

Kilometer-Scale GNSS-Denied UAV Navigation via Heightmap Gradients: A Winning System from the SPRIN-D Challenge

Michal Werner, David Čapek, Tomáš Musil, Ondřej Franek, Tomáš Báča, and Martin Saska

Faculty of Electrical Engineering, Czech Technical University in Prague

wernemic@fel.cvut.cz

Abstract—Reliable long-range flight of unmanned aerial vehicles (UAVs) in GNSS-denied environments is challenging: integrating odometry leads to drift, loop closures are unavailable in previously unseen areas and embedded platforms provide limited computational power. We present a fully onboard UAV system developed for the SPRIN-D Funke Fully Autonomous Flight Challenge, which required 9 km long-range waypoint navigation below 25 m AGL (Above Ground Level) without GNSS or prior dense mapping. The system integrates perception, mapping, planning, and control with a lightweight drift-correction method that matches LiDAR-derived local heightmaps to a prior geodata heightmap via gradient-template matching and fuses the evidence with odometry in a clustered particle filter. Deployed during the competition, the system executed kilometer-scale flights across urban, forest, and open-field terrain and reduced drift substantially relative to raw odometry, while running in real time on CPU-only hardware. We describe the system architecture, the localization pipeline, and the competition evaluation, and we report practical insights from field deployment that inform the design of GNSS-denied UAV autonomy. SUPPLEMENTARY MATERIALS: <https://gnssdenied.github.io/>

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are increasingly deployed for infrastructure inspection, logistics, and search-and-rescue. Many of these missions require long-range, low-altitude flight in areas where Global Navigation Satellite Systems (GNSSs) are unreliable, unavailable, or denied. Without GNSS, UAVs typically rely on visual-inertial or LiDAR odometry to navigate a previously unseen environment. These methods are consistent locally, but accumulate unbounded drift in previously unvisited areas, where loop closures cannot be used. When flying over kilometer-scale trajectories, this leads to position errors too large for waypoint-based navigation. The core problem addressed in our paper is therefore how to manage the drift using available geodata on a low-altitude resource-constrained UAV.

Existing methods address parts of the problem – deep-learning geolocalization achieves high accuracy but is too computationally heavy to run in real-time on embedded platforms, while odometry-only systems based on Visual-Inertial Odometry (VIO) or Lidar-Inertial Odometry (LIO) are efficient but accumulate excessive drift. At high altitudes, UAVs can use satellite data to correct their positions, but this becomes unfeasible close to the ground or under trees. Fully onboard systems that scale to kilometer ranges in diverse unseen environments remain rare, since outdoor geodata-based drift correction that is both lightweight and robust has not been solved. Addressing these challenges requires not only an individual algorithm but the integration of

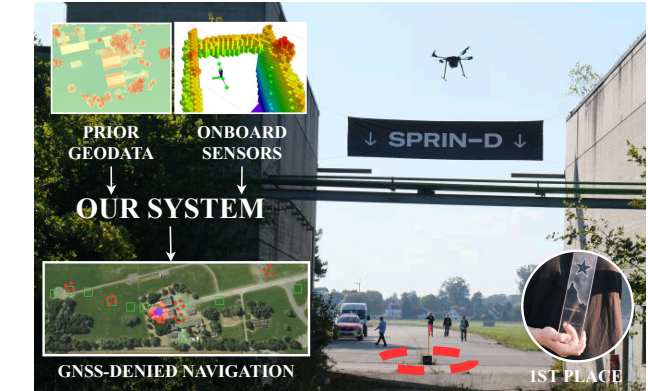


Fig. 1: The autonomous system for long-range GNSS-denied flight presented in this paper achieved 1st place in the international Fully Autonomous Flight Challenge.

perception, localization, planning, and control into a reliable onboard system.

We study this problem in the context of the *SPRIN-D Funke Fully Autonomous Flight Challenge*, which provided a stringent test of GNSS-denied autonomy. In response, we developed a UAV system that integrates onboard mapping, planning, and mission execution with a lightweight geodata-based localization method for drift correction. Our method’s key idea is to exploit heightmap gradients as a compact and robust structural signature—simple enough for real-time onboard use, yet distinctive enough to correct drift over kilometer-scale trajectories. The system was deployed during the competition, where it successfully executed kilometer-scale flights in mixed environments and achieved substantial drift reduction compared to raw odometry.

A. Problem definition

The SPRIN-D Challenge required an autonomous UAV with total mass below 25 kg to fly a prescribed 9 km long sequence of waypoints, each waypoint marked by a red flag on a 1 m pole, without relying on any GNSS-based localization. The waypoint positions were provided on a printed map with an uncertainty of approximately 20 m. Throughout the mission, the UAV had to remain below 25 m Above Ground Level (AGL) and autonomously avoid obstacles such as trees, buildings, or water curtains sprayed from a fire truck.

The competition environment combined urban areas, forests, and open fields. Although the area was announced beforehand, reconnaissance flights or custom dense mapping

were explicitly prohibited, precluding the use of pre-recorded maps or ground-truth data. Together, these requirements defined not only a localization problem but a full autonomy challenge, demanding a UAV system capable of reliable long-range navigation and mission execution using prior data of the deployment area, and onboard sensing with constrained computation.

