PixelPath: Predicting UAV Trajectories in GPS-Restricted Environments Using Image Feature Extraction and Machine Learning

Christos Petridis(✉), Abhudaya Shrivastava, Marijana Vacic, and Zoran Obradovic

Department of Computer and Information Sciences, Center for Data Analytics and Biomedical Informatics, Temple University, Philadelphia, PA, USA
{christos.petridis,abhudaya.shrivastava,marijana.vacic, zoran.obradovic}@temple.edu

Abstract. This study introduces a novel framework for predicting UAV trajectories in GPS-restricted environments using machine learning on features extracted from aerial images. The system employs a downward-facing camera to capture aerial images during flight, extracting key features with the ORB (Oriented FAST and Rotated BRIEF) algorithm to estimate relative movement. The movement data, combined with GPS coordinates obtained during signal availability, is used to train a linear machine learning model (Linear Regression) to estimate latitude and longitude in the event of GPS signal loss. Experimental results in various suburban and mountainous environments demonstrate that the proposed framework accurately estimates UAV positions, showing improved performance in visually diverse areas compared to less diverse environments like mountainous regions. This approach offers a promising solution for enhancing UAV operations in GPS-restricted scenarios.

Keywords: Drone's Trajectory · Feature Extraction · Machine Learning · GPS Restricted Environments

1 Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are aircrafts that operate without a human pilot onboard. UAVs can be controlled remotely by an operator or can function autonomously using onboard computers, sensors, and navigational systems. They are utilized in various fields, including Search and Rescue (SAR) operations [9], commercial sectors for delivery and infrastructure inspection [11], agriculture for crop monitoring [13], and mapping and data collection [4]. UAVs have become essential tools for many industries that require aerial capabilities without the risks and costs associated with manned aircraft.

The Global Positioning System (GPS) is a satellite-based navigation system that provides accurate geolocation and time information to a GPS receiver

I. Maglogiannis et al. (Eds.): AIAI 2025, IFIP AICT 757, pp. 1–15, 2026.
https://doi.org/10.1007/978-3-031-96231-8_20

anywhere on or near Earth's surface. GPS is essential for UAV navigation, as it helps maintain position during flight, follow predefined paths, and geotag data collected during missions. Many UAVs also rely on GPS for stability and autonomous functions, including Return-to-Home (RTH) features. However, this reliance makes UAVs vulnerable to situations where GPS signals are weak or intentionally disrupted. A GPS-restricted environment refers to any location or situation where GPS signals are limited or unreliable. Such environments exist in areas with significant obstructions, such as dense urban centers, forests, and indoor spaces [2]. Natural limitations, such as operating in remote areas with weak satellite coverage, also contribute to GPS denial. In GPS-restricted environments, navigation is difficult without a reliable external reference, leading to position drift and probably potential mission failure. For data collection missions, geo-referencing becomes impossible when there is no GPS at all. These limitations necessitate alternative approaches to ensure UAVs can operate effectively in such environments. Therefore, when GPS fails, there is a need for a system that can estimate a UAV position, allowing drones to complete their tasks or, at the very least, return safely to their starting point. This would make drone operations more reliable and safer, expanding their use in many areas. To address this need, we propose a novel mechanism to estimate UAV's position in case of a GPS loss, exploiting visual information and machine learning.

In this study, we have developed a framework that can infer the position of a drone based on information extracted from aerial images taken during flight using the drone's onboard camera. More specifically, while the GPS signal is available, we frequently collect aerial images and match them with their corresponding coordinates. We extract features between consecutive images to identify drone's movement and learn the impact that each movement has on drone's consecutive coordinates (latitude and longitude). Using this information, we can train a machine learning model to capture the dependencies between various movements and learn a relationship between movements and coordinates. When the GPS signal is lost, this framework estimates the drone's position by applying this learned relationship.

2 Related Work

While many studies exist in this field, this section focuses on the most relevant ones and the algorithms that can be applied to predict a drone's trajectory.

