

A Lightweight Modular Framework for Onboard Monocular Multi-View Satellite Imagery Analysis

Alberto Candela¹, Joao Passos^{1,2}, Jiho Lee^{1,3}, Evan Davis¹, Federico Rossi¹, Steve Chien¹,
Becca Bonham-Carter⁴, Luis Chavier⁴, Alexis Pascual⁴, Tim Heydrich⁴, Shannon Paul⁴,
Saleh Abdelrahman⁴, Andrew Macdonald⁴, and Michele Faragalli⁴

Abstract—Modern Low Earth Orbit (LEO) satellites can now execute AI workloads in orbit, turning raw sensor data into immediate, actionable insights. However, current onboard AI predominantly relies on single-image tasks, failing to exploit the spatiotemporal motion and geometric data inherent in multi-view imagery. Because traditional multi-view photogrammetry is computationally restricted to ground-based processing, we present a lightweight modular framework tailored for onboard monocular multi-view analysis. Our pipeline decouples 2D motion estimation—benchmarking classical and learning-based optical flow—from sequential 3D reconstruction, bypassing common heavy computational bottlenecks. Serving as an initial feasibility study, we validate our algorithmic design using real-world flight sequences from Mission Persistence, paving the way for upcoming in-orbit software integration and deployment.

I. INTRODUCTION

Recent developments in the space sector are bringing new capabilities to orbit through the proliferation of Low Earth Orbit (LEO) platforms, which improve spatiotemporal coverage while lowering observation costs. Driven by powerful Commercial Off-The-Shelf (COTS) processors, modern satellites can now transition from traditional “store-and-forward” downlinking toward onboard edge computing. This shift allows spacecraft to execute AI workloads directly in orbit [1], [2], [3], immediately converting raw sensor data into actionable knowledge at the source.

Despite these satellite hardware advancements, state-of-the-art onboard AI predominantly focuses on single-image 2D analysis tasks like scene classification, detection, and segmentation. These static methods fail to exploit the crucial motion and geometric information available across sequential multi-view imagery. Traditional multi-view photogrammetry remains bound to ground-based processing due to prohibitive computational requirements or a heavy reliance on large external datasets like global Digital Elevation Models (DEMs).

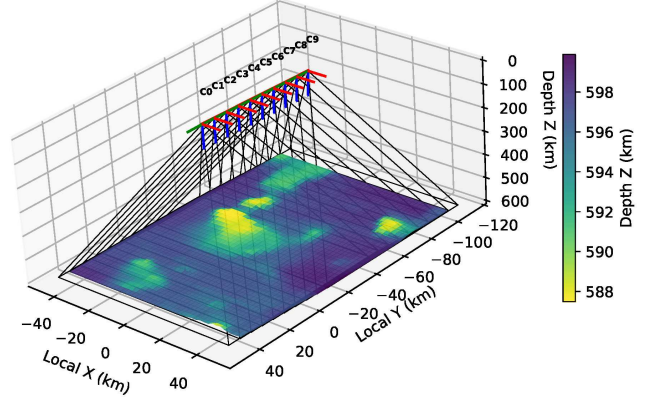


Fig. 1: Monocular 3D reconstruction from an orbital overpass. This example shows our framework’s ability to resolve cloud heights by leveraging orbital parallax on a Mission Persistence flight image sequence, targeting future in-orbit deployment for real-time onboard analysis.

In-orbit multi-view analysis could unlock vital spatiotemporal insights at the edge—allowing spacecraft to immediately characterize dynamic volumetric phenomena like clouds or volcanic plumes, and support disaster response or real-time autonomous navigation, where decisions must be made in seconds rather than days.

To address this gap, we present a lightweight modular framework for monocular multi-view satellite imagery analysis (Fig. 1). Our focus is on the algorithmic design required to execute multi-view geometry under strict orbital compute constraints. This work serves as an initial design and feasibility study before flight, paving the way for our method’s deployment on Mission Persistence, a satellite testbed for AI flight software [4], as well as other satellites with advanced onboard compute. Our core technical contributions are:

- We present a modular framework that decouples 2D motion estimation from 3D scene reconstruction, bypassing heavy computational and data bottlenecks to ensure efficiency.
- We perform a systematic evaluation of several classical and learning-based dense optical flow methods for 2D

*© 2026. All rights reserved.

