INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

Autonomous Navigation and Safe Landing of an Aerial Vehicle in a GPS-Denied Martian Environment

Khushi D Thakkar[1], Meet Jain[1], Harshid Rawal[1], Moulik Shaparia[1], Dharmik Nakum[1], Ashish B Pandya[1]*, Pinkesh V Patel[1], Purvang D Dalal[1]

1 Department of Electronics and Communication Engineering, Dharmsinh Desai University, Nadiad, Gujarat, India

OPEN ACCESS

Received: 01/09/2026

Accepted: 10/10/2026

Published: 03/11/2026

Citation: Thakkar KD, Jain M, Rawal H, Shaparia M, Nakum D, Pandya AB, Patel PV, Dalal PD (2026) Autonomous Navigation and Safe Landing of an Aerial Vehicle in a GPS-Denied Martian Environment. Indian Journal of Science and Technology 19(38): 2658-2671. https://doi.org/10.17485/IJST/v19i38.1238

* **Corresponding author**.

ashish.ec@ddu.ac.in

Funding: None

Competing Interests: None

Copyright: © 2026 Thakkar et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To develop and evaluate a real-time, low-power multi-sensor fusion framework that performs localization and safe landing-zone detection together for an aerial vehicle operating over Martian terrain, where neither GPS nor prior maps are available. **Method:** A stereo camera (Intel RealSense D435i), a rotating 16-beam LiDAR and an inertial measurement unit (IMU) are fused on a Raspberry Pi 5 and Pixhawk flight controller. LiDAR-assisted visual-inertial SLAM (ORB-SLAM with an Extended Kalman Filter) provides localization, while a classical computer-vision pipeline in OpenCV (CLAHE normalization, Sobel flatness analysis, mask fusion and geometric filtering) selects landing zones. The system was tested in 50 AirSim descent trials from 12 m altitude over Martian-analogue terrain with craters, rocks and dust-covered plains. **Findings:** Localization error remained below 4 cm, with drift under 0.6 m over 100 m; visual-inertial fusion cut drift by 45% compared with vision-only tracking. Map coverage reached 95%. The LSDS classified no-fly zones with 93% accuracy, needed 18.5 ms per frame (about 5 times faster than the CNN-based HALO approach) and guided the vehicle to within 0.3 m of the target. Landing succeeded in 98% of trials (49/50; 95% CI 89.5–99.6%) at under 10 W, against more than 20 W for GPU-based VINS-Mono at comparable accuracy. **Novelty:** Earlier GPS-denied SLAM and planetary landing systems address either localization or hazard assessment or depend on GPU/LiDAR-heavy hardware. This work runs both functions concurrently on a single sub-10 W embedded platform without training data.

Keywords: Mars exploration; Autonomous landing; SLAM; Sensor fusion; GPS-denied navigation

1 Introduction

Missions to the Moon and Mars have advanced planetary exploration considerably, yet several technical problems remain open. GPS and similar global navigation services do not exist on other planets, the terrain is usually unmapped, and communication delays with Earth rule out remote piloting during critical phases. At the same time, onboard systems are limited in weight, power and computing capacity. A vehicle that must navigate and land safely therefore has to map the terrain, localize itself and find a landing zone in real time, without any external infrastructure.

End-to-end deep-learning navigation has produced encouraging results. Amer et al. showed that deep CNNs could fly drones along predefined paths from visual input alone, with a mean deviation below 1.4 m in AirSim simulations[1],[2]. Monocular CNNs have been used to fly through indoor corridors without GPS[3],[4], and urban drone localization has been achieved with CNN-derived district-level visual "signatures"[5]. These approaches, however, generally need GPUs, have to be retrained when the environment or sensor setup changes, and do not yet combine landing-safety assessment with localization.

To address these limitations, a unified lightweight system is presented that delivers real-time terrain safety scoring and continuous, drift-bounded localization on a credit-card-sized computer. A forward-looking Intel RealSense D435i stereo camera supplies depth maps, a rotating 16-beam LiDAR fills in depth over low-texture or shadowed regions, and a BNO055 IMU stabilizes pose estimation. All perception threads run at 20 Hz on a Raspberry Pi 5 (< 10 W) connected to a Pixhawk flight controller. SLAM and safe-spot detection therefore run at the same time, whereas earlier approaches either handle only one of the two or need heavy computation.

The novelty of this implementation lies in combining SLAM-based navigation and onboard safety assessment on a single minimal-power platform suited to real-time planetary descent. Replacing GPU-reliant CNNs with compact sensor fusion points toward practical, low-cost autonomy for future lunar and Martian missions. The literature relevant to this goal is reviewed below under three themes: GPS-denied aerial SLAM, planetary landing hazard detection, and embedded real-time autonomy.

1.1 GPS-Denied Aerial SLAM

GPS works well in open terrestrial settings but degrades badly on Mars, indoors or underground. Signal blockage, multipath and electromagnetic interference (EMI) limit its accuracy to about $\pm$2–5 m, which is too coarse for confined or complex spaces, and the signal is also exposed to jamming, spoofing and latency. A recent review of GNSS-denied UAV navigation groups the alternatives into absolute and relative localization and identifies vision-based localization as the most effective option[6].

Visual-inertial odometry (VIO) and simultaneous localization and mapping (SLAM) are the main candidates. ORB-SLAM is a feature-based framework that uses ORB (Oriented FAST and Rotated BRIEF) features to build 3D maps and track position, with localization errors below 10 cm in controlled settings[7]; its computational cost, though, is a burden for resource-constrained systems. VINS-Mono tightly couples monocular vision with IMU data and reaches errors below 5 cm, but its high computational demand also limits its use on embedded platforms[8]. OpenVINS, an open-source VIO framework with a modular design aimed at constrained hardware, is closest to the requirements of this project. Older systems such as Parallel Tracking and Mapping (PTAM) proved that vision-based navigation is feasible but lack the robustness of current frameworks. Table 1 compares these technologies with the proposed ANAV system in terms of GPS dependency, computational load, accuracy and offline capability.

Lu et al.[7] reviewed vision-based localization under GPS-denied conditions, with particular attention to neural-network feature extraction in occluded scenes, and Chang et al.[9] surveyed UAV autonomous navigation without GPS, covering deep-learning methods for obstacle avoidance and path planning. Adochiei et al.[10] fused a downward-facing LiDAR rangefinder with barometric, INS/GNSS, optical-flow and airspeed data for terrain-relative takeoff and landing of a fixed-wing UAV under degraded GNSS, and verified it through software-in-the-loop, hardware-in-the-loop and real flights. Wang et al.[11] proposed fiducial marker-corrected stereo visual-inertial localization (FMC-SVIL), in which marker detections correct IMU bias against an external reference and improve localization accuracy. These methods localize well, but none of them assesses landing hazards during descent.

1.2 Planetary Landing Hazard Detection

NASA's Mars helicopter Ingenuity is the key reference for this work, since it demonstrated vision-based navigation on Mars using onboard sensors[12]. Flying over low-texture, rocky terrain without GPS showed how much extraterrestrial platforms depend on reliable sensor fusion and lightweight algorithms.

