drones

MDPI

Article

Dynamic Target Tracking and Following with UAVs Using Multi-Target Information: Leveraging YOLOv8 and MOT Algorithms †

Diogo Ferreira and Meysam Basiri *

Institute for Systems and Robotics, Instituto Superior Técnico (IST), 1049-001 Lisboa, Portugal; diogocostaf@tecnico.ulisboa.pt

* Correspondence: meysam.basiri@tecnico.ulisboa.pt

† This paper is an extended version of our paper published in IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC2024).

Abstract: This work presents an autonomous vision-based mobile target tracking and following system designed for unmanned aerial vehicles (UAVs) leveraging multi-target information. It explores the research gap in applying the most recent multi-object tracking (MOT) methods in target following scenarios over traditional single-object tracking (SOT) algorithms. The system integrates the real-time object detection model, You Only Look Once (YOLO)v8, with the MOT algorithms BoT-SORT and ByteTrack, extracting multi-target information. It leverages this information to improve redetection capabilities, addressing target misidentifications (ID changes), and partial and full occlusions in dynamic environments. A depth sensing module is incorporated to enhance distance estimation when feasible. A 3D flight control system is proposed for target following, capable of reacting to changes in target speed and direction while maintaining line-of-sight. The system is initially tested in simulation and then deployed in real-world scenarios. Results show precise target tracking and following, resilient to partial and full occlusions in dynamic environments, effectively distinguishing the followed target from bystanders. A comparison between the BoT-SORT and ByteTrack trackers reveals a trade-off between computational efficiency and tracking precision. In overcoming the presented challenges, this work enables new practical applications in the field of vision-based target following from UAVs leveraging multi-target information.

Keywords: unmanned aerial vehicle; vision-based tracking; YOLOv8; mobile target following; multi-object tracking; BoT-SORT; ByteTrack

check for updates

Citation: Ferreira, D.; Basiri, M. Dynamic Target Tracking and Following with UAVs Using Multi-Target Information: Leveraging YOLOv8 and MOT Algorithms. *Drones* **2024**, *8*, 488. https://doi.org/10.3390/drones8090488

Academic Editors: Liuguo Yin and Shu Fu

Received: 13 August 2024
Revised: 10 September 2024
Accepted: 11 September 2024
Published: 14 September 2024

1. Introduction

Airborne data is essential for accurately evaluating open environments, and unmanned aerial vehicles (UAVs) have become vital tools for providing this information quickly and efficiently. With advancements in UAV technology, particularly in autonomous operations, there is an increasing demand for higher-quality, broader and more detailed sensor information, which can significantly impact critical decision-making in various scenarios. This paper focuses on one key autonomous capability: vision-based mobile target tracking and following using UAVs. Tracking and following moving targets from the air has diverse applications, including search and rescue [1], police pursuits [2], and vehicle monitoring [3]. Additionally, this capability facilitates enhanced collaboration between aerial and ground robot systems, such as enabling UAVs to land on ground-based vehicles for recharging or other logistical support [4], or for inspection of moving structures [5]. In such dynamic environments, where multiple targets may be visible, it is crucial for UAV systems to not only track the primary target but also maintain awareness of other relevant objects in the field of view.

The problem of vision-based mobile target tracking (MTT) and following encompasses the detection of the target, followed by a target tracking algorithm that monitors the position of the target over time in the image. Finally, the position of the target in relation to the UAV is computed to give feedback to the controller and allow target following. Currently, there is a divide between two different strategies that perform target tracking in the image: single-object tracking (SOT) approaches, where only the target of interest is detected and tracked in the image, and multi-object tracking (MOT) approaches, where the system detects and tracks all the targets of interest in the image and assigns a specific ID to each of them.

Many works [6–10] have been developed in this field; however, the most accepted approach is to tackle the detection and tracking step of the problem by using SOT, namely the kernelized correlation filter (KCF) algorithm [11]. Correlation-based trackers such as KCF propose one-shot learning and show good performance without GPU acceleration, which makes them very appealing for embedded systems with computational limitations [12]. However, KCF relies on the appearance model of the target being tracked. When occlusions occur, significant changes in target appearance can hinder accurate tracking, potentially leading to tracking failures. Hence, the KCF method is very susceptible to partial target occlusions and can only track the target in the image after it is selected in the first frame, which is usually performed manually [6,7,10]. To partially tackle the problem of recovery after occlusion, an algorithm was developed in [7] that analyses the motion between frames to detect movement indicative of the target. However, this can be susceptible to noise and dynamic environments. Another common approach is to use the Kalman filter (KF) in conjunction with the KCF [9,10]. Additionally, deep learning methods (YOLOv3) can be used to initialize the KCF tracker and redetect the target after a full occlusion, which can be performed if there are no similar targets in the image [9].

On the other hand, MOT methods struggle with camera motion and view changes provoked by the UAV movement but work in dynamic multi-target environments. Recently, some works attempt to address this issue by improving camera motion models [13,14]. MOT applications have the advantage of working under a tracking-by-detection approach, which performs a detection step followed by a tracking step in every frame, instead of the single detection step at the beginning. The MOT approach allows for the consistent use of new deep learning-based detection methods (such as YOLOv8 [15]) in order to increase the reliability of the system as a whole. The accuracy of a tracking-by-detection approach using the previous YOLOv7 version and the state-of-the-art BoT-SORT algorithm has been demonstrated, showing an effective solution for target occlusion and identity switching in pedestrian target tracking, even under poor illumination conditions and complex scenes [16]. Furthermore, a similar system utilizing YOLOv8 and BoT-SORT within a synthetic aperture radar imaging framework has been shown to exhibit high precision in both detection and tracking, with real-time capabilities [17]. To the best of our knowledge, despite recent advancements and the growing application of MOT, there have been limited real-world implementations of these techniques for target following in UAV systems. This research aims to address this gap by demonstrating the potential of MOT for target following and showcasing its ability to manage dynamic scenarios, thereby overcoming some of the limitations associated with current SOT approaches. This approach can enable new applications where multi-target information is a valuable asset, such as easy conditional target switching, crowd following, or enhanced redetection methods.

YOLOv8 [18] was chosen for this work as it is one of the most recent additions to the YOLO family by Ultralytics on 10 January 2023, the original creators of YOLOv5, one of the most broadly used versions of the YOLO family. Moreover, the YOLO family is an ever-evolving field of research with some recent improvements shown with YOLOv9 [19] and YOLOv10 [20].

In addition to detection and tracking, accurately estimating the target's position relative to the UAV is critical for effective target following. The discrepancy between the target's pixel position and the centre of the image frame is often used to guide the UAV's camera or gimbal toward achieving line-of-sight (LOS) with the target [6–9]. Several studies leverage this deviation to adjust the UAV's orientation, either by steering a gimbal [6,7] or through lateral and vertical movements to maintain the target in focus [8,9]. Once LOS is established, maintaining a consistent distance between the UAV and the target becomes crucial. Distance estimation techniques, such as those based on the standard pinhole imaging model combined with extended Kalman filters (EKF), help manage observation noise and improve accuracy in dynamic scenarios [7]. Additionally, some methods adjust the UAV's following speed by calculating the proportion of the target's size in the image, providing another mechanism for precise target following [6,9].

Control algorithms for mobile target following vary across studies, from direct 3D position inputs [8,9] to the use of proportional navigation (PN) with cascade PID controllers [6]. Other works use a switchable tracking strategy based on estimated distance, transitioning between observing and following modes with different control methods [7]. Many existing distance estimation methods suffer from limitations in precise target positioning, highlighting the need for more accurate methods. Incorporating a depth sensing module, when feasible, presents a promising solution to address this issue, as suggested by [6].

This paper presents several novel contributions to the field of autonomous UAV-based target tracking and following systems. Firstly, we propose a vision-based system that leverages multi-target information for robust mobile target tracking and following. Secondly, we perform a comparative analysis of the state-of-the-art YOLOv8 object detection algorithm with leading multi-object tracking (MOT) methods: BoT-SORT and ByteTrack, evaluating their effectiveness in target following scenarios. Thirdly, we introduce a 3D flight control algorithm that utilizes RGB-D information to precisely follow designated targets. Fourthly, to address challenges like ID switches and partial/full occlusions in dynamic environments, we present a novel redetection algorithm that exploits the strengths of multi-target information. Finally, we comprehensively evaluate the proposed system through extensive simulations and real-world experiments, highlighting its capabilities and limitations.

This paper extend our previous work [21], which initially demonstrated the feasibility of our approach in simulation using the MRS-CTU system. Building on this foundation, the current research introduces several key advancements. We have refined the algorithms, leading to improved performance in target tracking and following. Additionally, we present a comparative analysis between the ByteTrack and BoT-SORT algorithms, which was not covered in the earlier study. Most importantly, we validate the system with real-world experiments, marking a substantial step forward from the simulation-based results of our prior work. These real-world results not only confirm the system's effectiveness but also demonstrate its practical applicability and robustness in actual drone operations.

The rest of the work is organized as follows: In Section 2, the proposed system is explained. Section 4 explains the experimental setup for simulations and real-world tests. Section 5 shows the simulation results, and Section 6 shows the real-world experiments. Concluding remarks and future work are discussed in Section 7.

2. Proposed System

The system configuration is shown in Figure 1. The overall system can be split in to four modules:

1. Visual detection and tracking module;
2. Distance estimation;
3. Following flight controller;
4. System mode switcher.

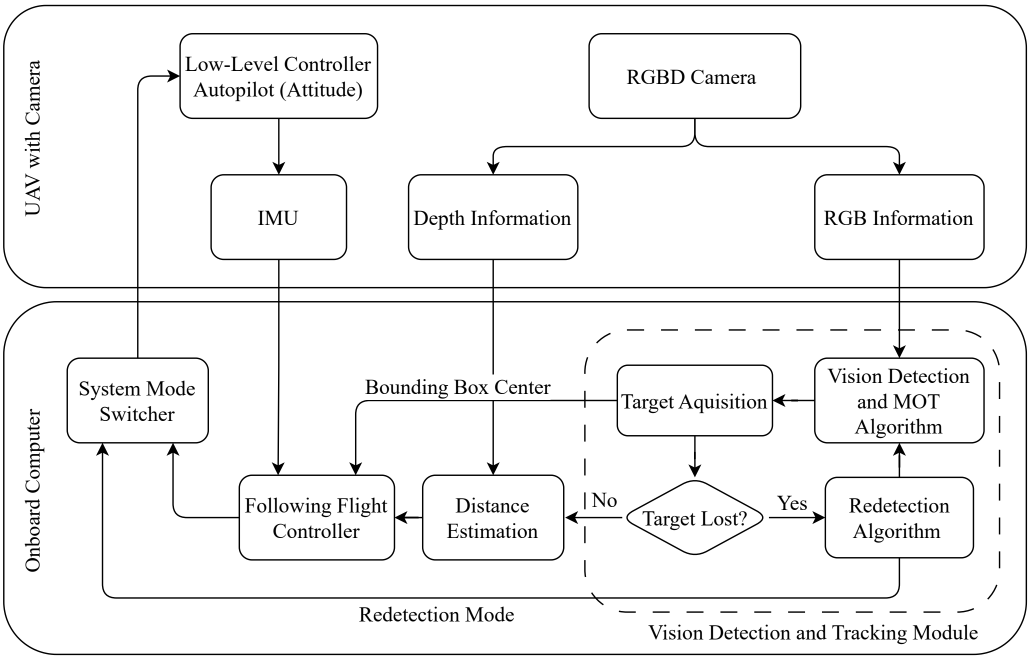


Figure 1. System configuration of the vision-based detection, tracking and following system.

