Monocular Geo-Localization of Terrestrial Objects from UAV Imagery with Spatial Refinement

Heitor Riquelme[1], João G. Melo[1], João P. Mafalado[1], Pedro César[1], Driele Pires[1], Mateus Ferreira[1], Bruno Eduardo[2], Edna Barros[1] and Paulo Rosa[2]
[1]Centro de Informática, Universidade Federal de Pernambuco,
Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife - Pernambuco, Brazil.
[2]Department of Computer Engineering, Instituto Militar de Engenharia,
Praça Gen. Tibúrcio, 80 - Urca, Rio de Janeiro - Rio de Janeiro, Brazil.

ABSTRACT

Geo-localization of ground objects from UAV platforms is a fundamental capability for applications such as autonomous inspection, search and rescue, and target tracking. This paper presents a fully online pipeline for estimating the geographic coordinates of multiple terrestrial objects from UAV monocular imagery. The approach integrates (i) a timestamp interpolation scheme that synchronizes camera frames with asynchronous UAV state estimates, (ii) a geometric projection model that maps image detections to ground-plane coordinates using UAV attitude and position, and (iii) a spatial tracking stage based on the Hungarian algorithm and Kalman filtering that associates and refines repeated noisy estimates into a consistent per-object geo-location. Experiments conducted in simulation with Gazebo and ArduPilot SITL across 3 scenarios and 6 flight paths demonstrate a mean localization RMSE of 0.603 m, with 97.9% of targets successfully identified.

Keywords: Geo-localization; Multi-Object Tracking; Spatial Tracking; State Interpolation

1 INTRODUCTION

Geolocation of objects from aerial platforms is a key capability for applications such as mapping, inspection, search and rescue [1]. When using monocular cameras, this task is challenging due to the lack of depth information and sensor asynchrony between the imaging and navigation systems.

An additional challenge lies in maintaining consistent associations between detections over time. As a UAV traverses its trajectory, each physical target is observed multiple times from different viewpoints and under varying localization noise, producing geo-localization estimates that are spatially dispersed around the true location [2]. Furthermore, false detections may contaminate this set with spurious estimates. Consequently, these raw observations must be aggregated, filtered, and associated across frames to yield a consistent and accurate representation of the scene. To this end, this paper presents a method for estimating the geographic coordinates of multiple ground objects detected in UAV imagery, addressing the inherent noise and association ambiguity through a spatial refinement pipeline.

The main contributions of this work are:

- A timestamp interpolation method that synchronizes monocular camera frames with asynchronous UAV state estimates.
- A projection model for monocular geo-localization of ground objects from UAV attitude and position.
- A spatial tracking and refinement pipeline that associates, filters, and consolidates repeated detections into consistent per-object geo-locations.

2 SYSTEM OVERVIEW

2.1 *Problem Statement*

The objective of this work is to estimate the geographic locations of multiple terrestrial objects using a monocular camera onboard a UAV. During flight, the UAV captures images while its onboard Flight Controller Unit (FCU) provides position and attitude estimates. Objects of interest are detected in the images, and their geo-locations are estimated assuming they lie on the ground plane.

A fundamental difficulty arises from the fact that each physical object is typically observed multiple times from different viewpoints along the UAV trajectory. These repeated observations produce multiple geo-localization estimates per object, as illustrated in Fig. 1. Due to sensor noise, detection uncertainty, and projection errors, these estimates are spatially dispersed around the true object location. Additionally, false detections may introduce spurious estimates that do not correspond to any real object.

Consequently, the raw set of geo-localization estimates does not directly provide a reliable representation of the scene. Instead, it consists of a mixture of:

- Multiple estimates corresponding to the same object.
- Estimates belonging to different objects.
- Outliers caused by incorrect detections or localization errors.

Therefore, the problem consists not only of estimating object locations, but also of maintaining consistent associations between observations over time, correctly identifying the number of distinct objects, and assigning a single, accurate geo-location to each of them.

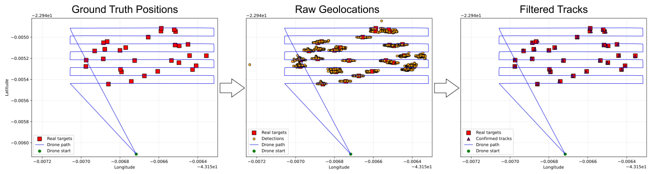


Figure 1: Illustration of the problem: multiple observations generate noisy geo-localizations that must be associated and refined into a single estimate per object.