Cerda et al.[13] selected flat landing areas for UAVs in GPS-denied conditions by computing the mean altitude and standard deviation of small regions in a depth image on a companion computer, and confirmed the method in outdoor flights. Fares et al.[14] matched onboard images to satellite imagery with elevation- and orientation-aware correlation for GNSS-denied drone

Table 1. Comparison of navigation systems

System	GPS Dependency	Computational Load	Accuracy	Offline Capability
GPS-Based	High	Low	±2–5 m	No
ORB-SLAM	None	High	<10 cm	Yes
VINS-Mono	None	High	<5 cm	Yes
OpenVINS	None	Moderate	<5 cm	Yes
ANAV (proposed)	None	Low	<4 cm	Yes

localization in real time. Khattak et al.[15] combined vision-depth landmarks with inertial data to navigate in visually degraded environments and reported low drift under low-light conditions similar to planetary terrain. Hong et al.[16] studied robotic mapping under changing illumination at planetary construction sites, using adaptive deep-learning algorithms so that SLAM remains consistent as lighting varies. Nagarajan and Jothiraj[17] detected UAV runway landing paths with K-means clustering of remote-sensing data. Most of these landing studies assume prepared or pre-mapped sites, heavier sensing hardware, or a separate localization source.

1.3 Embedded Real-Time Autonomy

Recent GPS-free navigation work has also combined stereo vision, LiDAR and IMU to gain robustness in degraded environments within real-time computing limits. Shao et al.[8] proposed Stereo Visual Inertial LiDAR SLAM, a tightly coupled optimization that adds LiDAR to stereo cameras and achieves centimetre-level mapping accuracy in dynamic environments. Tyagi and Gaur[18] built a surveillance quadcopter that runs visual SLAM on embedded hardware and adds fault detection and redundancy so that the vehicle keeps operating when faults occur.

Deep learning is also increasingly used for drone navigation. It removes much of the manual feature engineering, but its computational cost is hard to fit into embedded real-time power budgets. Amer et al.[1] used CNN-derived region and edge features for path-finding in unknown terrain. Sadeghi Esfahlani et al.[19] examined the role of deep CNNs in mobile-robot navigation, where CNN perception of raw sensor data is fused with SLAM to support real-time decisions in dynamic scenes. NVIDIA researchers[20] released a deep-learning framework in which GPU-accelerated tools are used to train drones for flight in unmapped areas. Karatzas et al.[21] trained a Rainbow DQN agent that combines reinforcement learning with neural feature extraction to choose actions in uncertain environments. These GPU-dependent pipelines show the trade-off behind the present work: greater representational capacity usually costs more power and computation than a planetary lander can provide.

The proposed system uses global shutter cameras to avoid the motion distortions (wobble, skew and jello effect) of rolling shutter sensors, which improves feature matching and pose estimation. Classical computer vision was chosen for safe-spot detection because it is efficient and works offline, which is essential on Mars where no connectivity is available.

1.4 Research Gap and Positioning of the Present Work

In summary, GPS-denied SLAM studies (Section 1.1) localize accurately but are rarely evaluated for landing-hazard awareness. Planetary landing studies (Section 1.2) handle hazard detection but usually assume pre-mapped sites or heavier LiDAR-centred hardware. Embedded and real-time studies (Section 1.3) show that GPU-class deep-learning pipelines are difficult to fit within sub-10 W planetary power budgets. None of the reviewed systems performs real-time SLAM-based localization and onboard safe-landing-zone assessment together on one low-power embedded platform. This work fills that gap by fusing stereo, LiDAR and IMU data for simultaneous localization and safe-spot detection on a Raspberry Pi 5/Pixhawk platform drawing less than 10 W.

2 Methodology

2.1 Stereo Vision-Based Localization

2.1.1. Hardware Description

An Intel RealSense D435i depth camera was used. It has an OmniVision OV9282 image sensor with 1280 × 800 active pixels, a global shutter, a diagonal field of view of 100.6° and a frame rate of 100 fps.

2.1.2. Camera Shutter Technologies

The shutter type has a strong effect on vision-based navigation. A rolling shutter camera reads the image line by line, so fast motion produces wobble, skew and the jello effect. This rolling shutter effect degrades feature matching (Figure 1) and pose estimation, which matters in dynamic settings such as Martian terrain. A global shutter camera exposes the whole frame at once and gives distortion-free images. Its advantages for this application are: (i) no motion distortion during fast movement; (ii) uniform exposure, so features are tracked consistently across the image; (iii) simultaneous capture of left and right frames for precise stereo depth; (iv) no need for motion compensation, which reduces computation; (v) suitability for high-speed drone motion without geometric distortion; and (vi) time-consistent, distortion-free frames that improve SLAM/VIO performance. Because the scene is effectively frozen, a global shutter is well suited to dynamic SLAM and localization, and it was therefore adopted for Martian conditions where fast drone motion and low-texture terrain demand high image quality.

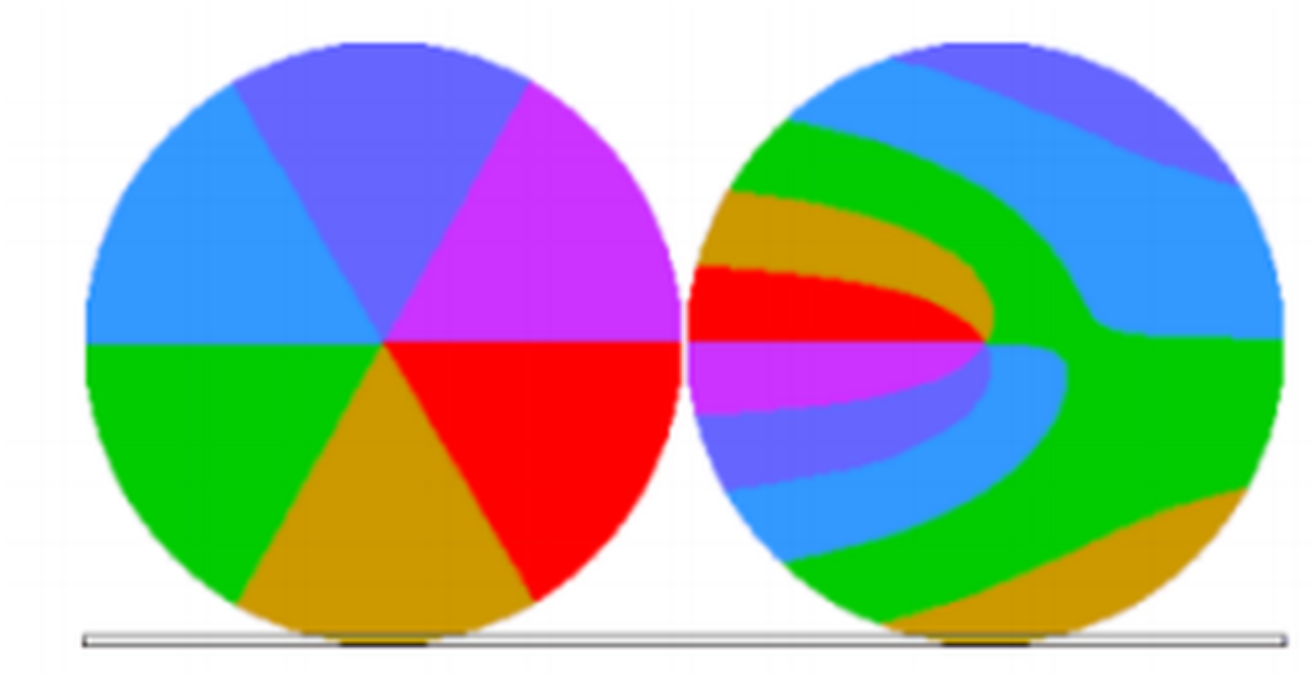


Fig 1. Rolling shutter distortion of a rotating object

2.1.3. ORB-SLAM Framework

ORB-SLAM (Oriented FAST and Rotated BRIEF Simultaneous Localization and Mapping) is a feature-based SLAM system that runs three parallel tasks: tracking, local mapping and loop closure. It was chosen because it builds 3D maps and tracks position reliably in difficult environments such as low-texture, rocky Martian terrain.