The "vision detection and MOT module" is responsible for detecting and tracking all objects of interest within the UAV's field of view, including the designated target to follow. This module integrates three main components: the vision detection and MOT algorithm, which detects and tracks all objects in the image; the target acquisition stage, where a Kalman filter is employed to estimate the position of the selected target; and the redetection algorithm, which monitors the status of the target and attempts to redetect it in the case of occlusion or loss. The module processes RGB data from the UAV's camera to differentiate between the primary target and other objects, such as bystanders, even during occlusions. When the redetection algorithm is triggered, a "redetection mode" signal is sent to the system mode switcher, which adjusts the control output accordingly.

The bounding box data produced by the vision detection and MOT module is then passed to the distance estimation stage, where the relative distance between the UAV and the target is calculated. This distance can also be determined using data from a depth sensor, such as an RGB-D camera, when available, particularly at larger distances where RGB data alone may be insufficient. Based on the position of the target relative to the image centre and the estimated distance, the controller computes the necessary yaw rate and velocity commands for the UAV. These commands are then relayed to the system mode switcher, which decides the appropriate inputs to send to the autopilot, depending on the current mode of the system. The autopilot controls the attitude of the UAV and sends the IMU data as input to the controller.

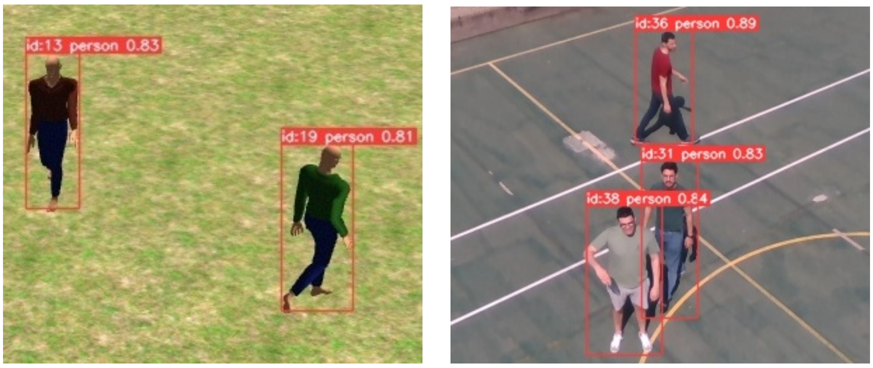
The proposed system will be tested in complex scenarios involving multiple moving pedestrians, where challenges such as occlusions and disturbances are likely to arise. However, the system is designed to be versatile and can be adapted for different objects of interest by modifying the training dataset used for the object detector and adjusting the control parameters as needed. Additionally, alternative distance estimation methods can be explored and applied to other object types, ensuring the system's effectiveness across a variety of applications.

3. Visual Detection and Tracking Module

The first module is the visual detection and tracking module responsible for processing the images from the camera, identifying the target and performing necessary redetections.

3.1. Object Detection and Tracking

The first step in achieving target following involves detecting the target as well as any bystanders in the image and assigning each a unique identifier (ID) for continuous tracking. This is performed using a track-by-detection approach, where a real-time object detector identifies and classifies objects in each frame for subsequent tracking. For the detections, the system takes advantage of the state-of-the-art detector YOLOv8. The YOLOv8 model will analyse the image in real-time and provide a list of detections, highlighting all the objects in the image that are identified with the "person" class above a threshold confidence level. These detections contain: bounding box surrounding the object, class of the object ("person", in this case) and confidence score in that result. An example of a detection scenario from simulation and in real-world scenarios is shown in Figure 2.



(**a**) Simulation detection. (**b**) Real-world detection.

Figure 2. An experimental instance showing detection and tracking frames, including IDs, class ("person"), and confidence scores (ranging from 0 to 1).

The detections (composed of the bounding boxes and classes) are then linked over time by the tracker, by attributing unique identifiers (IDs) for each object, as also shown in Figure 2. Two state-of-the-art trackers are considered: the BoT-SORT MOT and its predecessor, ByteTrack. YOLOv8 provides the detections via bounding boxes to the tracker which in turn performs data association with the previous frame to match each detection with a corresponding ID.

3.2. Target Acquisition and State Estimation

After the take-off and initialization of the detection and tracking algorithm, users can select a target to follow from the list of detected individuals. In this work, the algorithm automatically assigns the first detected person as the target to follow, storing their respective ID for tracking in subsequent frames. To fully leverage the capabilities of multi-object tracking (MOT), the positions of other detected individuals and their assigned IDs are also recorded to improve redetection in case the target is occluded.

After detection, a Kalman filter is used to predict the target's position even during occlusions [10]. The prediction model employed is based on a constant velocity motion model [22], where the velocities are derived from the movement of the target's bounding box centre in the image. The states used are the centre coordinates of the bounding box (C_x, C_y), the size of the bounding box (width - w, height - h) and the speed of movement in the image ($\dot{C}_x$, $\dot{C}_y$), calculated from the movement of the centre of the bounding box pixel coordinates. The states are defined in Equation (1).

$$\boldsymbol{s}(t) = (C_x, C_y, \dot{C}_x, \dot{C}_y, w, h). \tag{1}$$

The observations are shown in Equation (2).

$$\boldsymbol{z}(t) = (C_x, C_y, w, h). \tag{2}$$

The constant velocity model assumes the pixel coordinates of the centre of the target (C_x, C_y) move in the image with constant speed $(\dot{C}_x,\ \dot{C}_y)$ and direction. This allows the Kalman filter to predict the target's position during temporary occlusions. When the target is detected, the Kalman filter is updated with new observations. If the target is not visible, the Kalman filter's predictions are used by the controller to continue tracking, as the target is expected to reappear shortly. If the target does not reappear after a predetermined number of frames, the Kalman filter's predictions are incorporated into the redetection process to evaluate potential candidates for re-identification.

3.3. Redetection Algorithm

In vision-based detection and tracking of mobile targets from UAVs, occlusions pose a significant challenge, especially in densely populated areas. Redetection algorithms are crucial for reliable operations, particularly in scenarios involving multiple individuals. In such systems, the availability of multi-target information is a valuable asset to enhance the redetection algorithms and prevent the loss of the target.

Starting from the beginning of the target following process, the algorithm verifies detections using the assigned IDs from the YOLOv8+BoT-SORT or YOLOv8+ByteTrack setups by matching the received detections IDs with the locally stored detection list IDs. If the assigned target ID is absent in the detections, three possible situations may arise:

1. **Target ID Change:** A limitation of MOT algorithms that occurs when the target is still visible in the image; however, the MOT algorithm assigned it a different identity than the previous frame. This may happen due to sudden movements of the target or the UAV that cause the tracker to misidentify the target. Since the algorithm identifies the target via the assigned ID, an identity change would cause a system failure if left unattended. Hence, it is necessary to update the locally defined target ID with the new assigned ID from the MOT module for continuous target following. The redetection algorithm allows the system to keep accurate target tracking and following, despite the MOT limitations.
2. **Target Missing:** Initiates when the target is no longer detected in the image, potentially due to occlusions or obstructions by other objects, and a counter is initiated. During this step, the system continues to follow an estimate of the position of the target given by the Kalman filter. The brief window where the Kalman filter is used allows for a more robust system that will not immediately stop for short occlusions, like when two people cross paths. The Missing stage also adds some robustness to fast target movements that would cause the target to leave the image, by continuing the corrective manoeuvrer even after the target leaves the field-of-view.
3. **Target Lost:** Declared after a set of consecutive frames where the target is absent, suggesting a potential loss or concealment. To prevent further deviation from the hidden target, the UAV will stop and hover, looking for potential redetection candidates until the target is redetected. In this stage, a relocation strategy could also be considered in future works to position the UAV in a better view to redetect it.

To perform the redetection process, the algorithm searches all detections for possible redetection candidates. Firstly, if the target ID is not found, it will attempt to check if a "Target ID Change" is in place. If no match is found, then it will enter the "Target Missing" mode and later the "Target Lost" mode.

A viable candidate for a successful redetection must fulfil the following conditions:

1. It must represent a new detection not previously tracked, thereby excluding individuals already accounted for as they cannot be the target. This effectively excludes all the bystanders in the area and allows for redetections even in dynamic areas.
2. Candidate detections are evaluated based on a minimum interception threshold with the latest estimated position of the missing target, determined using an IoU approach (Equation (3)). The bounding boxes are scaled to allow for a greater recovery range.

3. Among all the candidates that fulfil step 1 and step 2, it must be the one that scored the highest on step 2.

The IoU approach is a common evaluation method to determine similarity between two bounding boxes in an image. It is given by the relation in Equation (3) where B_A and B_B are the respective bounding boxes which can be visualized in Figure 3.

$$IoU = \frac{Area(B_A \cap B_B)}{B_A \cup B_B} \tag{3}$$

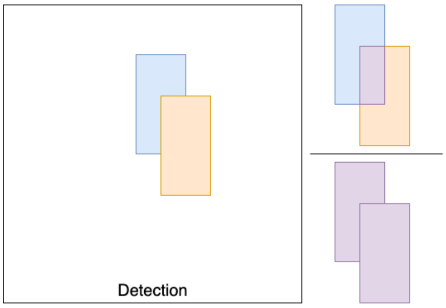


Figure 3. Representation of the IoU calculation.

If the target is redetected during the "Target ID Change" or the "Target Missing" modes, the UAV will immediately resume regular operations and continue following the target. If the redetection happened during the "Target Lost" state, then the system mode switcher will change the input to the controller accordingly, in order to resume target following operations after a complete stop. Further explanation will be given in the system mode switcher section—Section 3.6.

3.4. Distance Estimation

Estimating distance is essential for effectively tracking a target. This estimation can be roughly determined using the detection bounding box from the "visual detection and tracking module" or more precisely obtained using an installed depth sensor. However, depth information is not always available, either due to the absence of a depth sensor or when the target is beyond the sensor's detection range.