¹A. Candela, E. Davis, Federico Rossi, and S. Chien are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA {FirstName.LastName}@jpl.nasa.gov

²J. Passos is with Carnegie Mellon University, Pittsburgh, Pennsylvania, USA joaopassos@cmu.edu

³J. Lee is with the University of Colorado, Boulder, Colorado, USA jiho.lee@colorado.edu

⁴B. Bonham-Carter, L. Chavier, A. Pascual, T. Heydrich, S. Paul, S. Abdelrahman, A. Macdonald, and M. Faragalli are with Mission Control, Ottawa, Ontario, Canada {FirstName or LastName}@missioncontrolspace.com

motion estimation.

- We introduce a geometric reconstruction pipeline that supports both a vision-only mode and a telemetry-refined hybrid mode to handle variable data availability.
- We validate the framework’s capabilities using real-world data from flight sequences collected by Mission Persistence across diverse conditions.

II. RELATED WORK

A. Edge AI on EO Satellites

Onboard autonomy for EO satellites was pioneered by the Autonomous Sciencecraft Experiment (ASE) on the Earth Observing One (EO-1) mission [5], which utilized flight software to detect and respond to dynamic science events but was constrained by legacy hardware limitations. Modern platforms leverage powerful Commercial Off-The-Shelf (COTS) AI processors running Linux environments to execute intensive workloads directly in orbit. Key experimental milestones include ESA’s Φ -sat-1, which utilized a Vision Processing Unit (VPU) to validate CNN-based cloud filtering (*CloudScout*), wildfire detection, and land cover classification [1], and the ESA OPS-SAT mission for onboard TensorFlow Lite image classification [2].

Onboard satellite autonomy is transitioning from passive data filtering toward active decision-making. Orbital Edge Computing (OEC) established a framework enabling deep learning workloads to be parallelized across nanosatellite constellations to process data at the source [6]. The CogniSAT-6 mission [3] demonstrated the concept of “dynamic targeting” [7], employing an onboard CNN to identify cloud-free zones and autonomously reorient the spacecraft to optimize image acquisition [8].

B. Ground-Based Multi-View Photogrammetry

Multi-view photogrammetry remains predominantly ground-bound. Standard pipelines rely on dense matching algorithms like Semi-Global Matching (SGM) or Multi-View Stereo (MVS) methods, which impose prohibitive memory requirements and computational costs at the edge, or depend on high-resolution terrestrial DEMs. Multi-view missions such as NASA’s Multi-angle Imaging SpectroRadiometer (MISR) [9] downlink large raw radiance datasets for terrestrial processing to retrieve geometric attributes like cloud-top heights and volcanic plume dynamics [10]. This reliance on ground processing persists in modern commercial constellations like Maxar’s WorldView [11] and Airbus’s Pléiades Neo [12] for high-resolution 3D mapping.

C. Learning-Based Multi-View Analysis

State-of-the-art approaches are transitioning from traditional photogrammetry to learning-based methods. The Visual Geometry Grounded Transformer (VGGT) [13] leverages a large-scale transformer backbone to directly infer 3D camera poses and depth maps in a single forward pass. Deep learning has also redefined dense correspondence and motion estimation, spanning from earlier CNN architectures like FlowNet2 [14], PWC-Net [15], LiteFlowNet2 [16], and

IRR-PWC [17], to robust recurrent models like RAFT [18] and FlowFormer++ [19].

In the satellite domain, learning-based optical flow has been investigated for low-latency onboard tasks, including autonomous lunar landing navigation [20] and visual-inertial odometry over featureless terrain [21], offering superior robustness against low-contrast and repetitive orbital textures. Conversely, deep satellite MVS architectures such as SP-MVS [22] and DC-SatMVS [23] preserve sharp visual boundaries like building outlines but remain incompatible with edge processors due to their massive memory footprints and high power and compute demands.