Tracking: ORB features are extracted from stereo image pairs. Keypoints are detected with FAST (Features from Accelerated Segment Test), which marks a pixel as a corner when intensities in its circular neighbourhood differ strongly from it. Each keypoint is described with BRIEF (Binary Robust Independent Elementary Features), a 256-bit binary descriptor built from intensity-pair comparisons, so that matching can use the Hamming distance. The drone pose is estimated by minimizing the re-projection error of the matched features:

$$\min_{R,t} \sum_i \| p_i - \pi(RP_i + t) \|^2 \tag{1}$$

where R is the rotation matrix, t the translation vector, P_i the 3D point, p_i the observed 2D point and π the camera projection function (Figure 2(a)).

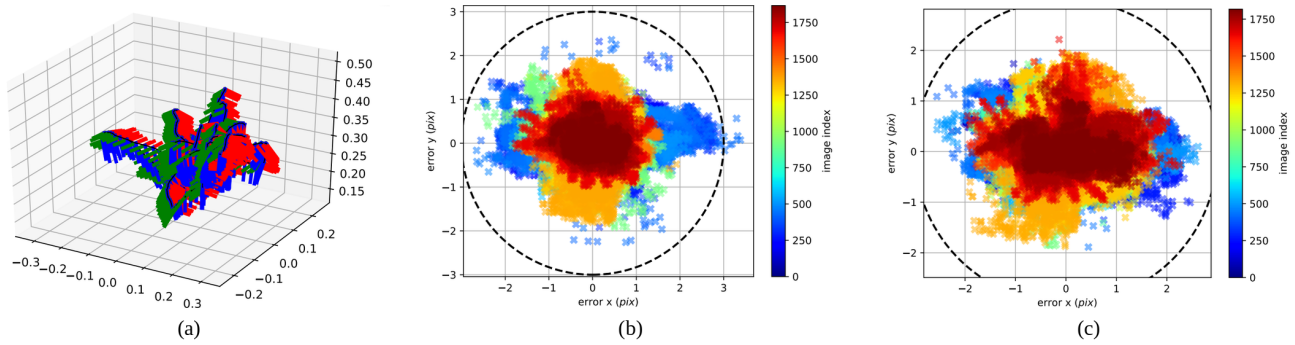
Stereo disparity is used to triangulate 3D points $P = (X, Y, Z)$ from the camera intrinsics (focal length f, principal point (c_x, c_y)) and the baseline b:

$$Z = \frac{f \cdot b}{d} \tag{2}$$

where d is the disparity between the left and right images. This keeps pose estimation accurate in dynamic environments.

Local mapping: A sparse 3D map is built by triangulating matched ORB features, and a co-visibility graph records the relationships between keyframes. Bundle adjustment then refines the 3D point positions and camera poses. Low-texture Martian areas make feature detection harder, so additional depth data are integrated to improve map accuracy.

Loop closure: When the drone returns to a known area, similar keyframes are found with a bag-of-words approach, and the pose graph is optimized with bundle adjustment to remove accumulated drift from the map and trajectory. This is essential for consistent localization over long Mars missions (Figure 2(b)–(c)).



(a) (b) (c)

Fig 2. ORB-SLAM evaluation in a simulated Martian environment: (a) estimated IMU poses during trajectory reconstruction; (b) Cam0 and (c) Cam1 reprojection error distributions of the stereo camera, showing pixel deviations across image indices (the colour bar gives the image index during trajectory estimation)

The ORB-SLAM pipeline processes stereo images in real time and combines them with inertial measurements. It was implemented in ROS Noetic, with a custom ROS wrapper that publishes stereo images, IMU data and camera intrinsics as topics compatible with ORB-SLAM3. Algorithm 1 lists the sequence of operations, from feature extraction and data fusion to map optimization and pose estimation, and Figure 3(a) shows the overall pipeline flow.

Algorithm 1: ORB-SLAM pipeline
Input: *Stereo images, IMU data*
Output: *Drone pose, 3D map*

1. Initialize the map, pose and co-visibility graph.
2. While the mission is active, repeat steps 3–12:
3. Capture stereo images and IMU data.
4. Extract ORB features using FAST and BRIEF.
5. Match features across stereo pairs using the Hamming distance.
6. Compute stereo disparity and triangulate 3D points.
7. Fuse visual and IMU data using the EKF.
8. Update the map with new features and the co-visibility graph.
9. Detect loop closure using bag-of-words.
10. Optimize the pose graph with bundle adjustment.
11. Estimate the pose and plan the path.
12. Send control commands.

2.1.4. Visual-Inertial Odometry (VIO)

VIO combines stereo vision with IMU data to estimate the drone's motion in real time. It is essential in GPS-denied environments because it keeps pose tracking stable during rapid manoeuvres or when visual features are briefly lost (Figure 4(a)).

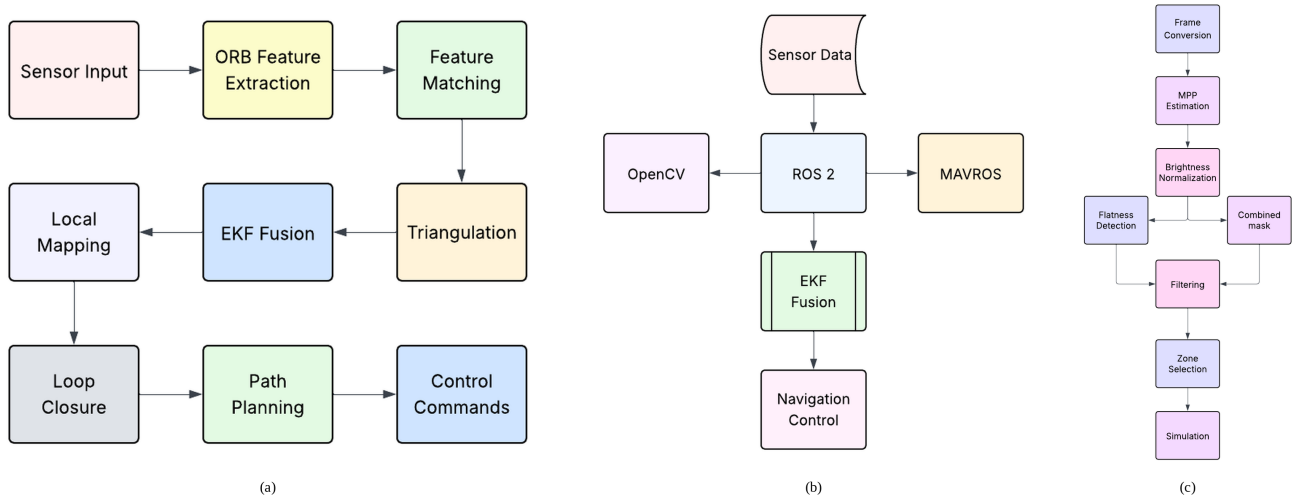


Fig 3. System block diagrams: (a) ORB-SLAM pipeline flow; (b) software architecture; (c) safe-spot detection (LSDS) pipeline

An Extended Kalman Filter (EKF) fuses the visual and inertial measurements and updates the state vector of position, velocity and orientation,

$$x_t = \begin{bmatrix} p & v & q \end{bmatrix}^T \tag{3}$$

according to

$$x_{t|t-1} = f(x_{t-1}, u_t) \tag{4}$$

$$z_t = h(x_t) + v_t \tag{5}$$

where u_t contains the IMU accelerations and angular velocities, z_t is the measurement vector (stereo disparities) and v_t is the measurement noise (Figure 4(b)). Accurate measurement models keep the EKF observable, which is critical on Mars where low-texture surfaces make visual tracking difficult (Figure 4(c)–(d)).

