

The external sensor module allows the connection of sensors such as sonar directly to the onboard computer for additional state estimation information. In our implementation, only an ultrasonic module is used as an external sensor to compensate for altitude measurement errors induced by strong gusts in barometric height measurement used in typical low cost autopilots.

3.5. Main Module

A finite state machine (FSM) model is used to implement the main module of the system. Figure 6 shows the finite state machine. The state of the system is controlled by following an OODA loop. Here,

a state represents the current belief of the system. According to the observations received from other modules, the system orients and transitions to the next suitable state.

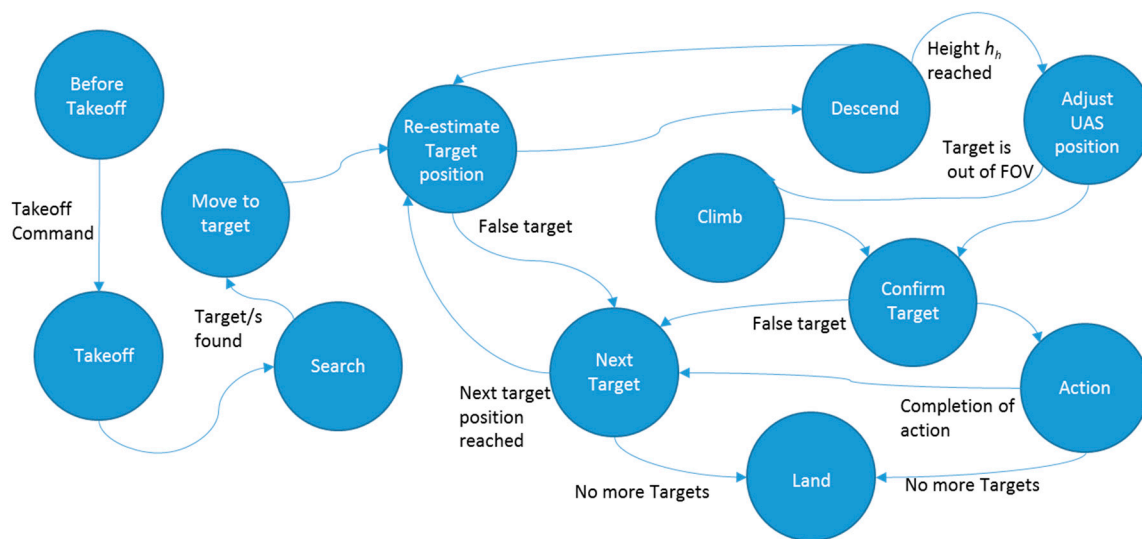


Figure 6. The finite state machine (FSM) of the main module.

If a target or targets are observed in the search state, the system is updated to the *move to target* state. In the *re-estimate target position* state, the UAS moves laterally towards the target and updates the target position. In the *descend* state, the UAS descends by a predefined amount of height. In the *adjust* state, the UAS aligns its x, y position above the target by the proportional controller defined by Equations (10) and (11). This controller also assists the UAS to maintain its hovering position against external disturbances such as wind.

$$x_n = x_c + (U_o - u)K/R_u \tag{10}$$

$$y_n = y_c + (V_o - v)K/R_v \tag{11}$$

In these equations, (x_n, y_n) is the output position of the controller, (x_c, y_c) is the current UAS position and (U_o, V_o) is the optical center of the camera sensor in pixels. (u, v) is the center of the target in pixels, K is proportional gain, and R_u and R_v are the horizontal and vertical resolutions of the image.

In the *climb* state, the UAS increases its height by a small amount (<2 m) and transitions into the *confirm target* state. The *confirm target* state confirms the availability of the target. If the availability of the target is not confirmed, it will be removed from the internal map, considering it as a false positive. Actions such as inspection or spraying are performed in the *action* state. After completing the action, target T_i is selected as the next target if:

$$\exists i, j, l \in \mathbb{Z}^+, (D_i = \min_{j=(1...l)} D_j) \wedge V_i > H \tag{12}$$

where D_i is the distance to the target T_i and V_i is the number of votes held by the target T_i . H is the minimum number of votes that qualify the target as a valid target for a visit. If there is no target to satisfy the criterion (12), the UAS lands at a predefined position.

3.6. Autopilot Driver Module

The autopilot driver module acts as an intermediary between the autopilot firmware and the onboard computer system. This module provides data such as autopilot state, inbuilt sensor data and

battery state to the main module. Commands from the main module are also translated into autopilot specific control commands.

4. Experiments

4.1. Hardware System

A quadrotor UAS was developed to conduct outdoor field experiments using a Pixhawk 2 open source autopilot and a DJI F450 frame. The Pixhawk consists of accelerometers, gyroscopes, barometer sensors and other devices on a single board. Figure 7 shows the hardware system used in the experiments. The hardware system architecture shows the components integrated into the UAS and their interconnection, with images of important components. The Raspberry Pi3 onboard computer, camera and sonar are attached under the UAS.

