Pixel to Geocoordinate Mapping in Oblique and Nadir UAV Imagery

Michal Aibin*, Suchang Cao†, Siqi Chen†, Cuiting Huang†, Sujit Nashik†, Victor Wu†, Zhiyuan Yang†
*Department of Computing, British Columbia Institute of Technology
†Khoury College of Computer Sciences, Northeastern University

***Abstract*—We present a practical pipeline for geolocating pixel-level detections from UAV imagery using camera geometry, drone metadata, and machine learning-based offset correction. Our method supports both nadir and oblique views, addressing challenges such as perspective distortion, altitude uncertainty, and orientation noise. By integrating angular projection with elevation correction and a trained offset model, we achieve accurate and scalable pixel-to-GPS coordinate conversion. Evaluated on 36 targets using five metrics—geodesic error, CEP50, heatmaps, cluster compactness, and directional decomposition—our method demonstrates substantial improvements over the baseline. Elevation sampling reduces average error from 27.19m to 20.21m, while the offset model achieves 10.39m with lower variance. CEP50 and compactness metrics confirm tighter prediction spread, while heatmaps show reduced spatial error. Directional analysis reveals a consistent eastward bias. Results show enhanced accuracy and consistency across diverse UAV scenes.**

***Index Terms*—UAV imagery, geolocation, oblique images, pixel-to-coordinate mapping, drone metadata, computer vision, elevation correction, machine learning**

I. Introduction

Precise geolocation of visually detected objects in UAV imagery is increasingly vital for applications such as disaster relief, urban planning, infrastructure inspection, and environmental monitoring [1]–[5]. UAVs have increasingly been used in time-sensitive emergency contexts, where rapid aerial data collection and geolocation support critical operations such as search and rescue, damage assessment, and situational awareness [6]. The global UAV market is rapidly expanding, with the commercial drone industry forecasted to exceed US $24 billion by 2029 (BCC Research, 2024) [7]. UAVs capture detailed images from various perspectives, such as nadir (directly overhead) and oblique (angled), significantly enhancing observational capabilities [8]. Although oblique imagery enriches spatial context, it complicates accurate mapping due to perspective distortion and changing ground elevation [9]. These distortions introduce scale variation and angular inconsistencies, making feature localization and geometric interpretation significantly more difficult compared to nadir imagery [10].Consequently, accurately translating pixel-level detections—such as trees, rooftops, or road damage—into precise geographic coordinates remains a fundamental challenge.

Traditional geolocation methods often rely on simplified assumptions such as flat terrain, constant drone altitude, and nadir-only imagery, limiting their effectiveness in realistic scenarios [11]. These methods inadequately handle real-world factors like elevation variations, camera orientation errors, and sensor inaccuracies—including GPS drift, barometric altitude errors, and orientation noise from inertial measurement units (IMUs)—leading to substantial spatial inaccuracies [12].In particular, GPS signals are susceptible to loss or multipath distortion near buildings or terrain, while IMUs accumulate integration drift over time, degrading position and orientation estimates unless corrected through sensor fusion or external references. This can result in meter-level errors during flight, especially in GPS-denied environments [13].

More advanced photogrammetric techniques, like Structure-from-Motion (SfM), offer improved accuracy but typically require extensive ground control points (GCPs) or computationally demanding 3D reconstructions, making them impractical for rapid or large-scale deployments [10], [14].

Therefore, there is a critical need for practical and scalable solutions that bridge the gap between pixel-level detections and accurate geographic locations without extensive field setups or heavy computational demands. To address that, our paper provides a comprehensive pipeline that: (a) calculates azimuth and elevation angles directly from image pixels and drone metadata; (b) applies elevation corrections utilizing publicly available terrain elevation data such as the Google Elevation API; and (c) employs a machine learning regression model to predict and correct systematic positional offsets

The core challenge lies in bridging the gap between 2D pixel coordinates and 3D geographic coordinates while addressing three key technical issues:

- *Sensor Limitations:* GPS inaccuracies, altitude drift, and orientation noise from IMUs.
- *Geometric Complexities:* Non-linear distortions from camera perspective, lens curvature, and terrain relief [15].
- *Coordinate System Alignment:* Converting between camera-centered local coordinates and global geodetic systems (e.g., WGS84) [16].