Visual Odometry (VO) [10] is a key technique in robotics and autonomous systems, enabling the estimation of a UAV position and orientation using a sequence of onboard camera images. VO is especially useful in GPS-restricted environments, offering an alternative to traditional localization methods. It can be approached in three ways [1]: feature-based, appearance-based, and hybrid. The feature-based approach relies on detecting and tracking distinct points like corners or edges across frames, using algorithms such as ORB [14], SIFT [6], and SURF [3]. The appearance-based approach focuses on analyzing the overall image appearance, employing methods like dense optical flow (e.g., Horn-Schunck [5]) or sparse optical flow (e.g., Lucas-Kanade [7]) to capture pixel displacement. Finally, the hybrid approach combines both methods, leveraging feature-based precision and appearance-based techniques' robustness, making it appropriate for low-texture or dynamic lighting conditions.

W. Power et al. [12] presented a structured learning framework for enabling drone swarms to navigate autonomously in GPS-denied environments. They used a structured learning approach combined with a recurrent neural network to predict drone positions. They also incorporated inter-drone communication and simplified sensor inputs for trajectory data to enable coordination and collision avoidance in GPS-denied environments.

P. Shu et al. [15] introduced a trajectory prediction model for UAVs using Long Short-Term Memory (LSTM) networks. The model uses historical trajectory data to forecast future UAV positions to enhance operational safety, aiding collision avoidance and dynamic path planning. The stacked Bidirectional and Unidirectional LSTM (SBULSTM) network uses the four positions before the current time of UAV to predict the position of the next time during cruise operation. A limitation of this model is its reliance on large volumes of labeled historical data, which may reduce its adaptability in scenarios with sparse data or highly dynamic environments. Additionally, the training dataset plays a key role in this work, and capturing all possible paths in a dataset is impossible.

Zhang et al. [15] proposed a method that enhances UAV trajectory prediction by integrating flight state recognition. In this approach, UAV flight data is preprocessed, including position, velocity, acceleration, direction, and curvature, and the Principal Component Analysis (PCA) is used to construct flight features. A Support Vector Machine (SVM) model is employed on constructed features to classify UAV flight states into climb, level flight, hover, turn, and descent categories. A neural network model is developed for trajectory prediction for each flight state. Experimental results from this approach demonstrate significant improvements in prediction accuracy, with errors reduced to within 0.5 m, showcasing the effectiveness of combining flight state recognition with neural network-based prediction models.

However, to the best of our knowledge, no published work matches our novel approach of estimating UAV trajectories using a straightforward combination of feature extraction, matching, and machine learning, showcasing its simplicity and uniqueness.

3 Our Methodology

In this section, we present our proposed methodology, discussing the drone's camera placement, aerial images, feature extraction mechanism, data gathering, and machine learning aspects.

3.1 Camera Placement

Our proposed framework involves utilizing a downward-facing camera mounted on the drone to capture images (there is no need for colorful images; grayscale is also compatible). The camera must adjust its capture rate dynamically based on the drone's velocity, with faster velocities resulting in higher image capture frequency. The capture rate in our experiments is 2 m, which means we capture an image when the drone moves 2 m. Figure 1 shows what the downward-facing camera is capturing during a flight.

3.2 Capturing Aerial Images

During the drone flight, we collect aerial images for analysis, processing, and applying feature extraction. Conducted experiments are based on high-fidelity simulated environments (capturing aerial images through the Google Maps interface). We have implemented a framework to capture such aerial images given a set of specific coordinates (flight path). When GPS is available, we know the coordinates of each image, corresponding to the center of the image. An example of six consecutive aerial images is presented in Fig. 1. The differences between two consecutive images are minimal, but this is very important for the project because we want to have mostly the same objects in consecutive images.

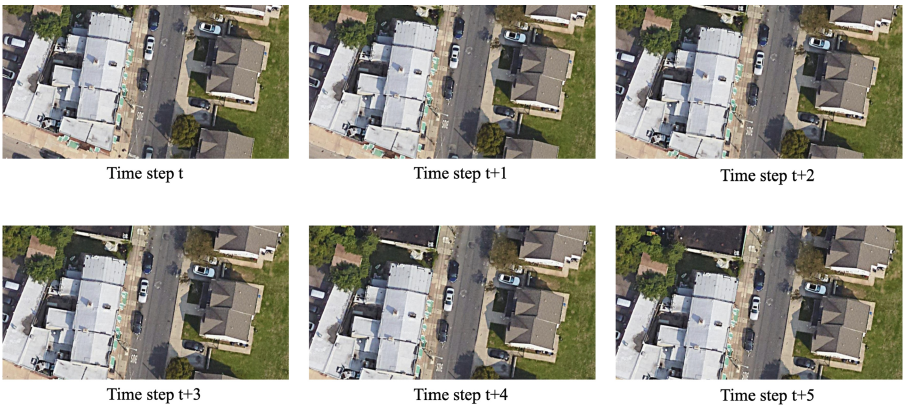

Fig. 1. An example with six consecutive images in a suburban environment.

