A Vision-Based Navigation System for GPS-Denied Unmanned Aerial Vehicle Localization

Real-Time Camera-to-Satellite Image Matching with Classical and Learned Feature Correspondence and Ground-Station Integration

Yakup Tekeli

2026

Abstract

Global Navigation Satellite Systems (GNSS) such as GPS are the primary source of absolute position for unmanned aerial vehicles (UAVs), yet they are fragile: the signal can be jammed, spoofed, or lost in urban canyons and indoor environments. This report presents the Visual Navigation System (VNS), a software framework that estimates a UAV's absolute geographic position using its downward-looking camera alone, without any GPS input. Each camera frame is matched against a pre-acquired satellite reference map through a coarse-to-fine pipeline: a city-wide search acquires an initial lock, after which a fast per-frame tracker maintains it. Two complementary matching engines are provided. A classical engine based on the Scale-Invariant Feature Transform (SIFT), fast approximate nearest-neighbour search, and random sample consensus (RANSAC) homography estimation handles satellite-to-satellite imagery, while a learned, detector-free engine (LoFTR) handles the substantially harder cross-domain problem of matching real drone video against satellite maps. The matched map pixel is converted to World Geodetic System 1984 (WGS-84) coordinates through a Web Mercator model, temporally smoothed by a constant-velocity Kalman filter, and streamed over the MAVLink protocol to the QGroundControl ground-control station, where the aircraft is displayed moving live on the map. The system is designed to be honest: the true position is never revealed to the ground station before a genuine visual lock is achieved. On a synthetic flight over a wide-area Ankara satellite map the system achieves an essentially perfect lock rate with a few metres of error; on independent cross-source imagery it remains largely reliable; and on real drone video the learned matcher raises the lock rate from roughly two per cent (classical) to more than ninety per cent over the mapped segment. The report describes the architecture, methodology, implementation, experiments, and the principal engineering findings, and outlines a path toward closed-loop GPS-denied autonomy.

Keywords: GPS-denied navigation; visual localization; image-to-map matching; SIFT; RANSAC; LoFTR; Kalman filter; geo-referencing; MAVLink; QGroundControl.

Table of Contents

1. Introduction

1.1 Motivation

Autonomous and semi-autonomous UAVs rely almost universally on GNSS receivers to obtain their absolute position. Although GNSS is accurate and inexpensive, it is also a single point of failure. Civilian and military operators alike face intentional interference (jamming and spoofing) and unintentional degradation in dense urban environments, forests, canyons, and indoor or subterranean spaces where the satellite signal is attenuated or reflected. When the position fix is lost, an aircraft that depends solely on GNSS must either return to a known state using inertial dead reckoning, which drifts unbounded over time, or abort the mission. A navigation capability that does not depend on any external radio signal is therefore of considerable practical value.

Vision offers such a capability. A downward-looking camera observes the same ground that is recorded in widely available satellite imagery, so in principle the aircraft can determine where it is by recognizing what it sees. This report describes a complete, working system that realizes this idea end to end, from raw camera frame to a live position on a professional ground-control station map, using no GPS input whatsoever.

1.2 Problem Statement

Given a stream of nadir (downward-looking) camera frames and a pre-acquired georeferenced satellite map of the operating area, estimate the aircraft's absolute geographic position (latitude and longitude) for each frame in real time, without using GPS. The estimate must be produced from a cold start, that is, without any prior on the aircraft's location within the map, and it must degrade gracefully and honestly when matching temporarily fails.

1.3 Contributions

The principal contributions of this work are the following:

- An end-to-end GPS-denied localization pipeline that converts a single camera frame into an absolute WGS-84 position and streams it to a standard ground-control station for live display.
- A dual matching architecture that pairs a classical SIFT/FLANN/RANSAC engine for satellite-to-satellite imagery with a learned, detector-free LoFTR engine for cross-domain matching of real drone video against satellite maps, exposed through a single interchangeable interface.
- An honest evaluation methodology, including a cross-source protocol in which the camera and reference imagery come from different providers, and a real-video experiment that exposes the limits of classical matching and the benefit of learned correspondence.
- A leak-free design principle whereby the true position is never transmitted to the ground station before a genuine visual lock, ensuring that the displayed track reflects only what the vision system actually inferred.
- A set of reproducible engineering findings concerning ground-station telemetry compatibility, temporal filtering, input scale matching for learned features, and display-window behaviour on the target platform.