D. Onboard Multi-View Analysis

In-orbit multi-view analysis remains limited. The OPS-SAT Multi-Frame Super Resolution experiment performed multi-view fusion for 2D image enhancement and compression but did not recover motion or 3D geometry [24]. Onboard stereo vision for On-Orbit Servicing and Manufacturing (OSAM) utilizes CNNs for rigid-body pose estimation against high-contrast backgrounds [25], but requires known target models for proximity operations rather than EO. The StereoBit project demonstrated 3D cloud height retrieval on a SpaceCube v3.0 Mini testbed rather than in orbit [26]; additionally, its FPGA-accelerated sparse matching fails to generate continuous data in textureless regions and lacks the flexibility of software-defined pipelines.

The upcoming Multi-view Onboard Computational Imager (MOCI) mission [27] represents a key effort in orbital 3D reconstruction, utilizing an NVIDIA Jetson TX2i to implement a classical Structure-from-Motion (SfM) pipeline via the custom SSRLCV library [28]. This yet-to-be-flown system relies on common computationally intensive operations to generate 3D products.

III. METHODOLOGY OVERVIEW

The proposed framework addresses the computational and data bottlenecks inherent in traditional photogrammetry. Conventional Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipelines rely on resource-intensive operations—such as sparse feature matching, global bundle adjustment, and intensive interpolation for mesh generation—which often exceed the strict compute budgets of small satellites. To overcome this, we utilize a direct pipeline that operates on low-level data. This design processes imagery without requiring prior geographic corrections and is compatible with varying degrees of radiometric calibration.

By treating the dense optical flow field as a continuous correspondence map, we follow geometric principles for recovering monocular 3D structure from motion [29]. This pipeline leverages dense matches to generate a semi-dense 3D point cloud, balancing geometric fidelity with edge efficiency. We adopt this hybrid vision-geometric workflow over a fully end-to-end learning approach due to the scarcity of high-resolution 3D ground truth for dynamic atmospheric phenomena in LEO imagery. Decoupling 2D flow estimation from 3D reconstruction ensures the framework remains

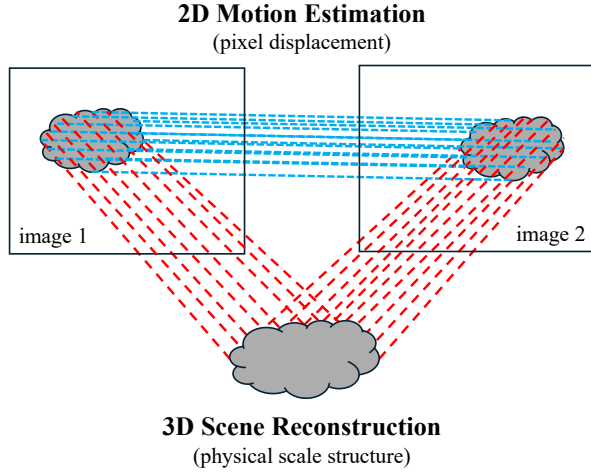


Fig. 2: Pixel displacements tracked via 2D motion estimation are triangulated to reconstruct the 3D scene geometry and resolve physical scale.

interpretable, enforces physical constraints, and allows for algorithm-agnostic flexibility depending on available on-board compute (Fig. 2).

The framework is a modular, sequential pipeline that consists of six steps: a) image preprocessing, b) 2D motion estimation (via dense optical flow), c) camera parameter estimation (intrinsic and extrinsic), d) correspondence outlier filtering, e) point cloud triangulation, and f) point cloud postprocessing. The technical description is divided into two sections: Section IV details the 2D motion estimation phase (steps a–b), while Section V delineates the 3D scene reconstruction process (steps c–f). Following this methodology, Section VI introduces Mission Persistence, as well as the datasets and experiments used to evaluate the framework. The results are presented and analyzed in Section VII.

IV. 2D MOTION ESTIMATION

A. Image Preprocessing

First, raw digital numbers are converted to radiometric or color values. To ensure a pinhole projection, non-linear lens distortions (e.g., radial or tangential) may need to be rectified. Finally, for deep learning architectures, images are normalized and reshaped to meet model input requirements.

B. Dense Optical Flow

We leverage the concept of optical flow to perform monocular 2D motion estimation. Given a pixel at $[u, v]^T$ in a frame, the objective is to determine its coordinates $[u', v']^T$ in the subsequent frame using a flow vector $[\Delta u, \Delta v]^T$:

$$[u', v']^T = [u, v]^T + [\Delta u, \Delta v]^T. \quad (1)$$

Traditional optical flow estimation relies on assumptions such as brightness constancy and spatial smoothness to describe pixel displacement. Classical approaches often struggle with geometric distortions and large displacements, which are common in orbital imagery.