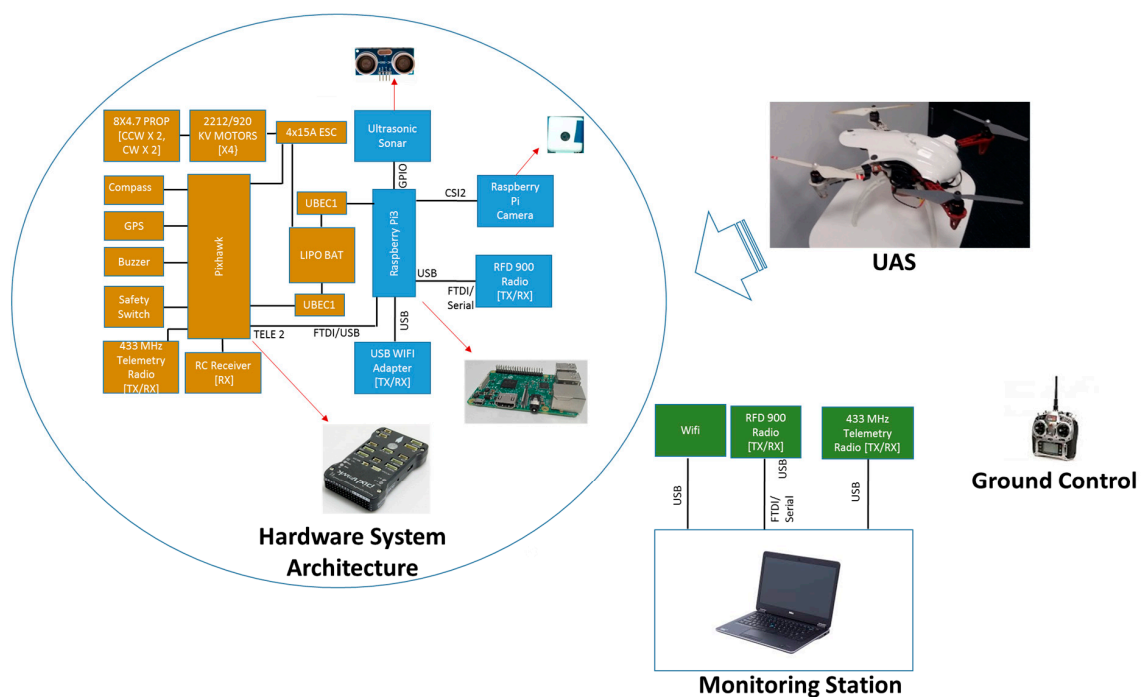


Figure 7. Hardware system used in the experiments.

The implemented framework was fully executed in the onboard Raspberry Pi3 computer. A ground control RC transmitter was used only to trigger the system.

4.2. Field Experiments

As mentioned already, our focus is to develop and demonstrate a framework for vision-based multiple ground target finding and action using a multirotor UAS that is robust to the (i) errors and uncertainties in localization of the UAS; (ii) errors and uncertainties in localization of the target; (iii) target detection errors; and (iv) effects of external disturbances such as wind. In this paper, we have validated our framework using two different test cases: (1) Finding and inspection of various type of objects scattered on the ground in search and rescue after a plane crash. (2) Finding and inspection of multiple red color ground objects.

Figure 8 shows the scenario used for both test cases. The UAS has to take off to the search height h_s (e.g., 40 m) and fly from the waypoint A to the waypoint B and search for targets. If any target/s are detected, the UAS picks a target and changes its flight path, descends and hovers above each target autonomously, as shown in Figure 1, steps 3, 4 and 5. Six targets were scattered randomly. The search boundary was 50 m \times 50 m. The test was repeated multiple times, and the exact positions of the targets

were randomly changed for different tests. However, a minimum separation distance of 4 m was kept between the targets, and the visibility of all targets in the FOV of the camera at search height h_s (40 m) was maintained. Arbitrary shapes in different colors indicates the targets in Figure 8.

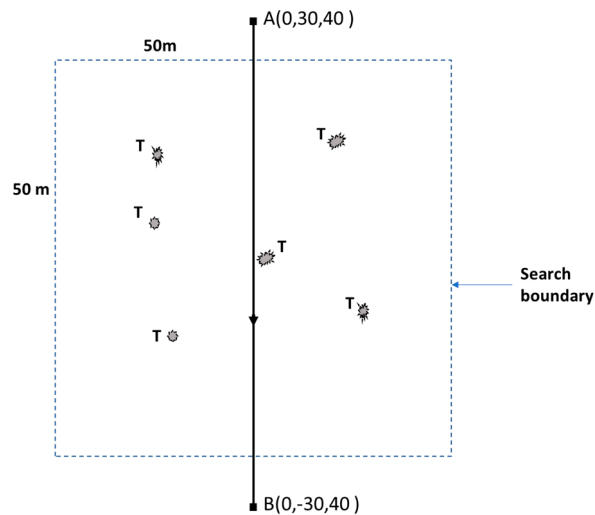


Figure 8. Test case scenario. The dotted square (50 m \times 50 m) indicates the search boundary. T indicates scattered targets within the search boundary. Arbitrary shapes of the targets are used to indicate that the targets are not defined by their shape. Waypoints are indicated by (●).

The experiments presented here used several parameters related to the framework and the hardware system used in the experiments. Table 1 lists these parameter values.

Table 1. Parameters used in the experiments.