2.2 *Overview of the Proposed Approach*

To address this problem, we propose the workflow illustrated in Figure 2, a fully online pipeline composed of two main components:

- **Object Detection and Position Estimation**
- **Spatial Refinement with Tracking and Filtering**

The first component processes each incoming frame independently. Objects are detected as bounding boxes in the image using a deep learning-based object detector; then, the UAV position and attitude are interpolated at the frame timestamp, and each detection is projected into geographic coordinates using geometric projection.

The second component maintains a set of object tracks over time. New geo-localization estimates are associated with existing tracks using a data association method based on the Hungarian algorithm [3]. Tracks are progressively refined, filtered, and either confirmed or discarded based on their temporal and spatial consistencies.

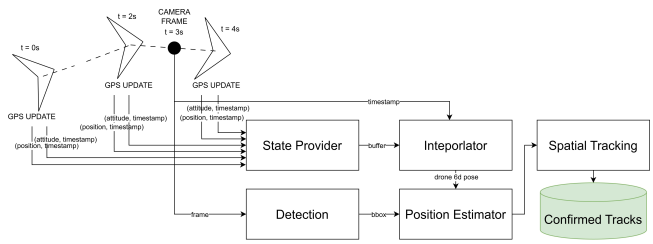


Figure 2: Workflow of the proposed approach.

3 GEO-LOCALIZATION

3.1 *Object Detection*

For each frame, objects are detected and represented as bounding boxes. The center of each bounding box is used as the representative image point. The timestamp at which each frame was captured is also recorded, enabling interpolation and estimation of the UAV's position and attitude at that moment.

We adopt a YOLOv11 model for object detection [4], owing to its suitability in real-time systems with high-throughput with low-power constraints [5, 6]. For each detected object, we record the bounding box center, capture timestamp, detection confidence, and object class. This information can be used further to improve multi-object tracking and association.

3.2 *State Provider and Interpolation*

We employ the ArduPilot's Software In The Loop (SITL) for UAV control and state estimation. By default, it provides estimations of the current position and attitude at 10 Hz. However, position and attitude data are published in separate messages, so they need to be read asynchronously. The state interpolator maintains two timestamped buffers, one for position and one for attitude, defined as:

$$(\mathbf{p}_i, t_i^p) \tag{1}$$

$$(\mathbf{a}_i, t_i^a) \tag{2}$$

where $\mathbf{p} = (x, y, z)$ denotes position in local ENU coordinates, $\mathbf{a} = (\psi, \theta, \phi)$ denotes the attitude as yaw, pitch, and roll angles, and t_i is the timestamp of the i-th measurement.

These position and attitude messages are saved in buffers in the State Provider, with their respective timestamps. This way, we keep track of the UAV's movements for up to a configurable number of messages, currently configured as $buffer\_size = 200$.

The interpolations are implemented as linear movements between timestamps, i.e., given a moment in time t, we compute a weighted average of the measurements taken at the neighboring timestamps. Position is interpolated linearly between consecutive measurements:

$$\mathbf{p}(t) = (1 - \alpha)\mathbf{p}_i + \alpha \mathbf{p}_{i+1} \tag{3}$$

where α is determined from timestamps, by:

$$\alpha = \frac{t - t_i}{t_{i+1} - t_i} \tag{4}$$

where $t_i \leq t \leq t_{i+1}$.

Attitude angles are circular quantities, so direct linear interpolation would produce incorrect results near discontinuities (e.g., wrapping from 359° to 1°). Instead, each angle is interpolated by averaging its sine and cosine components separately, then recovering the result via arctangent. Equation (5) shows how this applies to the yaw angle ψ:

$$\psi(t) = arctan\left(\frac{(1-\alpha)sin(\psi_i) + \alpha sin(\psi_{i+1})}{(1-\alpha)cos(\psi_i) + \alpha cos(\psi_{i+1})}\right) \tag{5}$$

3.3 Position Estimator

This module takes as inputs the coordinates of a given pixel (u, v), the camera's intrinsic and extrinsic parameters, and the UAV's position and attitude at the time each frame was captured, as illustrated in Figure 3. Using a geometric projection model, it computes the geo-localization of the given pixel, which should correspond to the position of a desired object.