To tackle these challenges, we introduce a step-by-step pipeline that improves how pixel-level detections in UAV imagery are converted to accurate coordinates. Our method integrates:

- Azimuth and elevation calculation from pixel locations in each image,
- Elevation correction using Google Maps Elevation API for accurate ground height estimation,
- A machine learning regression model to estimate pixel-to-GPS offsets using image metadata,

- Multi-view image matching to consolidate redundant detections and reduce duplication error.

II. Algorithm: Pixel-to-Geographic Coordinate Transformation using UAV Heading and Camera Geometry

We propose an algorithm to estimate real-world geographic coordinates (latitude and longitude) from image pixel locations captured by a UAV 1. The method leverages drone metadata—such as heading, altitude, and gimbal pitch—along with the camera's intrinsic parameters, including field of view and image resolution. By computing angular displacements relative to the image center, the algorithm projects these angles onto the Earth's surface to derive the geographic position of the detected object. This approach is designed to support oblique

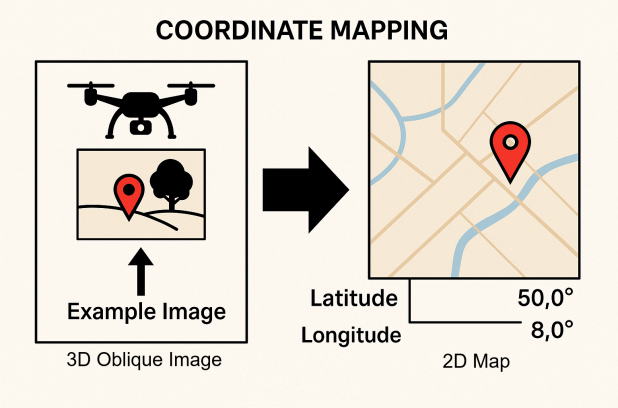


Fig. 1: 3D coordinates mapping on 2D map.

imagery and enables real-time, accurate pixel-to-coordinate transformation, making it suitable for dynamic UAV-based mapping and monitoring applications.

To estimate the geographic location of a detected object in a UAV image, the algorithm first converts the object's pixel coordinates (u, v) into angular displacements relative to the center of the image frame. We assume that the input images have undergone prior lens distortion correction, and the camera follows a standard pinhole projection model. The angular displacements are computed based on the camera's horizontal and vertical fields of view (HFOV, VFOV) and image resolution (W, H):

$$\begin{aligned} \Delta\theta_{\text{az}} &= \left(\frac{u - W/2}{W/2}\right) \cdot \frac{\text{HFOV}}{2} \\ \Delta\theta_{\text{el}} &= \left(\frac{H/2 - v}{H/2}\right) \cdot \frac{\text{VFOV}}{2} \end{aligned} \tag{1}$$

This formulation is suitable for oblique imagery because it accommodates differences in pixel distances from the optical axis in both the horizontal and the vertical directions such that angular consistency is maintained even if the camera is tilted.

These displacements are then combined with drone metadata:

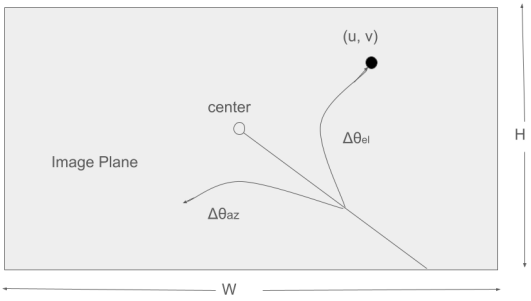


Fig. 2: Angular displacements $\Delta\theta_{az}$ and $\Delta\theta_{el}$ relative to image center (u, v).

- Heading (yaw) of the UAV, defining the reference azimuth direction;
- Gimbal pitch, indicating camera tilt below horizontal;
- Altitude, representing UAV height above ground.

In this paper, we define three coordinate frames: (1) the UAV body frame, aligned with the drone's forward direction; (2) the camera frame, oriented relative to the gimbal, where gimbal pitch indicates camera tilt from horizontal; and (3) the world frame, aligned with the local North-East-Down (NED) reference system. All angles and projections are resolved into the world frame using telemetry data, unless otherwise specified.