Paramete	Value
Search height (h_s)	40 m
Threshold rotation rate (α)	0.8 deg/s
Gating distance (d)	2 m
Camera frame rate	10 Hz
Camera resolution	640 \times 480
Proportional gain (K)	2
Votes for a detection (w_d)	1
Votes for a non-detection (w_n)	-1
Removal cut-off (G)	-2
Valid target cut-off (H)	5

Parameters camera resolution and frame rate describe the camera used. Other parameters, such as threshold rotation rate, gating distance, proportional gain, votes for a detection, votes for a non-detection, removal cut-off and valid target cut-off are related to the framework and determined using simulation and trial-and-error experiments.

These framework-related parameters have impact on the performance of the framework. For example, the gating distance d was set to 2 m in our experiments. Setting a larger value may label two different targets as the same target while setting a smaller value may label the same target in adjacent frames as different targets. The valid target cut-off H was set as 5. Therefore, if a target holds less than five votes, it will not be considered as a possible candidate for a visit or inspection. Hence, a target that appeared in less than five images will be ignored by the UAS. Reducing this value

may increase the chance of false targets to become a valid target. Similar effects are possible with the value for removal cut-off G . These values can be adjusted depending on the accuracy of the target detection method used. For an example, a lower value of valid target cut-off H is adequate for more accurate target detection algorithm.

4.2.1. Test Case 1: Finding and Inspection of Objects Scattered on the Ground in Search and Rescue after a Plane Crash

In this test case, a scenario was formulated to represent objects scattered in a search and rescue mission after a plane crash. The unmanned aerial system (UAS) must autonomously find these targets and conduct a close inspection at a low altitude. This capability is very useful in search and rescue operations in a larger area with limited human resources or in difficult terrains, such as searching for survivors or a plane’s black box. Figure 9a shows objects scattered at the experimental site from the ground view. Figure 9b shows the test site and targets from a UAS image. Images from Malaysia Airlines flight MH17 crash site [45–47] were used as references. Multiple objects such as clothing to represent people, and suitcases and backpacks were used as targets. In this test case, the targets have different shapes, colors and sizes, and the possibility of false detection is relatively high because of the generalized target detection algorithm.



Figure 9. (a) Objects scattered at the experimental site in the test scenario; (b) Top view of the experimental site, and target objects from a UAS image.

In order to detect various types of objects present in a vegetated area, the target detection algorithm illustrated in Figure 10 was implemented in the target detection function of the target detection module (see Figure 3).

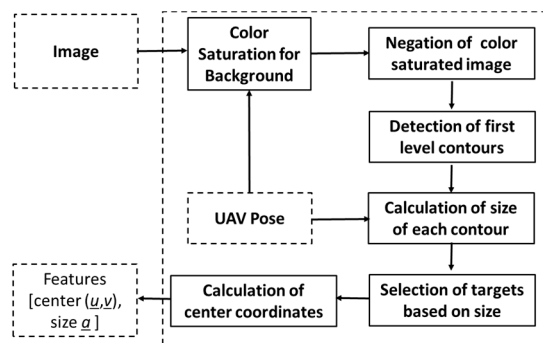


Figure 10. Implementation of the target detection function in the search and rescue test case.

In the first step, the image is saturated for the background color (e.g., green). In the second step, the resulting binary image is negated. In the third step, first level contours are detected. Next, the approximate size a (area on the ground) of each contour is calculated using Equation (13).

$$a = m_{00} \frac{z^2}{f^2} \quad (13)$$

Here m_{00} is the 0th moment, z is the altitude of the UAS, and f is the focal length of the camera. In a binary image, 0th moment m_{00} is defined by:

$$m_{00} = \sum_{u=0}^{u=w} \sum_{v=0}^{v=h} J(u, v) \quad (14)$$

where w and h are the width and height of the blob, respectively. $J(u, v)$ is the value of the pixel at (u, v) .

In the next step, the targets were selected based on size. Very small and large targets were omitted. The size of targets must be configured according to the specific application. The primary step of the detection algorithm is color saturation for the background, such as green vegetation. A simple detection algorithm was selected in order to show the capability of the framework in the presence of a high number of false positives.

Test case 1 was repeated five times. Table 2 summarizes the results. The UAS visited all six targets in three of the five flight tests. One target was missed in two tests because the number of votes acquired by the target was less than the minimum number of votes H to qualify to visit the target. An analysis of experimental data also showed that in three tests, the system (main module) did not activate the action state and considered one of the targets as a false detection. A detailed investigation revealed that this was due to detection failure at low altitude (2–4 m). A possible reason might be the auto adjustment settings of the Raspberry Pi camera used.

Table 2. Summary of results from flight tests.

Test No.	Number of Targets	Number of Targets Visited	Number of Targets Not Inspected (Action Failure)
1	6	6	1
2	6	5	0
3	6	5	0
4	6	6	1
5	6	6	1

Figure 11 shows the selected images from test 5.

Figure 12 shows the results from test 5. In Figure 12a, the UAS searches for the targets while flying from waypoint A to B. At point C, the first target is recognized, and the finite state machine state changes to *move to target* state. The UAS moves towards the first target designated T_1 (step 2, Figure 1) and performs an inspection (hovering above the target for 3 s). Next, the UAS climbs to the cruising height h_c (4 m) and moves to the next nearest adjacent target T_5 . The process repeats until the UAS has visited all six identified targets.