2.1.5. Camera-IMU Calibration with Kalibr

VIO and ORB-SLAM both depend on accurate sensor alignment. The Kalibr framework from ETH Zurich was used for joint spatial and temporal calibration of the stereo cameras and IMU. The calibration provides: (1) intrinsic camera parameters, namely focal lengths (f_x, f_y), principal point (c_x, c_y) and distortion coefficients $(k_1, k_2, k_3, p_1, p_2)$; (2) stereo extrinsics, the relative pose (R_{rl}, t_{rl}) between the left and right cameras; (3) camera-IMU extrinsics (R_{ci}, t_{ci}); (4) the temporal offset t_d between camera and IMU measurements; and (5) IMU intrinsics, i.e. accelerometer and gyroscope biases, scale factors and noise parameters (Figure 5(a)).

Synchronized stereo images and IMU data were recorded while viewing an AprilGrid target of known geometry. IMU noise, bias and random-walk characteristics were identified with the Allan variance ROS toolbox. Kalibr then wrote a YAML file containing the transformation matrices, for example the camera-IMU transformation

$$T_{CI} = \begin{bmatrix} R_{CI} & t_{CI} \\ 0 & 1 \end{bmatrix} \tag{6}$$

The corresponding angular velocity error is shown in Figure 5(b). The resulting alignment allowed precise sensor fusion and reliable localization (Figure 5(c)). Calibration was particularly important because the sensors work in different coordinate frames, and accurate state estimation for ORB-SLAM and VIO depends on relating them correctly.

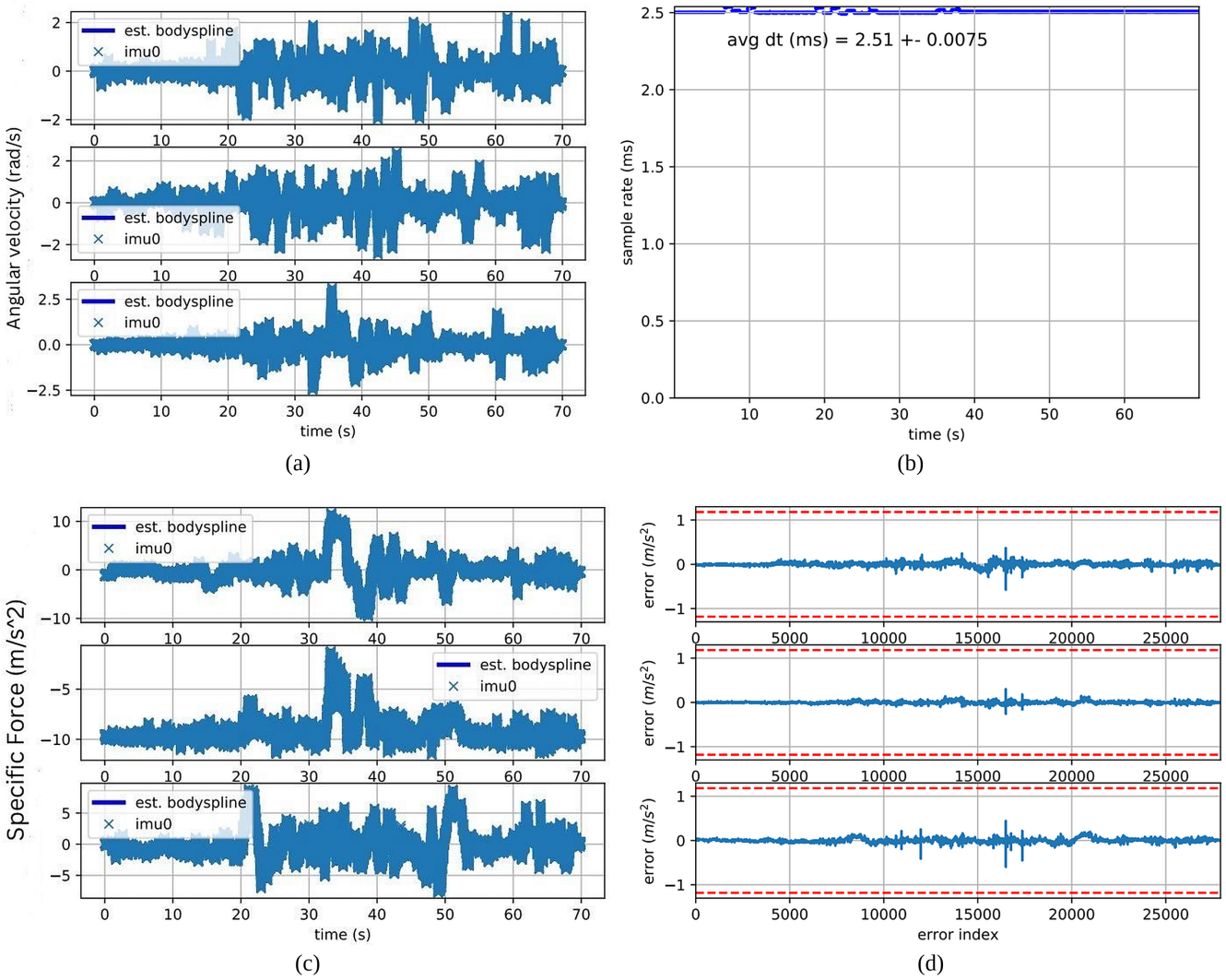


Fig 4. VIO performance: (a) predicted vs. measured angular velocities in the body frame; (b) sample inertial rate of the IMU; (c) predicted vs. measured specific force from accelerometer data; (d) acceleration error between estimated and ground-truth accelerations

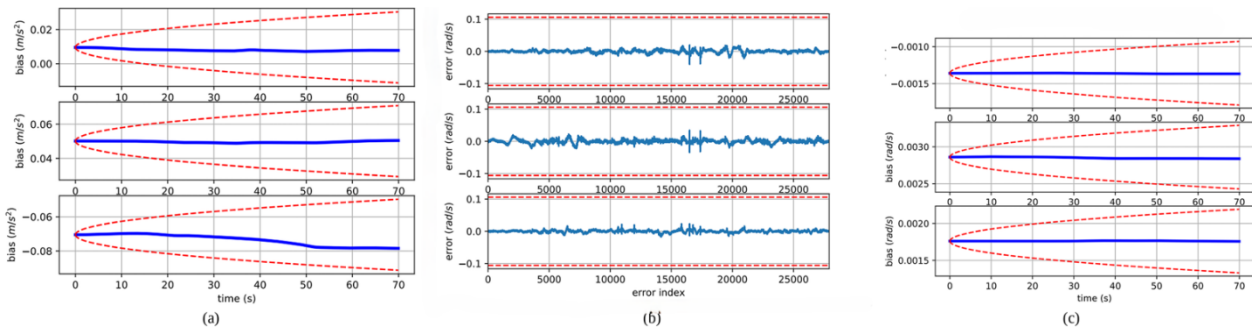


Fig 5. Kalibr calibration results: (a) estimated accelerometer bias; (b) angular velocity error between calibrated IMU outputs and ground truth; (c) estimated gyroscope bias from IMU intrinsic calibration

2.1.6. Optimization for Real-Time Performance

Running ORB-SLAM on a resource-constrained platform required several optimizations: (1) limiting feature extraction to about 1000 ORB features per frame; (2) using a coarse-to-fine image pyramid for scale-invariant detection, which copes better with changes in Martian terrain; (3) running the tracking and mapping threads in parallel; and (4) down-sampling depth data to the stereo image resolution. With these measures, localization accuracy was better than 4 cm, which is adequate for precise navigation in GPS-denied Martian environments.