For simplicity, we omit roll corrections in the projection model, assuming gimbal roll effects are minimal under typical flight conditions. While this simplification is generally valid under stable flight conditions, our directional error analysis (see Section IV) reveals a consistent southward bias—most vectors fall between 90° and 270°, with deviations reaching up to 8–12 meters. This suggests that unmodeled gimbal behavior or minor roll instabilities may still introduce measurable spatial errors in some cases.

The corrected azimuth and elevation angles are calculated as:

$$\begin{aligned} \theta_{\text{az}} &= (\text{heading} + \Delta\theta_{\text{az}}) \bmod 360 \\ \theta_{\text{el}} &= \text{gimbal\_pitch} + \Delta\theta_{\text{el}} \end{aligned} \tag{2}$$

We assume that azimuth angles are measured clockwise from true north, and elevation angles (pitch) are measured in degrees below the horizontal plane, consistent with standard UAV telemetry metadata. Here, δ_θ and δ_ϕ are empirically learned azimuth and elevation offsets to correct for systematic projection error due to sensor misalignment or calibration drift.

Given the drone altitude h and elevation angle ϕ, the algorithm computes the slant range r and horizontal ground-range distance d to the target:

$$\begin{aligned} r &= \frac{h}{\sin(|\theta_{\text{el}}|)} \\ d &= r \cdot \cos(\theta_{\text{el}}) \end{aligned} \tag{3}$$

Figure 3 illustrates the geometric relationship between the UAV's altitude h, the elevation angle θ_{el}, the slant range r, and the horizontal ground distance d. These variables form a right triangle used to compute the target's position on the ground as described in Equation 3.

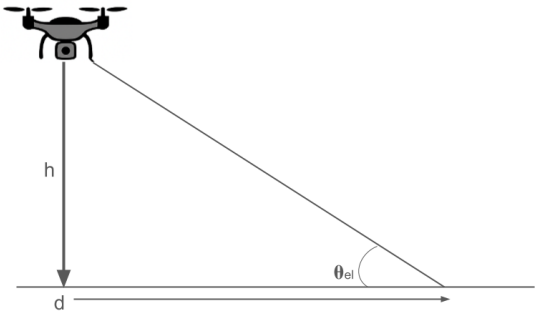


Fig. 3: Computation of slant range r and ground range d based on UAV altitude h and elevation angle θ_{el}.

The final geographic coordinates (ϕ_t, λ_t) of the detected object are computed by projecting the distance d along the corrected azimuth bearing θ from the drone's known GPS position (ϕ_0, λ_0) using spherical (great-circle) geometry.

This method assumes locally flat terrain for small fields of view, which is a common simplification in UAV-based geolocation when high-resolution digital elevation models (DEMs) are unavailable or unnecessary. In practice, this assumption holds well in urban or moderately flat regions. Future work could incorporate DEM data to improve accuracy in highly variable terrains.

A. Estimating UAV height above ground

A critical parameter for accurate target localization is the UAV's altitude above ground level (AGL). While most drone metadata reports altitude as height above mean sea level (AMSL), estimating object coordinates on the ground requires the relative height above the local terrain.

This introduces two key challenges:

- The reported GPS altitude may have significant error or drift;
- The terrain elevation varies between image locations.

To address this, we used the Google Elevation API to obtain the ground elevation at the UAV's GPS coordinates. The relative altitude is then estimated as:

$$h_{\text{rel}} = h_{\text{AMSL}} - h_{\text{ground}} \tag{4}$$

To reduce noise in the terrain elevation data, we implemented a smoothing strategy that samples the terrain elevation at the UAV's GPS point and its eight surrounding offsets (N, S, E, W, NE, NW, SE, SW). The smoothed elevation is calculated as the average of these samples.

$$h_{\text{ground}} = \frac{1}{N} \sum_{i=1}^{N} h_{\text{sample},i} \tag{5}$$

We then bounded the correction to the relative altitude to a maximum of ± 30 meters to prevent large anomalies due to elevation outliers or GPS error. This threshold was selected empirically by applying a sliding range of elevation offsets and observing which range most consistently brought the predicted coordinates closest to the ground-truth locations. A ± 30 meter bound yielded the most reliable improvements across our dataset and was therefore used in all subsequent coordinate estimations.