2. Background and Related Work

2.1 GPS-Denied Navigation

Approaches to navigation without GNSS fall into several families. Inertial navigation integrates acceleration and angular rate from an inertial measurement unit (IMU); it is fully self-contained but its error grows without bound because integration accumulates sensor bias and noise. Visual odometry (VO) and visual-inertial odometry (VIO) estimate incremental motion from consecutive images, optionally fused with an IMU; they are locally accurate but, being relative, also drift over long trajectories and provide no absolute position. Simultaneous localization and mapping (SLAM) builds and reuses a map to bound drift through loop closure, but the map it maintains is in its own arbitrary frame rather than in absolute earth coordinates. The present work belongs to the family of absolute visual localization, sometimes called map-based localization or geo-registration, in which live imagery is matched to a pre-existing georeferenced reference so that the resulting position is absolute and drift-free by construction.

2.2 Local Feature Matching

Classical image matching detects repeatable keypoints, describes their local appearance with invariant descriptors, and establishes correspondences between images. The Scale-Invariant Feature Transform (Lowe, 2004) remains a strong baseline because its descriptors are invariant to scale and rotation and robust to moderate illumination and viewpoint change. Candidate matches are found efficiently with approximate nearest-neighbour search (Muja and Lowe, 2009) and filtered with Lowe's ratio test. A geometric model, here a planar homography appropriate to near-nadir aerial views of largely flat terrain, is then fitted robustly with Random Sample Consensus (Fischler and Bolles, 1981), which simultaneously rejects outliers and yields the transformation relating the two images. This classical stack is fast and interpretable, and it performs well when the two images are visually similar.

2.3 Learned, Detector-Free Matching

Classical descriptors degrade sharply under large appearance change, for example when a real camera frame captured under one season, time of day, and sensor is matched against satellite imagery captured under entirely different conditions. Learned matchers address this by replacing hand-crafted detectors and descriptors with neural networks trained end to end. LoFTR (Sun et al., 2021) is a detector-free method that uses self- and cross-attention over dense feature grids to establish correspondences directly, without first committing to sparse keypoints. It is particularly effective in low-texture and repetitive regions where classical detectors fail, at the cost of higher computation. This system uses the outdoor-trained LoFTR weights through the kornia library for the real-video scenario.

2.4 Geo-Referencing and Ground Control

Web mapping services publish imagery in the Web Mercator projection on a quad-tree of tiles indexed by zoom level. This projection admits a closed-form conversion between pixel coordinates at a given zoom and geographic latitude and longitude, which the system uses to turn a matched map pixel into an absolute coordinate. On the ground-station side, MAVLink is the de facto lightweight messaging protocol between aircraft and ground-control stations, and QGroundControl is a widely used open-source station that renders vehicle position and telemetry on a map. Streaming the visual position estimate as MAVLink telemetry allows the system to be visualized with professional tooling rather than a bespoke viewer.

3. System Overview

The system is organized as a layered application. A text-based user interface presents the operator with a list of geographic regions and starts a flight in the chosen region. An orchestration layer arranges the on-screen cockpit and dispatches the appropriate flight routine. The core library performs the actual localization: it acquires and indexes the reference map, matches each frame, filters the resulting position, converts it to geographic coordinates, and publishes it. Two output surfaces are provided in parallel: a local matching dashboard that visualizes the correspondence and telemetry, and a MAVLink stream to QGroundControl that displays the aircraft on a professional map.

The end-to-end data flow for each processed frame is: camera frame, optional nadir rectification, coarse search or fine tracking to obtain a map pixel, Kalman smoothing, pixel-to-coordinate conversion, and finally telemetry publication and dashboard rendering. The single conceptual invariant that ties the components together is the mapping from an estimated map pixel to an absolute latitude and longitude, which is then transmitted to the ground station.

4. Methodology

4.1 Reference Map Acquisition and Geo-Anchoring

Each region is defined by a record that stores the path to its reference map together with its geographic anchor: the centre latitude and longitude, the Web Mercator zoom level, and a tile grid radius. These values must exactly match those used when the map was fetched, because the coordinate conversion assumes that the image was produced at the stated centre and zoom. Storing the anchor alongside the map in a single registry ensures that the pixel-to-coordinate conversion is consistent with the imagery. Reference maps are downloaded once from a public imagery service and cached locally, so the system operates without network access thereafter.