Sparse optical flow methods track a limited number of keypoints, often followed by interpolation, while dense methods compute this displacement for every pixel at the outset. We adopt a dense estimation approach to ensure a high-fidelity representation of the observed scene. Sparse feature tracking may lead to holes and miss critical geometric details. By computing a flow vector for every pixel, we provide a more complete dataset for subsequent geometric estimation and maintain robustness in low-texture areas where keypoint detectors often fail to find matches.

Recent advancements in deep learning have redefined dense estimation by utilizing hierarchical feature learning to handle complex lighting and non-rigid deformations. The emergence of relatively lightweight architectures offers a viable path for deployment within the power and memory constraints of the orbital edge. In this work, we compare classical methods against learning-based models.

1) *Classical Dense Methods*: We evaluate the following suite of dense optical flow methods available in the OpenCV library [30]: DIS (Fast), Farneback, SimpleFlow, DeepFlow, SparseToDense, RLOF, DualTVL1, and PCAFlow. These algorithms vary significantly in runtime and estimation quality.

2) *Learning-Based Models*: We evaluate six lightweight architectures: FlowNet2 [14], PWC-Net [15], LiteFlowNet2 [16], IRR-PWC [17], and both the large and small variants of RAFT [18]. To assess their out-of-the-box utility for orbital imagery, all models are used with off-the-shelf pretrained weights without additional fine-tuning. All architectures are leveraged from existing PyTorch implementations.

V. 3D SCENE RECONSTRUCTION

The geometric formulations and mathematical notation utilized throughout this section build on the standard principles of multi-view geometry [31]. Our method is formulated for framing cameras, which capture a 2D image simultaneously and can be described using the standard pinhole camera model. We adopt a quasi-static scene assumption, where the displacement in the dense correspondence field is attributed entirely to the geometric parallax induced by the spacecraft's orbital velocity. This simplification is highly effective for LEO platforms with high-cadence imaging (e.g., 1 Hz); because the orbital velocity (≈ 7.5 km/s) is orders of magnitude greater than typical atmospheric advection (10–50 m/s), independent motion becomes effectively negligible within the short temporal baselines between consecutive frames. This allows the 3D structure to be resolved through direct geometric triangulation, avoiding the computational overhead of more complex spatiotemporal modeling.

A. Camera Parameter Estimation

The intrinsic camera matrix K is defined by the focal lengths (f_x, f_y) and the camera center (c_x, c_y), all in pixels. These focal lengths can be estimated using the camera's horizontal and vertical Field of View ($\text{FOV}_h, \text{FOV}_v$) alongside the image width W and height H :

$$f_x = \frac{W/2}{\tan(\text{FOV}_h/2)}, \quad f_y = \frac{H/2}{\tan(\text{FOV}_v/2)}. \quad (2)$$

Many imaging systems have square pixels, such that $f = f_x = f_y$. Alternatively, given the platform altitude A and the Ground Sample Distance (GSD), the focal length can be approximated as $f \approx A/\text{GSD}$. The intrinsic matrix is:

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where $s \approx 0$ is a skew or shear parameter.

The extrinsic parameters, given by the rotation matrix R and translation vector $\mathbf{t}$, define the camera's pose. The full projection matrix P is formulated as:

$$P = K[R \mid \mathbf{t}]. \quad (4)$$

This matrix maps a 3D world point in homogeneous coordinates $\mathbf{X} = [x, y, z, w]^T$ to its corresponding 2D pixel coordinates $\mathbf{u} = [u, v, 1]^T$. Since the projection is defined up to a scale factor, the relationship is expressed as:

$$\lambda \mathbf{u} = P\mathbf{X}, \quad (5)$$

where λ is the scalar projective depth.