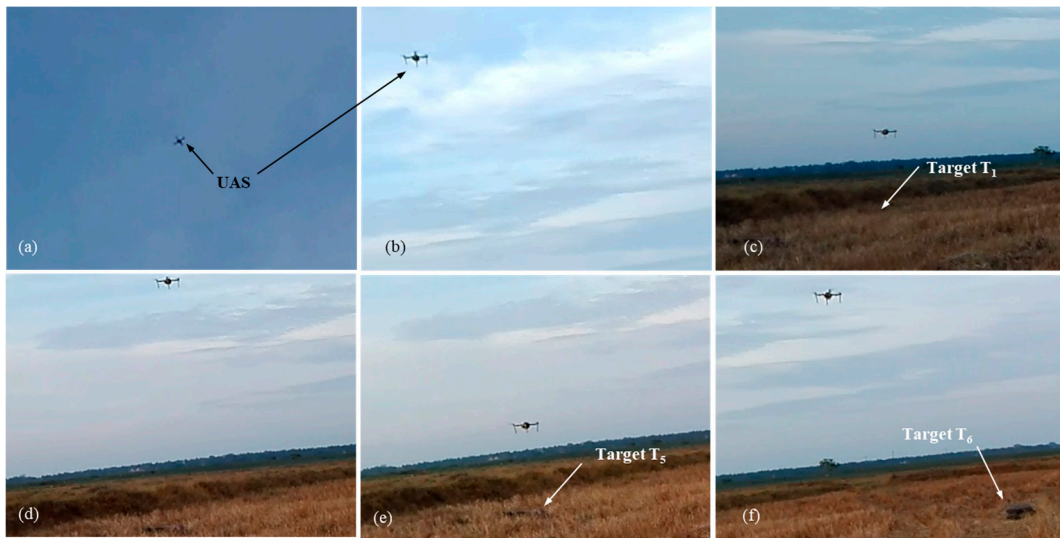


Figure 11. Pictures from test case 1-test 5: (a) UAS searches ground target; (b) UAS descends towards target T_1 ; (c) UAS inspects target T_1 ; (d) UAS moves towards target T_5 ; (e) UAS inspects target T_5 ; (f) UAS moves towards target T_6 .

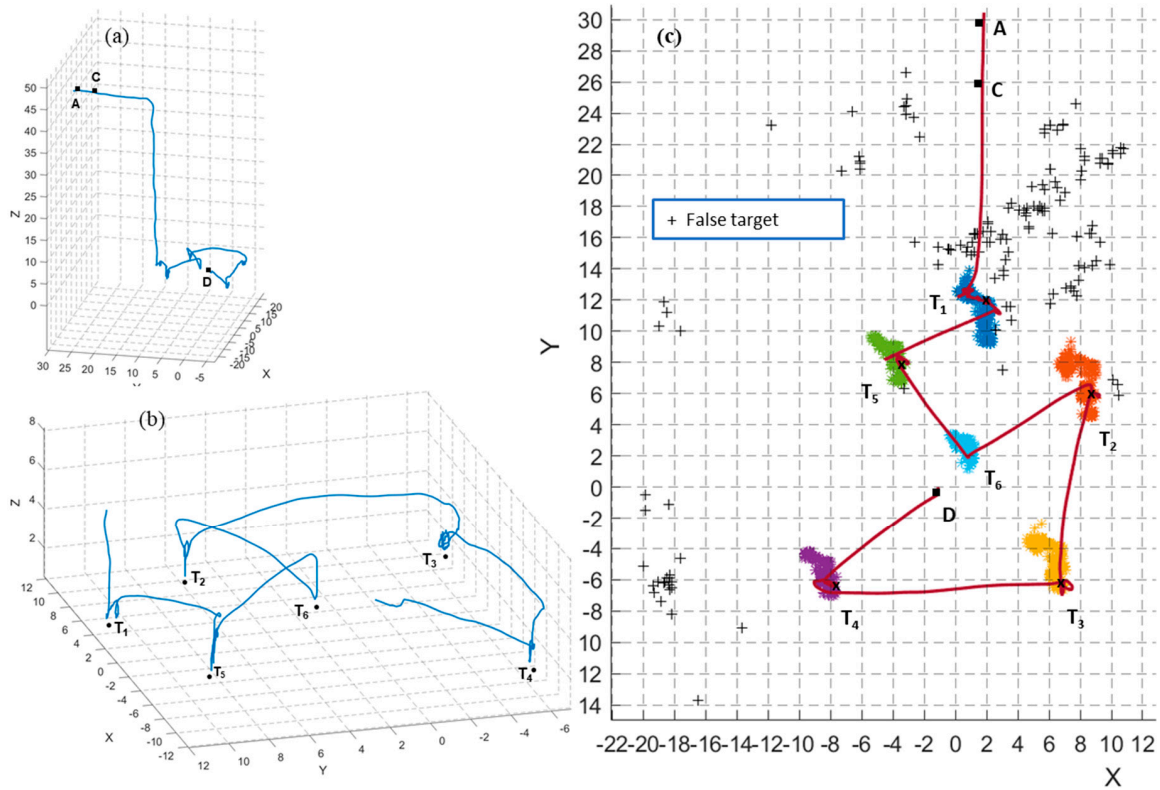


Figure 12. Results from flight test 5: (a,b) 3D view of the flight trajectory in full and zoomed view of the visiting targets and action part; (c) Top view of the flight trajectory and scatter plot of estimated target positions in the full mission of flight test 5. (+) indicates false targets. (x) indicates the estimated position of the target when the UAS was performing the inspection (hovering above the target). Waypoint B is omitted to increase the clarity of the plots.