B. Related work

Localization without GNSS is a widely studied problem in robotics, but remains challenging for long-range UAV missions. Odometry methods such as VIO and LIO provide local motion estimates, yet accumulate drift that is usually corrected through loop closures. In real-world missions, e.g. search and rescue, UAVs operate in previously unseen environments where revisits are rare and loop closures cannot be relied upon. We tested state-of-the-art visual Simultaneous Localization And Mapping (SLAM) systems such as RTAB-Map [1] and ORB-SLAM3 [2] in a high-fidelity simulator [3], and found that both performance degraded significantly in a long-range scenario (beyond 1 km), as their memory and compute demands grow with the size of the environment. Moreover, RTAB-Map was unable to maintain quality odometry in faster flight speeds (beyond 2 m s^{-1}), while ORB-SLAM3 suffered from tracking loss in textureless areas. This confirmed that conventional SLAM cannot serve as the basis for our system and motivates approaches that combine odometry with external geodata to maintain bounded error.

Several large-scale autonomy efforts, most notably the DARPA Subterranean Challenge [4], demonstrated advanced GNSS-denied navigation capabilities. SubT is the closest system-level reference, but its setting was fundamentally different: multi-robot teams operating primarily in underground environments, often supported by larger platforms and heterogeneous sensing. In contrast, the SPRIN-D Challenge focused on UAVs to operate autonomously in large outdoor environments at altitudes constrained below 25 m, without any prior mapping and without the support of other robots.

The absence of loop closures during the mission naturally leads to the problem of geolocalization within prior geodata. Existing geolocalization methods can be broadly divided into camera-based, LiDAR-based, and semantic approaches. Camera-based methods often align aerial or UAV imagery with satellite maps. High-altitude matching [5], [6] presents a viable solution in higher altitudes, but at low altitudes (below 25 m) the viewpoint differs drastically, making roofs, facades, and vegetation inconsistent with satellite imagery. A range of works [7]–[12] consider low-altitude or ground-based localization. [7] combine a 3D lidar with camera data and train an end-to-end matching model for localizing a grounded agent with a forward-facing camera. [8], [9] train Siamese networks to match ground and satellite image embeddings. The authors of [10] train a cross-view-matching network for satellite-ground localization and combine the matches with odometry using a particle filter [11]. The method in [12] uses panoramic ground imagery warped to bird’s-eye view. While effective in their respective domains and on certain

datasets, these methods assume ground agents or structured viewpoints and are not directly applicable to small UAVs without onboard GPU support deployed in cluttered outdoor mixed environments.

LiDAR-based methods use structural cues to improve robustness. Examples include learned place recognition fused with odometry [13], tree segmentation for forested areas matched with prior aerial scan of the area [14], or heightmap matching [15]. However, these are typically restricted to small scales or specific environments, and are not easily transferable to urban–forest missions. Semantic approaches aim to mitigate appearance variability by extracting higher-level categories. Roads [16], semantic maps [17], or canopy structures [14] have been exploited as robust cues, often fused with odometry inside a particle filter. These methods offer seasonal robustness, but are limited in scope and domain.

Our work follows the general paradigm of combining a similarity estimation front-end with odometry in a particle filter [11], [18], but adapts it to the unique constraints of the task we address: scalability for multi-kilometer flights, GNSS-denied operation, mixed urban–forest domains, and the requirement for reliable out-of-the-box functionality on test day. Unlike previous methods, we designed a new similarity estimation approach based on tall-object detection, which bridges LiDAR-based and semantic cues. Tall objects provide a stable and distinctive cue visible at low altitudes in both urban and forest environments, making them particularly suitable for the challenge setting. This choice proved to be computationally efficient, robust across diverse large environments, and suitable for real-world deployment without extensive parameter tuning.

C. Contributions

We present a complete UAV system capable of kilometer-scale GNSS-denied flight below 25 m AGL, operating fully onboard without a dedicated GPU acceleration. Moreover, we introduce a novel drift-correction method based on template matching of LiDAR-derived heightmap gradients, fused with odometry in a clustered particle filter. Finally, we validated the system in urban, forest, and open-field environments during the SPRIN-D Challenge, where the organizers fixed the conditions of the challenge which led to an objective evaluation of all teams’ solutions. As the only team, we achieved kilometer-scale flights in adverse conditions with substantial drift reduction relative to raw odometry, and we report practical lessons learned to guide future UAV design in GNSS-denied settings.

II. SYSTEM FOR GNSS-DENIED AUTONOMOUS FLIGHT

The task of GNSS-denied long-range navigation requires integration of perception, localization, planning, and control into a reliable onboard system. The structure of the proposed solution is shown in Figure 2.

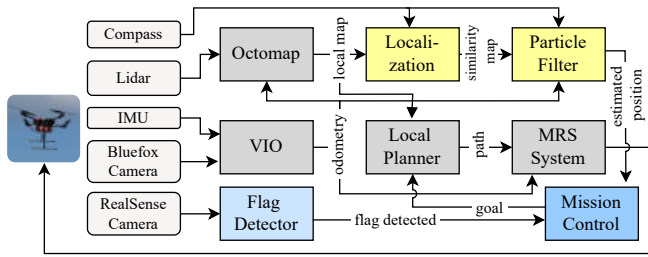


Fig. 2: System overview. Sensor inputs from LiDAR, cameras, IMU, and compass feed modules for visual-inertial odometry, obstacle mapping, and flag detection. A similarity-based localization module fuses odometry with prior geodata in a particle filter. Mission control coordinates planning and control for waypoint-based flight.