In order to estimate distance (d in Figure 4) from the bounding box, we use a relation between the pixel height (h) of the target in the image and a tuned constant value C as shown in Equation (4).

$$d = \frac{C}{h}. \tag{4}$$

The target's height in the image is chosen as the reference instead of the bounding box area ($w * h$) [10] because it mainly depends on the actual height of the target and the distance to the target. Conversely, the width in the image can vary based on movement direction and limb position. The constant value can be pre-tuned for an average human height and adjusted during operations using better estimates from onboard depth sensors. For non-human targets like vehicles or robots, the bounding box area can be used.

To reduce susceptibility to observation noise, an exponential low-pass filter in a discrete-time system is applied as shown in (5). d_{low_pass} is the filtered value, $d(t_{k-1})$ is the previous estimation and $d(t_k)$ is the current estimation. The α value is tuned to prevent higher frequency oscillations in the measurements that could disrupt the controller.

$$d_{low\_pass} = (1 - \alpha) * d(t_{k-1}) + \alpha * d(t_k), \ \alpha \in [0, 1]. \tag{5}$$

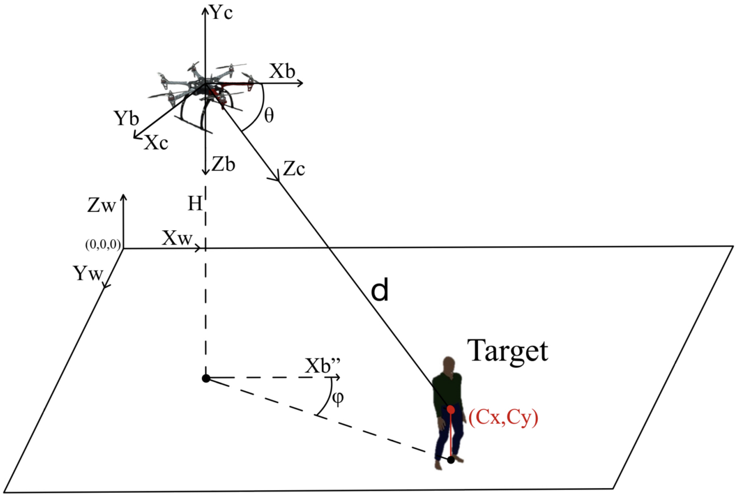


Figure 4. Relative distance to the target d and reference frames - UAV body frame F_b, camera frame F_c and world frame F_w.

3.5. Following Flight Controller

The following flight controller consists of three separate controllers to achieve precise 3D tracking of the target: one for yaw rate, one for altitude (vertical velocity), and one for the horizontal velocity of the UAV.

This approach is inspired by [8,9], using the deviation between the target's pixel location and the image frame's centre as feedback to keep the UAV focused on the target. The yaw rate and altitude controllers are proportional controllers aimed at centring the target in the image. Normalized references for the horizontal, and vertical pixel positions are used (6). The principle point coordinates ppx and ppy are obtained from the intrinsic parameters of the camera.

$$\begin{aligned} X_N &= \frac{C_x}{ppx} - 1, \\ Y_N &= \frac{C_y}{ppy} - 1. \end{aligned} \tag{6}$$

The heading reference for the controller and vertical velocity are then computed using (7) and (8), respectively, where K_ϕ and K_H are the proportional gains.

$$\dot{\phi}_{ref} = K_\phi * X_N. \tag{7}$$

$$V_z = K_H * Y_N. \tag{8}$$

The algorithm uses an aim-and-approach strategy similar to the PN strategy in [6] to achieve target following, by using the yaw rate and altitude controllers to aim and the horizontal velocity controller to approach the target. The control output for the velocity is calculated by using a PI controller taking the estimated distance as input. The horizontal velocity controller error is defined according to (9).

$$error = d - d_{desired}, \tag{9}$$

where the desired distance $d_{desired}$ is defined by the user. To achieve smooth control, the velocity value V is passed through a slew rate limiter, which prevents sudden and aggressive manoeuvrers that could cause a loss of line of sight. The slew rate limiter also ensures the initial control output is zero, allowing for controlled initial movement. The limited rate of change, SR, is defined by (10).

$$SR = \frac{V(t_k) - V(t_{k-1})}{t_k - t_{k-1}}. \tag{10}$$

In addition to the direct distance between the UAV and the target, the horizontal distance is considered a limiter to prevent the UAV from overshooting the target. This distance is calculated based on the current altitude of the UAV, an estimate of the target's height, and the assumption that the target is in the same plane as the measured altitude, as shown in Equation (11). This value can also be compromised by the quality of the altitude measurements, which is why it is only used as a limiter and not as the error feedback for the controller.

$$d_{horizontal} = \sqrt{d^2 - (H - \frac{target\_height}{2})^2}. \tag{11}$$

Figure 5 shows the workings of the high level controller. It takes the information from the RGB-D camera to compute the forward velocity, the vertical velocity and the yaw rate, which are then sent to the FCU—the Pixhawk autopilot.

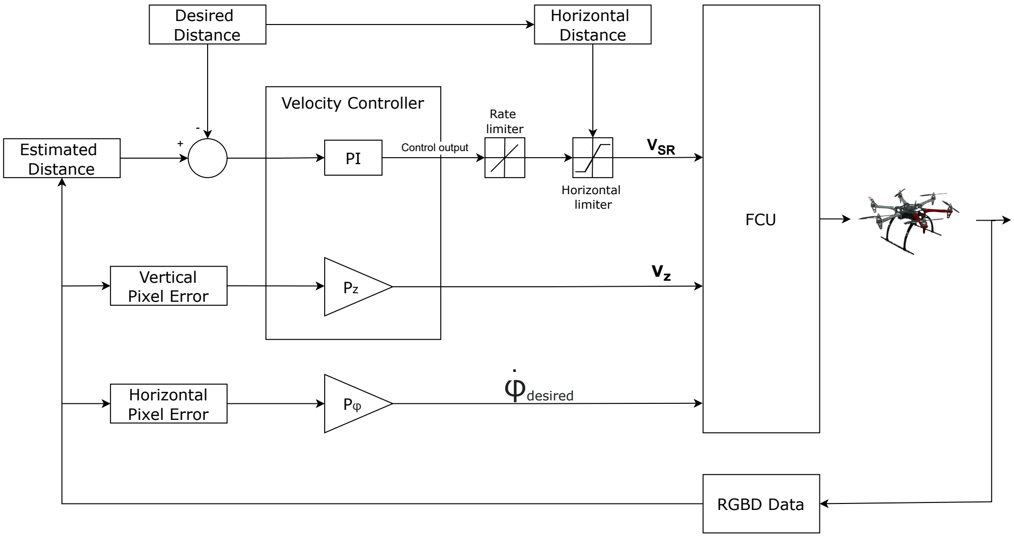


Figure 5. Representation of the UAV high−level controller.

3.6. System Mode Switcher

The system mode switcher is the responsible for providing the low-level controller with appropriate control references based on the mode of the system.

The system can operate in the following modes:

- **Search Mode:** no target has been detected and identified yet. The UAV performs a predetermined search pattern until the target is found.
- **Adjusting Mode:** only the yaw rate and vertical velocity controllers are used to centre the target in the image.
- **Following Mode:** this may be defined as the mode for standard operations, when the target is identified and followed in 3D. This is also the chosen mode if a Target ID Change occurred.
- **Target Missing Mode:** as previously defined in Section 3.3, the target is not visible in the image.
- **Target Lost Mode:** also defined in Section 3.3, if the target has been missing for several consecutive frames it is assumed lost/hidden. Adequate procedures are taken to prevent further deviations from a target hidden behind an obstacle.

Figure 6 presents a flowchart representing the system modes and their respective interactions.

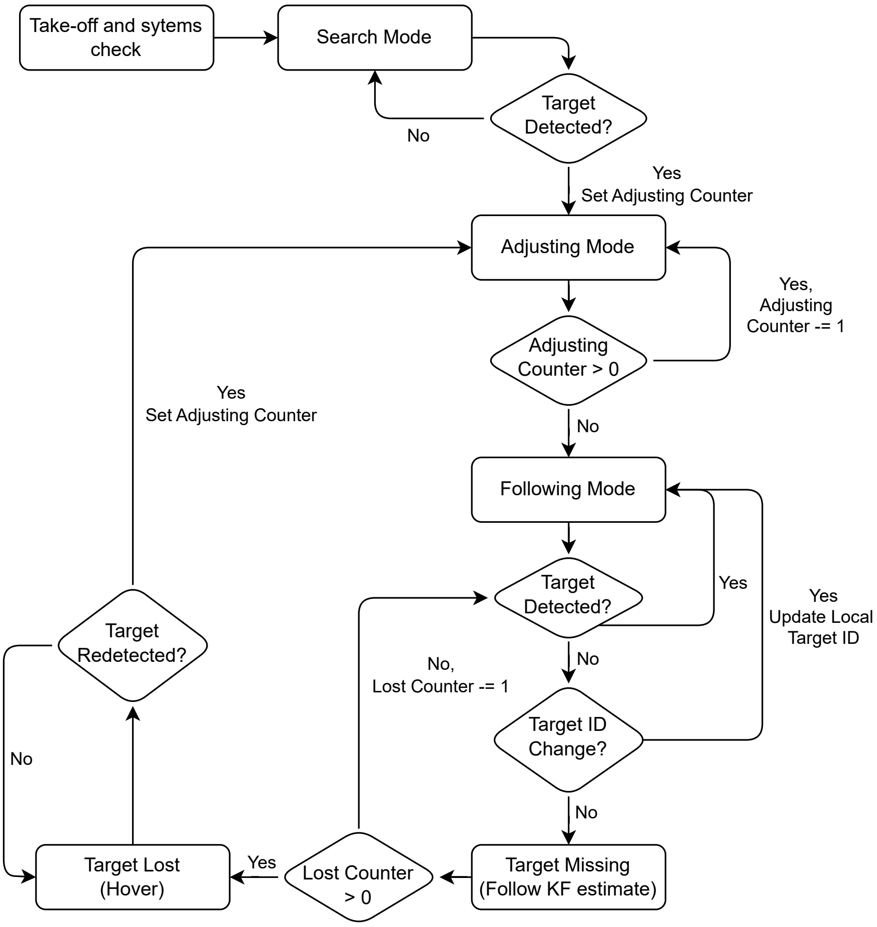


Figure 6. Flowchart of the system mode switcher.

Following take-off and systems check, the UAV will enter "Search Mode". In this mode, the UAV will perform a predetermined search pattern until the target is found. In this work, the UAV will perform a climb until the defined safety altitude and slowly rotate until a person is detected. The first person detected will be set as the the target to follow. The system allows the search mode to be redefined according to the specifications of the mission (e.g., making it possible for the user to select the specific target to follow).