3.3 Feature Extraction Mechanism

Having consecutive images, we estimate the relative positional displacement between them. Therefore, we implemented a feature-based approach using the ORB (Oriented FAST and Rotated BRIEF) [14] algorithm for keypoint detection and descriptor extraction. Our approach identifies distinctive features within each image and generates binary descriptors for matching. ORB descriptors are binary, and they were compared using a brute-force matcher with Hamming distance, which is very efficient when comparing binary entities. The matches were ranked based on their distance, with the most accurate pairs prioritized for further processing. A minimum threshold (10) was applied for a reliable estimation, retaining only the strongest matches. This threshold was chosen empirically to strike a balance between rejecting weak or ambiguous matches and retaining sufficient data for accurate localization. When the threshold is not so strict, the algorithm may include poor-quality matches, leading to increased positional noise. These selected key points were then used to compute a homography matrix through the RANdom SAmple Consensus (RANSAC) algorithm, which effectively identifies and excludes outliers. The matched points identified through this process allowed for accurate alignment of the images, making it possible to calculate the average movement in both horizontal and vertical directions based on the differences in the coordinates of the matched key points. We estimate the camera's relative movement measured in pixels by calculating the average displacement of matched key points. Figure 2 shows annotated images with matched features and their relative coordinates in image space. Finally, movement data is aggregated with associated GPS coordinates to track drone's trajectory, indicating an iterative approach for real-time motion estimation and spatial mapping.

Fig. 2. An example for feature extraction and matching between two consecutive images. The left image is at time step $t+1$, and the right one is at time step t.

3.4 Data Gathering

The dataset obtained after the feature extraction mechanism contains horizontal and vertical average movements between consecutive images and the corresponding changes in latitude and longitude (Table 2). The first row in Table 2 contains zeros for horizontal and vertical average movements because it cannot be compared with previous images since it is the first frame captured. Afterward, we calculate the difference in latitude and longitude between two consecutive images. Using this technique, we determine how the latitude and longitude change in relation to the average movements. The third row in Table 2 shows that the Longitude difference is equal to 0.0 and it makes sense since there is no change in longitude from img1.png to img2.png (as shown in Table 1). In order to enhance our framework, we split the dataset presented in Table 2 into two datasets: 1) One contains the average movements and Latitude Difference, and 2) the other contains the average movements and Longitude Difference. This separation is since horizontal and vertical movements affect latitude and longitude differently. Therefore, two separate models are needed to capture these relationships better.

Table 1. A snippet of the dataset with image coordinates during drone flight.

Image	Latitude	Longitude
img0.png	39.96715437762873	-75.1565855741501
img1.png	39.96716825267808	-75.15658020973207
img2.png	39.96718623908536	-75.15658020973207
...	...	...
img591.png	39.96700617406641	-75.15663385391238
img592.png	39.967022716179706	-75.15662587111866

Table 2. A snippet of the dataset following the comparisons between consecutive images.

Comparison	Horizontal average movement	Vertical average movement	Latitude Difference	Longitude Difference
img0 - img0	0.0	0.0	0.0	0.0
img1 - img0	−8.70	24.22	0.00001387504934768913	0.0000053644180297852
img2 - img1	−3.01	31.73	0.00001798640727912471	0.0
...	...	...	...	...
img591 - img590	−7.18	34.11	0.0000176890814316266	0.0000031513088032398
img592 - img591	−13.90	28.29	0.00001654211329338295	0.0000079827937184973