Our framework performs extrinsic parameter estimation based on data availability, using one of two options:

1) *Vision-Only (Nominal Scale)*: This mode assumes that satellite pose data is unavailable. Optical flow or sparse feature matching is used to measure the median pixel displacement between sequential images. To establish a metric scale, this displacement is mapped to a physical ground arc length using the nominal GSD. Under the assumption of a simplified circular orbital model and a spherical Earth—necessitated by the lack of an absolute global location—this ground arc is converted into an angular displacement along the orbital path, which remains a highly valid approximation for Low Earth Orbit (LEO) satellites. This angular step is then used to geometrically derive and iteratively update the current rotation R and translation $\mathbf{t}$, as the framework operates in a reference frame defined with respect to the first frame of the sequence. While this approach captures sensor disturbances and preserves relative geometry, relying on a fixed nominal altitude rather than the actual time-varying value introduces some scale uncertainty and altitude drift.

2) *Hybrid (Telemetry-Refined Scale)*: The Hybrid mode fuses optical flow with orbital telemetry to improve scale without compromising geometric stability. Median pixel displacement is still computed via optical flow. However, orbital telemetry—whether from direct onboard Global Navigation Satellite Systems (GNSS) measurements or propagated via Two-Line Elements (TLE) models—provides the instantaneous camera altitude, geodetic latitude, and corresponding GSD. The pixel displacement is scaled by this true GSD to calculate the physical ground arc traversed by the satellite. Because the absolute global position is known, the framework replaces the spherical approximation with the WGS84 Earth model. Using the WGS84 ellipsoid at the current latitude, the framework computes the local Meridian and Prime Vertical radii of curvature. The ground arc is then converted into angular rotations that track the local geodetic surface

normal. Combined with the telemetry-derived orbital radius, these angles define the current rotation R and translation $\mathbf{t}$. This fuses the absolute scale of telemetry with the geometric consistency of optical flow.

B. Correspondence Outlier Filtering

Prior to computing 3D points, we filter poor 2D correspondences using the *epipolar constraint*. Given the known camera intrinsics K , alongside the poses $(R_1, \mathbf{t}_1)$ and $(R_2, \mathbf{t}_2)$, we compute the relative rotation $R_{21} = R_2 R_1^T$ and translation $\mathbf{t}_{21} = \mathbf{t}_2 - R_2 R_1^T \mathbf{t}_1$. These define the Essential matrix $E = [\mathbf{t}_{21}]_{\times} R_{21}$ (where $[\mathbf{t}_{21}]_{\times}$ denotes the skew-symmetric matrix of the translation vector), allowing us to explicitly construct the Fundamental matrix F , bypassing iterative methods such as RANSAC:

$$F = K^{-T} \underbrace{([\mathbf{t}_{21}]_{\times} R_{21})}_E K^{-1}. \quad (6)$$

For a candidate pixel match $\mathbf{u}_1 \leftrightarrow \mathbf{u}_2$, the corresponding epipolar line in the second view is given by $\mathbf{l}_2 = F\mathbf{u}_1$. We evaluate the geometric consistency of the correspondence by calculating the perpendicular point-to-line distance d :

$$d(\mathbf{u}_2, \mathbf{l}_2) = \frac{|\mathbf{u}_2^T F \mathbf{u}_1|}{\sqrt{(F\mathbf{u}_1)_1^2 + (F\mathbf{u}_1)_2^2}}. \quad (7)$$

Matches where d exceeds a pixel threshold τ are discarded.

C. Point Cloud Triangulation

Using pixel disparities from Section IV, we reconstruct 3D points through sequential pair-wise triangulation. The framework ensures onboard efficiency while maintaining metric consistency by leveraging known mission parameters (such as GSD and altitude) to anchor the geometry, bypassing the need for computationally expensive global bundle adjustment. To resolve the triangulation, the scale factor λ in Eq. 5 is eliminated by rewriting the relationship as a homogeneous linear system $A\mathbf{X} = 0$ via the cross product $\mathbf{u} \times (P\mathbf{X}) = 0$. For a pixel (u, v) , this yields:

$$\begin{aligned} u(\mathbf{p}_3^T \mathbf{X}) - (\mathbf{p}_1^T \mathbf{X}) &= 0 \\ v(\mathbf{p}_3^T \mathbf{X}) - (\mathbf{p}_2^T \mathbf{X}) &= 0 \end{aligned} \quad (8)$$