A. Hardware

The UAV platform was based on [19], and equipped with a heterogeneous sensor suite (Figure 3) designed to enable robust state estimation, environment perception, and mission execution in GNSS-denied conditions. A Livox Mid-360 LiDAR sensor provided dense point clouds that served as the primary source for obstacle detection, local mapping, and heightmap generation, thereby forming the backbone of both planning and localization. Complementary visual sensing was provided by an Intel RealSense D435 depth camera for short-range obstacle perception, and a global-shutter Bluefox RGB camera paired with an inertial measurement unit (IMU) for visual-inertial odometry (VIO) using the OpenVINS framework [20]. To ensure the integrity of inertial measurements, the Inertial Measurement Unit (IMU) ICM42688 and VIO camera Bluefox2 are mounted on a custom battery case that was mechanically decoupled from the airframe by 3D-printed silent blocks, effectively attenuating high-frequency vibrations induced by the propulsion system. An onboard magnetometer was employed to provide absolute heading measurements for global frame alignment, although its reliability was observed to degrade in magnetically disturbed environments, such as steel-reinforced concrete runways. All computation was performed onboard by an Intel NUC i7 16 GB RAM computer, chosen for its balance of performance and weight, and critically without reliance on dedicated GPU acceleration. This hardware configuration enabled real-time execution of mapping, planning, localization, and mission control during fully autonomous flights.



Fig. 3: Overview of the UAV platform and integrated sensor suite. The multirotor airframe carries a Livox Mid-360 LiDAR for mapping and localization, an Intel RealSense D435 depth camera for short-range obstacle perception, a Bluefox RGB camera and IMU for VIO, and an onboard Intel NUC computer for real-time processing.

B. Visual Inertial Odometry

Accurate state estimation is essential for feedback control and for providing a prior to the localization module. For this purpose, we employ VIO, which offers a good balance between computational efficiency and robustness, making it well suited for real-time UAV operation in diverse outdoor environments. We have opted to utilize monocular VIO provided by OpenVINS [20]. We found that the reliability and practical usability of visual-inertial odometry critically depend on isolating the IMU from high-frequency vibrations generated by motors and propellers, which would otherwise corrupt the inertial measurements and make the odometry unusable. Moreover, it is practical to mount the camera and IMU as close to each other as possible. To address this, both the VIO camera and IMU are mounted on the battery case, mechanically decoupled from the drone body using additively manufactured silent blocks, as shown in Figure 4. The inherent mass of the battery further contributes to the damping effect, thereby reducing the transmission of vibrations.

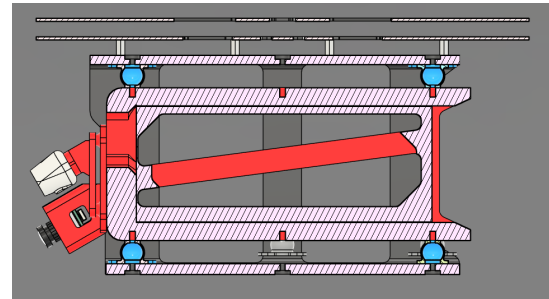


Fig. 4: Custom 3D-printed mount with integrated sensors, mechanically decoupled by silent blocks (blue) to dampen vibrations and improve VIO robustness.

C. Mapping, planning and feedback control

Safe low-altitude flight in complex environments requires the ability to continuously perceive surrounding obstacles and to compute collision-free trajectories in real time. To this end, point clouds from the Livox Mid-360 LiDAR are incrementally integrated into a local occupancy map using the OctoMap framework [21], [22]. The map is centered on the UAV body frame \mathcal{B} , spans an area of 60×60 m, and is updated at 10 Hz, thereby providing a compact yet sufficiently detailed representation of the immediate surroundings. Collision-free paths are found using the A* search algorithm on the euclidean signed distance field constructed from the local occupancy map. The selected path is subsequently transformed into a dynamically feasible trajectory using a polynomial trajectory generation module [23]. The trajectory is then tracked by the underlying model predictive reference tracking and control pipeline [24]. In this way, the mapping and planning subsystem forms the essential link between perception and control, enabling safe navigation in cluttered GNSS-denied environments. Further details of our mapping, planning and feedback control pipeline can be found in [21].