3.5 Machine Learning Aspect

Our framework uses movement data and machine learning. We employ two machine learning models: one for predicting latitude and one for predicting longitude. A snippet of the dataset that we use is presented in Table 2. When GPS signal is lost, the last known position is saved, and the system switches to prediction mode. The drone continues collecting data (horizontal and vertical average movements based on comparisons between consecutive images), which is the input into the trained models to predict the offsets for latitude and longitude. These offsets are then added to the previous position to estimate drone's current position. This iterative process enables the drone to navigate accurately in GPS-denied environments using only movement data and linear regression. After experimenting with several machine-learning algorithms, such as Random Forest Regressor and XGBoost Regressor, we decided to continue with Linear Regression since it outperforms the alternative algorithms when trained on very few data points (early GPS signal loss) for both latitude and longitude prediction. Moreover, linear regression is notably less computationally expensive in our case, where only two features are involved (horizontal and vertical average movement). The limited number of features means that the necessary computations, such as matrix multiplications and inversions, are performed on small matrices, resulting in significantly reduced processing time and resource consumption. More complex models, such as MLP Neural Networks and ensemble methods, tend to perform better with high-dimensional data and often require iterative optimization processes, which demand greater computational resources. We also need a computationally efficient solution, as the system may need to run on edge devices for real-time estimation instead of transmitting the images over a network to a control center. A flow chart of the system is presented in Fig. 3 to provide a clearer understanding of the main idea.

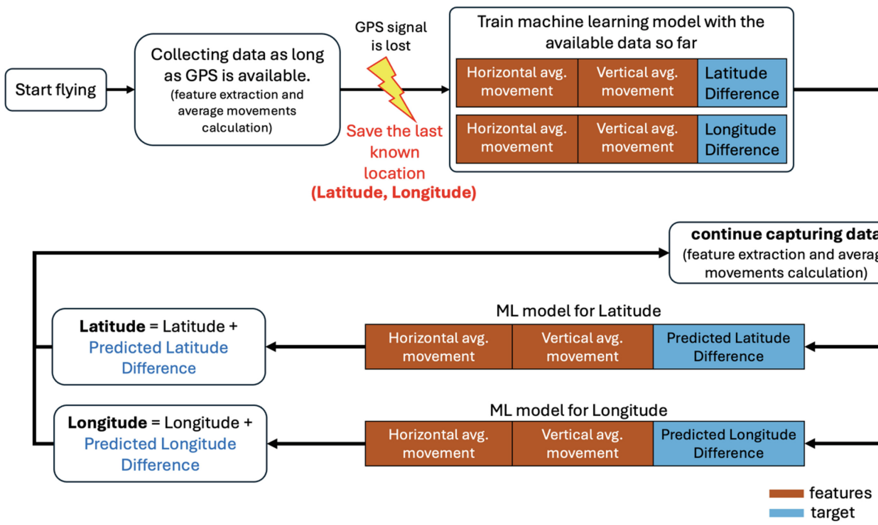

Fig. 3. A flow chart of the proposed framework.

4 Experimental Results

In this section, we present our methodology through various experiments in different scenarios, including suburban and mountainous areas. We utilize the Haversine distance to evaluate the predictions against the ground truth coordinates since it has been tested in previous experiments [8]. In real-life scenarios,

Table 3. Description for each color pin in figures.

Green Pin	It indicates drone's trajectory **with GPS.** It is the **training dataset**. This is what the model knows before the GPS loss.
Blue Pin	It indicates the **actual location** of the drone **after the GPS loss.**. This is the **test dataset** and is unknown in real-life scenarios.
Red Pin	It indicates the **predicted location** of the drone.
Orange Pin	It indicates the **entire trajectory** of the drone.

we cannot ascertain how close we are to the ground truth values because they are unknown. However, for experimental purposes, we assume that the GPS signal is lost at a specific point to showcase our approach. Table 3 explains the different colors used for the pins in our figures.

4.1 Environments for Experimental Setup

Figure 4 demonstrates three trajectories, two of which are located in suburban environments and one in a mountainous area. We evaluate our approach in these scenarios by following the described methodology, including the feature extraction mechanism and the described data pre-processing.

