LiDAR and Camera Sensor Fusion for Onboard sUAS Detection and Tracking

Daniel Justino
daniel.justino@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

January 2021

Abstract

The recent growth in numbers of small unmanned aerial systems (sUAS) has raised concerns among civilian and military organizations, as they can jeopardize critical infrastructure and threaten manned aircraft. Detecting and tracking intrusive drones is essential to construct reliable counter sUAS (C-sUAS) solutions. Onboard payload sensors typically include electro-optical (EO) cameras, which can be lightweight and provide high-resolution information of the surrounding scene. However, continually searching for targets across high-resolution images is computationally expensive and susceptible to an increase in false positives. Furthermore, EO cameras cannot measure distances directly, which light detection and ranging (LiDAR) sensors can, generating point clouds at a frequency up to $20Hz$ with ranges over $100m$. However, these are usually sparse and cannot recognize small targets. The present thesis studies each sensor's capabilities and develops a sensor fusion solution. It develops a *YOLO-based tracker* for visual detection and tracking and studies the ability of a LiDAR to detect sUAS onboard an aircraft. Additionally, it demonstrates an extrinsic calibration procedure to project 3D LiDAR points into the camera frame accurately. The proposed sensor fusion solution aims to create regions of interest (ROI) from these LiDAR projections to narrow the YOLO-based tracker's search window. Finally, experiments with multiple flying targets were performed onboard an aircraft, demonstrating that the sensor fusion solution improves the *YOLO-based tracker* baseline results, increasing precision from 17.0% to 91.2%, and framerate, from $24Hz$ to $57Hz$, keeping a similar recall of 41.9%, compared to 48.4%.

Keywords: YOLO-based tracker, regions of interest, extrinsic calibration, aerial systems, small UAS

1. Introduction

In the present world, it has become easier than ever to acquire and operate small unmanned aerial system (sUAS). The absence of a human pilot has enabled them to become a robust solution for many industries, such as infrastructure inspection or precise agriculture. Even so, consumer-grade sUAS dominate the commercial sector in the US with a 94% share, according to the Federal Aviation Administration (FAA) [1].

Their indiscriminate use can jeopardize critical infrastructure or even interfere with manned aircraft. A coordinated drone attack halted operations in the Gatwick Airport for three days, leading to millions in losses [2]. These 'off-the-shelf' sUAS platforms could be transformed into rudimentary weapons with relative simplicity, becoming an appealing tool to individuals with nefarious intentions. Recently, NATO has identified the urgent need to improve existing counter sUAS (C-sUAS) solutions [3]. Furthermore, a C-sUAS system is only as effective as its capability to detect possible threats.

This paper presents a sensor fusion methodology for detecting and tracking non-cooperative sUAS. The proposed approach aims to use a LiDAR point cloud to perform the acquisition of possible targets and then use a visual system to detect and track each vehicle. The objective is for the LiDAR to alleviate the visual detector's search task by providing it with regions of interest (ROI). Based on a YOLOv3 detector [4], and a DeepSORT tracker [5], the visual system keeps track of the vehicle location once it is acquired. The sensor system is experimentally tested onboard an aircraft and the sensor fusion methodology is evaluated on the captured data.

2. Related Works

Sensor fusion between LiDAR and camera for C-sUAS purposes was previously researched by Hammer et al. [6]. A pan-tilt camera would align itself towards a flying object detected by the LiDAR, using a deep learning algorithm to classify it. However, the presented solution is based on heavy sensor

equipment, unfeasible for an aerial detection solution, which this research aims to achieve.

A similar concept to the ROI is proposed by Opromolla et al. [7] to visually detect and track UAS while airborne. Knowing the GNSS position of a cooperative UAS and projecting it into the image frame created a search window to be processed by a YOLO-based detector. However, this cooperative assumption is not suitable for a C-UAS solution.

Localization of non-cooperative sUAS onboard an aircraft has been attempted by Vrba et al. [8][9] which combined a stereo camera and a YOLO-based detection method. The proposed approach reports reliable object tracking up to $32m$. Nevertheless, these techniques are not able to directly measure target distances.

The usage of onboard active sensors for sUAS detection was researched by Dogru and Marques [10], demonstrating that an airborne millimetre wave RADAR could measure a drone's bearing and range up to $25m$. De Haag et al. [11] performed a study on the capabilities of an airborne LiDAR to detect drones weighing less than $250g$, reporting a high detection probability for ranges smaller than $15m$, but leaving heavier vehicles still out of its scope. The main limitation of both these approaches is the inability to identify the flying vehicle.

This paper aims to fill the literature gap for an onboard sensor system that employs a LiDAR and camera sensors to achieve real-time detection and tracking of non-cooperative sUAS.

3. Sensor Fusion Methodology

As previously mentioned, electro-optical (EO) cameras can capture the environment in high-resolution. Based on such rich information, it is possible to identify and classify targets based on extracted features. When looking for aerial vehicles, targets are most often only small objects within a larger context. For example, a target measuring $45cm \times 45cm$, captured by an EO camera at $50m$, only occupies a 20×8 pixel area inside a 1280×720 image. Actively searching in the entire image is not only computationally expensive but can also lead to the acquisition of a substantial amount of false positives. Thus, refining the area of search would prove extremely valuable.