4.2 Coarse Localization

From a cold start the aircraft's location within the map is unknown, so the system performs a city-wide search. The reference map is divided into overlapping sectors, each indexed with SIFT features in advance. The incoming frame is searched against every sector over a small scale pyramid to tolerate altitude uncertainty, and the candidate that yields the strongest RANSAC homography, measured by inlier count, is accepted as the initial lock. Because coarse search is comparatively expensive, it is used only until a lock is established.

4.3 Fine Tracking

Once a lock exists, the system switches to fine tracking, in which each frame is matched directly against the pre-indexed reference map. Candidate correspondences from the SIFT descriptors are filtered by the ratio test and passed to RANSAC homography estimation with a geometric sanity check on the resulting quadrilateral. To keep the per-frame cost independent of map size, the camera view is synthesized or cropped from only a small window around the current estimate rather than warping the entire map, which keeps processing within real-time budget even for maps of eleven megapixels. The number of SIFT features is scaled with the map area so that feature density remains sufficient on large maps.

4.4 Learned Matching for Real Video

Matching a real drone video against satellite imagery is a cross-domain problem: the two images differ in date, season, illumination, resolution, and viewing geometry, and the scene may be dominated by repetitive texture such as parkland. Under these conditions the classical engine collapses to a lock rate of only a few per cent. The learned LoFTR engine replaces it for this scenario. The single most important practical requirement is scale matching: the drone frame and the corresponding map region must be presented at approximately the same ground scale, achieved here by resizing the map and the frame to a fixed, compatible pair of resolutions. With scale corrected, the number of reliable correspondences rises by an order of magnitude relative to the classical engine. It was also found that contrast-limited adaptive histogram equalization, a common pre-processing step that helps classical features, actually reduces the LoFTR lock rate, so raw grayscale input is used instead. The learned engine exposes exactly the same interface as the classical one, so the rest of the pipeline is unchanged.

4.5 Temporal Filtering

Per-frame position estimates are smoothed by a constant-velocity Kalman filter whose measurement weight is tied to the lock strength, so that confident matches influence the state more than weak ones. When a lock is temporarily lost, the filter coasts on its velocity estimate, providing an honest dead-reckoning bridge rather than a fabricated position. A subtle but critical detail is that the filter's time step must equal the real elapsed time per processed frame; using the nominal video frame rate instead causes the filter to lag the true motion and introduces position errors of hundreds of metres.

4.6 Coordinate Transformation

The smoothed map pixel is converted to geographic coordinates using the closed-form Web Mercator relationship between pixel position at the reference zoom level and latitude and longitude. Because the same projection and rounding are used both when the map is fetched and when the conversion is applied, the resulting absolute coordinate is consistent with the reference imagery. The conversion is anchored by the region's stored centre coordinate, zoom, and grid radius.

4.7 Telemetry and Ground-Station Integration

The geographic position, together with derived heading, ground speed, and altitude, is published as MAVLink telemetry to QGroundControl over a local UDP connection. The bridge emits a heartbeat, global position, attitude, a heads-up-display summary, a satellite-fix status, system status with healthy sensor flags, and a global origin so that the station can place the vehicle correctly. A decisive compatibility finding is that MAVLink protocol version 2.0 must be used: the station reads a version 1.0 heartbeat for vehicle detection but ignores version 1.0 position telemetry, so with the default protocol the vehicle appears but the map and telemetry remain empty. Before the first lock the bridge reports a no-fix status and sends no position, which enforces the leak-free principle.

4.8 Visualization

A local dashboard renders the camera view and the reference map side by side, overlays the matched correspondences and the localized footprint, and plots position error, lock strength, and the estimated trajectory. The dashboard is displayed in a freely resizable window implemented with a graphical toolkit, because the default image-display backend on the target platform forces the window back to the image size on every update and thus prevents free resizing; the graphical window scales the frame to the current window size while preserving aspect ratio, and falls back to the default backend when the toolkit is unavailable.

5. Implementation

The system is implemented in Python and organized into an application layer, a region registry, and a core library. The application entry point opens the region-selection interface and hands the chosen region to the orchestrator, which configures the cockpit and starts the flight routine. The core modules and their responsibilities are summarized below.