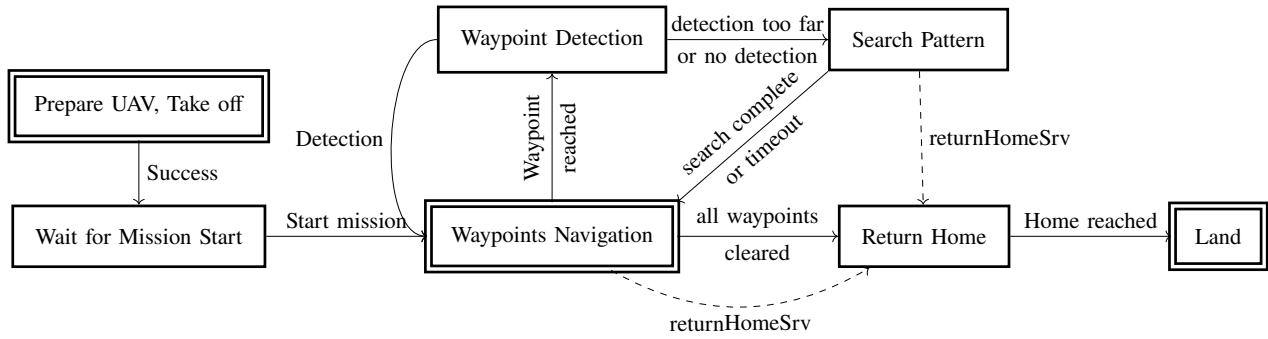


Fig. 5: The mission control state machine

D. Mission control

The execution of the mission is governed by a finite-state machine shown in Figure 5. The state machine starts by the preparation of the UAV and the take-off procedure. Once the mission starts the system enters the navigation state and the UAV starts to fly to the expected position of the first waypoint. Once the UAV enters a radius of 15 m around the expected waypoint position, it switches to the waypoint detection state. In the detection state the waypoint detection module is activated and images are being processed to detect the flags. If the flag is detected, the system switches to the overflight state, where the UAV flies over the waypoint to record the observation. If the flag is not detected at the estimated location, the UAV executes a square search pattern centered on the waypoint. Detection at any time interrupts the search and triggers the overflight state; Failure to detect after the search concludes results in a transition to the next waypoint. After reaching the last waypoint, the UAV returns to the take-off position and lands.

E. Digital twin driven development

The robotic development process requires extensive testing and simulation of the individual components, as well as the whole system. For that purpose, we have created a digital twin of the environment in the FlightForge simulator [3] as similar as possible to the deployment area. The environment was created on the basis of multiple publicly available geodata [25]. The terrain was modeled based on Digital elevation model (DEM), the vegetation (bushes, forest) was first identified in satellite images and then these 2D masks were used for the procedural generation of the corresponding vegetation. The same strategy was used for paved surfaces. The buildings were modeled thanks to the building model (LoD2) available in the area from [25]. The environment was extensively used for the development of the whole system, mainly the localization module as it is most strongly influenced by the environment. We have gone through multiple iterations of the localization module, testing different approaches and parameters in the simulator before deploying it in the real world. In addition to the localization, the simulated environment (Figure 6) was used to generate training data for the waypoint detector, as described in the next paragraph.

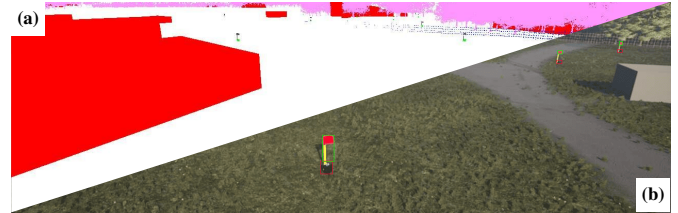


Fig. 6: Images from the digital twin environment: (a) segmentation mask, (b) RGB camera image with identified waypoints.

F. Waypoint detector

The waypoint detection module was responsible for detecting the combination of a black box and a red flag on a yellow pole, as shown in Figure 6. The module was based on the YOLOv8 architecture [26]. First, a larger model YOLOv8m was trained on synthetic datasets generated in the FlightForge simulator [3], where annotations are obtained automatically, as shown in Figure 6. The synthetic dataset contained all the variations of the weather conditions, improving the generalization of the model. After the initial training on synthetic data, the model was already able to detect the waypoints in real-world images on our own replica of the waypoint. The pretrained model was then used to assist with annotation of real-world recordings containing the exact replicas of the waypoints, however the inference time was approximately 300 ms. Finally, to reduce CPU load, a lightweight YOLOv8n model was trained on both synthetic and real data. This smaller model was selected to meet onboard computational constraints while still providing reliable real-time detection with inference times of approximately 100 ms using CPU only.

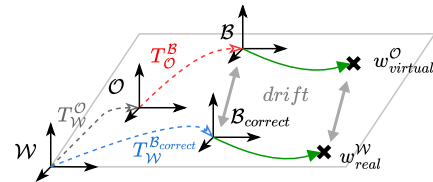


Fig. 7: Overview of the reference frames used in the system, where \mathcal{W} is the georeferenced world frame, \mathcal{O} is the odometry frame initialized on the known take-off position (i. e. known $T_{\mathcal{W}}^{\mathcal{O}}$), \mathcal{B} is the body frame used by the feedback control and $\mathcal{B}_{correct}$ represents the corrected body frame (i.e. true position) in the world frame. The transform $T_{\mathcal{W}}^{\mathcal{B}_{correct}}$ is published by the localization system described in Section III. Given the position of the next waypoint $w_{real}^{\mathcal{W}}$, the virtual goal $w_{virtual}^{\mathcal{O}}$ is created to guide the UAV to the correct position despite the odometry drift accumulated in $T_{\mathcal{O}}^{\mathcal{B}}$.

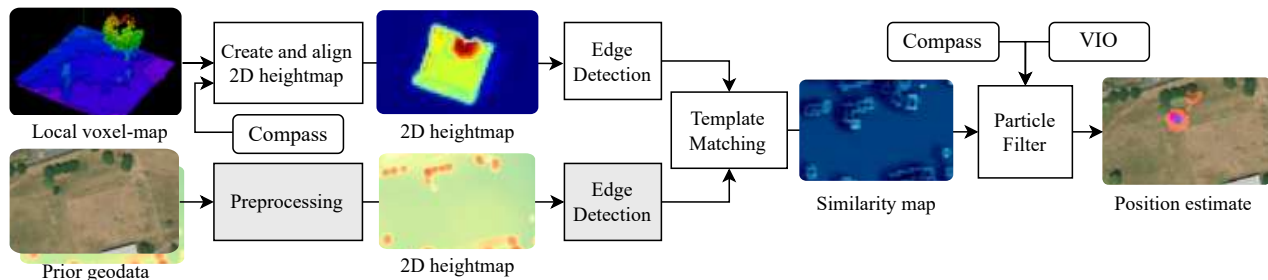


Fig. 8: LiDAR-derived local heightmaps are registered to prior geodata heightmaps through gradient-based template matching. The resulting similarity map is subsequently integrated with odometry and compass measurements within a clustered particle filter to produce drift-corrected position estimates. Precomputed steps are denoted by grey boxes, whereas the remaining components are executed onboard.