Fig. 4. Examples of drone trajectories in suburban (left and right) and mountainous (middle) environments. Yellow arrows indicate the drone's direction and starting/ending points are also labeled. (Color figure online)

4.2 Experimental Analysis of Suburban Environments

We present three scenarios in which GPS becomes unavailable at different time steps (Fig. 5), ordered from left to right based on the point at which the signal is lost. For evaluation purposes, we compare the actual trajectory with the predicted one and present the Haversine distance as a prediction error for reference. When only a limited amount of historical data is available (left figure), the

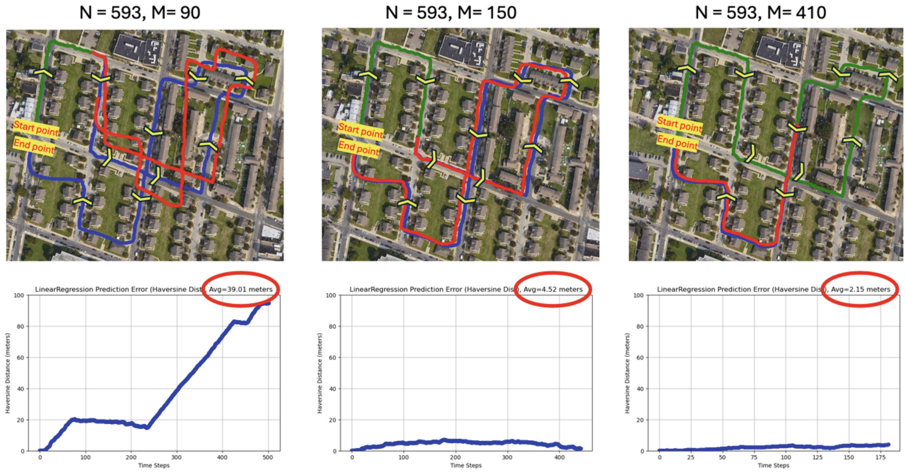

Fig. 5. Losing GPS signal in a suburban environment at different time steps. N represents the total points and M the training points (i.e., when GPS is available). Haversine distance is also presented as the prediction error for evaluation, yellow arrows indicate drone's direction, and starting/ending points are also labeled. (Color figure online)

model struggles to establish a strong relationship between the coordinates and the moving averages, leading to significant deviations between the predicted and actual positions (avg. prediction error = 39.01 m.). Despite this, the linear model still captures the general movement pattern of the drone. As more data becomes available (middle and right figures), the model benefits from a better understanding of the environment and motion dynamics, resulting in significantly improved performance (avg. prediction error = 4.52 m and 2.15 m.).

The problem with this approach is that we need a substantial amount of historical data. However, it seems that we don't need just a large quantity of historical data; we also require diverse data. If the historical data is sufficient in quantity but lacks diversity, we cannot accurately predict the drone's position. This is because the machine learning model fails to capture possible directions (north, east, west, south) during training if the training dataset is not diverse. This behavior is demonstrated in Fig. 6 where we present five different scenarios where GPS becomes unavailable, arranged in ascending order based on the point of GPS loss. In these scenarios, the drone initially flies in a straight line for almost 500 m before making three left turns. The problem is that the data collected from the straight-line segment lacks diversity. Specifically, it is clear that when M = 90, M = 105, and M = 275 (N = 508 constantly), the training dataset was insufficient due to its lack of diversity. Even though the scenario with 275 training instances (M = 275) includes almost three times more data than the first scenario with 90 training instances (M = 90), the performance in predicting future locations remains poor. However, as the training dataset becomes larger and more diverse (M = 350 and M = 410), we have very accurate predictions, and the prediction

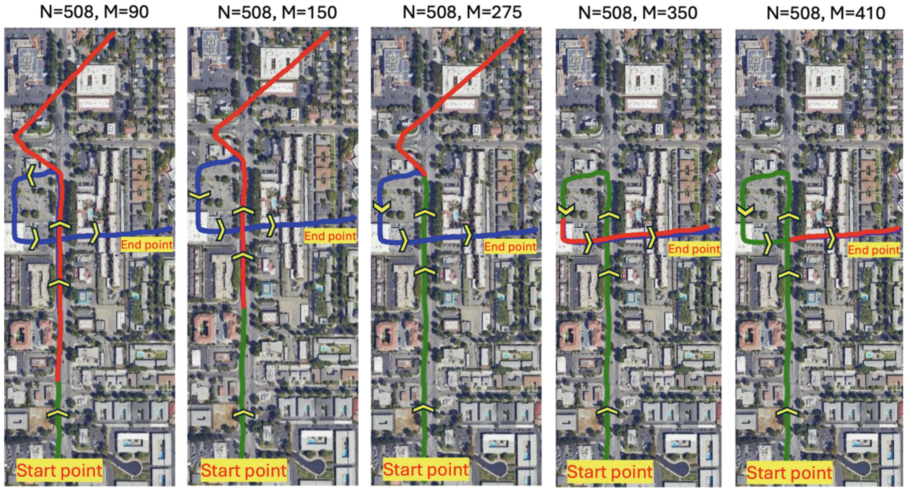

Fig. 6. Five examples to show that a large quantity of historical data is not enough if they are not diverse. N represents the total points and M the training points (i.e., when GPS is available). In total we have N = 508 historical data. Yellow arrows indicate drone's direction, and starting/ending points are also labeled. (Color figure online)