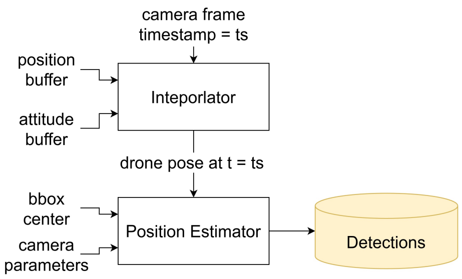


Figure 3: Inputs and outputs of the position estimator.

The geo-localization approach extends the method described in [7] to accommodate a lower-cost UAV platform without gimbal stabilization. In this setup, the camera is rigidly mounted and permanently oriented towards the ground. Therefore, the camera orientation is directly derived from the UAV's attitude using a fixed offset:

$$\begin{aligned} pitch_{cam} &= pitch - 90^\circ \\ yaw_{cam} &= yaw \\ roll_{cam} &= roll \end{aligned} \tag{6}$$

Since the camera is rigidly fixed to the UAV body without stabilization, any change in UAV attitude directly perturbs the camera pose and may introduce additional geo-localization error. This trade-off is accepted in exchange for a simpler, lower-cost hardware configuration compatible with custom-built platforms.

The method relies on three coordinate systems: the world coordinate system, the UAV coordinate system, and the camera coordinate system. The world coordinate system is defined in a metric georeferenced frame, such as UTM (Universal Transverse Mercator) or a local ENU (East-North-Up) frame, allowing positions to be expressed in linear units (meters).

In this formulation, all coordinate systems share the same origin, defined in the world reference frame. The UAV and camera coordinate systems differ from the world system only by their orientations:

- World coordinate system: $(x, y, z) = (\text{east}, \text{north}, \text{up})$
- Drone coordinate system: $(x, y, z) = (\text{forward}, \text{right}, \text{down})$
- Camera coordinate system: $(x, y, z) = (\text{right}, \text{down}, \text{forward})$

The relationship between these coordinate systems is defined by the camera pose. The rotation matrix R that maps points from the world coordinate system to the camera coordinate system is computed following the method described in [7]. This matrix encodes the camera orientation as derived from the UAV attitude and the fixed mounting configuration.

The camera center expressed in the world coordinate system is given by $v_0 = -R^T t$. Therefore, the translation vector t does not need to be computed independently, as the camera position is directly obtained from the UAV position, estimated by the state interpolator.

Given the camera pose, the projection from 3D world points to image pixels is modeled using the pinhole camera model [8]:

$$K[R \mid t]P = \lambda p, \tag{7}$$

where K is the intrinsic matrix, R (rotation matrix) and t (translation vector) map points from the world coordinate system to the camera coordinate system, $p = (u, v, 1)^T$ is the homogeneous pixel coordinate, λ is a scale factor, and $P = (x, y, z, 1)^T$ is a 3D point expressed in homogeneous coordinates.

To perform geo-localization, the inverse mapping must be computed. By rearranging the projection Equation (7), it is possible to derive the 3D ray in the world coordinate system corresponding to a given image pixel. This ray can be written in parametric form as:

$$P(\lambda) = v_0 + \lambda v, \tag{8}$$

where:

$$v = R^T K^{-1} p. \tag{9}$$

Thus, each pixel defines a ray originating from the camera center and extending into the scene.

The geo-location is obtained by computing the intersection between this ray and the ground. Assuming a planar ground model, this intersection can be obtained analytically. Let Δh denote the height difference between the camera center and the ground plane, and let v_z be the vertical component of the direction vector v. The intersection point is then given by:

$$P_{ground} = v_0 - \frac{\Delta h}{v_z} v. \tag{10}$$

This formulation provides a direct and efficient way to map image pixels to 3D positions in the world coordinate system under the planar ground assumption.

4 SPATIAL REFINEMENT

The purpose of this stage is to maintain a consistent set of object tracks over time from noisy geo-localization estimates.

The tracker maintains and updates this set of object representations upon the arrival of new geo-localization estimates at each frame. The overall process is illustrated in Fig. 4.

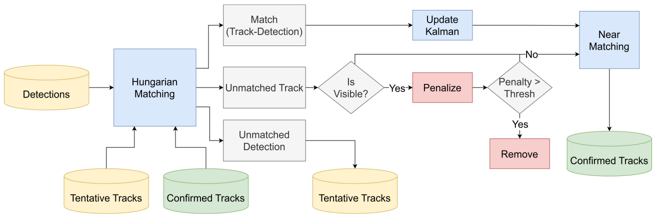


Figure 4: Overview of the spatial refinement pipeline. New detections are associated with existing tracks using the Hungarian algorithm. Tracks are updated, penalized, or removed depending on their consistency over time.