After the target is detected and identified, the UAV will enter "Adjusting Mode", adjusting the target to the centre of the image as best as possible, considering altitude safety limits by controlling the altitude and yaw rate. This allows the system to have a more controlled first approach to the target. After the "Adjusting Mode" timer is over, the system enters regular operations with the "Following Mode", which feeds the output from all the controllers to the autopilot.

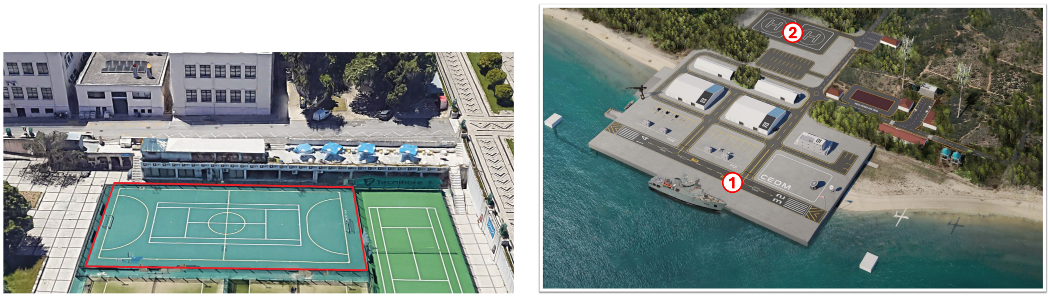
Regarding the first recovery mode—"Target Missing Mode"—the system mode switcher will continue to send the commands from all the controllers however, the estimated distance and subsequently the error will be calculated using the predictions from the Kalman filter.

Finally, in the second recovery mode—"Target Lost Mode"—the system mode switcher will set the UAV to hover. This is performed to prevent further deviation from the lost target, here assumed to be hidden behind some obstacle. The system will remain in this mode until a successful redetection is obtained. Unlike the "Target Missing Mode" that directly shifts to "Following Mode", here, once the target is redetected, the system mode switcher will initiate the "Adjusting Mode" for a brief period and reset the rate limiter and the low pass filter previous values. This will ensure the target is properly redetected and followed.

The github repository with the full implementation can be found in (https://github.com/diogoferreira08/Target-Following-from-UAV-using-MOT.git [23], accessed on 1 September 2024).

4. Experimental Setup

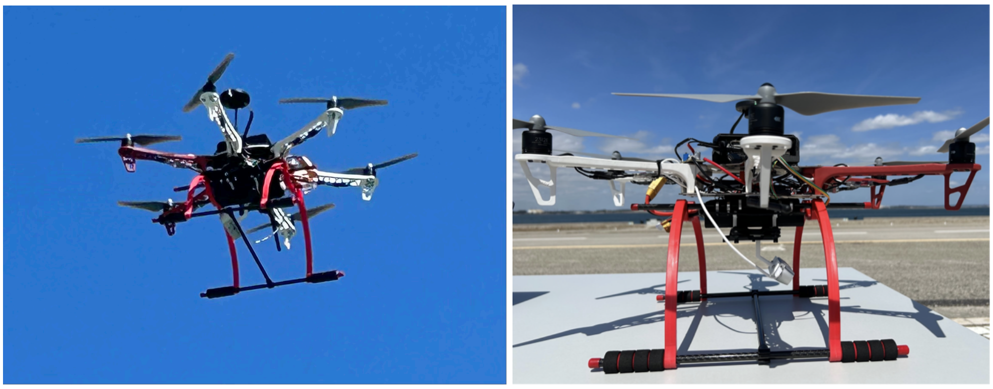
Experiments were conducted in two stages: initial simulation validation of the algorithm, followed by real-world experiments. These real-world tests took place at the football court at IST Lisbon and at the Centro de Experimentação Operacional da Marinha (CEOM), as shown in Figure 7.



(**a**) Football court at IST. (**b**) CEOM at Tróia.

Figure 7. Real-world experimental scenarios.

The UAV platform, shown in Figure 8 was built based on the DJI-F550 Hexacopter frame. For the on-board computer, a NVIDIA Jetson Xavier NX developer kit was used, which is capable of high-power performance on a portable light-weight system. The low-level control and IMU acquisition are managed by a Pixhawk Flight Control Unit (FCU). The Realsense D435 was installed using a 30 degree angle from the horizontal position as shown in Figure 8b. A scheme of the several connections and parts of the system is shown in Figure 9. In simulation, the system is tested using the MRS UAV System [24] within Gazebo.



(**a**) UAV platform. (**b**) Camera angle.

Figure 8. UAV developed for real-world tests.

The Hexacopter has six motors and six electronic speed controllers (ESCs) which control and regulate the speed of an electrical motor by transforming the digital signal from the Pixhawk Cube Orange FCU into a three-phase current. The Power Module powers the FCU and regulates the battery input for the ESCs. The FCU connects to a GPS module powered by VCC and GND cables and communicates using TX, RX cables (transmitter and receiver, respectively). The remainder components use similar connections. The RC receiver connects to the safety pilots controller and in manual mode controls the inputs for the ESCs and motors. The Nvidia Jetson Xavier NX is the on-board computer where the software is implemented. It connects to the FCU using a USB to serial converter and to the camera via USB.

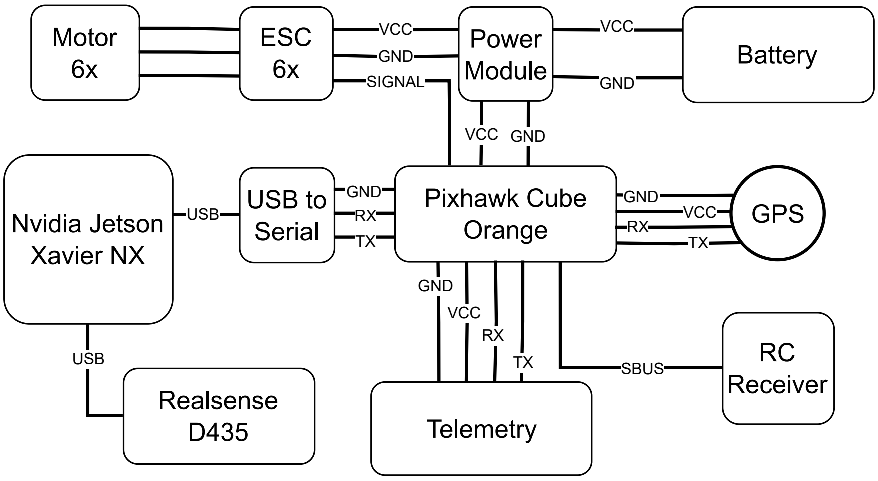


Figure 9. System scheme.

The performance metrics are as follows:

- **UAV and Target Distances**: Total travel distance for the UAV and the target, respectively, in each experiment.
- **Visual Accuracy**: The percentage of frames where the target is correctly identified.
- **Depth Use**: Percentage of time the depth sensor data is utilized for position estimation.
- **Estimation Error**: Average error and standard deviations between the estimated distance d and the real distance measured relative to ROS-Gazebo ground-truth; therefore, it is only used in simulation experiments.
- **Frames per Second (FPS)**: Rate of frames per second that are processed by the visual detection and tracking module, which in turn gives an accurate representation of the computational speed of the algorithm.
- **Number of ID Changes**: Measures target tracking performance, indicating how many times the ID assigned to the followed target changed, excluding corrections by redetection algorithms that adjust the target ID locally.

5. Simulation Experiments

To assess system performance in open environments with multiple pedestrians, five tests were conducted in a 900-square-meter obstacle-free area, featuring three to six pedestrians walking at an average speed of 1 to 1.5 m per second. Each test spans approximately 2 min, with randomized and not pre-set trajectories. The desired distance ($d_{desired}$) is set to 11 m. In order to perform a comparison between MOT methods, the same world conditions (target movements) are replicated in the tests conducted using BoT-SORT and ByteTrack.

Results are summarized for the BoT-SORT tests in Table 1 and ByteTrack in Table 2. Across all the experiments, the UAV successfully tracked and followed the target, covering a total travel distance of 883.5 m, while the target travelled for 1264.6 m. "Depth Use" shows that the depth estimation is used over 50% of the time in all the tests which is expected given the desired distance values of 11 m. Taking into consideration the dynamic use of both estimation methods for distance estimation, the error values are within a range that allows for accurate target following in dynamic scenarios. The standard deviation for the estimation values is relatively high compared to the average error mainly due to the differences in accuracy for both estimation methods (using the bounding box height or the depth information). Regarding computational load, the BoT-SORT algorithms averages around 12.07 Hz ± 1.57 Hz which is adequate for real-time applications. In addition, the BoT-SORT algorithm proves to be very effective in target following, maintaining the target ID in all but the fourth experiment where the "Target ID Change" redetection method had to be used once. In the ByteTrack experiments, there is a substantial tracking performance drop from the BoT-SORT results, which derives from the lack of camera movement compensation enhancements in the ByteTrack algorithm. Despite the higher number of Target ID Changes, the redetection algorithm developed is able to handle the limitations and continue to follow the target under these conditions. Also, the efficiency of the ByteTrack algorithm in comparison with BoT-SORT becomes evident with an average of 17.56 Hz ± 0.96 Hz which translates to a 31.25% decrease in runtime from the BoT-SORT results.

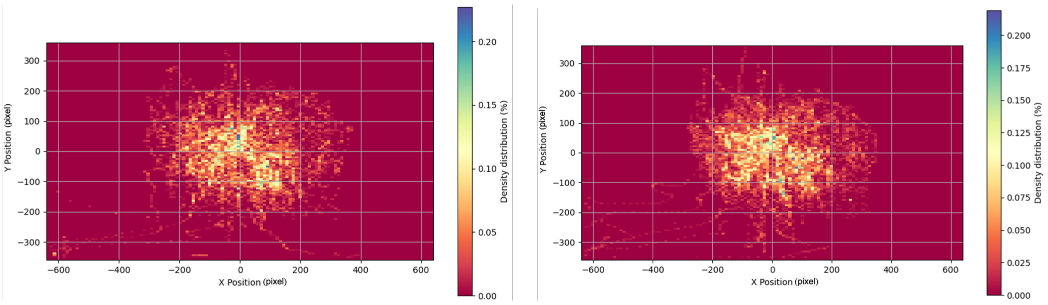
Table 1. BoT-SORT data from randomized tests.