LiDAR sensors can provide direct target measurements. The point clouds generated are fast to create (i.e. up to $20Hz$) and to process. Additionally, if employed in an aerial scenario, only a portion of the laser beams emitted will generate returns, as most will point towards the sky. So, there is a substantial likelihood that the observed returns correspond to an aerial target.

Nevertheless, point clouds usually contain non-desired points, which must be filtered. Also, since multiple points can represent a single target, a clustering method is also important to identify objects of interest among the cloud.

Developing a 3D-2D point projection method is fundamental to correlate information between the camera and the LiDAR. This enables LiDAR acquisitions to be projected into the camera frame for further inspection. Additionally, creating regions of interest (ROI) around projected clusters provides the visual detection algorithm with a refined search area to process.

The use of a visual tracker can establish temporal consistency between visual observations. Furthermore, by using an internal estimator, such as a Kalman filter, the visual tracker can predict future target locations and feed that information back into the ROI creation, enabling the formation of a closed tracking loop.

On the approach taken, the YOLOv3 [4] is chosen as the visual detector, complemented by a modified DeepSORT tracker [5]. The combination of both solutions creates the visual detection and tracking system nicknamed *YOLO-based tracker*.

Figure 1 presents a simplified diagram of the sensor fusion solution, which aims to detect and track sUAS.

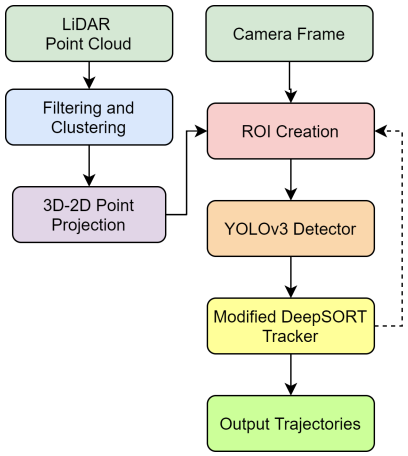

Figure 1: Diagram of the sensor fusion algorithm. The dashed line represents a link between consecutive iterations.

3.1. YOLO-Based Tracker

The *YOLO-based tracker* is the backbone of the sensor fusion algorithm and comprises a YOLOv3 detector [4] and a modified DeepSORT tracker [5].

The YOLOv3[1] algorithm is chosen as the detection framework since it presents a good compromise between accuracy and speed. It is trained on a dataset made available by Svanström [12], which includes 114 videos of a flying drone with a 640×512

[1] https://github.com/ultralytics/yolov3

resolution, totalling 15,133 labelled image frames. The videos were randomly divided into training and validation sets in an 80/20 proportion. The training process finished after 20 epochs presenting an overall $AP = 87.3\%$ and $F1_{score} = 83.1\%$.

Regarding the DeepSORT tracker, some key modifications are made to adapt the algorithm for sUAS tracking. Specifically, the CNN appearance descriptors are discarded, are they are not particularly useful when dealing with sUAS, as their small features against complex backgrounds are difficult to compare. Additionally, the proposed scenario deals with a small number of flying targets, making the distance metric ideal for the assignment problem between observations and tracked states.

3.2. Point Cloud Pre-Processing Methods

There are two pre-processing steps applied to the point cloud to obtain objects of interest from the LiDAR: filtering and clustering.

The filtering algorithm assumes that airspace boundaries representing admissible target locations are given relative to the sensor's X and Y position. Each point is transformed to compensate for the sensor system's angular movement, and removed if located outside the boundary conditions.

This clustering task is performed by the DBSCAN algorithm [13]. The algorithm requires two parameters: the maximum distance between two points for one to be considered in the neighbourhood of the other, and the minimum number of points to form a cluster. The first parameter is set to the largest dimension of the tested targets, which is $0.5m$. The minimum number of points is set to one, which assumes the filtering step removes every point not of interest to the algorithm.

3.3. LiDAR-Camera Pose Estimation

The pose between the camera and the LiDAR must be estimated before LiDAR points can be projected into a pixel frame. This estimation problem, known as perspective-n-point (PnP), is solved with the EPnP algorithm [14].

To find a solution, the EPnP algorithm requires the intrinsic camera matrix K, the radial and tangential distortion coefficients $(k_1, k_2, k_3, p_1, p_2)$, and several 3D-2D point correspondences.

The approach taken gathers calibration points by estimating the location of a calibration board's corners captured by the camera and LiDAR sensors. A rectangular board is rotated 45°, as seen in Figure 2, so the LiDAR horizontal scans can intersect its four edges. However, LiDAR point clouds can not directly provide the location of its corners, so their 3D coordinates have to be estimated.

The algorithm achieves this by computing the point cloud's convex hull and obtaining the minimum bounding box able to enclose it. The bounding box's corners provide an estimate for the calibration points in 3D. However, this approach applied to a single board does not offer much depth distinction between points. To diversify calibration point locations, a total of four different board positions are captured, as shown in Figure 2. In total, the procedure extracted 16 pairs of 3D-2D point correspondences at a maximum range of $9m$.