Each track represents a candidate object and is defined by:

- Estimated geographic position (latitude and longitude).
- State covariance (for uncertainty modeling).
- Track status (tentative or confirmed).
- A penalty counter used for track management.

Two sets of tracks are maintained:

- **Tentative Tracks**: recently initialized tracks that require validation.
- **Confirmed Tracks**: stable tracks corresponding to persistent objects.

At each time step, a set of new geo-localization estimates (detections) is obtained. These detections are associated with existing tracks (both tentative and confirmed) using the Hungarian algorithm.

A cost matrix is constructed based on the Mahalanobis distance between detections and track estimates:

$$d_M(x,y) = \sqrt{(\mathbf{x}_i^{det} - \mathbf{y}_j^{track})^T S^{-1} (\mathbf{x}_i^{det} - \mathbf{y}_j^{track})} \quad (11)$$

where $\mathbf{y}_i^{track}$ and $\mathbf{S}$ are, respectively, the mean and the covariance matrix of track i, and $\mathbf{x}_j^{det}$ is the position of detection j.

Assignments are accepted only if the distance is below a predefined gating threshold. The association step produces three sets:

- Matched track-detection pairs.
- Unmatched tracks.
- Unmatched detections.

4.1 Track Update with Kalman Filtering

For each matched track-detection pair, the track state is updated using a Kalman filter. The detection is treated as a measurement of the track position, allowing the filter to reduce noise and improve temporal consistency.

The update step refines both the estimated position and its associated uncertainty. This enables smoother trajectories and more robust associations in subsequent frames.

4.2 Handling Unmatched Detections

Detections that are not associated with any existing track are used to initialize new tentative tracks. These tracks represent potential new objects but require further validation before being confirmed.

4.3 Handling Unmatched Tracks

Tracks that are not matched with any detection are processed based on their visibility:

- If the track is expected to be visible in the current frame but is not detected, it is penalized by increasing its penalty counter.
- If the penalty exceeds a predefined threshold, the track is removed, as it is considered unreliable.

Visibility is determined by the camera's field of view and the estimated track position. This step helps to remove outliers from the set.

4.4 Near Matching and Duplicate Removal

To further improve consistency, a near-matching heuristic is applied to the set of confirmed tracks. Since repeated observations of the same physical object may occasionally generate multiple independent trajectories, tracks that are spatially close to one another are assumed to correspond to the same target and are therefore merged into a single representation.

This heuristic helps reduce duplicate detections and contributes to maintaining a compact and spatially consistent representation of the environment. However, its effectiveness depends on the selected distance threshold, particularly in scenarios where multiple targets are positioned very close to each other.

4.5 Final Output

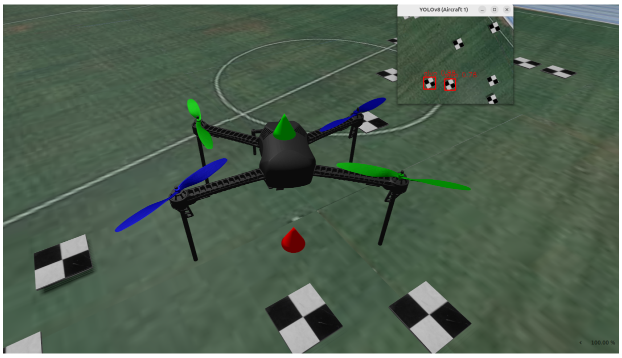
At any time step, the system maintains a set of confirmed tracks, each associated with a distinct physical object and identified by a unique ID. The geographic coordinates associated with these tracks are used as the system's final output.

5 EXPERIMENTS

5.1 Experimental Setup

Experiments were conducted in a simulated environment using Gazebo, with ArduPilot's SITL providing UAV control and state estimation. The UAV was equipped with a downward-facing monocular camera; frames were published to a ROS 2 topic and streamed via RTSP using GStreamer.

Figure 5 shows an upper view from the experimental environment and a view from the UAV's camera while flying.



(a) A view of the UAV and the camera frame.

(b) Superior view from the evaluation environment.

Figure 5: The experimental environment.

Multiple flights were performed over an area containing 30 instances of a black-and-white fiducial target, with targets placed in different positions across 3 scenarios and the UAV executing varied trajectories. These scenarios are illustrated in Figure 6, in which each plot shows the targets' geolocations and the vertices of the search area. In addition, 6 different paths were defined for the UAV to follow surrounding the targets' area, totalizing $3 \times 6 = 18$ runs. Figure 7 shows the predefined paths we used in the experiments.