Module	Responsibility
vnav_localizer	City-wide coarse search via overlapping SIFT-indexed sectors.
vnav_matcher	Fine real-time SIFT / FLANN / RANSAC matching against the indexed map.
loftr_matcher	Learned, detector-free LoFTR matching for real video (cross-domain).
vnav_kalman_filter	Constant-velocity position smoothing

	weighted by lock strength.
geo_converter	Pixel to WGS-84 latitude/longitude via the Web Mercator model.
vnav_rectifier	IMU-based rectification of oblique views to nadir.
mavlink_bridge	MAVLink 2.0 telemetry bridge to QGroundControl.
vns_dashboard	Matching overlay, telemetry and error plots.
display	Resizable graphical display window with a fallback backend.
map_fetcher	Reference satellite-map download and preparation.
regions	Region registry: map path, geographic anchor, and flight parameters.

The dependencies are numerical and computer-vision libraries for array processing and image handling; a graphical toolkit for the resizable display and the region-selection interface; a MAVLink library for telemetry; and a deep-learning stack for the learned matcher. The deep-learning stack is imported only on the real-video path, so the classical scenarios run without it. On Apple Silicon the learned matcher runs on the Metal Performance Shaders backend, which is markedly faster than the processor while producing identical results, with an automatic fallback to the processor if a required operation is unsupported.

6. Experimental Setup

6.1 Scenario A: Synthetic Flight over Ankara

A realistic flight is synthesized over a wide-area satellite map of central Ankara spanning approximately twelve kilometres. The trajectory is a smooth, randomly meandering path rather than a fixed circle, with a new route on each run, and it is paced so that the vehicle moves at a realistic cruise speed. Each frame is a rotated crop taken from a small window around the true position and matched against the full map. This scenario exercises the complete pipeline under favourable, same-source conditions.

6.2 Scenario B: Cross-Source Validation

To test whether the system localizes from genuinely independent imagery rather than from the map it was cropped from, the camera imagery is drawn from a different provider than the reference map, while both share the same geographic tiling. This makes the camera an image the system has not indexed, differing in date, colour, and processing, and therefore a fair test of cross-source robustness.

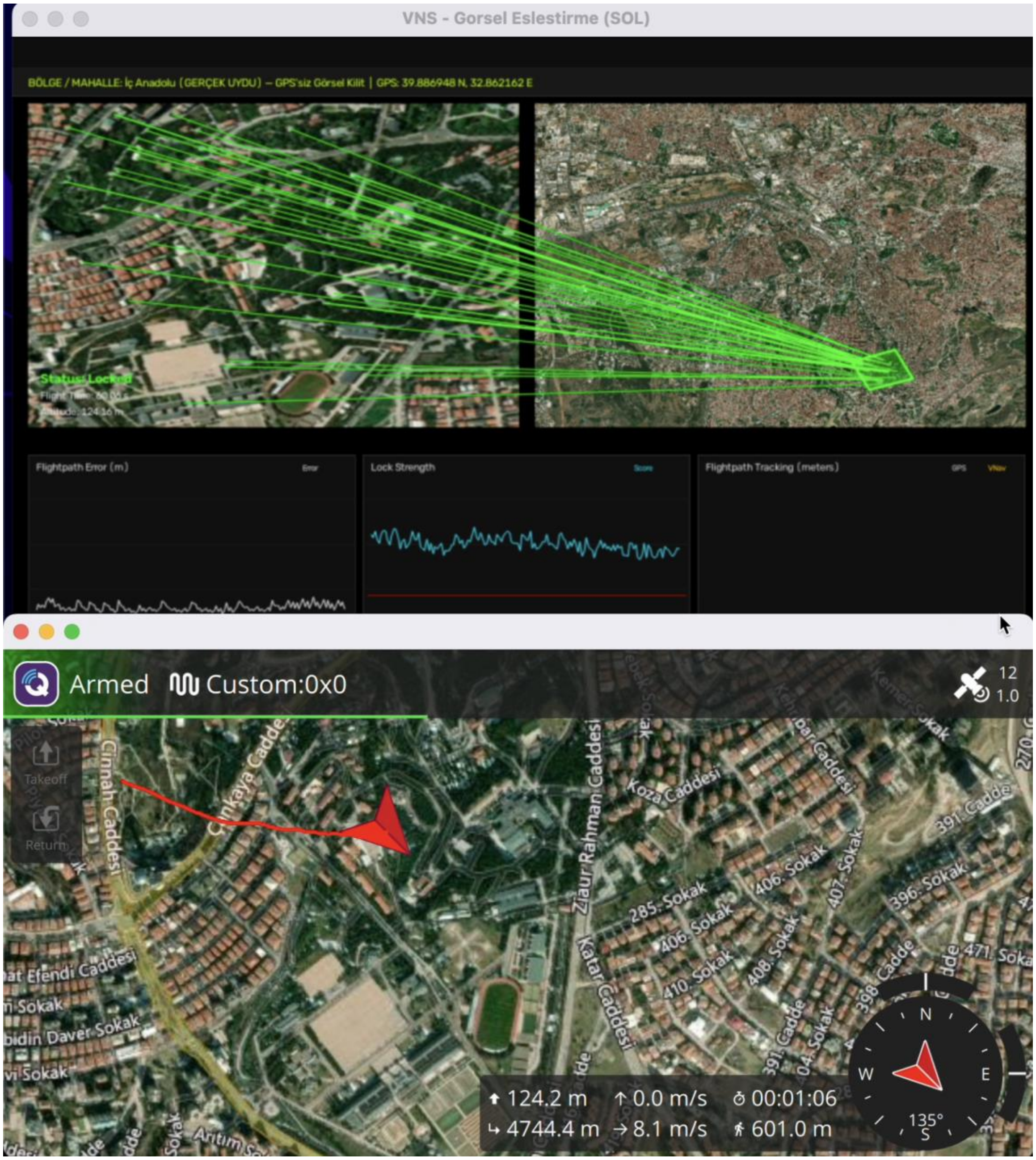
6.3 Scenario C: Real Drone Video over Munich