where $\mathbf{p}_i^T$ denotes the i -th row of P . For $n \geq 2$ views, we stack these equations into a matrix $A \in \mathbb{R}^{2n \times 4}$:

$$A = \begin{bmatrix} u_1 \mathbf{p}_{3,1}^T - \mathbf{p}_{1,1}^T \\ v_1 \mathbf{p}_{3,1}^T - \mathbf{p}_{2,1}^T \\ \vdots \\ u_n \mathbf{p}_{3,n}^T - \mathbf{p}_{1,n}^T \\ v_n \mathbf{p}_{3,n}^T - \mathbf{p}_{2,n}^T \end{bmatrix} \quad (9)$$

To resolve the 3D scene geometry, we utilize the Direct Linear Transformation (DLT) algorithm, solving for homogeneous world coordinates $\mathbf{X}$ via Singular Value Decomposition (SVD). The reconstructed 3D point corresponds to the singular vector $\mathbf{X}$ associated with the smallest singular value of A . As $\mathbf{X} = [x, y, z, w]^T$ is in homogeneous form, the final

3D coordinates $\mathbf{X}_{3D}$ are recovered by normalizing the vector by its fourth component:

$$\mathbf{X}_{3D} = \left[\frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right]^T. \quad (10)$$

D. Point Cloud Postprocessing

To refine the 3D point cloud generated through triangulation, we implement a post-processing stage based on voxel-grid filtering and statistical outlier removal. Specifically, we utilize a spatial binning approach where the 3D space is discretized into fixed-size voxels. By computing spatial statistics within each bin, we remove noise from residual matching errors while consolidating redundant points and pruning low-density clusters that likely represent triangulation artifacts. This process reduces the total point count, optimizing the data for storage and downlink. Furthermore, spatial interpolation techniques can be applied to fill remaining gaps in the point cloud. To accommodate onboard compute limitations, this densification stage can be optionally deferred to ground-based processing, depending on the available downlink bandwidth and mission requirements.

VI. EXPERIMENTS

A. Mission and Dataset

Mission Persistence [4] is a 6U CubeSat built on Spire’s LEMUR-2 bus (NORAD ID 64561). It launched on June 23, 2025, into a sun-synchronous orbit at an altitude of approximately 597 km. The mission is specifically designed as an orbital testbed for hosting and validating machine learning payloads on embedded hardware. Its Marsupial computing module uses an AMD Zynq UltraScale+ Multi-Processor System-on-Chip (MPSoC), specifically the ZU4CG-1E model.

The satellite carries the Gecko imager, a framing camera that captures snapshots with a 100 km swath and a GSD of 48 m. In this work, we utilize 25 flight sequences collected between late 2025 and early 2026. This dataset encompasses a diverse range of conditions, ranging from dense cloud cover to clear-sky terrestrial scenes. Each sequence consists of ten nadir-pointing frames captured at approximately 1 Hz. For each frame, the satellite’s position was estimated by propagating TLE sets. The original image resolution is 2048×2048 pixels, which we resized to 512×512 to improve computational efficiency while preserving visual details. Camera intrinsics were calculated using the Gecko imager specifications.

B. Experimental Setting

The framework’s performance is evaluated across two sets of experiments: 1) 2D motion estimation and 2) 3D scene reconstruction. We focus on benchmarking motion and geometric accuracy rather than memory or power utilization, as these metrics are highly dependent on specific flight hardware and software implementations. This approach ensures our findings regarding algorithmic performance remain applicable across various high-performance LEO compute

platforms. The experiments were conducted on a MacBook Pro 16 (M2 Max Processor, 64 GB RAM).

This first set of experiments evaluates a diverse suite of classical and deep learning-based methods described in Section III. Flow accuracy directly impacts 3D reconstruction fidelity. Performance is quantified—across all dense methods and the Lucas-Kanade baseline [32]—using three metrics:

- **SIFT Correspondence Error:** Flow accuracy against high-confidence sparse SIFT keypoints [33], expressed as the average Endpoint Error (EPE) (L_2 error).
- **Forward-Backward (FB) Consistency Error:** Reversibility of flow estimation (L_2 error).
- **Runtime:** Time to process each image pair.

The first two metrics are complementary: accurate SIFT matching may still fail the FB check in low-texture regions.