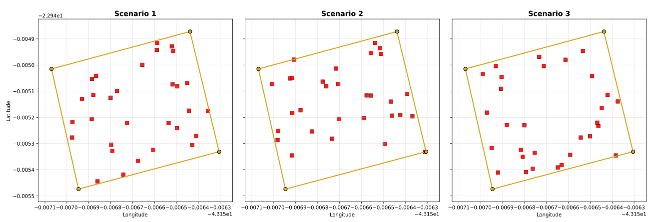
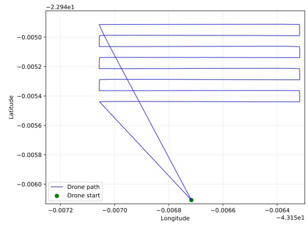
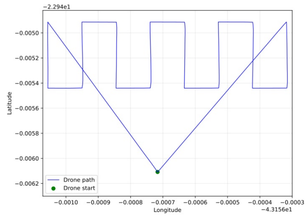


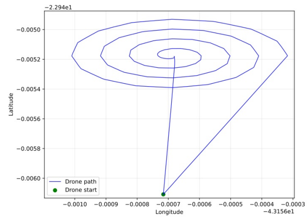
Figure 6: Scenarios of target distributions used in the experiments. The red squares indicate the targets' positions, while the yellow rectangle indicates the boundaries of the targets' area.



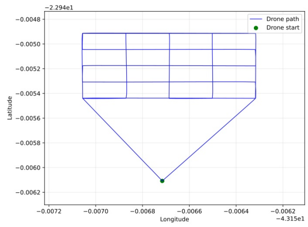
(a) Path 1



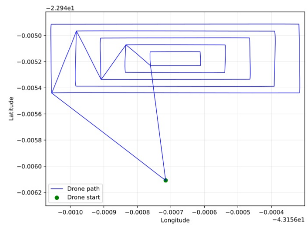
(b) Path 2



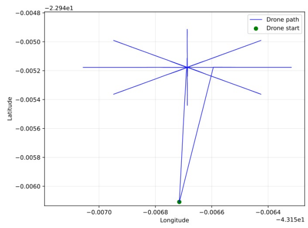
(c) Path 3



(d) Path 4



(e) Path 5



(f) Path 6

Figure 7: Pre-defined UAV paths used in the experiments.

In each run, we saved: the UAV's trajectory; the geo-coordinates (lat, long) of each detected object; and the final list of confirmed tracks.

For quantitative evaluation, each confirmed track was matched to its nearest ground-truth target. Regarding these matches, we measured the RMSE between the ground-truth targets and our estimated positions and the detections' recall, i.e., the fraction of real targets successfully identified by the system.

5.2 *Results*

Table 1 presents the quantitative results in terms of localization RMSE (in meters) and the number of correctly identified targets for each flight path and scenario. The results demonstrate the system's capability to perform geo-localization of ground objects using only a monocular camera and the UAV's standard state estimates.

Overall, the system achieved sub-meter localization accuracy in most experiments, with RMSE values ranging from 0.45 m to 0.84 m, demonstrating that the proposed spatial pipeline is capable of consolidating multiple noisy observations. Note that, both the robot's path and the targets' distribution have impacts in the accuracy.

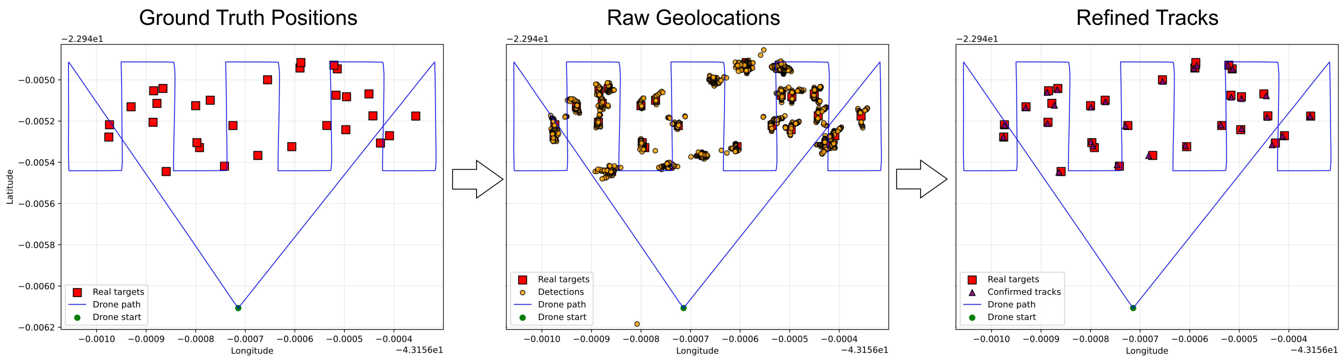


Figure 8: Results from path 2 being executed in scenario 1.