III. ONBOARD LONG-RANGE GNSS-DENIED LOCALIZATION

This section presents the onboard localization pipeline for long-range GNSS-denied flight. A local heightmap, derived from the local occupancy map, is matched against preprocessed geodata, and the resulting similarity is fused with odometry in a clustered particle filter to provide robust position estimates. The pipeline described in this section is shown in Figure 8.

A. Heightmap Pre-Processing

The prior data for the localization pipeline consist of a geo-referenced, north-aligned heightmap of the large-scale environment. For our experiments, we used a digital elevation model (DEM) derived from open-source point clouds provided by [25]. The point clouds were processed into a DEM by estimating the ground surface and computing the relative height above terrain of all points using the LAsTools software package [27]. Where high-accuracy DEMs are not available, obstacle heights above the terrain can alternatively be estimated from aerial RGB imagery using recent depth estimation models [28], [29], as illustrated in Figure 8 and applied in our experiments (Flight ID 60 and 61), where accurate remote sensing geodata were not available. To ensure comparability, we apply the same preprocessing steps (discussed in the next section) to the prior DEM and to the local heightmaps generated onboard from LiDAR data. The heightmaps are constructed by calculating the max heights of the given pointcloud (from DEM or from the local map occupied cells) in 1 m wide bins. This resolution allowed realtime operation at the multi-kilometer scale required by the competition while being precise enough for navigation.

The local heightmap itself is constructed from the online occupancy map. We assume that odometry drift remains bounded within the spatial extent of the local map, such that the resulting heightmap is not significantly distorted. In our experiments, the local map was configured as a square region with side lengths between 30 m and 60 m, depending on available computational resources. Finally, onboard compass measurements are used to align the heightmap with the north direction, ensuring consistency with the prior DEM, since the odometry frame O may be arbitrarily rotated relative to the Earth.

B. Heightmap Gradient Matching

Heightmaps can sometimes be matched using absolute elevation values, as demonstrated in prior work [15]. In our case, however, absolute height (e.g., Above Mean Sea Level (AMSL)) is not reliably available on the UAV, as barometric measurements are prone to high noise. Furthermore, the UAV might not see the ground at all times or the ground might be sloped, making it impossible to obtain absolute height using a ground plane model as in [15]. To address this, we perform matching based on *gradients* of the heightmaps rather than absolute heights. This eliminates the problem of vertical offsets between maps encountered in outdoor scenarios.

To further improve robustness, only gradients with an absolute value greater than 5 m were considered during the SPRIN-D challenge runs. This filtering emphasizes tall, stable structures such as buildings and trees, while discarding small or transient objects (e.g., fences, vehicles, or bushes). The resulting binary edge maps mark strong gradients with ones, which are then compared to binary edge maps obtained from prior geodata using template matching. For the template matching function, we use the non-normalized (since the maps are normalized) correlation coefficient, defined as

$$R(x, y) = \sum_{x', y'} (T(x', y') - \bar{T}) \cdot (I(x + x', y + y') - \bar{I}_{x, y}), \quad (1)$$

where T denotes the local binarized heightmap, I the prior map, \bar{T} the mean of the local map, and $\bar{I}_{x, y}$ the mean of the matched patch at (x, y) . This metric was empirically found to outperform alternative functions (e.g., squared difference or normalized correlation), particularly when parts of the local heightmap were incomplete, as during straight-line flight or immediately after takeoff. Finally, because the heightmaps are computed at relatively coarse resolution, a Gaussian blur is applied to the resulting similarity map to reduce discretization artifacts.

C. Particle filter

Odometry and similarity maps are fused in a particle filter to provide a unified probability estimate of the UAV's position. While orientation is directly obtained from the compass, the particle filter maintains multiple hypotheses of the translational state. To enable real-time operation, resampling is triggered only after the UAV has traveled 10 m according to odometry. Between resampling steps, particles