However, in this test, the action state is not activated for target T_6 . The target detection module failed to detect target T_6 in the confirm state of the finite state machine. T_6 was considered a false

target, and the UAS moved to the next target T_2 . This shows the autonomous re-planning ability of the system without getting stuck with target T_6 .

Figure 12c illustrates the top view of the flight trajectory and the scatter plot of the estimated target positions throughout the full mission. The plot shows several false targets. Moreover, ideally, the position of true targets T_1 – T_6 must appear as points. However, they appear like a cloud due to the variation in the estimates because of the localization errors. The highest variation calculated as the furthest distance between two estimates is 5 m in T_2 . In total, 117 different possible targets were detected throughout the mission, but only six of them—the actual targets—were visited by the UAS. Therefore, the developed framework can overcome localization errors and target detection errors in target finding and action tasks.

4.2.2. Test Case 2: Finding and Inspection of Multiple Red Color Ground Objects

In this test case, the UAS must fly according to the same scenario described in Figure 8 and find multiple red color ground objects and approach them and conduct a close inspection at a low altitude. Six similar red color circles with 0.2 m radius were used as targets. For this test case, only the target detection function of the target detection module was modified. Instead of the detection algorithm described in Figure 10, a simple red color detection algorithm was implemented. Similar targets were used in order to demonstrate that the framework's capability, even in the absence of additional information such as unique color and shape for tracking and control. Figure 13 shows the test site and targets from a UAS image. Figure 14 shows the selected images from one of the tests. Test case repeated five times as before. UAS has visited and inspected all six targets in all five tests. A link to the video of the test is given at the end of the paper. Due to the similarity to the test case 1 results, a detailed discussion of the results is avoided.

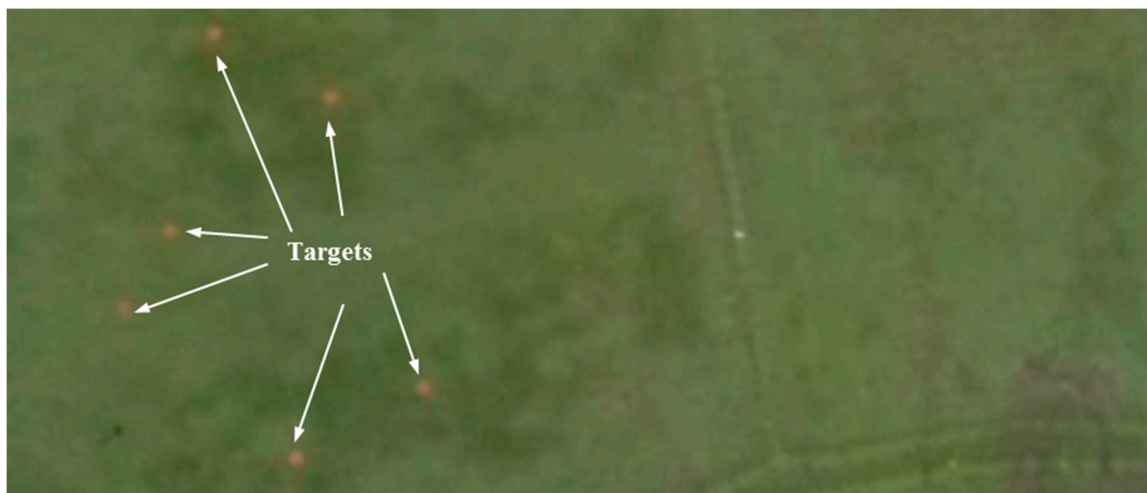


Figure 13. Top view of the experimental site and target objects from a UAS image.

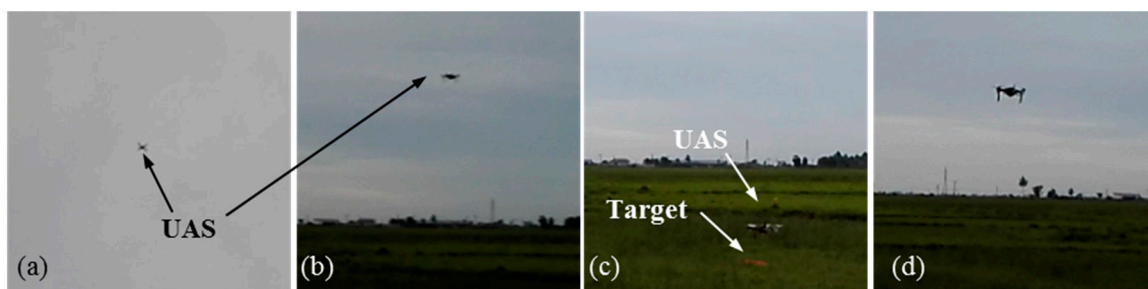


Figure 14. Pictures from a test in test case 2: (a) UAS searches ground targets; (b) UAS descends towards targets; (c) UAS inspects a target; (d) UAS moves towards another target for inspection.

The test case 2 was conducted under heavy rain cloud. However, all the tests were successful is implicates the robustness of the framework to such conditions as well.

