A Multi-UAV System for Detection and Elimination of Multiple Targets

Yurii Stasinchuk, Matouš Vrba, Matěj Petrlík, Tomáš Báča, Vojtěch Spurný, Daniel Hert, David Žaitlík, Tiago Nascimento, and Martin Saska

Abstract—The problem of safe interception of multiple intruder UAVs by a team of cooperating autonomous aerial vehicles is addressed in this paper. The presented work is motivated by the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020 where this task was simplified to an interaction with a set of static and dynamic objects (balloons and a UAV), and by a real autonomous aerial interception system of Eagle.One that our team has been working on. We propose a general control, perception, and coordination system for the fast and reliable interception of targets in a 3D environment relying only on onboard sensors and processing. The proposed methods and the entire complex multi-robot system were successfully verified in demanding desert conditions, with the main focus on reliability and fast deployment. In the MBZIRC competition, the proposed approach exhibited the greatest reliability and fastest solution. It was crucial to our team in winning the entire competition and achieving the second place in the intruder UAV interception scenario.

Index Terms—Target detection, Multi-UAV System, UAV Planning, Target Elimination.

I. INTRODUCTION

UAV interception by autonomous Unmanned Aerial Vehicles (UAVs) is a well-known task in the robotics community due to the importance of protecting critical infrastructure and non-flight zones against non-authorized aerial vehicles. One of the problems tackled in this field of research is the autonomous detection of intruder UAVs within a designated perimeter. When the target is detected, it needs to be precisely localized and tracked. An onboard mission planning system then takes care of autonomous following and interception of the target. To be able to protect large areas (e.g. borders, airports, etc.) against multi-UAV attacks, the interception task can be solved by a multi-UAV system. Multi-UAV systems can also increase the overall system reliability due to redundancy and can decrease the resulting elimination time if information is shared between the agents. High reliability and fast mission accomplishment are the key properties of AAIS.

Recent works tackling the detection of aerial vehicles of different shapes mainly rely on vision-based methods [1], [2], [3]. These works focus solely on the detection and visual tracking problem. A common approach for marker-less detection of UAVs includes Convolutional Neural Networks (CNNs) [2], [3], [4]. Other works propose the use of external detection of UAVs includes Convolutional Neural Networks (CNNs) [5], [6], [7], [8]. In a practical realization of an aerial interception [9], a ground-based detection station can be used to improve detection performance. In one of our previous works as part of the Eagle.One autonomous interception project, we proposed an approach for the detection, tracking, following, and interception of a target using a wide-baseline stereo camera [10]. However, due to the required baseline, the dimensions and weight of the UAV platform exceeded the size restrictions of the MBZIRC competition. The MBZIRC tasks are designed to be challenging, in order to push robotics beyond state-of-the-art and provide solutions of real worlds applications required by the society, like in previous works of our group [11], [12], [13], [14], [15].

Unlike our previously applied methods, in this work we tackled the problem of detection and visual tracking of an intruder UAV using a smaller monocular vision-based system. The UAV and the approach presented here were motivated by and tested in Challenge 1 of the MBZIRC 2020 robotics competition. In Challenge 1, a team of UAVs needed to autonomously detect and track an intruder UAV, and also interact with a set of ground targets represented by balloons (see Fig. 1). Thus, a much simpler color segmentation method is proposed throughout this work, exploiting the well-defined appearance of the targets (dimensions, color, shape). Although a CNN could perform such detections, neural networks are not practical in this case due to the limited
computational performance of the onboard systems of small UAVs. Multiple computationally expensive and time-critical subsystems (planning, mapping, control, state estimation) need to share a single dedicated computer onboard. This restricts the use of CNNs in this case. Moreover, the proposed solution enables combining input data from multiple sensors (RGB camera and small-baseline stereo camera) to increase reliability.