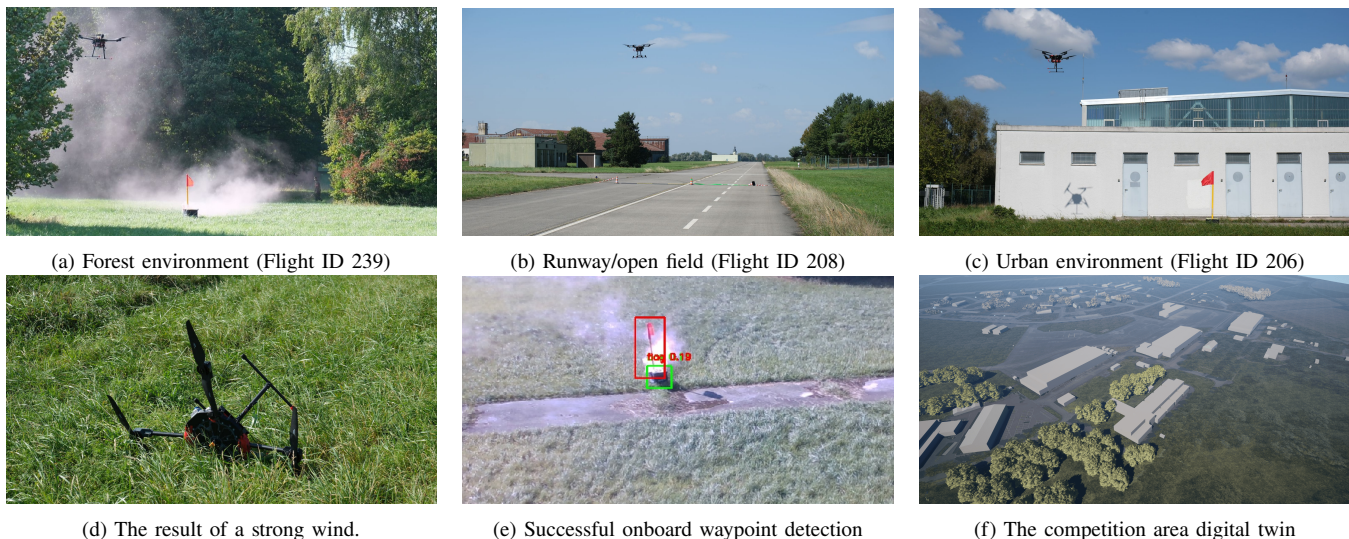


Fig. 9: Selected scenes from the system deployment, showing the three different competition environments, challenges we faced during the competition (strong wind, artificial smoke) and the simulated world used for the system validation and development.

are propagated by translating them according to the odometry estimate and aligning them with the compass heading.

During the particle filter update, each particle is assigned a weight based on the normalized value of the similarity map at its projected location. These weights are then used as probabilities during resampling. When a particle is resampled, its new position is perturbed by Gaussian noise drawn from an empirically estimated odometry covariance, simulating the uncertainty propagation.

Most prior works assume that particles form a single cluster and compute the position estimate as the mean of all particles. In practice, however, we frequently observed formation of multiple clusters caused by perceptual aliasing and odometry noise (see Figure 8). To address this, we apply K-means clustering to the particle set and select the centroid of the largest cluster as the final position estimate. The final global position estimate is used to calculate the direction towards the next waypoint. To navigate towards the waypoint, we periodically set the goal position of the local planner ($w_{virtual}^O$ in Figure 7) to be at a fixed distance from the UAV in the calculated direction to the next waypoint.

IV. EXPERIMENTS

The proposed robotic system was evaluated during the SPRIN-D Fully Autonomous Flight Challenge on the Erding airbase (Bavaria, Germany). The competition area covered a heterogeneous environment including (i) urban areas with buildings, fences, and individual trees, (ii) open fields, and (iii) semi-open forest sections. Each team was given two two-hour flight slots to validate their solutions under supervision of the jury. On the final day of the competition, a 9 km course was defined by a sequence of waypoints distributed across all three environments (printed on a paper map distributed 30 minutes prior the mission). Each team had 2 hours to demonstrate the capability of their system and visit as many waypoints as possible.

A. Evaluation Methodology

Since no GNSS-based localization was permitted, accurate ground truth of the UAV position was not directly available in the challenge. To estimate the real trajectory of the UAV, we manually estimated the approximate ground truth trajectory of the UAV from the VIO data corrected by the camera/lidar footage from distinctive places and objects (where the real position of the UAV is clearly identifiable with precision of 0–5 m). We also present data from two test flights prior the competition, where GNSS localization was available. However, it must be noted that the system should not be evaluated only on the basis of the Root Mean Square Error (RMSE) that our localization achieved, but rather as a system that was able to perform multiple kilometer-scale flights in challenging conditions while being able to successfully find the target waypoints.

B. Autonomous GNSS-denied flights

The experiments performed before and during the competition are shown in Table I. Test flights 60 and 61 were conducted outside the competition area and thus have accurate GNSS ground truth. To test the recoverability and drift correction of our localization method, the localization module was intentionally initialized approx. 32 m away from the true position in experiment number 61 shown in Figure 10. In this experiment, our method managed to correct the position estimate when the UAV observed the trees and in the end reduced its error to approx. 4 m, as shown in Figure 11.

During the actual competition, our system was able to autonomously perform multiple kilometer-scale flights with overall localization RMSE below 11 m while the compass-aligned odometry had RMSE of up to 53 m. Selected flights are shown in 12. Flights exceeding 1 km were consistently terminated due to battery constraints, rather than localization drift, indicating hardware limitations as the primary bottleneck. The main reason was typically drained battery or a hardware issue, such as faulty WiFi card that restarted the

Flight ID	Length [m]	Time [s]	RMSE _{odom} [m]	RMSE _{method} [m]	Waypoints detected	Area	Termination
60*	830	606	8	5	-	forest/open field	test finished
61*	495	343	36	13	-	forest/open field	test finished
205 [†]	939	622	31	10	1/1	urban	test finished
206 [†]	1023	621	27	6	2/2	urban	SW issue
208 [†]	872	500	14	11	1/1	urban/open field	low battery
232 [†]	1256	1021	53	7	1/2	forest	failsafe - HW issue
239 [†]	1371	977	8	9	4/4	forest	low battery
242 [†]	939	622	9	11	2/3	urban/open field	low battery