Test No.	Time (s)	Distance UAV (m)	Distance Target (m)	Visual Acc. (%)	Estimation Error (m)	Depth Use (%)	FPS (Hz)	No. ID Change
1	137.3	91.9	123.8	99.38	0.568 ± 0.425	57.96	13.11	0
2	120.8	68.5	112.3	99.72	0.590 ± 0.570	68.43	13.25	0
3	149.5	97.0	147.2	99.71	0.547 ± 0.467	65.16	13.04	0
4	127.1	93.9	130.6	99.00	0.765 ± 0.550	51.73	9.66	1
5	151.8	94.8	118.4	99.67	0.680 ± 0.693	64.42	11.30	0

Table 2. ByteTrack data from randomized tests.

Test No.	Time (s)	Distance UAV (m)	Distance Target (m)	Visual Acc. (%)	Estimation Error (m)	Depth Use (%)	FPS (Hz)	No. ID Change
1	137.1	90.2	123.7	98.85	0.474 ± 0.370	62.72	18.44	5
2	120.1	67.6	110.7	99.26	0.505 ± 0.505	73.61	18.51	7
3	147.7	94.0	146.8	98.08	0.468 ± 0.390	66.91	17.30	10
4	135.1	93.7	136.9	97.64	0.602 ± 0.468	63.71	16.18	7
5	147.4	91.9	118.7	98.86	0.574 ± 0.552	62.13	17.35	10

To evaluate the capabilities of the system to maintain line-of-sight and keep the target in the centre of the image, the heatmap of the target's bounding box centre position in the image was compiled across the five tests for each tracker and shown in Figure 10. Results are similar for both trackers with the target remaining predominantly centred. Target deviations along the vertical axis can be attributed to the lack of gimbal stabilization for the camera, coupling forward/backwards movement with a downward/upwards pitch manoeuvrer. It is believed that introducing camera stabilization would greatly improve these results by decoupling both movements.



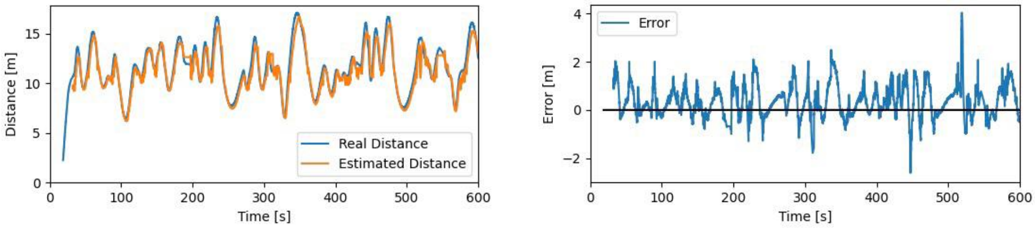
(**a**) BoT-SORT tracking.

(**b**) ByteTrack tracking.

Figure 10. Spatial heatmap depicting the trajectory of the mobile target's centre within the image frame during five experiments. This visualization highlights how the target was consistently tracked and kept within view during the target-following process, showing the areas of highest occurrence over time.

To further evaluate the full system, an experiment was conducted that includes partial and full occlusions up to 10 s where the system goes through the various redetection modes, such as Target ID Change, Target Missing and Target Lost. Similarly to the previous experiments, in this case the same scenario was also ran for both trackers in order to compare results which are expressed in Table 3.

Overall, the results align closely with those of obstacle-free experiments, with the only notable difference being a decrease in "Visual Accuracy." This decrease can be attributed to instances where the target experiences full occlusions during the course of the experiment. The differences between the YOLOv8+BoT-SORT setup and the YOLOv8+ByteTrack are even more pronounced in the long-term experiments with the BoT-SORT tracker performing much better in keeping the IDs of the targets while the ByteTrack tracker has the better frame rate. Considering the several instances of partial and full occlusions and the number of ID changes, especially in the ByteTrack experiments, it can be said that the redetection methods performed well during both experiments, maintaining target following throughout the full experiments. Nevertheless, the robustness of the BoT-SORT algorithm makes it more suitable for target following applications where the dynamic scenarios can provoke unforeseen circumstances. The distance to the target is correctly estimated with an average error of 0.599 m ± 0.502 m for the BoT-SORT and 0.587 m ± 0.640 m for ByteTrack experiments with some higher error when the target changes direction. The estimated distance versus real distance and the respective errors are shown for the BoT-SORT experiments in Figure 11. A demonstration video showcasing the entire experiment using YOLOv8+BoT-SORT is available at the following address (https://youtu.be/YrquRNc5tKM, accessed on 1 September 2024).



(**a**) Estimated vs. real distances from the target. (**b**) Error between real and estimated distances.

Figure 11. Estimated distance and respective error over time for the BoT-SORT experiment.

Table 3. Long term following experimental data.

Tracker Used	Time (s)	Distance UAV (m)	Distance Target (m)	Visual Acc. (%)	Estimation Error (m)	Depth Use (%)	FPS (Hz)	No. ID Change
BoT-SORT	575	366.8	504.2	93.28	0.599 ± 0.502	60.72	9.33	6
ByteTrack	518	348.5	453.0	91.31	0.587 ± 0.640	60.80	15.07	46

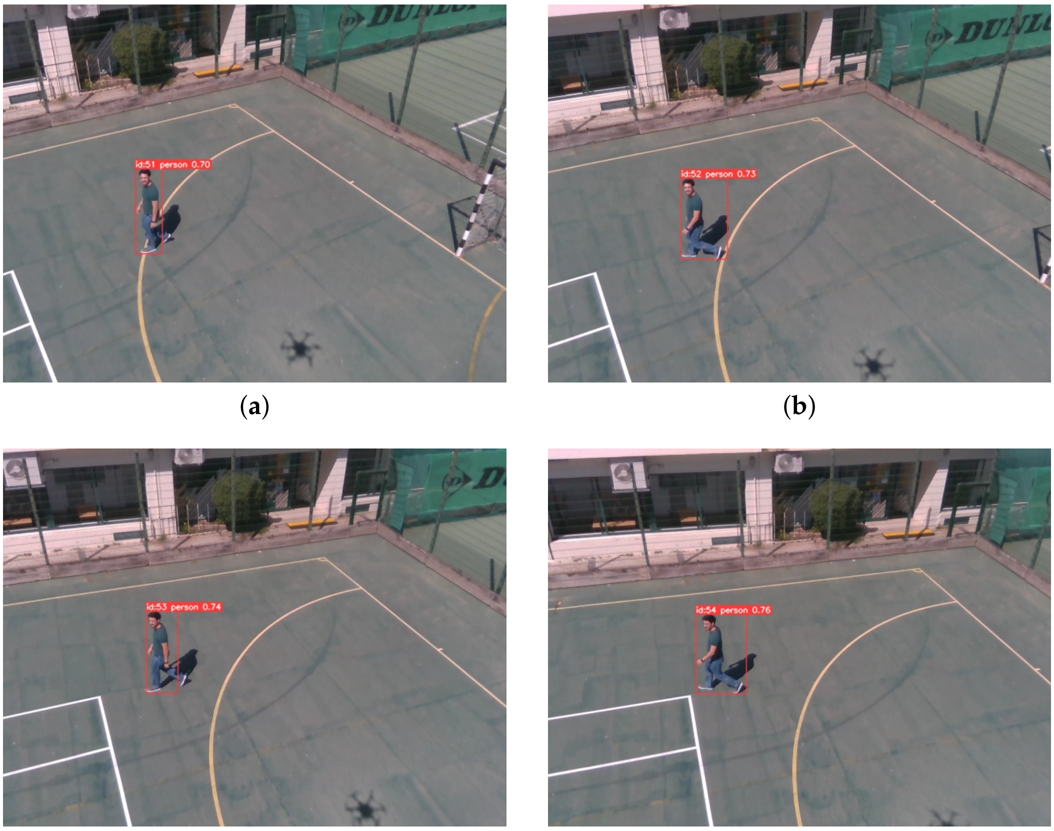
6. Real-World Experiments

To validate the target tracking and following system developed, extensive real-world experiments were conducted. The redetection algorithms were tested at the IST field Figure 7a in a 40 m × 20 m area. Then, the target following capabilities are shown in longer experiments conducted in CEOM Figure 7b. Video footage for the real-world experiments can be seen at this link (real-world video: https://youtu.be/eQuAWoovpI8, accessed on 1 September 2024). Real-world experiments were conducted using a desired distance d of 8 m considering the reduced field-of-view and depth range of the Realsense camera that slightly deviates from the simulations.

Redetection Results

To thoroughly evaluate the redetection capabilities of the proposed system, various real-world scenarios were considered in which the target encountered all redetection modes: Target ID Change, Target Missing, and Target Lost.

First, the effectiveness of the Target ID Change redetection mode was evaluated. As mentioned in Section 2, an ID change is a phenomenon in target tracking where the target is assigned a different identity by the MOT, in consecutive frames. This is more frequent in dynamic environments and can be exaggerated due to sudden camera movements. Hence, the Target ID Change redetection mode is vital step to make the system able to track the target in dynamic environments by overwriting the locally defined target ID with the newly assigned ID from the MOT algorithm, keeping accurate target tracking and following, despite MOT limitations. Figure 12 shows an example of several consecutive Target ID Change redetections where the target is successfully redetected without interference in target following. These results demonstrate that the system is robust to ID Changes, addressing one of the biggest limitations of MOT methods in these applications.

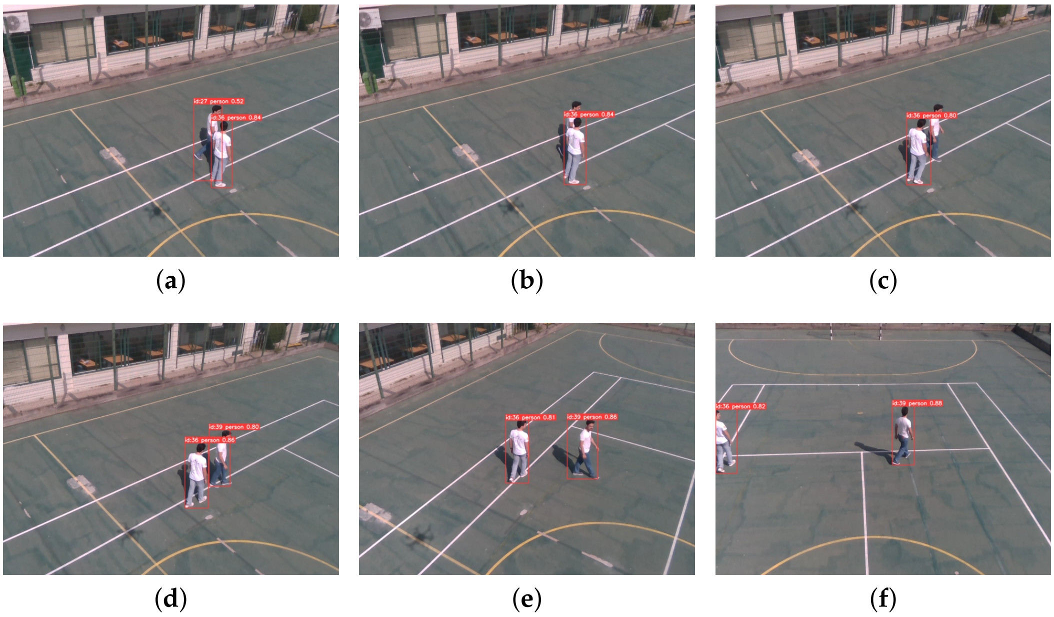


(**a**) (**b**)

(**c**) (**d**)

Figure 12. Target ID Change redetection experiment: system recognizes and adapts to ID changes from the MOT module by updating the locally defined target ID. (**a**) Target followed with ID 51. (**b**) Target ID Change—overwrite local target ID from 51 to 52. (**c**) Target ID Change–overwrite local target ID from 52 to 53. (**d**) Target ID Change—overwrite local target ID from 53 to 54.