B. Machine learning-Based offset correction

To improve the accuracy of pixel-to-geographic coordinate transformation in UAV imagery, we apply a machine learning-based offset correction framework. This model addresses systematic angular deviations introduced by factors such as drone orientation, camera pitch (especially in oblique views), and environmental conditions.

Our approach involves training two separate Random Forest regression models to predict angular corrections—specifically, azimuth and elevation offsets. These offsets represent the difference between the estimated angles from UAV metadata and the actual ground-truth angles derived from annotated target positions. The models are trained using labeled data that includes pixel coordinates and metadata such as UAV heading and gimbal pitch. The training process is described in detail in Section III.

At inference time, the trained models predict the azimuth and elevation offsets for each new pixel detection. These predicted values are then integrated into the geolocation pipeline by adjusting the original UAV angular metadata:

$$\theta_{\text{az}}^{\text{corrected}} = (\theta_{\text{heading}} + \Delta\theta_{\text{az}}) \bmod 360 \tag{6}$$

$$\theta_{\text{pitch}}^{\text{corrected}} = \theta_{\text{pitch}} + \Delta\theta_{\text{el}} \tag{7}$$

These corrected angles are used in the forward projection model to compute the real-world geographic coordinates of the detected object. This correction framework allows for greater spatial accuracy, particularly in challenging oblique-view conditions, while retaining the interpretability of the geometric model.

III. Methodology: Machine Learning-Based Offset Correction

This section outlines the end-to-end workflow designed to transform pixel-level detections from UAV imagery into accurate geographic coordinates. The proposed methodology integrates geometric projection and machine learning-based offset modeling to address the challenges in oblique image geolocation.

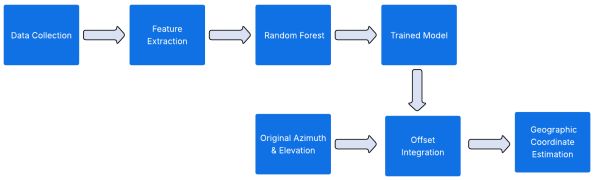


Fig. 4: Flowchart of the offset correction and coordinate estimation system for UAV imagery.

A. Data collection

To train and validate the machine learning-based offset correction model, we curated a dataset of 50 labeled pixel points along with their corresponding azimuth and elevation offsets. The pixel points were strategically selected to ensure broad spatial coverage across the image, thereby enhancing diversity in the dataset despite the limited number of samples. This, combined with our high dimensional feature space derived from drone metadata, allowed the model to generalize effectively and achieve robust performance. The imagery was captured using the DJI FC3682 camera, providing high-resolution RGB images with a resolution of 4000×3000 pixels. The dataset includes a mix of nadir and oblique images.

Each data point in the dataset comprises:

- Pixel Coordinates: Indicate the position of a target object within the UAV image frame.
- Ground-Truth Geographic Coordinates: The corresponding latitude and longitude of the target object, obtained through precise georeferencing.
- Azimuth and Elevation Offsets: Calculated as the difference between the UAV metadata-based angular projections and the actual angles derived from ground-truth positions.

This dataset serves as the foundation for training the Random Forest regressors to learn systematic deviations in azimuth and elevation angles, which are later used to refine projection accuracy during inference.

B. Feature extraction

To train the offset correction model, a diverse set of metadata-based features was extracted from each of the 50 annotated UAV image samples. These features were selected to capture the geometric, positional, and sensor characteristics that influence angular deviations in oblique UAV imagery.

Each data point is represented by the following features:

- **Pixel Coordinates**: The location of the detected object within the image.
- **Drone Heading**: The UAV's yaw angle at the time of image capture.
- **Gimbal Pitch**: The tilt angle of the camera during capture.