Applications with multi-Autonomous Aerial Interception Systems (AAISs) intercepting multiple targets present a higher degree of complexity than tracking a single UAV by a single AAIS as it is necessary to manage target selection and robot coordination. Regarding deployment of multi-robot systems relevant to target hunting, we mention the work of Faigl et al. [16], which was designed for the collaborative search, track, pick, and place scenario of MBZIRC 2017. This task may be considered a 2D predecessor of the scenario addressed here. Other works also propose approaches for planning and gathering information about targets [17, 18]. These methods rely on a priori knowledge of the environment and focus mostly on surveillance and planning for UAVs. However, obstacles close to the targets (see Fig. 1 (c)) and fast interaction with the target objects can increase the probability of UAV malfunction, which is also the case in real AAIS missions. Thus, a reliable system for task reallocation must be employed when using multiple UAVs for multiple targets and to allow task completion if any UAV team-member fails.

To the best of our knowledge, all existing state-of-the-art works focus on subproblems of the intruder localization and elimination task by a single UAV. In contrast, our proposed system goes beyond the current state-of-the-art by designing all the necessary building blocks for onboard execution of the task using multiple cooperating UAVs. We propose a complete methodology for executing multi-robotic scenarios reliably, from the takeoff, throughout the autonomous search and elimination, and to the landing. The proposed solution has been extensively tested in various realistic outdoor environments and time-constrained conditions, such as Challenge 1 of the MBZIRC competition. Thus, our main contributions are:

1) A methodology designed for solving the problem of multiple UAV agents hunting multiple 3D targets. The solution is designed to provide a general approach for the control and coordination of multiple AAIS in real UAV hunting scenarios.
2) A multi-sensor fusion approach for precise control and state estimation to increase the robustness of the overall system.

II. AUTONOMOUS MULTI-UAV SYSTEM

Multi-UAV systems often impose special requirements on control reference generation. In our case, the diversification and modularity of the MRS UAV System made it possible to provide agile constraints, resulting in high maneuverability of the UAVs. The constraints have been empirically selected for each mission profile (i.e. target search, target approach, and target elimination).

Fig. 2 shows the connection between the UAV system and the proposed Mission & Navigation pipeline. The Mission & Navigation pipeline commands the MRS UAV system by providing desired trajectory references \( \{(r_d, \eta_d)_1, (r_d, \eta_d)_2, \ldots, (r_d, \eta_d)_k\} \). A trajectory reference consists of a sequence of desired 3D positions and headings of the UAV sampled at regular intervals. The navigation pipeline also specifies a set of dynamics constraints that serve to limit the speed, acceleration, jerk, and snap of the flight maneuvers produced. The MRS UAV system ensures robust tracking of the desired trajectory (modified by the Model Predictive Control (MPC) tracker to respect the current dynamic constraints) and provides estimates of the UAV state. The general structure of the proposed Mission & Navigation pipeline is shown in Fig. 2 and consists of the following blocks: the Target detection, which is a visual perception module for detecting and estimating the positions of the targets; the Color picker, which is a semi-automatic color selection tool; the Kalman filter, which filters the position estimates of the selected target during the elimination phase of the state machine; and the State machine, which controls the mission execution, selects the current target to be eliminated, and plans the search, approach, and elimination trajectories.

A. Target detection and position estimation

A crucial element of an autonomous robotic system designed for the elimination of intruder UAVs is the target position and state estimation method, which needs to be precise, reliable, and fast. In the case of this work, the balloons are detected from an RGB camera image using a binary color segmentation algorithm to achieve a high frame rate with the limited computational resources onboard. A 3D position of the detected balloons relative to the camera is estimated from their apparent size in the image, optionally using an aligned depth image for better distance estimation and false-positive rejection. The steps of the detection and position estimation algorithm are:

1) The RGB image is segmented pixel-wise using a Look-Up Table (LUT) that maps the RGB color space to...
a binary value, indicating the validity of each color. To determine the set of valid colors \( C \), we created the Color picker semi-automatic tool for selecting an area in an image and generating an LUT of the colors represented in the area. The valid colors may be selected interactively using several methods in the HSV [20] or L*a*b* [21]. The result of the segmentation is a binary image where the value of each pixel represents whether its original color is a valid color of the target.

2) The algorithm, as presented in the work of Suzuki and Abe [22], is used to extract contours from this binary image.

3) Extracted contours are filtered based on several shape descriptors (namely their area, circularity, convexity, and inertia). Contours conforming to the empirically determined bounds of the shape descriptors are considered as detection candidates.