Second, the Target Missing Mode was assessed through scenarios with temporary occlusions, since this mode serves as an intermediary phase between the initial detection failure of the target and its categorization as lost. During this phase, an estimation of the position of the target is pursued to navigate temporary occlusions without completely stopping the UAV. Figure 13 shows an instance where the target was obscured as it crossed behind a bystander. Results demonstrate a continuous target following behaviour even during the occlusion period by utilizing the Kalman filter predictions during the Target Missing Mode.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 13. Target Missing redetection experiment: the target is redetected after temporary occlusion when crossing behind a bystander. (**a**) Following the target (left). Bystander in the right. (**b**) Target crosses behind the bystander. Initiate Target Missing Mode. (**c**) Target Missing—following Kalman filter estimate. (**d**) Target (right) redetected and followed. (**e**) Target (right) followed. (**f**) Target (right) followed.

To test more complex redetection scenarios with full occlusions, three progressively more challenging scenarios were considered. Firstly, the target gets behind a non-person static obstacle in the terrain. In this scenario, the capabilities of redetection are tested on a basic level without any interference from bystanders. Figure 14 shows the sequential steps during the first experiment. The system tracks the target in the Following Mode for Figure 14a. During Target Missing Mode (from Figure 14b until Figure 14c) the UAV attempts to follow the missing target by using the Kalman filter estimate. After some consecutive frames without redetecting the target, the system will switch to the Target Lost Mode where all movement is stopped (Figure 14c). Once the target is redetected (Figure 14d) the system mode switcher will activate the Adjusting Mode to centre the target in the image (Figure 14e) which is then followed by the Following Mode (Figure 14f).

In the second scenario, the target walks alongside another person and then hides behind that bystander for a brief amount of time. Here, the redetection algorithm is challenged in its ability to differentiate between a bystander and the target in a redetection scenario (Figure 15).

In the third experiment, the ability to handle dynamic scenes with multiple people is tested by having the target and two more bystanders walking randomly in the same area. The target then hides behind one of the bystanders in the centre while the other keeps moving around in the image (Figure 16c). It can be seen that the system ignores the bystanders and correctly waits for an accurate redetection candidate while the target is hidden. Once the target is redetected in the image (Figure 16d), it is correctly identified and followed (Figure 16e,f).

The target tracking and following capabilities are evaluated for both the BoT-SORT and the ByteTrack trackers during long term tracking experiments in CEOM (Figure 7b). Redetection conditions are also tested with partial and full occlusions. The trajectories of the experiments conducted in the runway are illustrated in Figure 17a,b and for the heliport in Figure 17c,d. Results are shown in Table 4.

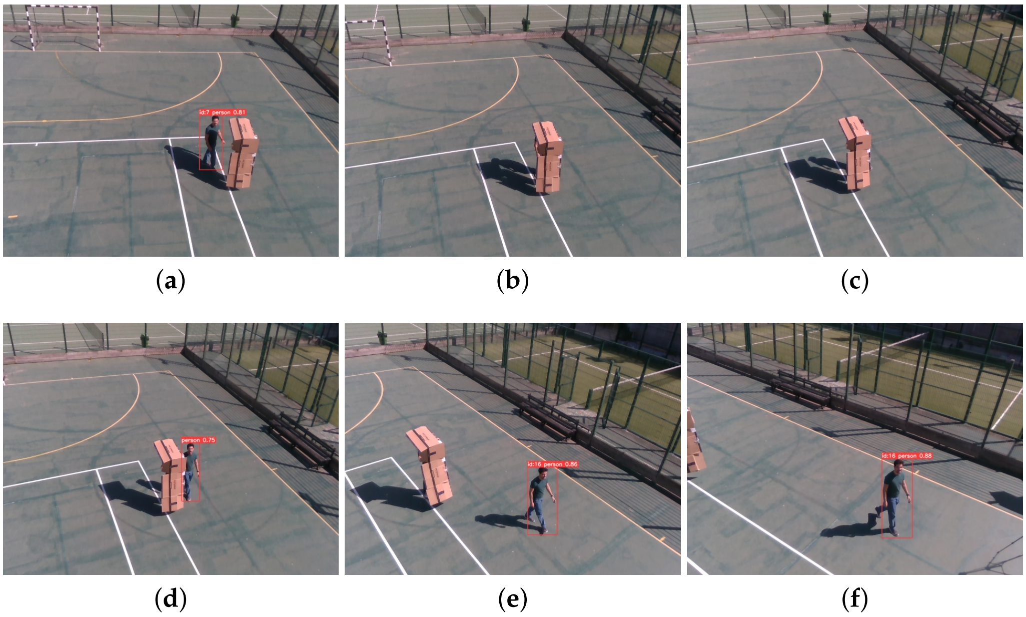


Figure 14. Target Lost Experiment 1—obstacle occlusion redetection scenario. Target successful redetection after full occlusion behind a non-identifiable object. (**a**) Target Following Mode ID 7. (**b**) Target hidden behind bystander. (**c**) Target lost. Stop movement. (**d**) Target reappears. (**e**) Target redetected ID 16. Adjusting Mode. (**f**) Target redetected ID 16. Following Mode.

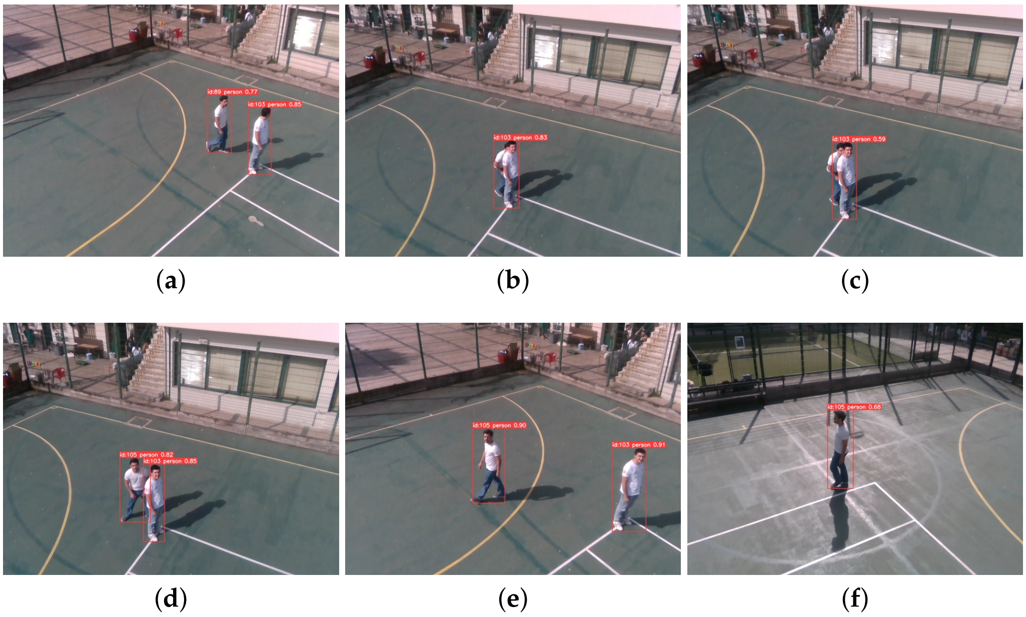


Figure 15. Target Lost Experiment 2—two person redetection scenario. Target successful redetection after full occlusion behind bystander. (**a**) Following the target (left). Bystander in the right. (**b**) Target hidden behind bystander. Target Missing Mode. (**c**) Target Lost Mode. (**d**) Target reappears in the detections. (**e**) Target Adjusting Mode. (**f**) Target Following Mode.