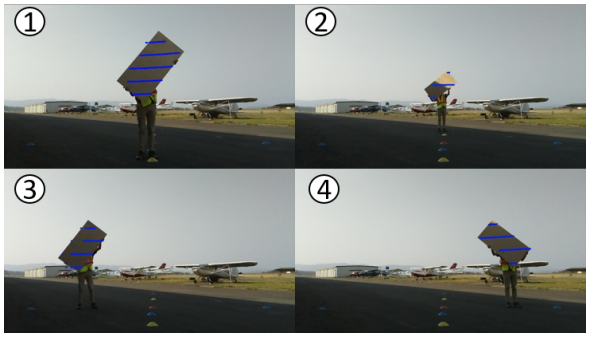

Figure 2: Reprojection of the calibration board's point cloud into the camera frame at four different locations. The distance of each location is: 1 - $4m$; 2 - $9m$; 3 / 4 - $7m$.

Additional calibration points are obtained at further distances using a different approach. From the flight test data, 3D-2D point correspondences are extracted by visual inspection, as presented in Figure 3. After a preliminary calibration procedure obtained from the previous method, 3D target points are visually examined for corresponding location in the pixel context. In total, 22 point-pairs are extracted, ranging between $10m$ and $50m$.

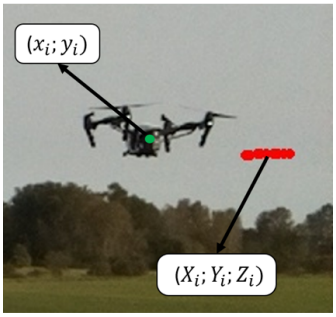

Figure 3: Example of the visual inspection used to extract calibration points at further distances. A LiDAR point $P = (X_i; Y_i; Z_i)$ is visually associated with the pixel point $p = (x_i; y_i)$ for the i^{th} point correspondence.

3.4. ROI Creation

The objective of creating a region on interest (ROI) is to narrow the search area of the detection algorithm.

Each ROI has fixed dimensions to simplify its creation process. They are chosen based on the maximum pixel size that the target presents during the flight tests. The primary target measures $45cm \times 45cm$. When captured by the camera at $10m$, it occupies an 80×50 pixel area inside a 1280×720 image. Since the YOLOv3 detector requires an input size multiple of 32, the ROI size is set to 128×128 pixels. This restriction is related with its convolutional network architecture.

When creating a ROI around a target, its location can come from two sources: LiDAR detections and tracker predictions. After the LiDAR point cloud is transformed into 2D clusters, a bounding box enclosing every point is created, being representative of the captured target. However, LiDAR clusters may only capture a small portion of the target, leading to uncertainty regarding its outer limits. On the other hand, bounding boxes from tracker predictions contain information on the target's pixel size, allowing a more informed decision on ROI placement. Suppose the centre of the bounding box created from the LiDAR data is inside an already existing bounding box in the tracker prediction list. In that case, the LiDAR detection is discarded since the target is already being visually tracked. Although LiDAR clusters do not consistently provide target detections, they are a crucial mechanism for target acquisition.

A ROI is created around each target if the object is not already entirely enclosed by an existing ROI. This simple solution presents some downsides, as one object in the borderline of another ROI will produce its own search region, resulting in overlapping ROI, leading to redundant computations. Additionally, overlapping ROI can cause the same target to be detected multiple times. As a solution, a non-maximum-suppression algorithm is applied to remove duplicate YOLOv3 detections. Given the present sUAS scenario has low vehicle density, the problem of overlapping ROI is not substantial.

Figure 4 represents a frame processed by the sensor fusion algorithm using the described approach. A LiDAR detection, the red dots, create the ROI, yellow bounding box, which is processed by the visual detector, obtaining the detection in dark blue.

Figure 4: Example of a ROI creation. The LiDAR detection is the red dots, the ROI is the yellow bounding box, the ground truth is the light blue and a detection is a dark blue bounding box.

4. Experiments

There are three main experiments performed during the flight tests: *moving target*, *moving sensor* and *multi-target free-flight*. A simplified diagram of these is presented in Figure 5. The *moving target* and *moving sensor* experiments are divided into two sub-experiments related to the sensor system position and movement. For the *moving target*, the sensor is either positioned on the ground or in a hovering aircraft. In the *moving sensor*, two different manoeuvres are performed: altitude and pitch angle variation. A detailed description of each experiment and of the remotly piloted aircraft (RPA) is made in the following subsections.

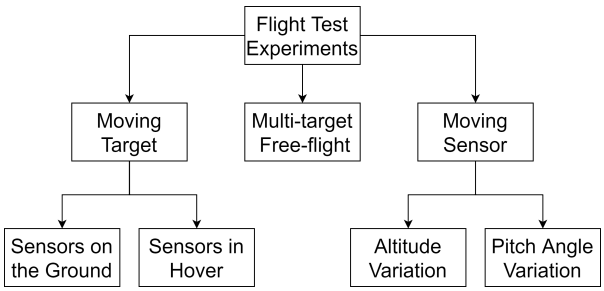

Figure 5: Diagram of the flight test experiments.

4.1. RPA Deployed

Three different RPA flew on the flight tests, as represented in Figure 8. The aircraft carrying the sensor payload was the DJI Matrice 600. It is the largest aircraft of the three and provides an ideal platform to support the sensor system. With a maximum take-off weight of $15.5kg$, the RPA has an endurance of 18 minutes while carrying a $5.5kg$ payload.