The input features, including pixel coordinates and heading values, were normalized to standardize the inputs for the regression model. Pixel coordinates were normalized based on the image dimensions, as described in Equation 9, while all heading values were wrapped to the range $[0°, 360°)$ using modulo operation, as shown in Equation 8.

$$\theta_{\text{normalized heading}} = \theta_{\text{heading}} \bmod 360 \tag{8}$$

C. Model training: random forest regressors

To estimate the angular correction offsets for azimuth and elevation, we used Random Forest regression, an ensemble-based method that combines the outputs of multiple decision trees to produce a more accurate and robust prediction. As introduced by Breiman, Random Forests consist of numerous decision trees, each trained on a randomly sampled subset of the training data. During the construction of each tree, a random subset of input features is considered at each split, which helps to reduce overfitting and improve generalization.

In the context of regression, each tree in the ensemble outputs a continuous value, and the final prediction is obtained by averaging these outputs. This averaging process reduces the influence of noise and improves stability, particularly when modeling complex or nonlinear relationships.

In our implementation, we trained Random Forest regression models to predict azimuth and elevation offsets based on spatial and metadata features extracted from UAV imagery. The input feature vector for each data point included normalized pixel coordinates and UAV heading. The raw pixel coordinates $(x_{\text{pixel}}, y_{\text{pixel}})$ were normalized with respect to the image center (2000, 1500) using the following formulas:

$$x_{\text{norm}} = \frac{x_{\text{pixel}} - 2000}{2000}, \quad y_{\text{norm}} = \frac{y_{\text{pixel}} - 1500}{1500} \tag{9}$$

These normalized values, together with the heading angle

$$\mathbf{X} = [x_{\text{norm}}, y_{\text{norm}}, \theta_{\text{heading}}] \tag{10}$$

Each model was trained using 100 trees and a fixed random seed of 42 to ensure consistency across runs.

IV. Metrics

The proposed geolocation pipeline is evaluated using five metrics that collectively assess positional accuracy, precision, spatial error distribution, multi-view consistency, and directional bias.

Geodesic error. Absolute positioning error is measured via the Haversine formula, which computes the great-circle distance d between predicted and ground-truth coordinates:

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\tfrac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\tfrac{\Delta\lambda}{2}\right)}\right)$$

where R is Earth's radius, (ϕ_1, λ_1) and (ϕ_2, λ_2) are the true and predicted latitudes and longitudes, and $\Delta\phi$, $\Delta\lambda$ are their differences.

This metric is widely used in UAV and GPS-based geolocation research for its simplicity and direct interpretability [16]. However, It doesn't reveal whether errors are predominantly north–south or east–west and amean can hide heavy

tails—hence we also report CEP50, heatmaps and standard deviation.

Circular error probable (CEP50). To evaluate the spatial accuracy of predicted coordinates, we employ the Circular Error Probable at the 50th percentile (CEP50) as a performance metric. CEP50 represents the radius of the smallest circle centered at the true (ground truth) location within which 50% of the predicted points fall.

Let $\mathbf{p}_i = (\text{lat}_i, \text{lon}_i)$ denote the i-th predicted location and $\mathbf{p}_{\text{true}} = (\text{lat}_{\text{true}}, \text{lon}_{\text{true}})$ be the true target location. The *geodesic error* d_i for each predicted point is computed as the great-circle distance (e.g., haversine or Vincenty) between $\mathbf{p}_i$ and $\mathbf{p}_{\text{true}}$:

$$d_i = \text{GeoDist}(\mathbf{p}_i, \mathbf{p}_{\text{true}})$$

where $\text{GeoDist}(\mathbf{p}_i, \mathbf{p}_{\text{true}})$ represents the geodesic distance function on the Earth's surface.

The set of all geodesic errors is then sorted in ascending order:

$$\{d_{(1)}, d_{(2)}, \ldots, d_{(N)}\}, \quad \text{such that} \quad d_{(1)} \leq d_{(2)} \leq \ldots \leq d_{(N)}$$

The CEP50 is defined as the value at the 50th percentile of the sorted error distances:

$$\text{CEP}_{50} = d_{(\lceil 0.5N \rceil)}$$

where N is the total number of predicted points and $\lceil \cdot \rceil$ denotes the ceiling function to ensure proper indexing.

This metric is ideal for map-based visualizations and requires no mathematical formula to interpret, offering an intuitive understanding of positional precision.