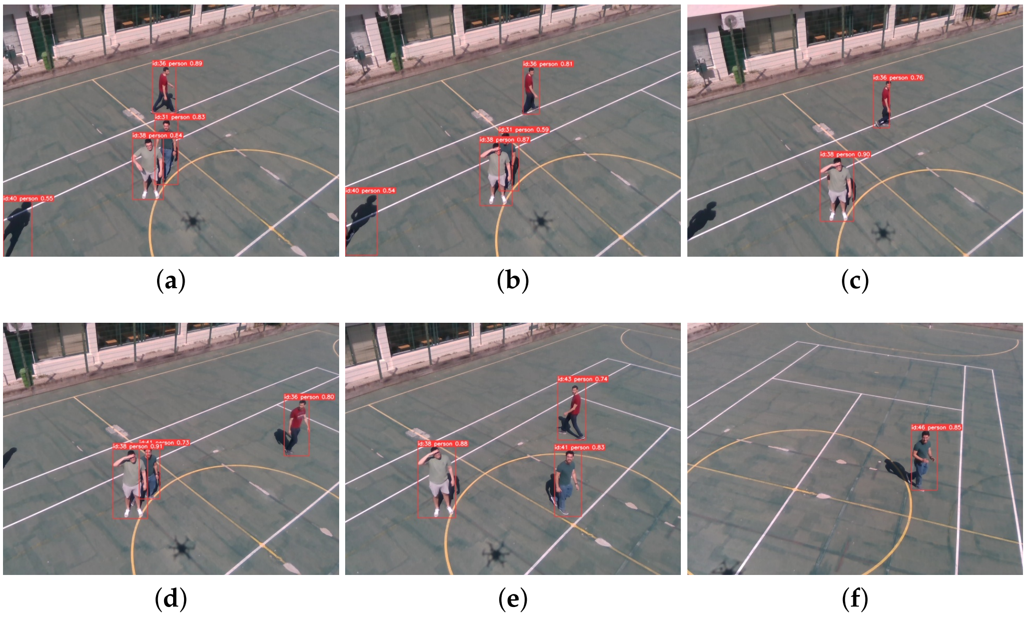
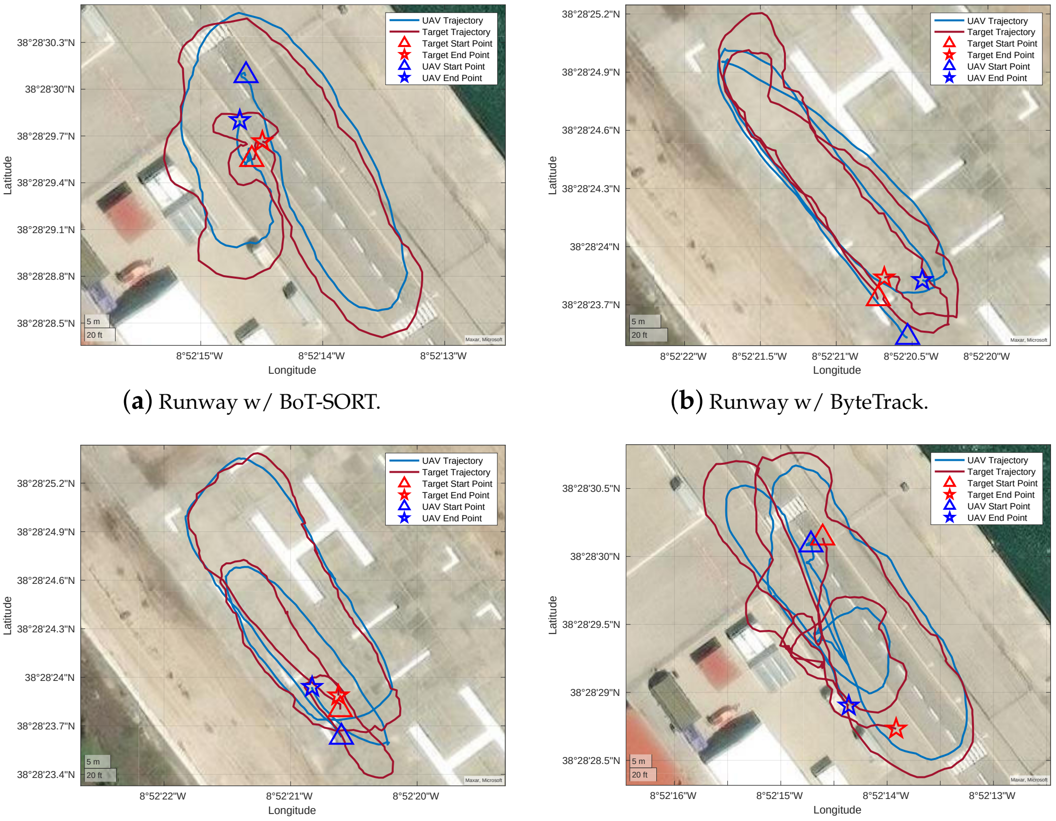

(**a**) (**b**) (**c**)
(**d**) (**e**) (**f**)

Figure 16. Target Lost Experiment 3—dynamic scenario with multiple people. Target successful redetection after full occlusion behind bystander while another person walks around in the image. (**a**) Following target green ID-31. Bystanders: grey ID-38 and red ID-36. (**b**) Following the target (green ID-31). (**c**) Target hides behind bystander ID-38 (grey). Bystander ID-36 (red) moving. (**d**) Target (green ID-41) reappears in the detections. (**e**) Target Adjusting Mode. (**f**) Target Following Mode.



(**a**) Runway w/ BoT-SORT. (**b**) Runway w/ ByteTrack.
(**c**) Heliport w/ BoT-SORT. (**d**) Heliport w/ ByteTrack.

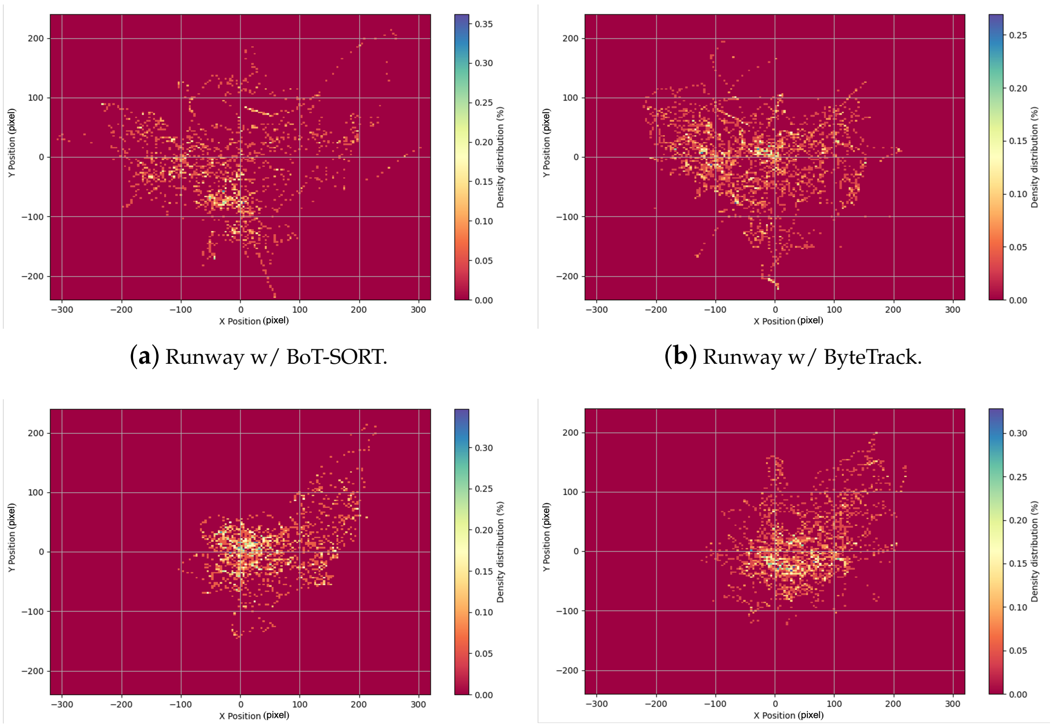
Figure 17. UAV and target trajectories in CEOM.

Table 4. Data from experiments at CEOM.

Exp. Scene	Tracker Used	Time (s)	Distance UAV (m)	Distance Target (m)	Visual Acc. (%)	Depth Use (%)	FPS (Hz)	No. ID Change
Runway	BoT-SORT	274	308	340	92.31	15.63	6.96	13
Runway	ByteTrack	376	439	510	90.27	10.35	8.05	77
Heliport	BoT-SORT	252	228	250	99.69	10.83	7.62	3
Heliport	ByteTrack	320	258	270	88.12	9.60	8.92	118

Over the four experiments, the UAV travelled a total of 1233 m while the target walked for 1370 m. Depth Use values are lower than simulation due to the reduced performance of the Realsense camera in the real world. However, this method still proves effective in preventing the UAV from overshooting the position of the target in the event of partial occlusions where the bounding box information is not reliable. Hence, the use of depth information is a vital cornerstone for the reliability and robustness of the system, even if only used between 9.60% and 15.63% of the time. Comparing both trackers, the trade-off between tracking efficiency and effectiveness is still present with BoT-SORT showing much better tracking results at the cost of slightly higher computational demands. In the real-world scenarios, however, ByteTrack only presents a 14.08% decrease in computational runtime for an average of 10 times more ID Changes which makes BoT-SORT the preferable choice when weighing both metrics. Overall, the system obtains frame rates capable of sustaining real-world performance, although it is noticeable a decrease in efficiency from the simulation experiments, due to lower on-board processing power.

To evaluate the capabilities of the system at maintaining LOS, the heatmaps representing the position of the centre of the bounding box of the target for the four experiments are shown in Figure 18, with the average and standard deviations shown in Table 5.



(**a**) Runway w/ BoT-SORT.

(**b**) Runway w/ ByteTrack.

(**c**) Heliport w/ BoT-SORT.

(**d**) Heliport w/ ByteTrack.

Figure 18. Spatial heatmap depicting the trajectory of the mobile target's centre within the image frame across four experimental runs. This visualization highlights how the target was consistently tracked and kept within view during the target-following process, showing the areas of highest occurrence over time.

Table 5. Average and standard deviation of the target pixel positions during the experiments.

Experiment	X Error (Pixel)	Y Error (Pixel)	Total Error (Pixel)
Runway w/BoT-SORT	76.47 ± 96.75	61.38 ± 74.63	68.92 ± 56.58
Runway w/ByteTrack	76.86 ± 88.27	47.11 ± 62.62	61.98 ± 50.26
Heliport w/BoT-SORT	63.21 ± 68.80	35.06 ± 47.54	49.14 ± 48.30
Heliport w/ByteTrack	61.18 ± 63.54	42.63 ± 56.33	51.91 ± 45.02

Firstly, a distinction is observed between the runway and heliport experiments, where the latter shows overall better results in maintaining line-of-sight. This difference can be attributed to the nature of the experiments conducted: while in the heliport, the target followed a circular trajectory with little speed variations and direction changes; in the runway, the trajectory was more erratic with speed and direction changes. In the heliport experiments, there is a visible shift in the heatmap distribution to the right, caused by the clockwise direction of the movement of the target while in the runway the distribution is more evenly spread. It can be determined that the horizontal deviations are mainly caused by the sideways movement of the target in relation to the UAV when turning. Investigating this correlation further, Figure 19 shows a slight delay between the yaw rate reference given by the developed yaw rate controller (in blue) and the actual actuation of the UAV (in orange). These limitations of the FCU can be minimized by fine tuning the PID parameters and filters of the autopilot.