error decreases significantly, as shown in Figs. 6 and 7. This increased diversity in the data enables the model to perform better, as it would have been trained in all four possible directions. Based on this observation, we propose a new technique where the drone needs to explore a square (approximately 70 m on each side) before the actual flight to collect a diverse training dataset no matter the actual flight.

To illustrate this, we present the results of another experiment where the drone performs the square in the beginning, and afterward, it starts the actual flight in a zig-zag trajectory to make it more difficult. Drone's trajectory is shown in the rightmost figure in Fig. 4. As presented in Fig. 6, we present four different scenarios in Fig. 8 where GPS becomes unavailable, arranged in ascending order based on the point of GPS loss. Similarly to the first example (Fig. 5), the linear model captures the general movement of the drone despite having a limited training dataset, although the predicted locations are not accurate. Figure 8 also shows the employed metric for evaluation, the Haversine distance, to quantify the accuracy of the predicted locations. After the first two turns, we already have more accurate predictions, and by the time that the square has been performed (when M = 160), the prediction error is less than 1 m. The zig-zag trajectory is quite challenging, yet our framework performs very well in this scenario. As expected, the more diverse and extensive the training data, the better the model performs, resulting in more accurate predictions and a lower prediction error (Haversine distance). This technique appears to be highly beneficial since it allows the model to learn about the environment and understand how average movements impact the coordinates.

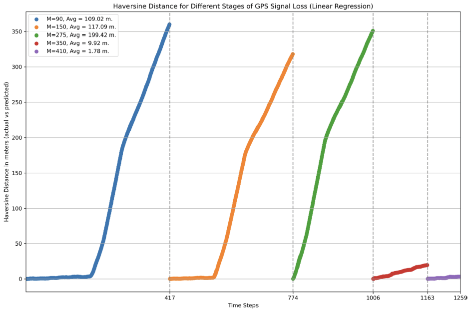

Fig. 7. The Haversine distance for the five scenarios of GPS signal loss, as shown in Fig. 6. The blue and orange scatter plots start with very small values because the model predicts accurately during the straight-line segments (errors increase when it turns). The red and purple scatter plots have much lower values overall, as the predictions are more accurate. (Color figure online)

4.3 Experimental Analysis of Mountainous Environments

Dealing with scenarios where the environment lacks diversity, such as flying over a forest, the feature extraction mechanism may theoretically encounter challenges in recognizing and matching key points across consecutive images. Here, we present the results of the performance of our proposed technique in a repetitive environment, and the trajectory is shown in the middle figure of Fig. 4. As illustrated in Fig. 9, it is quite difficult for human observers to detect consistent features in consecutive images. Despite the environment's repetitive and less diverse nature, our proposed technique performs well. Figure 10 illustrates four scenarios with GPS signal loss at different time intervals. As indicated, the drone explores a square before the actual flight for the reasons mentioned earlier. We also provide the Haversine distance for evaluation purposes, and as time progresses and the training dataset expands, performance improves significantly, and prediction error decreases. Similar to previous experiments, despite the poor performance with limited training data (M = 100), it is important to note that the model accurately captures the general movement even with limited training data.