The two remaining RPA were operated as flying targets. The DJI Inspire and DJI Mavic weight $2.8kg$ and $0.7kg$, respectively. They both are popular commercially available sUAS, becoming ideal test vehicles for C-sUAS experiments. Between them, the DJI Inspire has the largest size, and for that reason, it was considered the primary target, flown on all experiments. The DJI Mavic was only used during the *free-flight* experiments, where all three RPA were flown simultaneously.

4.2. Moving Target Experiments

The *moving target* experiments were performed while the sensor system is stationary relative to the

Table 1: Specification sheet of the M8 LiDAR [15]. For sUAS detection, the LiDAR is turned upside-down. Thus, experimentally the vertical FOV is inverted $(+18°/-3°)$.

Parameter	Specification
Wavelength	$905nm$
Nominal Weight	$940g$
Maximum Range	$> 100m$ (80% reflectivity)
Range Accuracy $(1\sigma$ *at* $50m)$	$< 3cm$
Frame Rate	$5 - 20Hz$
Vertical FOV	$21° (+3° / -18°)$
Horizontal FOV	$360°$
Vertical Channels	8
Angular Resolution	$0.03 - 0.13°$ (dependent on frame rate)
Output Rate	420,000 points per second

surrounding environment. Two different sensor locations were studied: on the *ground* and in *hover*. While in these positions, it would capture the RPA target performing a manoeuvre named *cross manoeuvre* at different distances. Figure 7 presents a visual representation of the manoeuvres performed. The airspace assigned to the DJI Matrice 600 and DJI Inspire is represented by the blue and yellow rectangles, respectively. The red region is a 'no-fly' zone to guarantee that no contact could occur between aircraft.

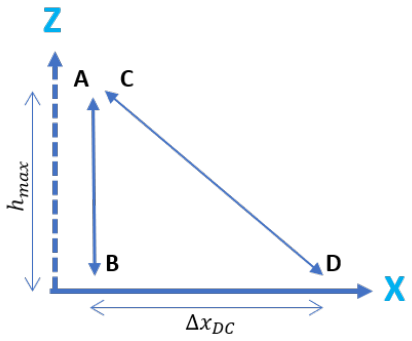

Figure 6: Cross manoeuvre diagram, with same frame of reference as in Figure 7. The target begins by moving between A-B-A, followed by C-D-C.

A concerning limitation of the M8 LiDAR, as presented in Table 1, is the sparse vertical resolution it provides, having only 8 vertical channels. A manoeuvre was created to increase the probability of the target being detected by the LiDAR. To do so, the RPA performs significant altitude variations to cross as many horizontal scanning lines as possible. For simplicity reasons, it is named the *cross manoeuvre*. In its essence, the manoeuvre consists of simple climb and descents, performed both vertically and diagonally. The objective of the diagonal movement is to diversify the detection location. The *cross manoeuvre* can be split into four different steps, presented in Figure 6. The RPA must move parallel to the X-axis, as seen in Figure7. The *cross manoeuvre* is then performed at different range intervals to evaluate the relationship between LiDAR detections and target distance. The *cross manoeuvre* has two parameters: the maximum height h_{max} and the horizontal distance Δx_{DC} between point D and C. To allow the drone a chance to pass over every possible horizontal scanning line, the maximum height h_{max} was different at each distance d, taken from the equation

$$h_{max} = h_s + d \cdot tan(\alpha_{max}) \tag{1}$$

where h_s is the height of the sensor system and α_{max} is the maximum LiDAR vertical scanning angle.

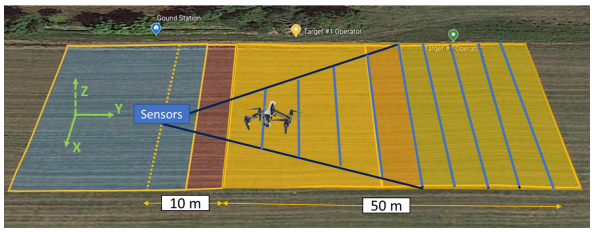

Figure 7: Visual representation of the *moving target* experiments. The DJI Inspire performs the *cross manoeuvre* along the light blue lines, staying parallel to the X-axis. The dark blue lines represent the FOV limits of the camera. The sensor system has the same X and Y values in every manoeuvre.

The RPA target performs the *cross manoeuvres* between $10m$ and $60m$ from the sensor system. Ideally, Δx_{DC} should be set as always to keep the RPA within the camera FOV. Markers were physically placed on the field to help the pilot-in-command visualize these boundaries, as seen in Figure 7.

During the experiments, the sensor system was kept at the same X and Y positions. The altitude varied, staying at $h_s = 1m$ while on the ground, and at $h_s = 10m$ while in hover.

4.3. Moving Sensor Experiments

The benefits of sensor mobility are evaluated by exploring two different techniques: altitude and pitch angle variation.