4) The distance of each detection candidate from the camera is estimated based on its known physical dimensions from its apparent size in the image and using a calibrated mathematical model of the camera.

5) If available, a median distance in the area of each contour is calculated from the depth image. However, this distance estimate may be unavailable, e.g. when the target is too far because of a limited range of the depth camera, in which case this step is skipped. The median distance is compared to the distance estimate from the previous step. If the difference is too large, this means that the apparent size of the object in the image does not correspond with its known dimensions and the measured depth. The detection candidate is therefore discarded as a false-positive. Otherwise, the distance estimate from the previous step is substituted by the median distance from the depth image.

6) A directional vector from the camera origin to the center of the contour is calculated for each detection candidate using the calibrated camera model.

7) Relative 3D positions of the detected balloons are calculated using the corresponding directional vectors and estimated distances from the camera. This set of detected target positions \( T \) is the final output of the detection and position estimation algorithm, which is repeated for each input RGB image.

The LUT is generated based on the selected set of valid colors \( C \) and is indexed using the red, green, and blue channels of color. For a 24-bit color representation, which was output from the Realsense D435 stereo camera that was used, this indexing pattern translates to a LUT size of \( 2^8 \times 2^8 \times 2^8 = 256 \times 256 \times 256 = 16777216 \approx 16.8 \cdot 10^6 \) elements. Each LUT element represents the validity (value of 1) or the invalidity (value of 0) of the corresponding indexed color and is saved as one byte. This implementation provided very good segmentation speed (approx. 28 Hz for an image size 1920 px x 1080 px on our onboard i7-8559U CPU and 100 Hz with hardware acceleration using the CPU’s integrated Iris Plus 655 GPU) at a cost of < 17 Mb of RAM space, which can be considered a negligible memory requirement on modern systems, even onboard UAVs.

High segmentation speed using the LUT method enabled our system to process each image from the input image stream with very low latency and without consuming much of the limited onboard processing power. The low latency and high refresh rate of the balloon position estimates are crucial for good feedback when popping the balloons. In contrast to our approach, several CNN-based methods were tested beforehand, but were found to be too slow and too computationally demanding (approx. 4 Hz for the tinyYOLO [23] and for the CenterNet [24] CNN architectures with hardware acceleration) and unnecessarily complex for this task. It is important to mention that this simplification enabled using balloons of known shape and color. In a real UAV-hunting scenario, the vision pipeline will be replaced by a heavy 3D Lidar that was found as the most efficient solution for UAV detection in our research of single AAIS scenario [3], [10].

B. Balloon position filtering

To stabilize the output of the balloon detector and to discard false-positive detections, a position filtering algorithm based on the Kalman filter is applied. It is assumed that the balloons are stationary. Although this assumption is not true in practice, as the balloons often oscillate around their attachment point due to wind, we filter out these perturbations to stabilize the popping maneuvers. The presented method was also tested with dynamically floating balloons tethered to the ground using a rope. The method proved to be robust enough to enable reliable elimination, even under such conditions. A discrete linear stochastic system model is used in the form:

\[
\begin{align*}
    x[k] &= A x[k-1] + v[k], \quad v[k] \sim N(0, Q), \\
    z[k] &= H x[k] + u[k], \quad u[k] \sim N(0, S[k]),
\end{align*}
\]

where \( x[k] \) is a state vector estimate at time-step \( k \), \( A \) is a state transition matrix, \( v \) is process noise, \( Q \) is a covariance matrix of the process noise, \( z[k] \) is a measurement vector, \( H \) is a state to measurement-mapping matrix, \( u \) is measurement noise, and \( S[k] \) is a measurement noise covariance matrix. The state vector \( x \in \mathbb{R}^3 \) is an estimate of the filtered balloon 3D position in the static global coordinate frame \( \mathbb{W} \). The measurement vector \( z \in \mathbb{R}^3 \) is its position, as detected by the method previously described, transformed to \( \mathbb{W} \). Let us define the matrices as