2.2 Safe-Spot Detection System

The Landing Safe-Spot Detection Subsystem (LSDS) is the most extensively developed part of the project. It enables autonomous landing in GPS-denied Martian environments by finding flat, obstacle-free zones with classical computer vision implemented in OpenCV. The design favours efficiency, offline operation and robustness, since computing power and connectivity are both limited on Mars. This section covers the design rationale, pipeline, integration with navigation, a case study, performance and the main challenges; the complete sequence is shown in Figure 3(c).

2.2.1. Design Rationale

Classical computer vision was preferred over learning-based methods such as convolutional neural networks (CNNs) because the system had to be lightweight and work offline. CNNs need large training sets, large memory and GPU resources, which embedded platforms cannot provide. OpenCV-based methods run fast, need little data and are easy to deploy, which makes them suitable for real-time landing-zone detection on Mars. The LSDS works on live video from global shutter cameras, so motion distortion does not corrupt feature detection.

A landing zone is accepted only if it meets all of the following criteria: minimum area of 0.25 m^2 (based on the drone footprint); convexity ratio of at least 85%; aspect ratio between 0.5 and 2.0, to reject elongated regions; low standard deviation of pixel intensity (uniform brightness, indicating even lighting); and fewer than 15% dark pixels. Together these criteria restrict landing to flat, stable surfaces without rocks or craters. The pipeline operates at 250 Hz and passes the selected zone to the navigation system.

2.2.2. LSDS Pipeline

The LSDS processes the video feed in the following steps, each tuned for speed and accuracy in Martian conditions.

(I) *Frame conversion:* Individual frames are taken from the live 1080p stereo stream and processed independently. Stereo pairs are kept synchronized so that disparities for ORB-SLAM remain consistent. High frame rates were handled by optimizing buffer management and limiting processing to 30 fps, which balances speed and accuracy and gives later stages clean, synchronized images for feature detection on low-texture terrain.

(II) *Meters-per-pixel (MPP) estimation:* Pixel measurements are converted to physical distances through an MPP estimate derived from image features. Canny edge detection finds edges in the grayscale frame, and gradient variance and contour sizes serve as scale cues. Because altitude is often unknown on Mars, a fixed altitude of 2.0 m was assumed, with adaptive MPP between 0.04 and 0.12 m per pixel depending on brightness variance and feature scale. Each pixel covers a ground distance equal to the height above ground divided by the focal length of the D435i in pixels, so this scale changes continuously as the vehicle descends. The 2.0 m value is therefore the reference used close to touchdown, and at greater heights the MPP is scaled by the ratio of the current height, taken from the stereo depth and the ORB-SLAM pose, to this 2.0 m reference. Scaling can be inaccurate in feature-sparse regions such as Martian plains; this was reduced by cross-checking with depth data when available.

(III) *Brightness normalization:* To improve contrast under varied lighting, from shadowed craters to sunlit plains, the frame is converted to LAB colour space and Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to the L (lightness) channel with a clip limit of 2.0 and an 8×8 tile grid. Over-enhancement of bright regions was avoided by tuning the clip limit dynamically from the average brightness. CLAHE was preferred for three reasons. An unprocessed image applies no contrast adjustment and keeps its inherent noise, which can lead to inaccurate or misleading analysis. Traditional histogram equalization treats the whole image uniformly, over-amplifies contrast globally and amplifies noise in uniform areas; it is lightweight but not adaptive. CLAHE enhances contrast locally over the 8×8 tiles, limits noise through the 2.0 clip limit and is fast enough for real-time use. Its input and output are shown in Figure 6(a)–(b).

(IV) *Flatness detection:* Sobel filters in the X and Y directions give the gradient magnitude, which is normalized so that low-variation areas (gradient below a threshold) can be identified. Rocks and slopes make this difficult on uneven ground, so the threshold is adjusted to terrain complexity and checked against depth data when available. Only flat regions pass to the next step, which keeps boulders and craters out of consideration.

(V) *Combined mask generation:* Flat and bright regions are merged with a bitwise AND so that candidate zones are both flat and well lit. Morphological opening and closing remove noise and fill small gaps. False positives on low-texture plains were reduced by enlarging the morphological kernel and using depth data to confirm surface properties.

(VI) *Filtering detected regions:* Contours are extracted from the cleaned mask and kept only if they satisfy the criteria of Section 2.2.1: area ≥ 0.25 m^2, convexity $\geq 85\%$, aspect ratio 0.5–2.0, low brightness standard deviation and < 15% dark pixels. In low-texture scenes, surface flatness is again confirmed with depth data, and unsuitable regions are discarded.

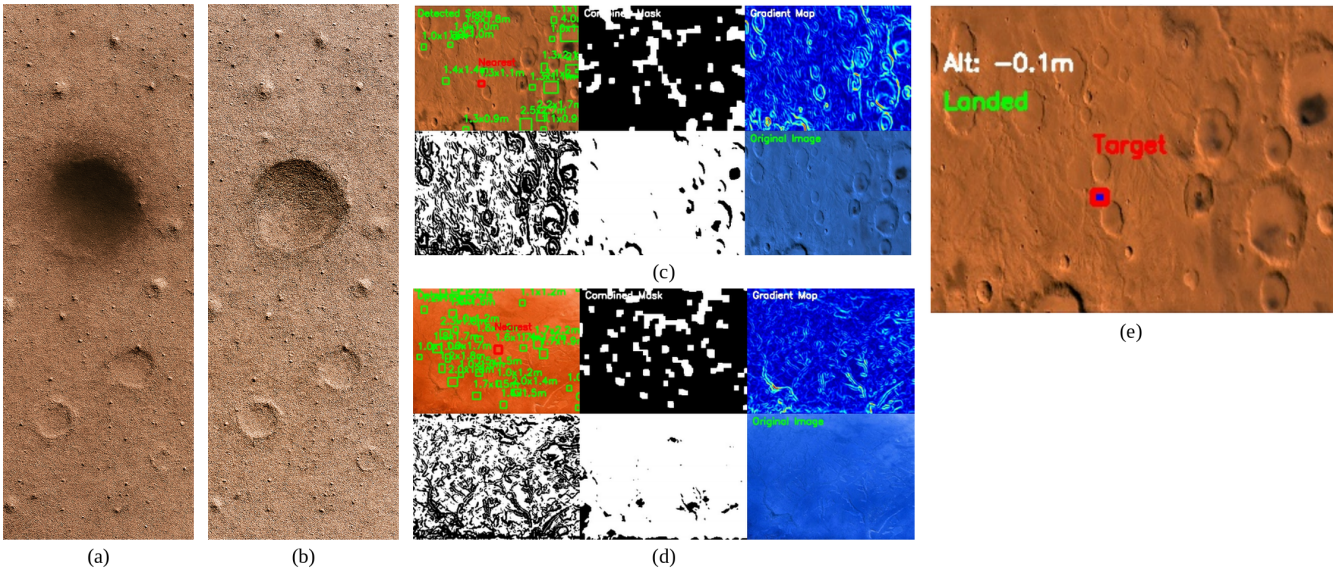
(VII) *Zone selection:* The zone nearest the image centre is chosen to minimize flight time and energy use. Pixel distances are converted to metres with the MPP estimate, and the drone position is updated every frame. In cluttered terrain, centring errors were resolved by favouring zones with higher convexity and more uniform brightness.