Spatial error heatmap. This metric visualizes the distribution of geodesic errors across the image frame. For each detection, the error value e_i (distance in metres) is mapped to its pixel coordinates (u_i, v_i), building an error matrix E that matches the image resolution $W \times H$.

$$E(u, v) = \begin{cases} e_i, & \text{if } (u, v) = (u_i, v_i)\,, \\ 0, & \text{otherwise.} \end{cases}$$

To reduce the influence of extreme outliers, we discard errors above the 95th percentile before visualization. Finally, we render $E(u, v)$ with a continuous color gradient from cool (low error) to warm (high error), producing a heatmap that highlights spatial patterns of high and low geolocation accuracy.

Prediction cluster compactness. The consistency of multiview predictions for the same target is quantified by computing pairwise geodesic distances among all predictions for that target, demonstrated in Figure 5. The mean intra-cluster distance is calculated for each object and then averaged across all objects. Lower values indicate tighter clustering.

Directional error decomposition. Systematic biases are assessed by decomposing each geodesic error vector into north–south (latitude) and east–west (longitude) components. Statistical summaries (mean, variance) of these components reveal directional skew in the predictions.

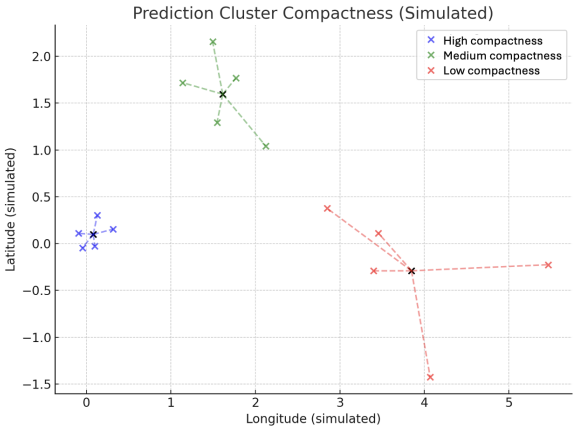


Fig. 5: Example of prediction cluster compactness.

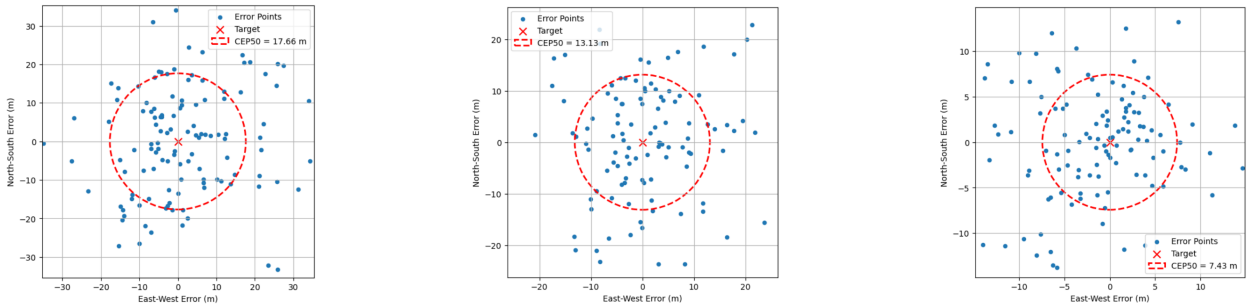
V. Results

We evaluated our geolocation pipeline on 36 diverse target locations using a combination of nadir and oblique UAV imagery. The data was captured using two drone-mounted cameras, the DJI FC3682 and DJI FC7303, to reflect real-world variability in hardware setups. For each target location, we selected between three and four images taken from different UAV viewpoints to ensure multi-angle coverage. Pixel coordinates were annotated using the LabelImg tool, achieving an inter-annotator agreement of 95%. Ground truth coordinates were obtained using Google Maps API v3, which reports an accuracy typically within 3 meters, ensuring high reliability for evaluation purposes. By comparing predicted coordinates from our geolocation pipeline to these verified ground truth positions, we assessed the spatial accuracy and consistency of different pipeline configurations.

To assess prediction accuracy, we plotted the spatial offsets between each predicted point and its corresponding ground truth. Figure 6 shows scatter plots of horizontal (east–west) and vertical (north–south) offsets for each pipeline variant. A red "×" marks the true location at the origin (0,0), and the **CEP50** radius—defined as the distance enclosing 50% of predictions—is used to quantify overall precision.