TABLE I: Summary of the real-world experiments. The a) ID of the flight, b) flight time, c) flight length, d) RMSE of the odometry and our proposed method compared to GNSS ground truth(*/approximate ground truth described in Section IV-A ([†]) (both rounded to meters), e) number of waypoints with flag successfully detected, f) type of the environment and g) termination of the mission are reported.

onboard computer during the flight shown in Figure 12c. Despite the lack of accurate ground truth, it is possible to conclude that the proposed drift correction method significantly improved the localization accuracy over the kilometer-scale distances. Our method showed better performance in urban environment, where there were more features and distinguishable objects that helped with the drift correction. In open field areas, the system mostly relied on the odometry. The odometry from VIO was fused with magnetometer measurements that helped to keep the odometry frame correctly oriented in the world coordinate frame. In some areas of the competition, the compass provided incorrect data in the form of a slowly changing bias (as we can see right after the start in Figure 12c) with an error of up to 30 deg. However, the proposed system was able to correct the magnetometer drift (when enough distinctive features were present).

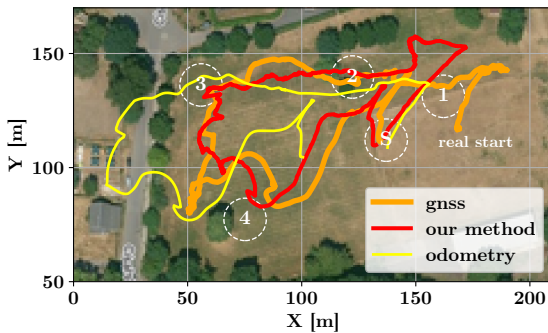


Fig. 10: Flight ID 61. The starting (and landing) position is marked with S and the waypoints are numbered in the respective order. During this experiment, the localization method was initialized (waypoint S) 32 m off the starting position (see "real start" in the map). Despite the initial error, the localization method successfully relocalized by the end of the test flight.

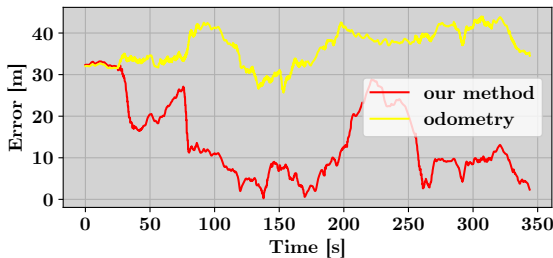


Fig. 11: Flight ID 61. The localization error error of the proposed localization method and odometry compared to GNSS ground truth.

C. Results of the competition

A total of nine teams from European countries participated in the challenge. The most significant difficulty proved to be establishing reliable odometry onboard the UAV, which is a prerequisite for stable autonomous flight. In addition, achieving accurate GNSS-denied geolocation at low altitudes proved to be inherently difficult. Some teams used RGB satellite image-based matching, but this has proved to be highly unreliable at such low altitudes. As a result, only us and 2 other teams managed to realize autonomous flights under these demanding conditions for more than 100 m. The two other teams accumulated critical drift and were not able to reach more than 2 waypoints. Our team succeeded in this task, ultimately achieving the best overall performance and being awarded first place in the competition.

D. Lessons learned

The challenge provided valuable insights into real-world GNSS-denied UAV deployment. Mechanical decoupling of the IMU and VIO camera from the drone frame proved essential: when the dampening mechanism failed, VIO drift increased sharply, showing the strong impact of parasitic vibrations. Although the compass appears a natural choice for global alignment, we confirmed that magnetometers are unreliable near buildings and reinforced concrete; however, our method was able to correct this drift when sufficient environmental features were available. More generally, this highlights the broader lesson in robotics that no single sensor can be fully relied upon, making sensor fusion essential for robust autonomy. A general lesson is that overall system performance is constrained by its weakest component. In our case, the limited size of the local map restricted flight speed, which in turn capped the effective mission range despite accurate localization. Equally important in time-critical scenarios is rapid redeployment: the system must support fast, structured diagnostics of its components to localize the weak points of the system within minutes.

V. CONCLUSIONS

We have presented and successfully demonstrated a fully onboard system for reliable long-range UAV navigation in GNSS-denied environments. Our approach corrects odometry drift by aligning locally generated LiDAR heightmaps with prior geodata using a gradient-matching technique within a clustered particle filter, proven robust across varied

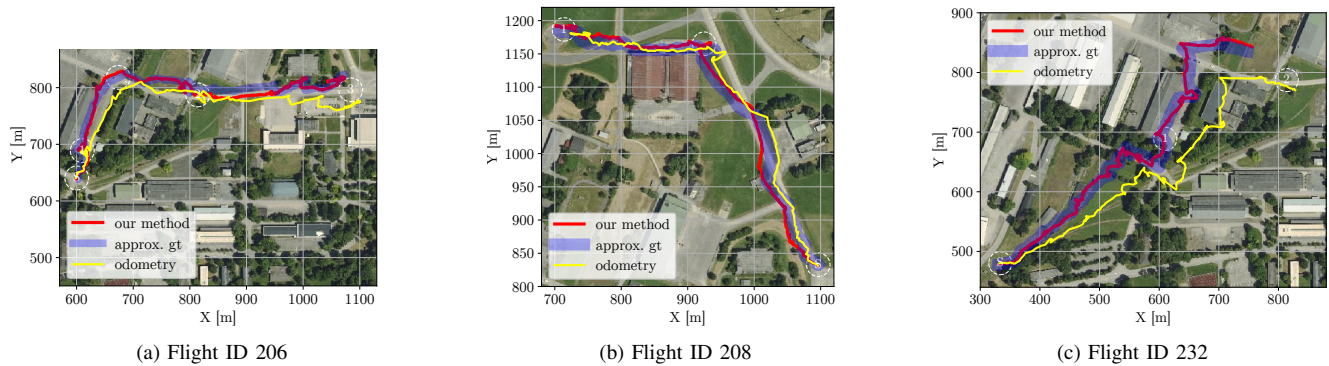


Fig. 12: Comparison of flight trajectories during the SPRIN-D competition. The results highlight the reduced drift of our method compared to raw odometry across different environments.