The second set of experiments evaluates the 3D reconstruction pipeline using spatial binning (192 m/pixel to maintain consistency with the downsampled 48 m GSD $\times$ 4 resolution) and linear interpolation during point cloud post-processing. Geometric outlier filtering applies an epipolar constraint threshold $\tau = 3$ pixels. We benchmark a selection of high-performing pipelines—the sparse SIFT baseline, the top classical dense algorithm, and the leading deep learning model—across both Vision-Only and Hybrid extrinsic estimation configurations detailed in Section V-A. Direct 3D validation for dynamic phenomena is exceptionally challenging; high-cadence geostationary platforms lack the fine spatial resolution of LEO assets, whereas high-resolution alternatives suffer from infrequent revisit rates that miss transient atmospheric events. Although Mission Persistence lacks active onboard depth sensors (e.g., LiDAR) to provide direct ground-truth profiles, we validate our pipeline by comparing a subset of clear-sky terrestrial sequences against the Copernicus global DEM (30 m/pixel). We use AstroLoc [34], [35] and visual feature matching to precisely align the scenes with Sentinel-2 reference imagery, enabling a fair topographic evaluation. Most metrics are computed across the entire 25-sequence dataset, whereas the topographic DEM error is evaluated over 4 clear-sky terrestrial scenes:

- **Epipolar Outlier Rate:** Percentage of feature matches rejected for violating epipolar constraints, measuring relative geometric consistency.
- **Reprojection Error:** Pixel-wise distance between original and reprojected features to quantify internal triangulation stability.
- **Geodetic Altitude Error:** Altitude mean absolute error (MAE) relative to the uncalibrated WGS84 ellipsoid surface, capturing raw vertical telemetry drift.
- **Topographic DEM Error:** Localized terrain Relative MAE computed against the Copernicus DEM following relative vertical alignment.
- **Processing Time:** Execution duration divided between main pipeline compute and optional point cloud post-processing for a ten-image sequence.

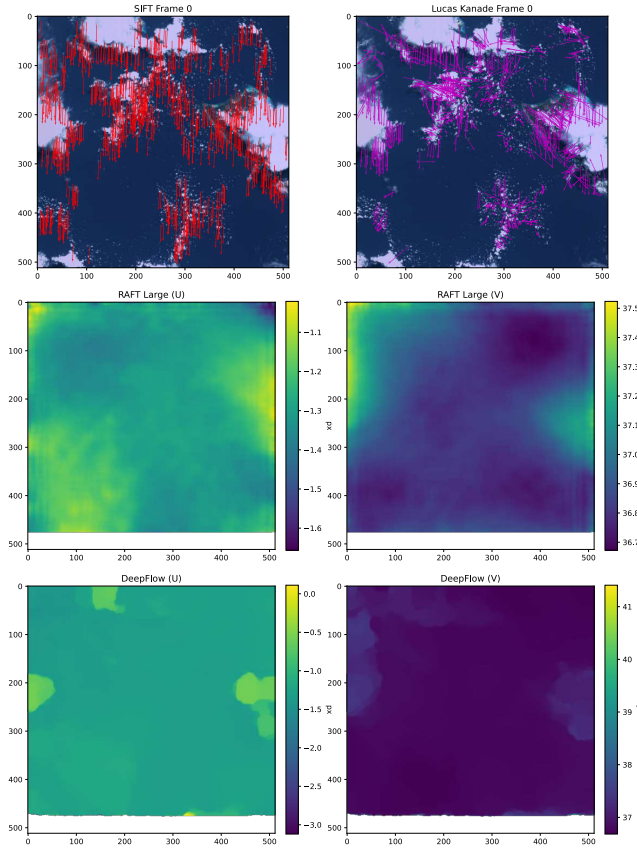


Fig. 3: Optical flow estimation on an image sequence. Top: SIFT and Lucas-Kanade correspondences. Middle: RAFT deep learning model. Bottom: DeepFlow classical algorithm.

VII. RESULTS AND DISCUSSION

A. Results: Optical Flow Estimation

Results indicate that dense optical flow effectively captures general motion trends while filling the data gaps inherent in sparse methods (Fig. 3). Subtle variations in the displacement unveil features such as clouds, especially along the pixel columns axis due to motion parallax effects. These features are instrumental in later resolving 3D structure.