The first experiment explores the sensor's altitude variation. By varying the altitude it is possible to ensure that the LiDAR horizontal scans search the airspace completely, up to a certain range. Thus, the height Δh between two consecutive scans at a certain range represents the same vertical displacement the LiDAR needs to achieve and is given by:

$$\Delta h = d \cdot (tan(\alpha_{i+1}) - tan(\alpha_i)) \tag{2}$$

where d is the horizontal distance between the target and the sensor, α_i is the angle between the horizon and the i^{th} horizontal scan. One obvious concern with this approach is that as α_i and α_{i+1} increase, so does the height Δh. For the M8 LiDAR, the separation between horizontal scans is $\Delta\alpha = \alpha_{i+1} - \alpha_i = 3°$, as shown in Table 1, and the elevation of the two highest horizontal arrays is $\alpha_8 = 18.25°$ and $\alpha_7 = 15.25°$. So, for $d = 60m$, an altitude variation of $\Delta h = 3.42m$ would be necessary to guarantee an overlap between scanned regions. Experimentally, the sensor system performs $\Delta h = 5m$ with an initial altitude of $h_0 = 10m$, theoretically achieving no blind regions up to $d = 87.8m$.

The second technique examines how the pitch angle movement of the LiDAR can be used to detect other RPA. If the sensor system pitch angle θ is tilted more than the angle $\Delta\alpha$ between the LiDAR horizontal scans, it allows the horizontal scans to overlap, allowing a thoroughly inspection of the airspace for targets. Experimentally, precise pitch angle motions of the sensor system is performed by a servo-controlled mechanism. This solution preserves the aircraft's position and stability. The experiments perform a pitch variation of $\Delta\theta = 5°$, guaranteeing an overlap between LiDAR scans, since $\Delta\alpha = 3°$. Additionally, the sensor system hovers at $h_s = 10m$.

The target is positioned at the centre line position in both experiments, along the Y-axis represented in Figure 7. Like the *moving target* experiments, the vehicle moves between $10m$ and $60m$ from the sensor system. At each segment, the target sustains a hover manoeuvre at different altitudes to keep it at a constant 10° elevation angle relative to the payload. The sensor system preserves the same X and Y values for the entire *moving sensor* experiments.

4.4. Free Flight

The objective of the *free-flight* experiments is to capture realistic flight test data. Each agent should have no predefined trajectories or manoeuvres. Nonetheless, the hazard of operating multiple RPA within share airspace raises safety concerns. As such, each RPA has a specific airspace region assigned to itself, with 'no-fly' zones in-between, as represented in Figure 8. The DJI Mavic, being the aircraft with the smallest dimensions, is placed in the airspace closest to the sensor system. Additionally, RPA pilots are placed inside the 'no-fly' regions to prevent the RPA from crossing to unwanted regions.

5. Results

This Section presents the results obtained during the flight test experiments explained in Section 4.

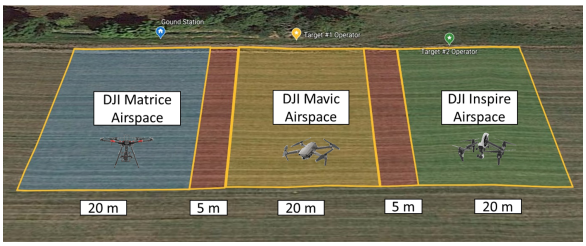

Figure 8: Unique airspace regions assigned to each RPA for the *free-flight* experiments. The DJI Matrice 600 carries the sensor system payload, while the DJI Mavic and DJI Inspire act as flying targets.

5.1. LiDAR Detection Performance

Figure 9 presents the LiDAR recall on the *moving target* experiments. The LiDAR recall is the percentage of total targets the LiDAR was able to detect. It shows comparable recall results between the LiDAR on the ground or in a hover position. Both scenarios show a drastic decrease in target recall throughout the $60m$ range. The LiDAR presents almost no detections at this last distance, except for a single point captured in the ground position.

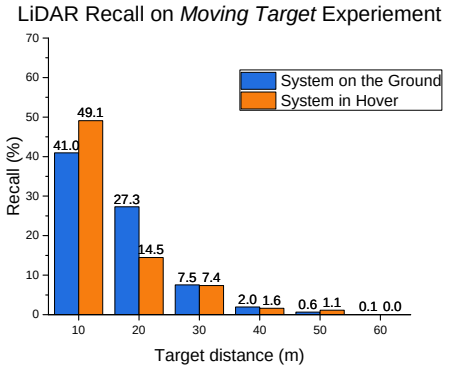

Figure 9: LiDAR recall with target distance on *moving target* experiments.

These results display the LiDAR sensor's inability to continuously capture the target location or act as a single sensor solution on a static platform. The discrepancies observed between the ground and hover experiments are not very significant. They seem to indicate no meaningful performance deterioration induced by the LiDAR sensor's presence onboard an aircraft compared to having it on the ground position.

The *moving sensor* experiments proved to be more consistent than the *moving target*. For example, the sensor's pitch movement is controlled by the onboard computer, granting a consistent and precise 5° pitch variation. In addition, the sensor system's altitude varied only between $10m$ and

$15m$ while maintaining the same X and Y positions. These techniques can remove some of the uncertainty associated with the target movement in the previous experiments.

Analysing the altitude variation experiment in Figure 10, it is observed a 27.6% target recall at $10m$, which is inferior to both *moving target* experiments. However, recall consistency is improved, as it only drops below 3% over the $50m$ range, while previous results show a drop below 3% at $40m$. Additionally, target recall values on the *pitch variation* experiment show the highest recall rates of all approaches, achieving 38% recall at distances up to $20m$, only dropping below 10% at ranges greater than $40m$.