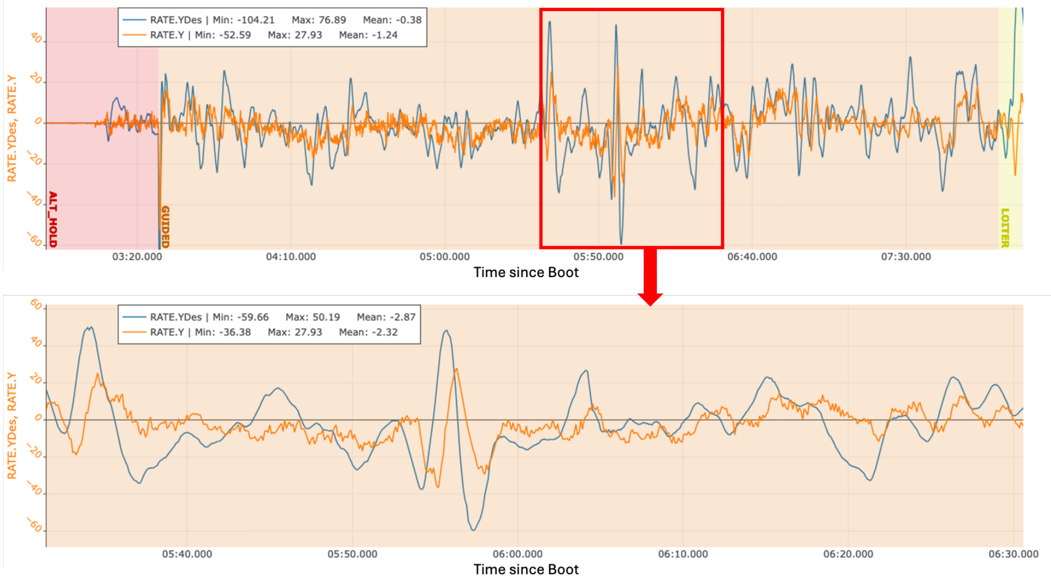
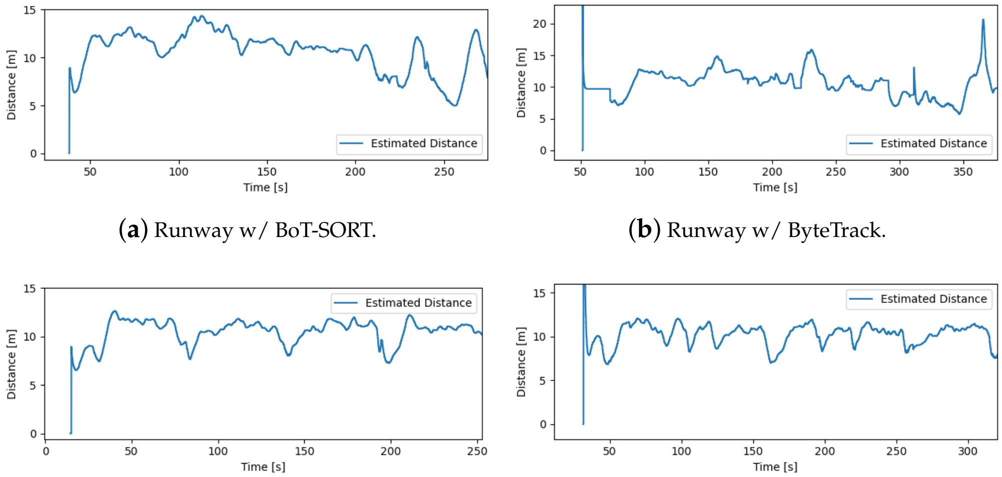


Figure 19. Desired yaw rate (blue) and respective yaw rate actuated by the UAV (orange) during the BoT−SORT experiment in the runway.

Distance estimation values from the four experiments are shown in Figure 20 with the respective averages and standard deviations shown in Table 6. The distance averages around 10 m rather than the pretended 8 m set as the desired distance, which explains the lower Depth Use than the simulation experiments. The deviation happens due to the difference in the behaviour of the target between the simulation and the real-world experiments. While in the simulation, the actors would often walk towards the UAV, in the real-world, the target often walks or even runs away from the UAV.



(**a**) Runway w/ BoT-SORT.

(**b**) Runway w/ ByteTrack.

(**c**) Heliport w/ BoT-SORT.

(**d**) Heliport w/ ByteTrack.

Figure 20. Estimated distance over time.

Table 6. Estimated distance over time.

	Runway w/BoT-SORT	**Runway w/ByteTrack**	**Heliport w/BoT-SORT**	**Heliport w/ ByteTrack**
Average	10.28	10.77	10.32	9.95
Std. Dev.	2.153	1.952	1.278	1.630

7. Conclusions

This work presents a practical approach in vision-based target tracking and following from UAVs leveraging multi-target information to enhance redetection capabilities and allow operations in dynamic scenarios. Both simulation and real-world results show that the system is capable of handling complex and dynamic scenarios where partial or full occlusions are common. Moreover, the 3D flight controller effectively follows the target while keeping it centred in the image showcasing robustness even during sudden direction changes. Comparing the results between the BoT-SORT and ByteTrack trackers reveals a trade-off between computational efficiency (ByteTrack) and tracking precision (BoT-SORT), pending in favour of BoT-SORT for the substantial improvements in camera motion models which substantially decrease the number of ID Changes in target tracking.

Current limitations include maximum target acceleration and direction changes which might cause LOS if there is a sudden and fast movement towards the UAV. Regarding the ability to maintain accurate target centring, limitations arise due to the lack of camera stabilization since the pitch and roll of the UAV are coupled with the pitch and roll of the camera. This can influence the position of the target in the vertical axis of the image when moving forward or backwards and in the horizontal axis when rolling. Future works may consider the application of a gimbal which would decouple the movement of the UAV from the movement of the camera. In addition, path planning methods could also be integrated to generate efficient trajectories while avoiding obstacles [25].

By providing a solution for ID changes and demonstrating a practical application of MOT in unmanned aerial systems, this work establishes a foundation for future research and facilitates the exploration of new applications in mobile target following using UAVs. The integration of multi-target information in our system contributes to advancing the field and opens avenues for further investigation and development.

Author Contributions: Conceptualization, D.F. and M.B.; methodology, D.F. and M.B.; software, D.F.; validation, D.F., M.B.; formal analysis, D.F. and M.B.; investigation, D.F. and M.B.; resources, D.F. and M.B.; data curation, D.F. and M.B.; writing—original draft preparation, D.F.; writing—review and editing, D.F. and M.B.; visualization, D.F. and M.B.; supervision, M.B.; project administration, M.B.; funding acquisition, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by LARSyS (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020 and 10.54499/UIDB/50009/2020) and Aero.Next project (PRR—C645727867-00000066).

Data Availability Statement: The original contributions presented in the study are included in the article and Github repository: https://github.com/diogoferreira08/Target-Following-from-UAV-using-MOT, accessed on 1 September 2024. further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Qi, J.; Song, D.; Shang, H.; Wang, N.; Hua, C.; Wu, C.; Qi, X.; Han, J. Search and Rescue Rotary-Wing UAV and Its Application to the Lushan Ms 7.0 Earthquake. *J. Field Robot.* **2016**, *33*, 290–321. [CrossRef]
2. Prabhakaran, A.; Sharma, R. Autonomous Intelligent UAV system for Criminal Pursuit—A Proof of Concept. *Indian Police J.* **2021**, *68*, 1–20.
3. Liu, X.; Zhang, Z. A Vision-Based Target Detection, Tracking, and Positioning Algorithm for Unmanned Aerial Vehicle. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5565589. [CrossRef]
4. Morales, J.; Castelo, I.; Serra, R.; Lima, P.U.; Basiri, M. Vision-Based Autonomous Following of a Moving Platform and Landing for an Unmanned Aerial Vehicle. *Sensors* **2023**, *23*, 829. [CrossRef] [PubMed]
5. Pimentel, M.; Basiri, M. A Bimodal Rolling-Flying Robot for Micro Level Inspection of Flat and Inclined Surfaces. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5135–5142. [CrossRef]
6. Liu, X.; Yang, Y.; Ma, C.; Li, J.; Zhang, S. Real-Time Visual Tracking of Moving Targets Using a Low-Cost Unmanned Aerial Vehicle with a 3-Axis Stabilized Gimbal System. *Appl. Sci.* **2020**, *10*, 5064. [CrossRef]
7. Cheng, H.; Lin, L.; Zheng, Z.; Guan, Y.; Liu, Z. An autonomous vision-based target tracking system for rotorcraft unmanned aerial vehicles. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1732–1738. [CrossRef]
8. Feng, Y.; Wang, D.; Yang, K. Research on Target Tracking Algorithm of Micro-UAV Based on Monocular Vision. *J. Robot.* **2023**, *2023*, 6657120. [CrossRef]
9. Luo, D.; Shao, P.; Xu, H.; Wang, L. Autonomous Following Algorithm for UAV Based on Multi-Scale KCF and KF. In Proceedings of the 2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 16–18 June 2023; pp. 430–436. [CrossRef]
10. Wei, H. A UAV Target Prediction and Tracking Method Based on KCF and Kalman Filter Hybrid Algorithm. In Proceedings of the 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 14–16 January 2022; pp. 711–718. [CrossRef]
11. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]
12. Yadav, S.; Payandeh, S. Critical Overview of Visual Tracking with Kernel Correlation Filter. *Technologies* **2021**, *9*, 93. [CrossRef]
13. Liu, S.; Li, X.; Lu, H.; He, Y. Multi-Object Tracking Meets Moving UAV. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8866–8875. [CrossRef]
14. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv* **2022**, arXiv:2206.14651.
15. Hussain, M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines* **2023**, *11*, 677. [CrossRef]
16. Li, T.; Li, Z.; Mu, Y.; Su, J. Pedestrian multi-object tracking based on YOLOv7 and BoT-SORT. In Proceedings of the Third International Conference on Computer Vision and Pattern Analysis (ICCPA 2023), Hangzhou, China, 7–9 April 2023; Shen, L., Zhong, G., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 2023; Volume 12754, p. 127541I. [CrossRef]
17. Yan, S.; Fu, Y.; Zhang, W.; Yang, W.; Yu, R.; Zhang, F. Multi-Target Instance Segmentation and Tracking Using YOLOV8 and BoT-SORT for Video SAR. In Proceedings of the 2023 5th International Conference on Electronic Engineering and Informatics (EEI), Wuhan, China, 30 June–2 July 2023; pp. 506–510. [CrossRef]
18. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO, 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 1 February 2024).

19. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
20. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
21. Ferreira, D.; Basiri, M. Leveraging Multi-Object Tracking in Vision-based Target Following for Unmanned Aerial Vehicles. In Proceedings of the 2024 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Paredes de Coura, Portugal, 2–3 May 2024; pp. 88–93. [CrossRef]
22. Baisa, N.L. Derivation of a Constant Velocity Motion Model for Visual Tracking. *arXiv* **2020**, arXiv:2005.00844.
23. Mori, K. ultralytics_ros. 2023. Available online: https://github.com/Alpaca-zip/ultralytics_ros (accessed on 1 February 2024).
24. Baca, T.; Petrlik, M.; Vrba, M.; Spurny, V.; Penicka, R.; Hert, D.; Saska, M. The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles. *J. Intell. Robot. Syst.* **2021**, *102*, 26. [CrossRef]
25. Basiri, M.A.; Chehelgami, S.; Ashtari, E.; Masouleh, M.T.; Kalhor, A. Synergy of Deep Learning and Artificial Potential Field Methods for Robot Path Planning in the Presence of Static and Dynamic Obstacles. In Proceedings of the International Conference on Electrical Engineering (ICEE), Tehran, Iran, 17–19 May 2022; pp. 456–462. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.