Beyond spatial accuracy, we evaluated the consistency of multi-view predictions using the **Prediction Cluster Compactness** metric, which measures how tightly predictions cluster around the target. As shown in Table I, the introduction of elevation sampling reduced the average intra-cluster distance from 27.19 m to 20.21 m. However, the offset correction model achieved the most significant improvement, reducing both the mean and variance of compactness.

Figure 7 presents **spatial error heatmaps** for each variant. Lower intensity indicates smaller geolocation errors across the



(a) Original Algorithm (b) Elevation Sampling (c) Offset Model

Fig. 6: CEP50 radii for three pipeline variants: (a) Original, (b) with Elevation Sampling, and (c) with Offset Model.

TABLE I: Prediction Cluster Compactness Consistency

Method	Mean	Median	Std Dev
Original Algorithm	27.19	20.33	24.33
Algorithm w/ Elevation Sampling	20.21	16.04	24.55
Algorithm with Offset Model	10.39	8.24	5.08

image frame. The offset model (Prediction 3) yields the most compact and evenly distributed errors. The elevation sampling variant (Prediction 2) performs moderately well but shows greater variation, while the original pipeline (Prediction 1) suffers from widespread high-error regions.

We also examined the **directionality** of prediction errors. As shown in Figure 8, most error vectors fall between 90° and 270°. This south bias warrants further investigation, particularly into potential camera pose errors from SLAM or SfM systems, as well as comparisons between error directionality and factors such as flight direction, sun angle, or camera gimbal orientation. This could stem from systematic issues such as GPS offset, gimbal tilt, or consistent UAV orientation during flight.

VI. Conclusion

We found that incorporating the elevation sampling strategy slightly improved localization accuracy compared to the original method, which used the drone's constant height without correction. The original approach often resulted in inaccurate target projections, particularly in regions with varied terrain or GPS noise, due to incorrect assumptions about the drone's actual height above the ground.

By estimating the relative altitude with localized elevation smoothing, our model produced more consistent and spatially accurate predictions across different viewpoints. However, the offset model strategy significantly improved upon both the original method and the original method coupled with elevation sampling. The offset model reduced both the mean and variance of geolocation errors by explicitly learning systematic biases present in the projection process. This led to the tightest clustering of predicted coordinates across all methods, as seen in the CEP50 plots and heatmaps.

While the offset model significantly improves geolocation accuracy and cluster consistency, it introduces a key limitation: the model must be retrained for each specific UAV camera setup. This reduces its generalizability and scalability. Moreover, the exact cause of the observed offsets is still uncertain, whether due to camera intrinsic misalignment, gimbal behavior, or GPS calibration error. Future research should focus on identifying the root causes of these systematic offsets. Understanding these sources of error could enable the development of a model that generalizes across different camera setups, reducing the need for recalibration and mitigating overfitting.

Finally, while our method performed well across a range of conditions, its applicability may vary. For example, results could differ in urban environments where or occlusions are more common, or in low-altitude flights where small changes in terrain elevation have a greater impact on relative altitude. Weather conditions like fog or strong winds may also affect GPS accuracy or image quality. Future work should explore how these environmental factors influence prediction reliability and determine the model's robustness across different operating scenarios.

References

[1] Z. Yang *et al.*, "Next-gen remote airport maintenance: Uav-guided inspection and maintenance using computer vision," *Drones*, vol. 8, no. 6, p. 225, 2024.

[2] M. Aibin *et al.*, "Advancing forest fire risk evaluation: An integrated framework for visualizing area-specific forest fire risks using uav imagery, object detection and color mapping techniques," *Drones*, vol. 8, no. 2, p. 39, 2024.

[3] J. D. Maunder *et al.*, "Ai-based general visual inspection of aircrafts based on yolov5," in *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2023, pp. 55–59.