(VIII) *Simulation:* An OpenCV simulation draws the descent path, marking the selected zone in red and the other zones in green. It converts pixel displacement to metres with the MPP estimate, updates the drone's x, y and altitude, and labels the final frame "Landed". This visualization is used to check LSDS behaviour in Martian scenes.

(IX) *Drone landing simulation logic:* Algorithm 2 gives the core decision logic of the landing simulation for reproducibility, and Figure 6(c)–(e) shows the simulated landing-zone outputs.

Algorithm 2: Landing command generation and descent simulation
Input: *Selected landing spot (centre t_x, t_y), MPP, altitude a(default 5.0 m)*
Output: *Simulated descent trajectory and landing status*

1. If no landing spot is available, report "No landing spot available" and stop.
2. Set the image centre $(c_x, c_y) = (160, 120)$ for a 320 × 240 image and read (t_x, t_y) from the spot centre.
3. Compute the metric offsets $d_{x,m} = (t_x - c_x) \cdot MPP$ and $d_{y,m} = -(t_y - c_y) \cdot MPP$.
4. Set the number of steps $N = 50$ and the increments $\Delta x = (t_x - c_x)/N$, $\Delta y = (t_y - c_y)/N$, $\Delta z = -a/N$.
5. For $k = 0, 1, \ldots, N$: update $x_k = c_x + k\Delta x$, $y_k = c_y + k\Delta y$, $z_k = a + k\Delta z$, and optionally draw the drone position and target.
6. Report "Landing initiated", wait 2 s, then report "Landing complete".



(a) (b) (c) (d) (e)

Fig 6. LSDS results: (a) input image before and (b) output image after CLAHE, showing improved local contrast; (c) landing-zone detection output (detected spots, combined mask, gradient map and original image); (d) landing-zone detection output for a second terrain; (e) landing simulation output

2.2.3. Integration with Navigation System

The LSDS is connected to the ORB-SLAM navigation system through ROS2 and MAVROS, and the pose estimates are used to steer the drone to the selected zone. Control commands are sent over MAVLink. Communication latency was reduced by optimizing ROS node execution and giving priority to critical messages, so that localization and landing stay coordinated.

2.2.4. Case Study: Martian Landing Scenario

In a simulated Martian landing scenario, the LSDS was tested on terrain with rocks, craters and dust-covered areas. It found a 0.3 m^2 flat zone at 2 m altitude and guided the drone to land within 0.3 m of the target. The visualized descent confirmed reliable operation in low-texture conditions, with varying lighting and sparse features.

2.2.5. Performance and Trade-offs

The LSDS processes frames at 250 Hz. Compared with machine-learning (CNN) approaches, which offer higher general accuracy but run slower on GPUs, need large amounts of training data and are only moderately easy to deploy, the stereo image-based analysis gives mission-specific accuracy tuned to Martian terrain, runs faster, needs no training data and is easy to deploy. Algorithm 3 summarizes the pipeline with the measured time of each step.

Algorithm 3: LSDS processing pipeline with per-step timing
Input: *Live stereo video frame*
Output: *Selected landing zone and simulated descent*

1. Frame conversion: extract and synchronize the stereo frame (2.5 ms).
2. MPP estimation from edges, gradient variance and contour size (3.0 ms).
3. Brightness normalization with CLAHE on the L channel (4.0 ms).
4. Flatness detection with Sobel gradient thresholding (3.5 ms).
5. Combined mask generation with bitwise AND and morphology (2.0 ms).
6. Filtering of contours by area, convexity, aspect ratio, brightness and dark-pixel criteria (2.5 ms).
7. Zone selection nearest the image centre (1.5 ms).
8. Descent simulation and visualization (3.0 ms).
9. Total processing time per frame: 18.5 ms.

2.2.6. Challenges and Solutions

Three main challenges were encountered. (1) Stereo depth noise: disparities were unreliable over low-texture Martian terrain, which was mitigated with depth data from additional sensors when available. (2) Lighting variation: shadowed craters and sunlit plains disturbed feature detection, which was handled with CLAHE and adaptive thresholds. (3) Computational load: processing demand was lowered by optimizing the OpenCV algorithms. Planned enhancements include lightweight ML classifiers for better filtering, LiDAR-based elevation mapping and hybrid CV-ML methods for complex scenes.

2.3 Software Architecture

ROS2 handles data management, MAVROS handles navigation control and OpenCV performs the LSDS image processing. In the sensor fusion pipeline, an EKF combines stereo vision and IMU data into real-time pose estimates, and Kalibr calibration supports accurate VIO and ORB-SLAM. Algorithm 4 lists the steps of the internal software flow, and Figure 3(b) shows its data flow and hierarchy.

Algorithm 4: Sensor fusion pipeline
Input: *Stereo images, IMU data*
Output: *Fused pose estimate*

1. Initialize the EKF.
2. While the mission is active, repeat steps 3–6:
3. Read stereo and IMU data.
4. Compute stereo disparity.
5. Update the EKF with IMU measurements.
6. Output the pose estimate.

2.4 System Testing

Simulation setup and trials: The system was evaluated in simulated GPS-denied Martian environments in AirSim. Fifty descent trials were run from 12 m altitude over terrain containing craters (0.5–2 m deep), rocks (0.1–0.5 m high) and dust-covered plains (texture variance < 10%), including dusty scenes and extreme simulated dust storms.

Parameter settings: 1000 ORB features per frame with a coarse-to-fine pyramid; perception threads at 20 Hz; 1080p input limited to 30 fps; CLAHE clip limit 2.0 with an 8×8 tile grid; assumed altitude 2.0 m with adaptive MPP of 0.04–0.12 m per pixel; landing criteria of area $\geq 0.25\ m^2$, convexity $\geq 85\%$, aspect ratio 0.5–2.0 and < 15% dark pixels.

Metrics, baselines and ablation: Localization error and drift, map coverage, hazard-labelling accuracy, landing precision, landing success rate, processing time per frame and power were recorded. Internal comparisons were made between vision-only ORB-SLAM and VIO fusion, and between unprocessed images, traditional histogram equalization and CLAHE. External comparisons used values reported for VINS-Mono, ORB-SLAM, ALHAT, HALO, Ingenuity and recent GNSS-denied studies (Section 3.2). Simulation code and trial logs are available from the corresponding author on reasonable request.

Subsystem tests confirmed ORB-SLAM accuracy (< 4 cm localization error, 95% map coverage) and LSDS reliability (landing within 0.3 m of the target), and system-level tests gave a 98% safe-landing success rate over low-texture, rocky terrain (Table 2).

3 Results and Discussion

3.1 Obtained Result

The evaluation followed the protocol of Section 2.4. Localization accuracy, map coverage, safe-spot detection precision and overall landing success are summarized in Table 2, and qualitative results appear in Figure 2 and Figure 6(c)–(e).