A real nadir drone video is matched against a satellite map of the Englischer Garten area of Munich. This is the most demanding scenario because it combines every cross-domain difficulty at once. It is used to compare the classical and learned matching engines directly. All experiments were run on an Apple Silicon workstation; the learned matcher used the Metal Performance Shaders backend.

7. Results and Evaluation

7.1 Synthetic Ankara Flight

Under same-source conditions the system achieves an essentially perfect lock rate across the flight, with a mean position error of only a few metres and a maximum of well under ten metres. Because the per-frame cost is decoupled from map size by cropping a small window, real-time pacing is maintained even on the eleven-megapixel wide-area map.



Top: the drone camera view (left) is matched against the satellite reference map (right); the green lines are inlier feature correspondences and the green quadrilateral is the localized footprint on the map. Bottom: QGroundControl shows the drone (red) moving live on the Ankara map, with its position, heading, altitude and speed derived entirely from image-to-map matching — no GPS input.

7.2 Cross-Source Validation

With independent camera imagery the average lock rate is approximately seventy to eighty per cent and the position error, when locked, is on the order of seven metres. The lock rate is markedly route-dependent: trajectories that pass through texture-rich areas match reliably, whereas trajectories over low-texture or strongly changed regions can fail for stretches, during which the Kalman filter bridges the gap. This variance is an honest reflection of the difficulty of cross-source matching and motivates the learned engine for the hardest cases.

7.3 Real Munich Video: Classical versus Learned Matching

On real drone video the classical engine effectively fails, achieving only about two per cent lock across the video because of the cross-domain gap. Replacing it with the learned engine, and critically matching the input scales, transforms the result: the number of geometric inliers rises from a classical maximum of about thirteen to a learned median of roughly fifty and a maximum above one hundred, and the lock rate over the well-mapped segment of the flight rises to approximately ninety-three to ninety-five per cent. On the Metal Performance Shaders backend the learned matcher processes a frame in roughly seven hundred milliseconds, about five times faster than on the processor, with identical accuracy.

7.4 Engineering Findings

Several findings emerged that materially affected correctness and are reusable beyond this project:

- Ground-station telemetry requires MAVLink protocol version 2.0; with version 1.0 the vehicle is detected but no position or telemetry is displayed.
- The Kalman filter time step must equal the real per-frame elapsed time; using the nominal frame rate introduces position errors of hundreds of metres.
- For the learned matcher, matching the ground scale of the drone frame and the map region is decisive, and contrast-limited histogram equalization reduces rather than improves its performance.
- On the target platform the default image-display backend forces the window size on every update, so a graphical toolkit is required for a freely resizable display.