4.4 Experiments with Less Frequent Image Capture Rate

In our experiments, images were captured every 2 m of drone movement. We also tested less frequent capture rates of 3.5, 5, 8.5, and 10 m. When changing the frequency from 2 m to 3.5 m in the environment shown in Fig. 8, we observed a general decrease in prediction error over time, as expected (Fig. 11). However, the error remained higher than in the 2-m frequency approach which resulted in significantly better outcomes. While the 3.5-m frequency may be a viable alternative in cases of hardware limitations, less frequent capture rates (5, 8.5,

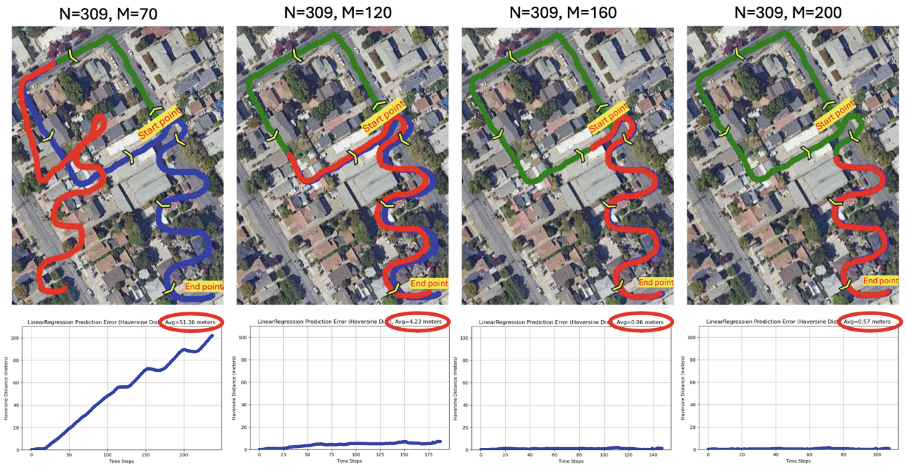

Fig. 8. Four scenarios in a suburban environment with GPS signal loss at different time steps. N represents the total points, and M represents the training points (i.e. when GPS is available). Yellow arrows indicate the drone's direction, and starting/ending points are also labeled. Below each scenario, we present the Haversine distance as the prediction error for evaluation. (Color figure online)

(a) Flying over a forest at time step t.

(b) Flying over a forest at time step t+1.

Fig. 9. Two consecutive images over a forest. The repetitive and homogeneous nature of the forested area makes it challenging to detect and recognize consistent features across the images.

and 10 m) lead to poor performance. The capture frequency may also be linked to altitude, which we plan to address in future work.

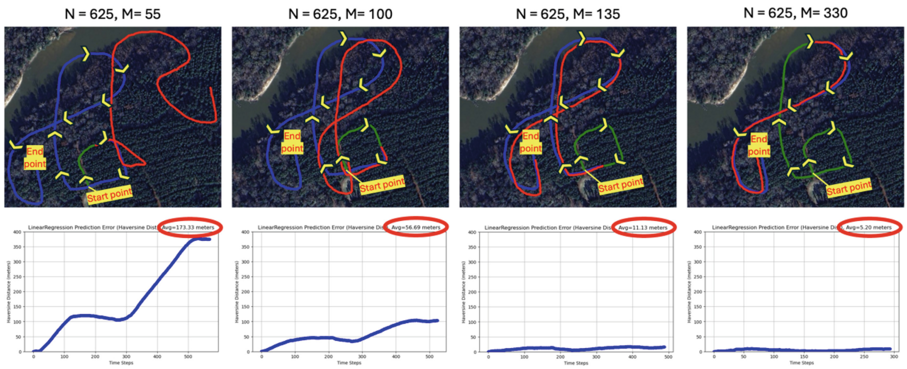

Fig. 10. Four scenarios in a mountainous environment with GPS signal loss at different time steps. N represents the total points and M the training points (i.e., when GPS is available). Yellow arrows indicate drone's direction, starting/ending points are also labeled and below each scenario, we present the Haversine distance as the prediction error for evaluation. (Color figure online)

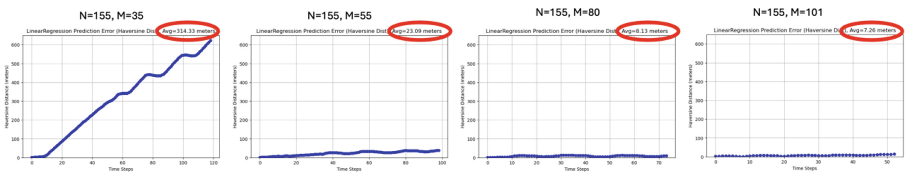

Fig. 11. Four scenarios in a suburban environment with GPS signal loss at different time steps. N represents the total points and M the training points (i.e., when GPS is available).