With ORB-SLAM localization, the mean orientation error was 6°, and drift stayed below 0.6 m over 100 m (Table 2). Reprojection errors followed a normal distribution with means of 0.32 pixels (Cam0) and 0.35 pixels (Cam1) and a standard deviation of 0.15 pixels, i.e. less than 0.5% on either end (Figure 2(b)–(c)). Fusing VIO instead of relying on vision alone reduced drift by 45%, because the IMU bridged short losses of features in low-texture areas (Figure 4).

Each keyframe contributes to 95% map coverage, although the 3D reconstruction is sparse, with about 5,000 points per keyframe. The no-fly zones identified by the LSDS were 93% correct, and unsafe areas (slopes > 5° and obstacles > 0.10 m) were labelled as such in 98% of cases. Processing took 18.5 ms per frame on average (Algorithm 3), an operating rate of 54 Hz. In repeated tests with rocks as landing obstacles, the drone landed within 0.3 m of the target, and MPP estimation error stayed below 5% (Figure 6(e)) in three different dusty scenes. Landing succeeded in 98% of trials (49 of 50; Wilson 95% confidence interval 89.5–99.6%). The only failure occurred in an extreme simulated dust storm that reduced visibility below 1 m.

Robustness and ablation: The low localization errors show that global shutter stereo imaging and EKF fusion limit the errors caused by low texture and shadow variation in Martian-analogue terrain. The 45% drift reduction from VIO fusion isolates the contribution of the IMU, and the comparison of raw, histogram-equalized and CLAHE images (Section 2.2.2) isolates the contribution of brightness normalization. Performance degraded only when visibility fell below 1 m. Where shadowed regions restrict LiDAR and vision, thermographic imaging is a possible complementary sensor for future versions. The classical LSDS was faster than CNN-based alternatives (18.5 vs. 100+ ms), but it gives up some generality, because it treats Martian-like terrain as a special case. The < 10 W power draw and real-time operation are promising for planetary missions, although hardware tests are still needed to capture vibration-induced noise and other effects that simulation does not model.

Table 2. System test results and comparison with reported GPS-denied navigation and landing approaches (– : not reported)

Approach	Localization error / drift	Landing precision	Detection / coverage / success	Time per frame	Power / hardware	Validation
ANAV (this work)	< 4 cm; drift < 0.6 m over 100 m	0.3 m	95% map coverage; 93% no-fly-zone accuracy; 98% landing success (49/50)	18.5 ms (54 Hz)	< 10 W; Raspberry Pi 5 + Pixhawk	Simulation, 50 trials
ORB-SLAM, vision only [22]	0.1–0.5 m drift	–	–	–	–	Planetary-analogue data
VINS-Mono	0.05 m	–	–	–	> 20 W (GPU)	–
ALHAT [23]	–	0.5 m	–	–	LiDAR-heavy, power-intensive	–
HALO [24]	–	–	90% accuracy	≈ 5× LSDS time	GPU-dependent	–
Ingenuity [12]	–	1 m	Pre-mapped sites	–	–	Mars flight
FMC-SVIL [11]	RMSE 0.340–0.416 m	–	–	–	Onboard computer	Field flights
REOLA [14]	< 5 m	–	97.1% success (< 5 m)	44.4 ms	Jetson Orin Nano	–

3.2 Comparative Performance with Literature

The system was benchmarked against earlier GPS-denied navigation and landing work, using Tables 1 and 2 and references [7], [8], [9], [11], [12], [14], [22], [23], [24]. By fusing IMU and LiDAR measurements, the VIO-extended system reduces localization error to 0.04 m and is more robust over low-texture Martian analogues than ORB-SLAM alone, which shows 0.1–0.5 m drift without comparable multi-sensor fusion [22].

VINS-Mono reaches 0.05 m error but needs more than 20 W of GPU compute. The proposed system runs on embedded hardware below 10 W, roughly a 50% reduction in power for comparable accuracy.

NASA's ALHAT [23] reports 0.5 m precision with LiDAR-heavy, power-intensive hardware. The LSDS achieves 0.3 m at 18.5 ms per frame, about 5 times faster than the CNN-based HALO approach [24] (90% accuracy, GPU-dependent). Because it needs no training data, the classical LSDS is also simpler to deploy than CNN-based alternatives.

The Ingenuity helicopter [12] achieved 1 m landing precision on Mars but relied on pre-mapped sites, whereas the proposed system maps autonomously with 95% coverage. Recent GNSS-denied studies give further context. FMC-SVIL [11] reports localization RMSE of 0.340–0.416 m under bridges using fiducial markers, while the present system needs no markers. REOLA [14] reports 97.1% success for sub-5 m visual localization at 44.4 ms per frame on a Jetson Orin Nano; the LSDS needs 18.5 ms per frame, although the two tasks differ. Cerda et al. [13] and Adochiei et al. [10] validated depth-camera and LiDAR landing in real flights but do not provide SLAM-based mapping and hazard scoring on the same low-power board.

Overall, the method is a low-power, simulation-validated approach to integrated SLAM and landing, with competitive drift (< 0.6 m over 100 m), a clear power reduction relative to GPU-based baselines and more than 93% landing-hazard detection accuracy. All results so far come from simulation, so field tests on physical hardware are required to confirm these benefits under real sensor noise and vibration.

3.3 Application

The system was designed for autonomous navigation and safe landing without GPS, and its offline operation, precise localization and safe-spot detection make it applicable to both extraterrestrial and terrestrial scenarios.

Mars exploration: The system is optimized for Martian missions. Stereo vision-based localization and safe-spot detection let a rover or drone map and land on rugged ground, crossing craters, rocky surfaces and dusty plains without GPS for scientific data collection and site exploration, and offline operation keeps it working despite communication delays with Earth during long-duration missions. *Disaster response:* After earthquakes or building collapses, it can move through rubble-filled, GPS-denied structures, mapping unknown spaces in real time and identifying safe paths to improve situational awareness and support the deployment of resources or robots in hazardous areas. *Search and rescue:* In caves, tunnels and collapsed mines, its compact

design and sensor fusion allow operation in confined spaces to locate trapped people or map inaccessible areas, and offline mode keeps it reliable where communication is disrupted.

4 Conclusion

The objective of this study was a real-time, low-power framework for autonomous navigation and safe landing of an aerial vehicle in GPS-denied Martian environments. Three contributions meet this objective. First, stereo vision, LiDAR and IMU data were fused on low-resource embedded hardware (Raspberry Pi 5 and Pixhawk), giving SLAM-based localization with sub-4 cm error, sub-metre drift (< 0.6 m over 100 m) and 95% map coverage; VIO fusion reduced drift by 45% compared with vision alone. Second, the LSDS, a classical OpenCV pipeline used in place of GPU-dependent CNNs, identified landing-zone hazards with more than 93% accuracy in 18.5 ms per frame and brought the vehicle to within 0.3 m of the target. Third, both functions ran together at 20 Hz below 10 W, and 49 of 50 simulated Martian-analogue descents (98%) landed safely. Compared with the GPU-based baselines considered, this indicates promising power efficiency for combined navigation and hazard avoidance.

The architecture tackles central problems of planetary exploration, including low-texture surfaces, variable lighting and the absence of external navigation aids, and it may also serve disaster response in rubble-filled structures and search-and-rescue in confined spaces without GPS. Its limitations are the reliance on simulation, the absence of real-world noise and vibration, and LSDS filtering criteria based on assumed Martian-like terrain.