\[
\begin{align*}
    A &= I_{3 \times 3}, \quad Q = \sigma_v^2 I_{3 \times 3}, \\
    H &= I_{3 \times 3}, \quad S[k] = R_m[k] \begin{bmatrix}
    \sigma_v^2 & 0 & 0 \\
    0 & \sigma_y^2 & 0 \\
    0 & 0 & \sigma_z^2
\end{bmatrix} R^T_m[k],
\end{align*}
\]

where \( I_{3 \times 3} \) is a \( 3 \times 3 \) identity matrix, and \( \sigma_v \) and \( \sigma_y, \sigma_z \) are empirically determined parameters. The value of \( \sigma_z[k] \) is scaled according to the distance of the detection as

\[
\sigma_z[k] = \max \left\{ \sigma_{z,\text{min}}, \sigma_{z,\text{coeff}} \cdot \| z[k] - r[k] \| \right\},
\]
The state machine controls the target elimination mission based on data from the various system modules (see Fig. 2). The output of the state machine is a setpoint trajectory for the UAV control pipeline. The state machine is presented in Fig. 3. It manages the search and destroy plan execution and deals with possible false positive detections, hardware issues, and emergencies. Each state can switch back to Init or to a previous state to recover from a false positive detection, or in the event of a module failure (e.g., balloon position estimation or camera failure). To improve robustness and prevent a deadlock of the state machine in the event of a hardware failure or a software glitch, certain states have a maximal execution time. These states are namely Detection confirmation, Target approaching, and Target elimination. Whenever a state exceeds its respective limit, the currently selected target $b_c$ is marked as a forbidden zone and the state is switched back to Init. Detections classified as false positives by the Kalman filter subsystem are also marked as a forbidden zone. A forbidden zone is defined as a sphere with a center $x_f$, a radius $r_f$, and an expiration duration $t_f$. Any detection inside a forbidden zone is ignored. This gives flexibility to the state machine, so that the UAV can skip problematic targets (e.g., when it took too much time to eliminate such a target, or when there was a false positive detection) and can search and focus on other targets.

Furthermore, for each UAV, the vision-based detection pipeline produces estimates of the target positions, which are then filtered by the Kalman filter and are used by the state machine to select a primary target. During the Target approaching state, the UAV approaches the balloon to the distance $d_b$ (see Fig. 4). The delayed descent to the target height during the approach is implemented to reduce the risk of collisions with the metal poles to which the targets were tethered. Similarly, the trajectory during the Target elimination state also changes height based on the distance between the UAV and the target (see Fig. 5).

The position of the target estimated by the Target detection
subsystem is noisy in practice (e.g., \( \approx 10 - 20 \) cm jumps in the estimated position of a single target were observed). Because the estimated position of the target is used for planning the trajectory of the UAV, this noise may lead to positive feedback with the position of the robot. This may then lead to an oscillation of the UAV together with the estimated reference object. To prevent the UAV from oscillating, dead banding of the position reference is used. A neutral zone around the current UAV position is defined, and if the newly-calculated reference position would be inside this zone, only the heading of the UAV is changed and not its position. This modification completely mitigates the oscillations and prevents destabilization of the system.

D. Multi-UAV target hunting coordination

Our proposed approach in the Mission & Navigation pipeline considers the coordination of a multi-AAIS multi-target hunting task. Our coordination strategy divides the search area equally according to the number of UAVs in the group. To ensure high robustness of the system, the redundancy of the UAV team is leveraged to compensate for the potential failure of some of the agents. The UAVs use wireless communication to periodically report their status to other members of the team. When an UAV has to prematurely end its mission, be it due to sensor failure, expended battery, or any other reason, it notifies the other team members and its operational area is taken over by its neighbor with the highest priority. Similarly, when no message is received from an UAV for a duration of \( t_s \), it is assumed that its onboard computer is no longer operational and the same method is applied to compensate for its failure. The algorithm is illustrated in Fig. 6. It can be applied in the general shape of the workspace map in a particular UAV hunting mission.