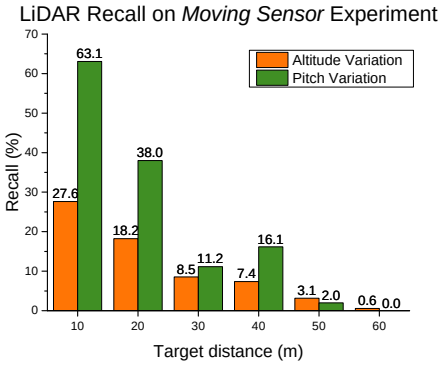

Figure 10: LiDAR recall on *altitude variation* and *pitch variation* experiments with target distance.

Observing Figure 11, both manoeuvres present similar trends in the number of points per detection and are on par with the *moving target* experiments. Multiple points per detection are presented until the $30m$ range, dropping only to a single point per detection above this threshold. The exception is the *altitude variation* experiment, which presents two points instead of one at $40m$.

These results reinforce the evidence of the LiDAR's drastic performance degradation with the increase in target distance. They also demonstrate the LiDAR's potential to detect hovering sUAS through both dynamic movements.

5.2. Pose Estimation Accuracy

The calibration results for the *free-flight* experiment are shown in Table 2. It presents the two different techniques explored to capture 3D-2D point correspondences, plus the scenario of both methods combined.

Points projected from the 3D point cloud into the 2D image frame are classified based on their overlap with the ground truth bounding box. A projected point can either land 'inside' or 'outside' this area. A 'near' label is adopted to specify if a

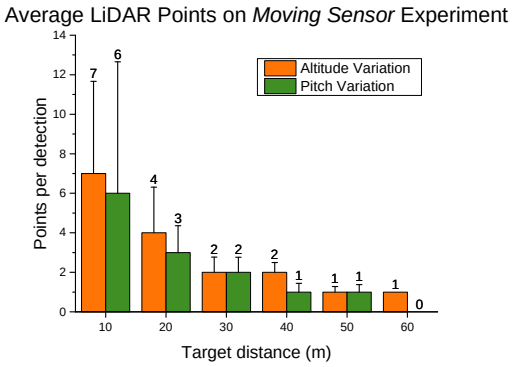

Figure 11: Points per detection on *altitude variation* and *pitch variation* experiments with target distance.

point lands in the proximity of the ground truth. This classification is reserved for points that land in an area with twice the ground truth's height and length.

The PnP solution obtained with the calibration board showed poor accuracy results with only 8.4% of the LiDAR points landing in the 'near' label, and none 'inside' the ground truth. This effect was mainly caused by the close distances at which the calibration board was captured. In *free-flight* experiment targets flew over this threshold, as seen in Figure 8. The PnP solution obtained with the visual inspection method shows higher accuracy, with only 2.6% of the points landing in the 'near' label and none 'outside' the ground truth. However, the best calibrations results observed happened when the PnP solution was obtained with the combination of both methods, leading to 99.0% of the points landing 'inside' and 1.0% landing 'near' the ground truth.

Table 2: Extrinsic calibration accuracy on the *free-flight* experiment. Three point extraction methods are used to solve the PnP problem: calibration board, visual inspection and both combined. Each solution presents three labels for a LiDAR point projection: inside, near or outside the target's ground truth.

Point Extraction Method	Inside	Near	Outside	Total
Calibration Board	0	115	1243	1358
Visual Inspection	1322	36	0	
Both Combined	1344	14	0	

5.3. YOLOv3 Detector

The YOLOv3 is the algorithm responsible for detecting sUAS targets. Metrics for this section are

based on [16]. A precision-recall (P-R) curve was created to evaluate the algorithm's ability to detect targets captured during flight testing. The non-maximum-suppression parameters of the YOLOv3 algorithm were set with an intersection over union (IOU) threshold of 50% and a minimum confidence threshold of 0.01. The P-R curve obtained is presented in Figure 12, showing an average precision (AP) of 32.0%. The AP serves as an overall metric to describe the detector's performance. However, the value obtained is low, and a close inspection of the P-R curve indicates the probable causes.

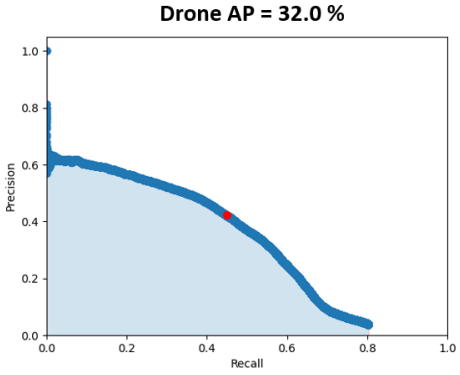

Figure 12: Precision-recall curve for the YOLOv3 detector. The presented results are based on flight test data collected. The red point represents the highest $F1_{score}$ achieved of 0.44.

The negative slope in Figure 12 indicates an increasing proportion of false positives with a decrease of confidence threshold. On examining detection samples presented in Figure 13, the various types of false targets collected become apparent. Obvious wrong detections, like the treetop, expose some flaws in the training process. Other false positives present background clutter that is difficult to distinguish between a true and a false positive, even for the human eye, since the target and background clutter can appear very similar. The bottom right sample in Figure 13 represents a safety cone. However, when the target is above the horizon, its features contrast more with the simple background, leading to more consistent detections.