8. Discussion

The results support the central claim that a UAV can determine its absolute position from vision alone when a georeferenced reference is available. The same-source scenario establishes that the geometry, geo-referencing, filtering, and telemetry are correct end to end. The cross-source scenario shows that the system generalizes to genuinely independent imagery, albeit with route-dependent reliability, and the real-video scenario demonstrates that the residual gap to practical operation is primarily one of appearance robustness, which learned matching addresses. The honest, leak-free design is essential to this argument: because the true position is never sent before a genuine lock, the displayed track is a faithful record of what the vision system inferred, not of ground truth.

The engineering findings are as important as the algorithmic ones. Correct telemetry versioning, correct temporal bookkeeping in the filter, correct input scaling for the learned matcher, and a correct display backend each made the difference between a system that appears to work and one that actually does. Recording them explicitly is intended to save others the same diagnosis.

9. Limitations

- The system operates open-loop: the position estimate is injected for visualization and the aircraft is not physically controlled, so true attitude dynamics are not exercised.
- Each region uses a single flat reference map; large areas or long flights require a tiled, pyramidal coarse search, and the real-video map presently covers only a short segment of the flight.
- In the same-source scenario the camera is derived from the reference map, which makes the lock partly self-confirming; the cross-source and real-video scenarios exist precisely to counter this circularity.
- Cross-source reliability is route-dependent and can degrade over low-texture regions.
- Satellite imagery used for cross-source validation is subject to provider terms of service and is excluded from distribution.

10. Future Work

- Closed-loop GPS-denied autonomy in a software-in-the-loop simulator, feeding the visual estimate to the flight-control estimator and reporting error against simulator ground truth.
- A tiled, pyramidal coarse search to cover arbitrarily large areas and full flights at constant per-frame cost.
- Quantitative accuracy evaluation on a ground-truthed dataset to report root-mean-square error rather than lock rate alone.
- Extension of the region registry to additional geographic areas with their own georeferenced maps.
- Acquisition of real drone footage over the primary operating region to replace synthetic and cross-source proxies with a fully realistic evaluation.

11. Conclusion

This report presented a complete vision-based navigation system that localizes a UAV in GPS-denied conditions by matching its camera imagery to a georeferenced satellite map, converts the result to absolute coordinates, and displays it live on a professional ground-control station. By combining a classical matching engine for favourable imagery with a learned engine for hard cross-domain cases, and by adhering to an honest, leak-free design, the system achieves near-perfect localization under same-source conditions, remains largely reliable across independent imagery, and succeeds on real drone video where classical matching fails. The architecture, methodology, and engineering findings reported here provide a foundation for the next step toward closed-loop GPS-denied autonomy.

References

[1] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91-110.

[2] Fischler, M. A., and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.

[3] Muja, M., and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), 331-340.

[4] Sun, J., Shen, Z., Wang, Y., Bao, H., and Zhou, X. (2021). LoFTR: detector-free local feature matching with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8922-8931.

[5] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82(1), 35-45.

[6] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

[7] Riba, E., Mishkin, D., Ponsa, D., Rublee, E., and Bradski, G. (2020). Kornia: an open source differentiable computer vision library for PyTorch. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), 3674-3683.

[8] Fonder, M., and Van Droogenbroeck, M. (2019). Mid-Air: a multi-modal dataset for extremely low altitude drone flights. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

[9] MAVLink contributors. MAVLink micro air vehicle communication protocol. https://mavlink.io.

[10] QGroundControl contributors. QGroundControl ground control station. https://qgroundcontrol.com.

[11] Esri. ArcGIS World Imagery. https://www.arcgis.com.

[12] OpenStreetMap contributors and Nominatim. https://nominatim.openstreetmap.org.

Appendix A. Key Parameters

Parameter	Value / Note
İç Anadolu map coverage	~12 km, Web Mercator zoom 15, 3328 x 3328 px.
İç Anadolu cruise speed	10 m/s (~36 km/h).
Kalman time step	Real per-frame elapsed time (not the nominal video frame rate).
MAVLink connection	UDP 127.0.0.1:14550; protocol version 2.0.
LoFTR input scale	Map ~1024 px, drone frame ~384 px (scale matched).
LoFTR pre-processing	Raw grayscale (no contrast-limited histogram equalization).
LoFTR throughput	~712 ms/frame on Metal Performance Shaders (~5x the processor).
Same-source lock rate	~100% with a few metres of error.
Cross-source lock rate	~70-80% average, route-dependent; ~7 m error when locked.
Real-video lock rate	~93-95% over the mapped segment (learned matcher).