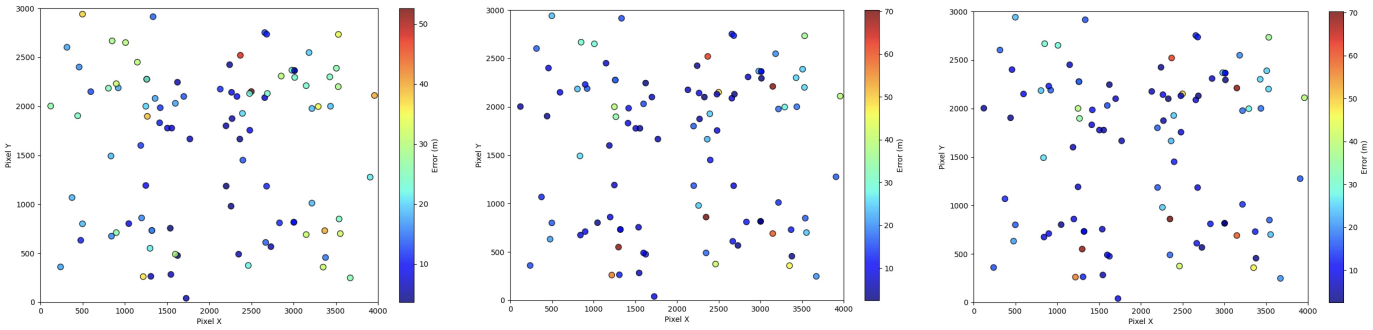
[4] A. Lee *et al.*, "Ocean medical waste detection for cpu-based underwater remotely operated vehicles (rovs)," in *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2022, pp. 0385–0389.

[5] R. Sharma *et al.*, "Aerial footage analysis using computer vision for efficient detection of points of interest near railway tracks," *Aerospace*, vol. 9, no. 7, p. 370, 2022.

[6] M. Erdelj *et al.*, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[7] BCC Research, "The commercial drone market: Trends, opportunities and future growth," https://blog.bccresearch.com/the-commercial-drone-market-trends-opportunities-and-future-growth, 2025, accessed: 2025-06-17.

[8] M. Aibin *et al.*, "Exploring the potential of oblique uav imagery for wildfire risk assessment: Comparative analysis between nadir and oblique images."



(a) Original Algorithm (b) Elevation Sampling (c) Offset Model

Fig. 7: Spatial error heatmaps for each pipeline variant. Lower intensity indicates lower error across the image frame.

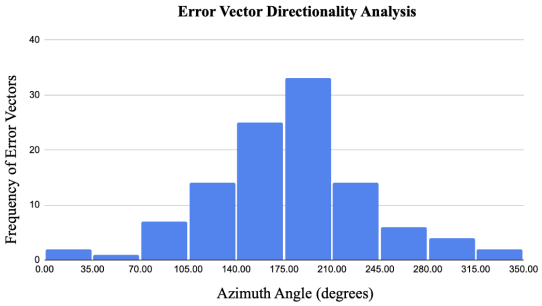


Fig. 8: Error Vector Directionality Analysis.

[9] Y. Lyu *et al.*, "Bidirectional multi-scale attention networks for semantic segmentation of oblique uav imagery," *arXiv preprint arXiv:2102.03099*, 2021.

[10] S. Jiang *et al.*, "Efficient structure from motion for oblique uav images based on maximal spanning tree expansion," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 132, pp. 140–161, 2017.

[11] D. Turner *et al.*, "An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (uav) imagery, based on structure from motion (sfm) point clouds," *Remote sensing*, vol. 4, no. 5, pp. 1392–1410, 2012.

[12] C. Stöcker *et al.*, "Review of the current state of uav regulations," *Remote sensing*, vol. 9, no. 5, p. 459, 2017.

[13] M. Erbaş *et al.*, "Review of gps and imu system performance in unmanned aerial vehicles (uavs)," *ALKÜ Fen Bilimleri Dergisi*, vol. 7, pp. 25–42, 04 2025.

[14] M. J. Westoby *et al.*, "'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology*, vol. 179, pp. 300–314, 2012.

[15] F. Nex *et al.*, "Uav for 3d mapping applications: A review. applied geomatics, 6 (1), 1–15," 2014.

[16] K.-W. Chiang *et al.*, "The development of an uav borne direct georeferenced photogrammetric platform for ground control point free applications," *Sensors*, vol. 12, no. 7, pp. 9161–9180, 2012.