Additionally, with a minimum confidence threshold of 0.01, the detection algorithm could only achieve a recall of around 80%, contributing to the low AP number. A closer inspection reveals that the vast majority of the non-detected targets happen on occasions where the RPA is far and in locations with complex backgrounds, as shown in Figure 13.

Furthermore, the dataset used to train YOLOv3 is predominantly composed of vehicles above the horizon, as a ground-based system captured it. This would not prepare the neural network for scenarios

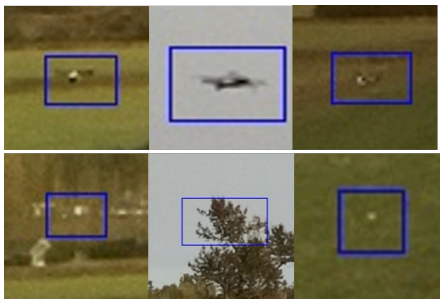

Figure 13: YOLOv3 detection examples. The top row presents true positives, and the bottom row false positives.

with sUAS against a textured background.

Finally, to determine with which confidence threshold the algorithm performs best, the P-R curve's point is chosen based on the highest $F1_{score}$, which represents an harmonic mean between precision and recall. It provides a balanced representation of the algorithm's precision and recall. The best $F1_{score}$ is 44%, obtained with a confidence threshold of 0.26. This point, represented by a red dot in Figure 12, has a precision of 42% and a recall of 46%. The following results present the YOLOv3 detector set on this confidence threshold.

5.4. Sensor Fusion and YOLO-based tracker

A comparison between the sensor fusion methods and the *YOLO-based tracker* follows. Metrics for multiple object tracking were based on [17].

Figure 14 shows that the sensor fusion recall on the hover experiment is comparable to the *YOLO-based tracker* on distances up to $30m$, with the sensor fusion presenting the lower values. One cause for this effect is the sensor fusion's idleness while waiting for the LiDAR sensor to capture a target, which leaves initial frames without detections. The increase of target distance amplifies this effect, as a significant decline in recall values can be observed on distances over $40m$. Even with a LiDAR recall of 1.2% for the $50m$ range, the sensor fusion solution can still achieve a recall of 39.6% for the hover experiment.

Precision results are shown is in Figure 15. The sensor fusion solution shows a tremendous increase in precision with respect to the *YOLO-based tracker*. This effect is related to the search region being focused on the target location, leading to the algorithm being less susceptible to false positives. There is a decrease in precision between $30m$ and $40m$ to 75.6%, which rises to 93.8% at $50m$. This anomaly might be influenced by the presence of ground clutter on the hover experiments. However, the differences in precision between the sensor fusion and the *YOLO-based tracker* are very sig-

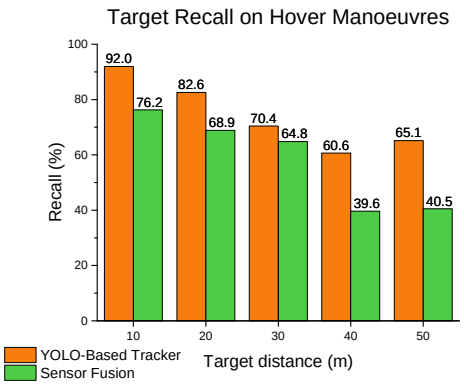

Figure 14: Recall comparison on the *hover* experiments.

nificant, having the sensor fusion almost ten times higher precision across all distances.

The *YOLO-based tracker's* low precision value is related to a large amount of ground clutter being miss detected as a drone. Such an example is the bottom right image in Figure 13, where safety cones placed in the flight test field can have a similar appearance to an aircraft in the distance.

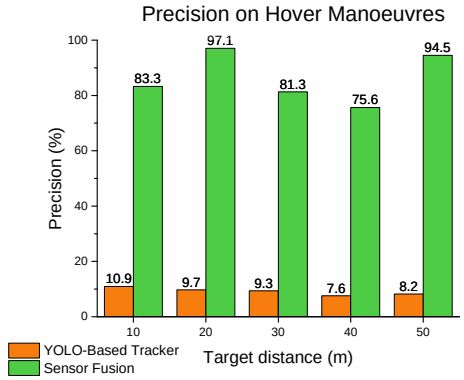

Figure 15: Precision comparison on the *hover* experiments.

Similar trends are shown for the *free-flight* experiment, seen in Table 3. The sensor fusion and the *YOLO-based tracker* present similar recall values, of 48.4% and 41.9%, with the sensor fusion presenting the lowest one. Once again, the sensor fusion solution shows a drastic increase in target precision, from 17.0% to 91.2%. Both solutions present similar MOTP, which means the average IOU overlap between detections and ground truths is above 70%. Sensor fusion presents an increase from -1.88 to 0.38 in the MOTA metric, which accounts for the overall tracking ability by combining the number of misses, false positives and ID switches. The closer it is to 1.0 the better. However, the sensor fusion solution presents a lower ability to track the targets for more extended periods, dropping by one the number of trajectories *mostly tracked* (MT) and increasing by four the number of trajectories *mostly lost* (ML). A reason for this effect can be linked to some tracks being made of a lower number of frames, giving a smaller change for the LiDAR to detect the vehicle and begin the tracking procedure. Each time a RPA enters and leaves the camera field of view, a new trajectory is created, leading to a high variability in the length of each one. Nonetheless, the sensor fusion method presents half of the trajectory fragments FM, 50 instead of 99, showing it has more tracking consistency than the *YOLO-based tracker*.