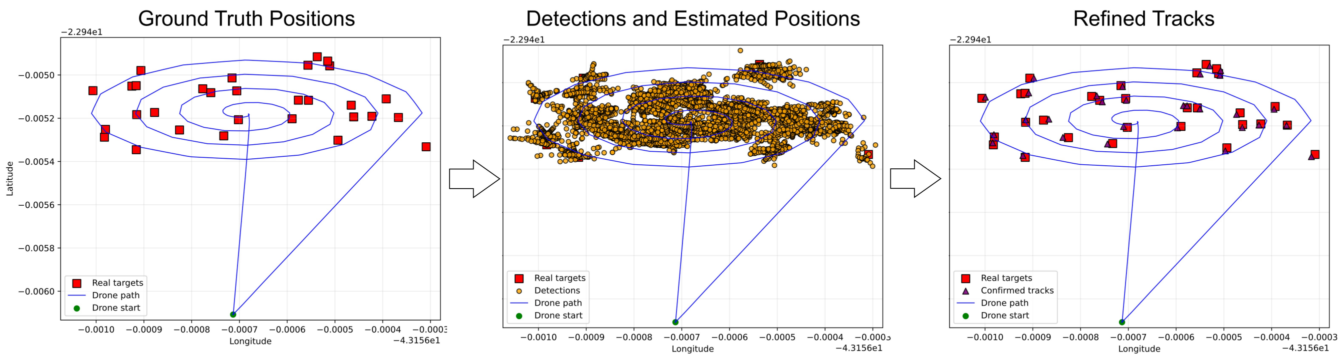


Figure 9: Results from path 3 being executed in scenario 2.

As shown in the Figures 8, 9 and 10, the raw geo-location estimates form highly dispersed point clouds around the true target positions. This dispersion is mainly caused by projection inaccuracies, GPS noise, perspective distortions, false positives from the object detector, and small UAV attitude oscillations. Nevertheless, the proposed tracking and filtering stages progressively reduce this spatial dispersion through repeated Kalman filter updates and trajectory association, resulting in a more reliable representation of the environment.

Table 1: Final geo-localization results (RMSE / Number of Targets per run).

	Scenario (RMSE [m] / Targets)		
Path	**1**	**2**	**3**
1	0.49 / 30	0.52 / 29	0.72 / 30
2	0.59 / 30	0.45 / 29	0.60 / 30
3	0.66 / 30	0.70 / 29	0.78 / 29
4	0.50 / 30	0.47 / 28	0.48 / 29
5	0.84 / 30	0.49 / 28	0.60 / 30
6	0.65 / 29	0.64 / 29	0.67 / 30

The results also show a high target recovery rate, with most runs successfully identifying all 30 targets, as illustrated in Figure 8, which shows the resulting detections and confirmed tracks when Path 2 was executed in Scenario 1. In this experiment, all targets were identified and have their corresponding tracks, with no duplicates. This behavior also happens in Scenario 3, which has one of its runs shown in Figure 10, on which all targets were correctly identified.

Experiments in Scenario 2 consistently presented a lower number of confirmed targets, with several runs detecting only 28 or 29 targets. This behavior is directly related to the spatial arrangement of the targets in this scenario, where multiple targets were positioned extremely close to one another, as shown in Figure 9. Under these conditions, small localization errors become significant enough to cause ambiguity during the association stage. As a consequence, detections originating from neighboring targets may be incorrectly associated with the same trajectory, leading the system to merge two distinct objects into a single confirmed track.

Despite the success in filtering and spatial association, the methodology presents vulnerabilities that affect accuracy. Since the camera is rigidly attached to the drone body without a stabilization system, any perturbation in the UAV attitude directly affects the camera pose, introducing additional errors into the geolocation process. Furthermore, the geometric projection model relies on a flat-ground assumption and is sensitive to inaccuracies in the onboard GPS system.