terrains including urban, forest, and open fields. The system’s performance in the SPRIN-D Challenge, where it achieved first place with RMSE below 11 m over kilometer-scale flights, validates its capability for real-world autonomous operation on CPU-only hardware. Crucially, our results demonstrate that for long-range missions, the ability to recover from periods of high uncertainty and re-localize is more critical than maintaining consistently low instantaneous RMSE. While challenges in endurance and operation in low-feature environments remain, this work provides a foundational blueprint for developing resilient GNSS-denied autonomy systems required for practical deployment.

ACKNOWLEDGMENT

This work was funded by the German Federal Agency for Disruptive Innovation (SPRIN-D), by the Czech Science Foundation (GAČR) under research project no. 25-17779M, by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590) and by CTU grant no SGS23/177/OHK3/3T/13.

REFERENCES

- [1] M. Labbé *et al.*, “Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446.
- [2] C. Campos *et al.*, “ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [3] D. Čapek *et al.*, “Flightforge: Advancing uav research with procedural generation of high-fidelity simulation and integrated autonomy,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 1611–1617.
- [4] K. Ebadí *et al.*, “Present and future of slam in extreme environments: The darpa sub challenge,” *IEEE Transactions on Robotics*, vol. 40, pp. 936–959, 2024.
- [5] N. Samano *et al.*, “Global aerial localisation using image and map embeddings,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5788–5794.
- [6] H. Goforth *et al.*, “Gps-denied uav localization using pre-existing satellite imagery,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2974–2980.
- [7] F. Fervers *et al.*, “Continuous self-localization on aerial images using visual and lidar sensors,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 7028–7035.
- [8] L. M. Downes *et al.*, “City-wide street-to-satellite image geolocalization of a mobile ground agent,” *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11 102–11 108, 2022.
- [9] —, “Wide-area geolocalization with a limited field of view camera,” *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10 594–10 600, 2022.

- [10] S. Hu *et al.*, “Image-based geo-localization using satellite imagery,” *International Journal of Computer Vision*, vol. 128, pp. 1205–1219, 2019.
- [11] F. Dellaert *et al.*, “Monte carlo localization for mobile robots,” *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, pp. 1322–1328 vol.2.
- [12] A. Viswanathan *et al.*, “Vision based robot localization by ground to satellite matching in gps-denied situations,” *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [13] L. Sun *et al.*, “Localising faster: Efficient and precise lidar-based robot localisation in large-scale environments,” *2020 IEEE International Conference on Robotics and Automation (ICRA)*.
- [14] L. C. de Lima *et al.*, “Air-ground collaborative localisation in forests using lidar canopy maps,” *IEEE Robotics and Automation Letters*, vol. 8, pp. 1818–1825, 2023.
- [15] R. Kaslin *et al.*, “Collaborative localization of aerial and ground robots through elevation maps,” *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 284–290, 2016.
- [16] A. Viswanathan *et al.*, “Vision-based robot localization across seasons and in remote locations,” *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4815–4821, 2016.
- [17] G. A. Christie *et al.*, “Semantics for ugv registration in gps-denied environments,” *ArXiv*, vol. abs/1609.04794, 2016.
- [18] S. Thrun *et al.*, “Robust monte carlo localization for mobile robots,” *Artif. Intell.*, vol. 128, pp. 99–141, 2001.
- [19] D. Hert *et al.*, “MRS Drone: A Modular Platform for Real-World Deployment of Aerial Multi-Robot Systems,” *Journal of Intelligent & Robotic Systems*, vol. 108, pp. 1–34, July 2023.
- [20] P. Geneva *et al.*, “OpenVINS: A research platform for visual-inertial estimation,” in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020.
- [21] M. Petrlik *et al.*, “UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT,” *Field Robotics*, vol. 3, pp. 1–68, January 2023.
- [22] A. Hornung *et al.*, “Octomap: An efficient probabilistic 3d mapping framework based on octrees,” *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [23] C. Richter *et al.*, “Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments,” in *Robotics Research*. Springer, 2016, pp. 649–666.
- [24] T. Baca *et al.*, “The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 05 2021.
- [25] Bayerische Vermessungsverwaltung, “Geodaten bayern opendata,” <https://geodaten.bayern.de/opengeodata/>, accessed: 2025-09-12.
- [26] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics yolov8,” 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [27] M. Isenburg, “Lastools: Efficient lidar processing software,” <http://rapidlasso.com/LAStools>, 2022, accessed: 2025-09-14.
- [28] L. Yang *et al.*, “Depth anything v2,” *arXiv:2406.09414*, 2024.
- [29] J. Tolan *et al.*, “Very high resolution canopy height maps from rgb imagery using self-supervised vision transformer and convolutional decoder trained on aerial lidar,” *Remote Sensing of Environment*, vol. 300, p. 113888, 2024.