5 Discussion

Using feature extraction and a linear machine learning model, our proposed framework effectively estimates UAV trajectories in GPS-restricted environments. We firmly believe that Haversine distance is the most suitable metric for evaluating performance across all experiments. Predictions in suburban environments improve with more diverse training data, demonstrating that the quantity and diversity of historical data are crucial for more accurate results. When incorporating a square flight path before the actual flight, the performance is enhanced significantly, with predicted positions closely matching actual locations (mean prediction error = 0.57 m). Despite the repetitive terrain, which makes the feature extraction more challenging in mountainous environments, our framework maintained a mean prediction error of 5.20 m. Since our approach heavily relies on feature extraction, its performance is sensitive to the diversity of the environment. This reliance explains the less accurate predictions observed in mountainous regions. In contrast, more diverse environments, such as suburban areas, provide richer visual features that improve the overall performance. Additionally, our framework assumes a constant drone altitude throughout the flight, simplifying the model but not reflecting real-world conditions where altitude can vary. Incorporating altitude changes could enhance the robustness and accuracy of our system. Our future work will focus on improving feature extraction in low-diversity environments and integrating dynamic altitude adjustments to create a more reliable framework for UAV navigation in GPS-restricted environments.

6 Conclusions

This study presents a reliable and efficient method for UAV navigation in GPS-restricted environments by combining image feature extraction with machine learning. The results highlight 1) the importance of diverse training data for accurate trajectory prediction, 2) that a linear model can capture the motion dynamics, and 3) repetitive environments pose challenges, indicating room for improvement.

Acknowledgments. This research was sponsored by the U. S. Army Engineer Research and Development Center (ERDC) and was accomplished under Cooperative Agreement Number W9132V-23-2-0002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Engineer and Development Center (ERDC) or the U.S. Government.

References

1. Aqel, M., Marhaban, M.H., Saripan, M.I., Ismail, N.B.: Review of visual odometry: types, approaches, challenges, and applications. Springerplus **5**(1), 1–26 (2016). https://doi.org/10.1186/s40064-016-3573-7
2. Balamurugan, G., Valarmathi, J., Naidu, V.P.S.: Survey on UAV navigation in GPS denied environments. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 198–204. IEEE (2016)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
4. Gillan, J.K., Ponce-Campos, G.E., Swetnam, T.L., Gorlier, A., Heilman, P., McClaran, M.P.: Innovations to expand drone data collection and analysis for rangeland monitoring. Ecosphere **12**(7), e03649 (2021)
5. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**(1–3), 185–203 (1981)
6. Lindeberg, T.: Scale invariant feature transform (2012)
7. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI'81: 7th International Joint Conference on Artificial Intelligence, vol. 2, pp. 674–679 (1981)
8. Maria, E., Budiman, E., Taruk, M., et al.: Measure distance locating nearest public facilities using haversine and Euclidean methods. J. Phys. Conf. Ser. **1450**, 012080. IOP Publishing (2020)

9. Mayer, S., Lischke, L., Woźniak, P.W.: Drones for search and rescue. In: 1st International Workshop on Human-Drone Interaction (2019)
10. Nister, D., Noriditsky, O., Bergen, J.: Visual odometry. In: IEEE Conference Computer Vision and Pattern Recognition CVPR, vol. 4 (2004)
11. Park, J., Kim, S., Suh, K.: A comparative analysis of the environmental benefits of drone-based delivery services in urban and rural areas. Sustainability **10**(3), 888 (2018)
12. Power, W., Pavlovski, M., Saranovic, D., Stojkovic, I., Obradovic, Z.: Autonomous navigation for drone swarms in GPS-denied environments using structured learning. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) AIAI 2020. IAICT, vol. 584, pp. 219–231. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49186-4_19
13. Rejeb, A., Abdollahi, A., Rejeb, K., Treiblmaier, H.: Drones in agriculture: a review and bibliometric analysis. Comput. Electron. Agric. **198**, 107017 (2022)
14. Rublee, E., Rabaud, V., Konolige, K., Bradski, G., Orb: an efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision, pp. 2564–2571. IEEE (2011)
15. Shu, P., et al.: Trajectory prediction of UAV based on LSTM. In: 2021 2nd International Conference on Big Data and Artificial Intelligence and Software Engineering (ICBASE), pp. 448–451. IEEE (2021)