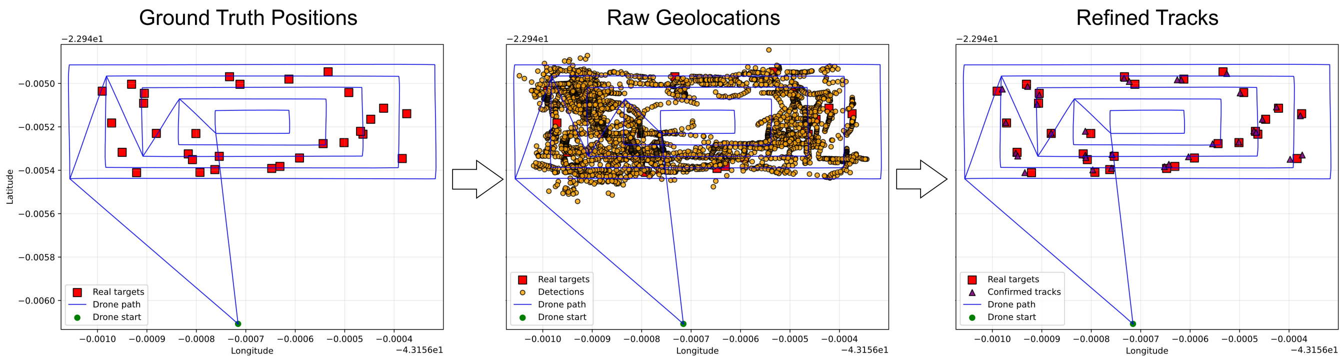


Figure 10: Results from path 5 being executed in scenario 3.

Finally, the system is affected by information latency. Although the pipeline includes a linear interpolation method to mitigate the fact that the flight controller's position and attitude data are published asynchronously, the inherent latency between image acquisition by the camera and the processing of state messages may still generate small temporal misalignments, degrading the estimation performance during high-speed flights or abrupt maneuvers.

6 Conclusion

This paper presented an online pipeline for geo-localizing multiple terrestrial objects from UAV monocular imagery. The proposed approach combines three tightly integrated components: a timestamp interpolation scheme that synchronizes camera frames with asynchronous UAV state estimates, a geometric projection model that maps image detections to ground-plane coordinates using UAV attitude and position, and a spatial refinement stage that employs the Hungarian algorithm and Kalman filtering to associate, filter, and consolidate repeated noisy observations into consistent geolocations. Together, these components enable accurate estimation of ground object positions, eliminating false positives and consolidating repeated observations into consistent geolocations through a spatial refinement pipeline.

Experiments conducted in simulation using Gazebo and ArduPilot SITL across 3 target distribution scenarios and 6 distinct flight paths demonstrated the robustness and accuracy of the approach. The system achieved a mean localization RMSE of 0.603 m while successfully identifying 97.9% of the 30 targets per scenario. These results validate the effectiveness of the spatial refinement pipeline in reducing localization noise and suppressing false positives, even under the constraints of a rigidly mounted camera and a planar ground assumption.

Future work include replacing linear interpolation with higher-order or learning-based state estimation to reduce temporal misalignments during aggressive maneuvers. Additionally, conducting experiments in real outdoor environments will be essential for further validation of the proposed methodology, as well as extending the evaluation to a broader variety of target objects beyond the fiducial markers used in this work.

References

[1] Fangbing Zhang, Tao Yang, Yi Bai, Yajia Ning, Ye Li, Jinghui Fan, and Dongdong Li. Online ground multitarget geolocation based on 3-d map construction using a uav platform. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17, 2022.

[2] Ziang Ma, Chuanzhi Chen, Jinbao Chen, and Yuhan Jiang. Stc-sort: A dynamic spatio-temporal consistency framework for multi-object tracking in uav videos. *Applied Sciences*, 16(2), 2026.

[3] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[4] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024.

[5] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements, 2024.