5. Discussion

To our knowledge, no previous work has attempted multiple ground target finding and action using UAS. Related work on vision-based landing provides some insights. However, the techniques used in those studies are mostly limited to a single target and are only demonstrated at low altitudes (<5 m) where the effects of detection and localization errors have minimal impact. One possible technique for multiple target finding and action is to estimate the targets' positions from the image and guide the UAS towards each target. However, this technique requires a more precise and accurate localization of the target as well as the UAS and a 100% accurate target detection. Given the dynamic nature of the task and the outdoor environment, a 100% accurate target detection is nearly impossible [34,48]. For an example, we observed a large number of false detections when sudden lighting changes occurred due to the movement of rainy clouds. Moreover, image-based target positioning had errors due to factors such as lens distortion, imaging sensor resolution, GPS accuracy, barometric altimeter resolution and target height [49]. GPS sensors used in typical, low-cost UASs are also subject to drift [27]. This study has developed a framework to address these challenges with less computational requirements and validated the framework using two different test cases with different implementations of the target detection module. The navigation framework proposed in this paper can address these challenges independent from detection algorithm thus a reasonable accuracy from detection algorithms is adequate to perform the task.

When we consider one potential application of the framework such as search and rescue, in earlier works [33–37], UAS was limited in its use as only as a monitoring device. The framework proposed in this research further extends the capabilities of UASs to more active participation, such as conducting a detailed closer inspection or delivering emergency kits. In other studies on railway semaphore inspection [23] and pole inspection [24], the UAS was manually positioned at a workable distance, and researchers did not address the part of finding and approaching the target for inspection. The framework proposed in this research may be used in conjunction with those techniques to increase autonomy. This framework has opened the whole new avenue of research where UASs can be used for close inspection of ground targets for more informed decision-making or can be used to perform an action on the target.

The system presented in this paper uses a higher level decision-making approach OODA for control and has no operator involvement from the mission start to end, and can automatically re-plan the task in the absence of identified targets or if new targets appear. This ability meets level 4 autonomy, according to the autonomy level evaluation method of NASA [38]. However, the framework's lack of the capability to avoid obstacles means it cannot perceive the environment widely as required by level 4 autonomy. This requirement needs to be addressed in the future to achieve a fully compatible level 4 autonomy.

The proposed framework has limitations. The voting-based scheme employed to remove false detections assumes that false detections are not always persistent. However, this might not be true for all target detection methods. Techniques used for target tracking assume errors and drift in sensors are gradual over time, which might fail when abrupt changes occur. The framework also assumes the targets are located in reasonably flat terrain. Although this assumption may be appropriate for agricultural land, further research is needed to address targets located in hilly or sloping terrain.

6. Conclusions

This paper proposed a framework for autonomous, multiple ground target-finding and action using a low-cost, multirotor unmanned aerial system. The framework can be effectively used in a variety of UAS applications with suitable detection algorithms in fully and supervised autonomous missions such as inspection, search and rescue and spot spraying in agriculture. Uncertainties and errors introduced by the onboard sensors and external disturbances such as wind pose significant challenges

in real outdoor environments. A framework was proposed using the high-level decision-making approach of OODA, a series of techniques, and a modular software system to address these challenges.

Two different test cases were formulated to search and inspect multiple ground objects scattered on the ground. Each test case was repeated multiple times to show that the framework can overcome the errors and uncertainties and perform the task successfully. The detection algorithm is simple compared to the state-of-the-art algorithm used in precision agriculture or search and rescue work. There are two reasons for the simple algorithm: one is to demonstrate the capability of the framework with a high number of false detections, and the other one is the limited computation capacity of the Raspberry Pi used in our UAS.

Ongoing work focuses on integrating a sprayer system to a suitable UAS platform, identifying a test site, and obtaining approvals to test the system in paddy fields for brown planthopper control application.

A video of the flight tests can be found at https://www.youtube.com/watch?v=QVmHo_uUkFU.