III. EXPERIMENTAL EVALUATION

In this paper, we propose an approach for perimeter surveillance performed by multiple UAVs. During the experimental evaluation, the testing flight area was shaped as a non-convex polygon 100 meters in length and 30 meters in width, with a 20 m ceiling. The UAV platform designed for the given task is a Tarot T650 quadcopter frame with Tarot motors, which are connected to a Pixhawk 4 flight controller unit and an Intel NUC onboard computer (see Fig. 7). The computer runs the Ubuntu 18.04 LTS operating system. Pixhawk 4 contains several integrated sensors, in-
including a GPS with a magnetometer, a gyroscope, an IMU, and a barometer, which are used for basic stabilization and localization of the vehicle. An Intel RealSense D435 stereo camera is used to provide color and depth images for the detection of the targets. The color images have a 1920 × 1080 pixel resolution and a 69.4° × 42.5° horizontal × vertical field of view (FOV), while the depth images are with a 87.4° × 58° FOV in 848 × 480 resolution. The color and depth images are supplied at a maximum of 30 FPS.

During the experiments, the proposed system was extensively tested under extreme conditions and demonstrated resistance to unexpected situations, such as a sudden strong wind or when the target balloon did not pop, producing a large control error. The graphs in Fig. 6 show the performance of the UAV system in real robot experiments. When the search for a target is performed by two or more UAVs, the motion planning must consider the division of the perimeter. This is shown in Fig. 9, where two UAVs were deployed to search for potential targets. The plot shows the process of scanning the area, spotting the target, executing the approaching trajectory, and then destroying the target, as done repeatedly by two UAVs simultaneously. Videos and data from the experiments are available online.

Numerous simulations and experiments were performed in various environments to evaluate the robustness of the state machine to various failure scenarios. The resulting algorithm worked as planned state-by-state, thanks to the robust design and the built-in fail-safe mechanisms. Furthermore, two performance criteria may be considered for an evaluation of the task under discussion in this paper: reliability and total mission time. The proposed system was deployed more than twenty times in real-world experiments and all targets were destroyed in all the trials. This performance and reliability significantly contributed to achieving second place in Challenge 1 of the MBZIRC 2020 competition and first place in the MBZIRC 2020 Grand Challenge. The results are shown in Table I. Finally, Fig. 10 compares the performance elimination times of individual balloons during the different trials. With both UAVs functional during the entire trial, the 5 balloons are eliminated up to 20% faster than is the case when only one UAV is performing the task.

![Fig. 10: Elimination times of the n-th balloon in different trials. The number of active UAVs during each trial is shown in the legend brackets.](image)

![Fig. 11: A sequence of pictures where UAV performs target elimination during the MBZIRC 2020 trial.](image)

### Table I: MBZIRC 2020 resulting scores.

<table>
<thead>
<tr>
<th>Team</th>
<th>Challenge 1</th>
<th>Grand Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beijing Institute of Technology</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>CTU in Prague, UPENN, and NYU</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>University of Tokyo</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>University of Bonn</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>UPM, UPF, PUT, and CNRS</td>
<td>18</td>
<td>40.5</td>
</tr>
</tbody>
</table>

#### IV. Conclusion

In this paper, a detailed description and evaluation of our solution to MBZIRC 2020 Challenge 1, motivated by multi-target multi-UAV interception task, was presented. The MBZIRC tasks are designed to be challenging in order to push robotics beyond state-of-the-art and provide solutions of real worlds applications required by society. Moreover, a critical point of the MBZIRC is to design reliable, fast-to-deploy systems for the tight time constraint during the trials, with no exceptions being given to weather factors - both of which perfectly meet the requirements of AAIS. To tackle this innovative task, we proposed a multi-robot approach for cooperative search and elimination of multiple targets relying on onboard sensors and computational resources. The system has proven itself to be robust, as it provided repeatable performance in different environments (i.e. terrain, lighting, weather conditions) and has demonstrated the viability of the multi-robot scheme for this task. This system resulted in our team placing first in the Grand Challenge and second in Challenge 1 among 240 competitor teams registered for MBZIRC. The proposed system was designed to be general enough to initiate research of cooperative aerial interception of multiple targets. We are actively contributing to research and development in this field, thereby accomplishing the primary purpose of the competition in bringing novel robotic solutions. To further facilitate the research in this arising field, the software is released as open-source.

**REFERENCES**