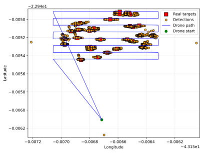
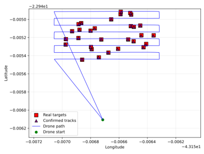
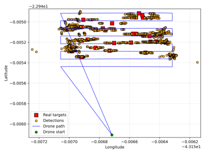
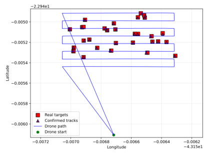
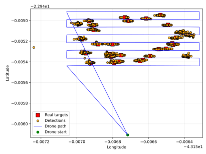
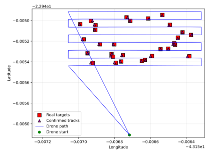
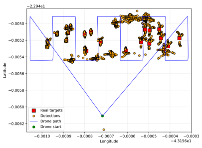
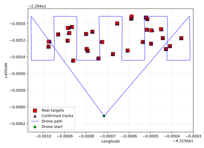
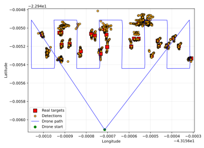
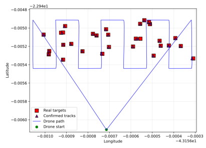
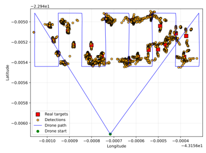
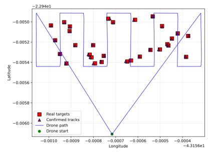
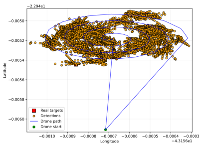
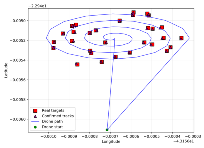
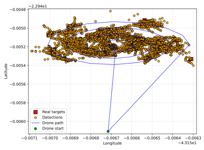
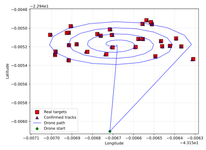
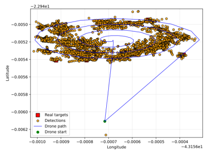
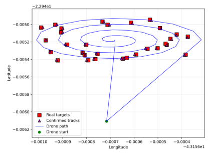
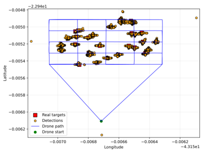
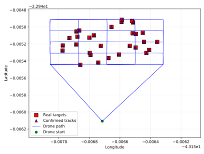
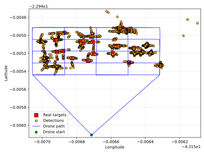
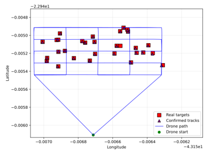
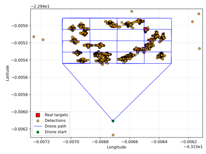
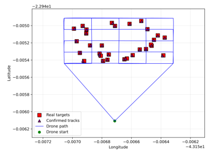
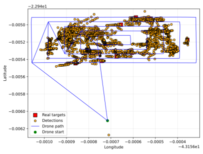
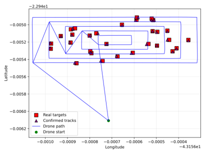
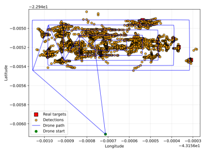
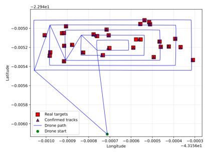
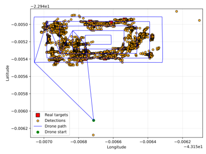
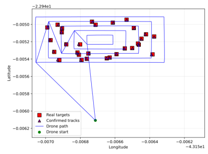
[6] Daghash K. Alqahtani, Muhammad Aamir Cheema, and Adel N. Toosi. Benchmarking deep learning models for object detection on edge computing devices. In Walid Gaaloul, Michael Sheng, Qi Yu, and Sami Yangui, editors, *Service-Oriented Computing*, pages 142–150, Singapore, 2025. Springer Nature Singapore.

[7] Bruno Eduardo de Oliveira Carvalho, Fabio Luiz Junior, Jauvane Cavalcante de Oliveira, and Paulo Fernando Ferreira Rosa. Lightweight AR System for Precise Object Geolocation in Real-Time Drone Video. In *2025 27th Symposium on Virtual and Augmented Reality (SVR)*, pages 28–38, Los Alamitos, CA, USA, October 2025. IEEE Computer Society.

[8] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, March 2004.

APPENDIX A: DATA

Table 2: Comparative analysis of targets detections vs. confirmed tracks across all paths and scenarios.

	Scenario 1		**Scenario 2**		**Scenario 3**	
Path	Detections	Conf. Tracks	Detections	Conf. Tracks	Detections	Conf. Tracks
1						
2						
3						
4						
5						
6