Additionally, both solutions present the ability to process sensor data in real-time, presenting a processing speed higher than $20Hz$, which is the framerate at which sensor data is recorded. Furthermore, the sensor fusion methods can more than double the processing speed of the visual algorithm, to $57Hz$. However, these results are achieved on powerful computational resources, using an *NVIDIA Tesla P100 GPU* to improve the processing speed of the YOLOv3 detector. Real-time onboard capabilities are still a topic left to be researched.

Table 3: Tracking results of the *YOLO-based tracker* and sensor fusion methods on the *free-flight* experiment.

	Recall	Precision	MOTP	MOTA	FM	Runtime
YOLO-Based Tracker	48.4%	17.0%	72.0%	-1.88	99	24 Hz
Sensor Fusion	41.9%	91.2%	74.0%	0.38	50	57 Hz

6. Conclusions

In this paper, a multi-sensor methodology was investigated to visually detect and track multiple sUAS onboard an aircraft.

Experiments have revealed that the LiDAR sensor presents a drastic decline in performance with an increase in target distance. Furthermore, they have shown that attaching the sensor onto an airborne platform can lead to the detection of targets that otherwise would be unobservable. Also, by using a combination of two calibration methods to estimate the camera pose with respect to the LiDAR sensor it was possible to project LiDAR detections inside their respective ground truth labels accurately.

The YOLOv3 detector proved to have its limitations when detecting sUAS against textured backgrounds. It also acquired a large number of false positives by miss detecting ground clutter. Results could be improved by training the algorithm on a more appropriate dataset.

It was shown that the sensor fusion approach could drastically increase the overall precision of the *YOLO-based tracker* while maintaining similar recall values and doubling its framerate.

Proposed future work includes incorporating a visual search on random image locations when the LiDAR does not return any targets and when no target is being visually tracked to improve recall results. Additionally, future work could include creating an algorithm capable of locating a vehicle in a 3D environment, taking advantage of LiDAR measurements. This information would be precious for a C-sUAS system.

References

[1] Rina Valeur Simonsen, Malene Hartung, Kirstine Clemens Brejndal-Hansen, Stig Yding Sørensen, Kristian Oluf Sylvester-Hvid, and David Klein. Global Trends of Unmanned Aerial Systems. *Danish Technological Institute [DTI]*, page 44, 2019. `https://www.auvsi.org/global-trends-unmanned-aerial-systems` [Online: accessed on December 2020].

[2] The Guardian. Gatwick drone disruption cost airport just £1.4m, 2018. `https://www.theguardian.com/uk-news/2019/jun/18/gatwick-drone-disruption-cost-airport-just-14m` [Online: accessed on December 2020].

[3] NATO Industrial advisory group. Low, slow and small threat effectors. *final report on NATO NIAG study group 200*, 2017.

[4] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[5] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[6] Marcus Hammer, Björn Borgmann, Marcus Hebel, and Michael Arens. A multi-sensorial approach for the protection of operational vehicles by detection and classification of small flying objects. In *Electro-Optical Remote Sensing XIV*, volume 11538, page 1153807. International Society for Optics and Photonics, 2020.

[7] Roberto Opromolla, Giuseppe Inchingolo, and Giancarmine Fasano. Airborne visual detection and tracking of cooperative uavs exploiting deep learning. *Sensors*, 19(19):4332, 2019.

[8] Matouš Vrba, Daniel Heřt, and Martin Saska. Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system. *IEEE Robotics and Automation Letters*, 4(4): 3402–3409, 2019.

[9] Matouš Vrba and Martin Saska. Marker-less micro aerial vehicle detection and localization using convolutional neural networks. *IEEE Robotics and Automation Letters*, 5(2):2459–2466, 2020.

[10] S. Dogru and L. Marques. Pursuing drones with drones using millimeter wave radar. *IEEE Robotics and Automation Letters*, 5(3):4156–4163, July 2020. ISSN 2377-3766. doi: 10.1109/LRA.2020.2990605.

[11] Maarten Uijt de Haag, Chris G Bartone, and Michael S Braasch. Flight-test evaluation of small form-factor lidar and radar sensors for suas detect-and-avoid applications. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–11. IEEE, 2016.

[12] Fredrik Svanström. Drone Detection and Classification using Machine Learning and Sensor Fusion. Master's thesis, Halmstad University, 2020.

[13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[14] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.

[15] M8 ™ Sensor User Guide, 2019. Available at `http://quanergy.com` [Online: accessed on December 2020].

[16] Rainer Stiefelhagen, Keni Bernardin, Rachel Bowers, R Travis Rose, Martial Michel, and John Garofolo. The CLEAR 2006 evaluation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4625 LNCS, pages 3–34, 2006. ISBN 9783540695684. doi: 10.1007/978-3-540-69568-4_1.

[17] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.