Author Contributions: A.H. developed the software and contributed to methodology, investigation, data curation, formal analysis, validation and writing (original draft). R.R. contributed to supervision and writing (review and editing). J.R. contributed to supervision. F.G. conceptualized the overarching idea and contributed to funding acquisition, project administration, supervision and writing (review and editing). All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by University Grant Commission Sri Lanka grant number ucg/dric/qut2015/seusl/01 and the APC was funded by Felipe Gonzalez.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kontitsis, M.; Valavanis, K.P.; Tsourveloudis, N. A Uav Vision System for Airborne Surveillance. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04), New Orleans, LA, USA, 26 April–1 May 2004; Volume 17, pp. 77–83.
2. Gonzalez, L.F.; Montes, G.A.; Puig, E.; Johnson, S.; Mengersen, K.; Gaston, K.J. Unmanned aerial vehicles (uavs) and artificial intelligence revolutionizing wildlife monitoring and conservation. *Sensors* **2016**, *16*, 97. [[CrossRef](#)] [[PubMed](#)]
3. Villa, T.F.; Salimi, F.; Morton, K.; Morawska, L.; Gonzalez, F. Development and validation of a uav based system for air pollution measurements. *Sensors* **2016**, *16*, 2202. [[CrossRef](#)] [[PubMed](#)]
4. Bandara, P.T.; Jayasundara, M.U.P.; Kasthuriarachchi, R.; Gunapala, R. *Brown plant hopper (bph) on rice In Pest Management Decision Guide: Green and Yellow List*; CAB International: Oxfordshire, England, 2013.
5. Alsalam, B.H.Y.; Morton, K.; Campbell, D.; Gonzalez, F. Autonomous Uav with Vision Based on-Board Decision Making for Remote Sensing and Precision Agriculture. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017; pp. 1–12.
6. Gps Accuracy. Available online: <https://www.gps.gov/systems/gps/performance/accuracy/> (accessed on 6 June 2019).
7. Lieu, J. Australia Is Spending \$200 Million to Make Its GPS More Accurate. Available online: <https://mashable.com/2018/05/09/gps-australia-investment-accuracy/#SsV9QnDCksqI> (accessed on 22 September 2018).
8. Mindtools.com. Ooda Loops Understanding the Decision Cycle. Available online: https://www.mindtools.com/pages/article/newTED_78.htm (accessed on 10 May 2016).
9. Thoms, J. Autonomy: Beyond ooda? In Proceedings of the 2005 The IEE Forum on Autonomous Systems (Ref. No. 2005/11271), Stevenage, UK, 28 November 2005; p. 6.
10. Plehn, M.T. Control Warfare: Inside the Ooda Loop. Master's Thesis, Air University, Maxwell Airforce Base, AL, USA, 2000.
11. Yang, S.; Scherer, S.A.; Zell, A. An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *J. Intell. Robot. Syst.* **2013**, *69*, 499–515. [[CrossRef](#)]
12. Zarudzki, M.; Shin, H.S.; Lee, C.H. An image based visual servoing approach for multi-target tracking using an quad-tilt rotor uav. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 781–790.

13. Greatwood, C.; Bose, L.; Richardson, T.; Mayol-Cuevas, W.; Chen, J.; Carey, S.J.; Dudek, P. Tracking control of a uav with a parallel visual processor. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4248–4254.
14. Vanegas, F.; Bratanov, D.; Powell, K.; Weiss, J.; Gonzalez, F. A novel methodology for improving plant pest surveillance in vineyards and crops using uav-based hyperspectral and spatial data. *Sensors* **2018**, *18*, 260. [[CrossRef](#)] [[PubMed](#)]
15. Perez-Grau, F.J.; Ragel, R.; Caballero, F.; Viguria, A.; Ollero, A. An architecture for robust uav navigation in gps-denied areas. *J. Field Robot.* **2018**, *35*, 121–145. [[CrossRef](#)]
16. Zhang, J.; Wu, Y.; Liu, W.; Chen, X. Novel approach to position and orientation estimation in vision-based uav navigation. *IEEE Trans. Aerosp. Electron. Syst.* **2010**, *46*, 687–700. [[CrossRef](#)]
17. Zhang, J.; Liu, W.; Wu, Y. Novel technique for vision-based uav navigation. *IEEE Trans. Aerosp. Electronic Syst.* **2011**, *47*, 2731–2741. [[CrossRef](#)]
18. Stefan, N.; Bayram, H.; Isler, V. Vision-based uav navigation in orchards. *IFAC Pap.* **2016**, *49*, 10–15. [[CrossRef](#)]
19. Brandao, A.S.; Martins, F.N.; Sonaguetti, H.B. A vision-based line following strategy for an autonomous uav. In Proceedings of the 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Colmar, France, 21–23 July 2015; SciTePress: Colmar, Alsace, France; pp. 314–319.
20. Ramirez, A.; Espinoza, E.S.; Carrillo, L.R.G.; Mondie, S.; Lozano, R. Stability analysis of a vision-based uav controller for autonomous road following missions. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Piscataway, NJ, USA, 28–31 May 2013; IEEE: Piscataway, NJ, USA; pp. 1135–1143.
21. Araar, O.; Aouf, N. Visual servoing of a quadrotor uav for autonomous power lines inspection. In Proceedings of the 22nd Mediterranean Conference on Control and Automation, Palermo, Italy, 16–19 June 2014; pp. 1418–1424.
22. Mills, S.J.; Ford, J.J.; Mejías, L. Vision based control for fixed wing uavs inspecting locally linear infrastructure using skid-to-turn maneuvers. *J. Intell. Robot. Syst.* **2011**, *61*, 29–42. [[CrossRef](#)]
23. Máthé, K.; Buşoniu, L. Vision and control for uavs: A survey of general methods and of inexpensive platforms for infrastructure inspection. *Sensors* **2015**, *15*, 14887–14916. [[CrossRef](#)] [[PubMed](#)]
24. Máthé, K.; Buşoniu, L.; Barabás, L.; Iuga, C.I.; Miclea, L.; Braband, J. In Proceedings of the Vision-based control of a quadrotor for an object inspection scenario. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 849–857.
25. Sa, I.; Hrabar, S.; Corke, P. Inspection of pole-like structures using a visual-inertial aided vtol platform with shared autonomy. *Sensors* **2015**, *15*, 22003–22048. [[CrossRef](#)] [[PubMed](#)]
26. Choi, H.; Geeves, M.; Alsalam, B.; Gonzalez, F. Open source computer-vision based guidance system for uavs on-board decision making. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–5.
27. Hinas, A.; Roberts, J.; Gonzalez, F. Vision-based target finding and inspection of a ground target using a multirotor uav system. *Sensors* **2017**, *17*, 2929. [[CrossRef](#)] [[PubMed](#)]
28. Hinas, A.; Ragel, R.; Roberts, J.; Gonzalez, F. Multiple ground target finding and action using uavs. In Proceedings of the IEEE Aerospace Conference 2019, Big Sky, MT, USA, 2–9 March 2019.
29. Hinas, A.; Ragel, R.; Roberts, J.; Gonzalez, F. A framework for vision-based multiple target finding and action using multirotor uavs. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA, 12–15 June 2018; pp. 1320–1327.
30. Theunissen, E.; Tadema, J.; Goossens, A.A.H.E. Exploring network enabled concepts for u(c)av payload driven navigation. In Proceedings of the 2009 IEEE/AIAA 28th Digital Avionics Systems Conference, Orlando, FL, USA, 23–29 October 2009; pp. 3–15.
31. Sari, S.C.; Prihatmanto, A.S. Decision system for robosoccer agent based on ooda loop. In Proceedings of the 2012 International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 11–12 September 2012; pp. 1–7.
32. Karim, S.; Heinze, C. Experiences with the design and implementation of an agent-based autonomous uav controller. In Proceedings of the 4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, The Netherlands, 25–29 July 2005; Association for Computing Machinery: Utrecht, The Netherlands; pp. 59–66.