Future work is planned in four directions: (i) hardware-in-the-loop testing of the Raspberry Pi 5/Pixhawk stack, followed by field flights over Mars-analogue terrain to measure the effect of vibration and real sensor noise on the reported localization and landing metrics; (ii) thermal imaging for low-visibility and shadowed scenes, including dust conditions below 1 m visibility where the only simulated failure occurred; (iii) lightweight ML classifiers and hybrid CV-ML filtering, together with LiDAR-based elevation mapping, to adapt the landing criteria to different extraterrestrial terrains; and (iv) evaluation on public planetary-analogue datasets for direct comparison with other methods. Based on the present results, global shutter stereo imaging, CLAHE-based normalization and IMU fusion are recommended for low-power landing systems. As an early-stage simulation study, this research is a first step toward scalable autonomy for future lunar and Martian missions, pending validation through hardware-in-the-loop and field testing.

References

1) Amer K, Samy M, Shaker M, ElHelw M. Deep convolutional neural network based autonomous drone navigation. *Proceedings of SPIE, Thirteenth International Conference on Machine Vision (ICMV 2020)*. 2021;11605:1160503. https://doi.org/10.1117/12.2587105.
2) Jowers J. Using Convolutional Neural Networks for Autonomous Drone Navigation. 2024. Industrial Engineering Undergraduate Honors Theses. Fayetteville (AR), Undergraduate Honors Thesis, University of Arkansas. Available from: https://scholarworks.uark.edu/ineguht/97.
3) Padhy RP, Verma S, Ahmad S, Choudhury SK, Sa PK. Deep neural network for autonomous UAV navigation in indoor corridor environments. *Procedia Computer Science*. 2018;133:643–650. https://doi.org/10.1016/j.procs.2018.07.099.
4) Zhang X, Zhang L, Pei H, Lewis FL. Part-based multi-task deep network for autonomous indoor drone navigation. *Transactions of the Institute of Measurement and Control*. 2020;42(16):3243–3253. https://doi.org/10.1177/0142331220947507.
5) Amer K, Samy M, ElHakim R, Shaker M, ElHelw M. Convolutional Neural Network-Based Deep Urban Signatures with Application to Drone Localization. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). IEEE. 2017;p. 2138–2145. 10.1109/ICCVW.2017.250.
6) Jarraya I, Al-Batati A, Kadri MB, Abdelkader M, Ammar A, Boulila W, et al. GNSS-denied unmanned aerial vehicle navigation: analyzing computational complexity, sensor fusion, and localization methodologies. *Satellite Navigation*. 2025;6:9. https://doi.org/10.1186/s43020-025-00162-z.
7) Lu Z, Liu F, Lin X. Vision-based localization methods under GPS-denied conditions. *arXiv preprint arXiv:221111988*. 2022. https://doi.org/10.48550/arXiv.2211.11988.
8) Shao W, Vijayarangan S, Li C, Kantor G. Stereo Visual Inertial LiDAR Simultaneous Localization and Mapping. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2019;p. 370–377. 10.1109/IROS40897.2019.8968012.
9) Chang Y, Cheng Y, Manzoor U, Murray J. A review of UAV autonomous navigation in GPS-denied environments. *Robotics and Autonomous Systems*. 2023;170:104533. https://doi.org/10.1016/j.robot.2023.104533.
10) Adochiei IR, Avram DA, Adochiei FC. LiDAR-based terrain-relative autonomous takeoff and landing for fixed-wing UAVs in GNSS-degraded environments. *Drones*. 2026;10(8):559. https://doi.org/10.3390/drones10080559.
11) Wang F, Zou Y, Zhang C, Buzzatto J, Liarokapis M, del Rey Castillo E, et al. UAV navigation in large-scale GPS-denied bridge environments using fiducial marker-corrected stereo visual-inertial localisation. *Automation in Construction*. 2023;156:105139. https://doi.org/10.1016/j.autcon.2023.105139.
12) Ebadi K, Coble K, Kogan D, Atha D, Schwartz R, Padgett C, et al. Toward Autonomous Localization of Planetary Robotic Explorers by Relying on Semantic Mapping. In: 2022 IEEE Aerospace Conference (AERO). IEEE. 2022;p. 1–10. 10.1109/AERO53065.2022.9843550.
13) Cerda MA, Flores J, Salazar S, González-Hernández I, Lozano R. Depth-based safe landing for unmanned aerial vehicles in GPS-denied environment. *Drones*. 2025;9(11):764. https://doi.org/10.3390/drones9110764.
14) Fares H, Mohanna A, Kaddouh B. Real-time elevation and orientation-aware visual localization for GNSS-denied drone navigation. *Drones*. 2026;10(6):445. https://doi.org/10.3390/drones10060445.
15) Khattak S, Papachristos C, Alexis K. Vision-Depth Landmarks and Inertial Fusion for Navigation in Degraded Visual Environments. In: Advances in Visual Computing – 13th International Symposium, ISVC 2018;vol. 11241 of Lecture Notes in Computer Science. Cham. Springer. 2018;p. 529–540. 10.1007/978-3-030-03801-4_46.

16) Hong S, Shyam P, Bangunharcana A, Shin H. Robotic mapping approach under illumination-variant environments at planetary construction sites. *Remote Sensing*. 2022;14(4):1027. https://doi.org/10.3390/rs14041027.
17) Nagarajan N, Jothiraj S. An innovative runway landing path detection using UAV implementation of the K-means clustering algorithm. *Indian Journal of Science and Technology*. 2024;17(15):1527–1534. https://doi.org/10.17485/IJST/v17i15.2495.
18) Tyagi A, Gaur C. SLAM-based navigation and fault resilience in a surveillance quadcopter with embedded vision systems. *arXiv preprint arXiv:250415305*. 2025. https://doi.org/10.48550/arXiv.2504.15305.
19) Sadeghi Esfahlani S, Sanaei A, Ghorabian M, Shirvani H. The deep convolutional neural network role in the autonomous navigation of mobile robots (SROBO). *Remote Sensing*. 2022;14(14):3324. https://doi.org/10.3390/rs14143324.
20) Nemire B. NVIDIA researchers release trailblazing deep learning-based framework for autonomous drone navigation. *NVIDIA Technical Blog*. 2017. Available from: https://developer.nvidia.com/blog/nvidia-researchers-release-trailblazing-deep-learning-based-framework-for-autonomous-drone-navigation.
21) Karatzas A, Karras A, Karras CN, Giotopoulos KC, Oikonomou K, Sioutas S. On Autonomous Drone Navigation Using Deep Learning and an Intelligent Rainbow DQN Agent. In: Intelligent Data Engineering and Automated Learning – IDEAL 2022;vol. 13756 of Lecture Notes in Computer Science. Cham. Springer. 2022;p. 134–145. 10.1007/978-3-031-21753-1_14.
22) Giubilato R, Stürzl W, Wedler A, Triebel R. Challenges of SLAM in extremely unstructured environments: the DLR Planetary Stereo, Solid-State LiDAR, Inertial Dataset. *IEEE Robotics and Automation Letters*. 2022;7(4):8721–8728. https://doi.org/10.1109/LRA.2022.3188118.
23) Epp CD, Robertson EA, Brady T. Autonomous Landing and Hazard Avoidance Technology (ALHAT). In: 2008 IEEE Aerospace Conference. IEEE. 2008;p. 1–7. 10.1109/AERO.2008.4526297.
24) Hayner CR, Buckner SC, Broyles D, Madewell E, Leung K, Açıkmeşe B. HALO: Hazard-Aware Landing Optimization for Autonomous Systems. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2023;p. 3261–3267. 10.1109/ICRA48891.2023.10160655.