33. Doherty, P.; Rudol, P. A uav search and rescue scenario with human body detection and geolocalization. In Proceedings of the 20th Australian Joint Conference on Artificial Intelligence, Gold Coast, Australia, 2–6 December 2007; pp. 1–13.
34. Rudol, P.; Doherty, P. Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery. In Proceedings of the 2008 IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2008; IEEE: Piscataway, NJ, USA, 2008; p. 8.
35. Sun, J.; Li, B.; Jiang, Y.; Wen, C.-y. A camera-based target detection and positioning uav system for search and rescue (sar) purposes. *Sensors* **2016**, *16*, 1778. [[CrossRef](#)] [[PubMed](#)]
36. Hoai, D.K.; Phuong, N.V. Anomaly color detection on uav images for search and rescue works. In Proceedings of the 2017 9th International Conference on Knowledge and Systems Engineering (KSE), Hue, Vietnam, 19–21 October 2017; pp. 287–291.
37. Niedzielski, T.; Jurecka, M.; Miziński, B.; Remisz, J.; Ślopek, J.; Spallek, W.; Witek-Kasprzak, M.; Kasprzak, Ł.; Świerczyńska-Chłaściak, M. A real-time field experiment on search and rescue operations assisted by unmanned aerial vehicles. *J. Field Robot.* **2018**, *35*, 906–920. [[CrossRef](#)]
38. Young, L.; Yetter, J.; Guynn, M. System analysis applied to autonomy: Application to high-altitude long-endurance remotely operated aircraft. In *Infotech@aerospace*; American Institute of Aeronautics and Astronautics: Arlington, VA, USA, 2005.
39. *Unmanned Aerial Vehicles Roadmap 2000–2025*; Office Of The Secretary Of Defense: Washington, DC, USA, 2001.
40. Huang, H.-M. Autonomy levels for unmanned systems (alfus) framework: Safety and application issues. In Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems, Gaithersburg, MD, USA, 28–30 August 2007; pp. 48–53.
41. Durst, P.J.; Gray, W.; Nikitenko, A.; Caetano, J.; Trentini, M.; King, R. A framework for predicting the mission-specific performance of autonomous unmanned systems. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 1962–1969.
42. Ros.org. Mavros. Available online: <http://wiki.ros.org/mavros> (accessed on 2 June 2017).
43. Ros.org. Ros. Available online: <http://wiki.ros.org> (accessed on 6 June 2017).
44. Sieberth, T.; Wackrow, R.; Chandler, J.H. Automatic detection of blurred images in uav image sets. *ISPRS J. Photogramm. Remote Sens.* **2016**, *122*, 1–16. [[CrossRef](#)]
45. Oliphant, R. Mh17 Crash Site. The Long Grass Hides the Carnage. Many Other Photos Cannot in Conscience Be Published. Available online: <https://twitter.com/RolandOliphant/status/999596340949680128> (accessed on 28 June 2019).
46. Valmary, S. Shelling Adds to Mh17 Nightmare for East Ukraine Village. Available online: <https://news.yahoo.com/shelling-adds-mh17-nightmare-east-ukraine-village-193407244.html> (accessed on 28 June 2019).
47. Mh17 Ukraine Plane Crash: Additional Details Revealed. Available online: <https://southfront.org/wp-content/uploads/2016/11/CIMG0997.jpg> (accessed on 28 June 2019).
48. Pena, J.M.; Torres-Sanchez, J.; Serrano-Perez, A.; de Castro, A.I.; Lopez-Granados, F. Quantifying efficacy and limits of unmanned aerial vehicle (uav) technology for weed seedling detection as affected by sensor resolution. *Sensors* **2015**, *15*, 5609–5626. [[CrossRef](#)] [[PubMed](#)]
49. Nisi, M.; Menichetti, F.; Muhammad, B.; Prasad, R.; Cianca, E.; Mennella, A.; Gagliarde, G.; Marenchino, D. Egnss high accuracy system improving photovoltaic plant maintenance using rpas integrated with low-cost rtk receiver. In Proceedings of the Global Wireless Summit, Aarhus, Denmark, 27–30 November 2016.