Our evaluation reveals a clear performance divide between classical and deep learning approaches in this orbital context (Table I). As expected, Lucas-Kanade is the fastest but suffers from low accuracy due to large pixel displacements in the orbital sequences. Among dense classical methods, DeepFlow is the accuracy leader with the lowest errors, while DIS (Fast) provides the best runtime with competitive accuracy. Conversely, simpler baselines like Farneback and SimpleFlow yielded much larger errors. Notably, all deep learning architectures achieved sub-pixel accuracy across both metrics, despite being used “zero-shot”. Domain-specific fine-tuning could yield even higher precision. While RAFT Large consistently achieves great estimation quality, it is also the slowest, highlighting a critical trade-off where model selection must be driven by specific mission constraints.

Ultimately, the best-performing classical and deep learning

TABLE I: Aggregated Optical Flow Performance Results (Mean $\pm$ Std. Dev.) on the Persistence Dataset

Method	SIFT (px)	FB Err (px)	Runtime (s)
<i>Classical Methods</i>			
Lucas-Kanade	26.34 ± 27.29	29.18 ± 1.47	0.006 ± 0.000
DIS (Fast)	0.208 ± 0.204	0.253 ± 0.04	0.024 ± 0.001
Farneback	4.966 ± 12.44	7.173 ± 0.40	0.046 ± 0.001
SparseToDense	0.209 ± 0.203	0.091 ± 0.01	0.065 ± 0.003
PCAFLOW	0.778 ± 1.884	1.050 ± 0.13	0.079 ± 0.023
RLOF	0.211 ± 0.205	0.135 ± 0.02	0.252 ± 0.007
SimpleFlow	28.42 ± 6.985	3.119 ± 0.31	0.579 ± 0.012
DeepFlow	0.199 ± 0.202	0.074 ± 0.01	1.035 ± 0.006
DualTVL1	31.66 ± 8.781	5.430 ± 0.69	14.10 ± 0.387
<i>Deep Learning</i>			
RAFT Small	0.268 ± 0.206	0.144 ± 0.00	0.873 ± 0.015
RAFT Large	0.234 ± 0.202	0.097 ± 0.01	2.385 ± 0.031
LiteFlowNet2	0.234 ± 0.201	0.280 ± 0.01	0.758 ± 0.024
IRR-PWC	0.227 ± 0.199	0.127 ± 0.01	4.913 ± 0.132
PWC-Net	0.237 ± 0.200	0.137 ± 0.01	0.768 ± 0.018
FlowNet2	0.321 ± 0.198	0.268 ± 0.02	2.284 ± 0.040

models generally achieved subpixel accuracy. Choosing a method depends on the limitations of a particular compute system on a mission. For instance, selecting a classical method like DeepFlow depends on the overhead of integrating or reimplementing non-standard OpenCV libraries into flight software. Similarly, deep learning adoption hinges on the ease of porting and optimizing specific architectures to the target environment and the availability of specialized hardware, such as GPUs, to handle the computational load.

B. Results: 3D Scene Reconstruction

The 3D reconstruction results across the Mission Persistence dataset demonstrate high fidelity and geometric consistency (Fig. 4). While sparse methods like SIFT anchor the geometry, they leave significant spatial gaps requiring extensive interpolation. In contrast, dense approaches maintain superior structural continuity. DeepFlow yields sharper boundaries but increased outlier noise, whereas RAFT produces a smoother surface that is attributable to its iterative CNN downsampling layers. These results confirm the framework’s capacity to recover complex scene structures.

As summarized in Table II, the hybrid mode resolves scale ambiguity and minimizes global altitude drift compared to the vision-only mode while maintaining relative geometric consistency. Dense methods outperform the sparse baseline, yielding sub-kilometer topographic errors and tighter reprojection errors. By tracking continuous motion fields rather than isolated features, dense approaches smooth localized matching noise to minimize geometric jitter, although with higher processing times. Overall, DeepFlow represents the best balance of accuracy and efficiency for this edge pipeline; despite a slightly higher outlier rate than RAFT, it matches its physical stability while achieving lower reprojection error with lower latency. Meanwhile, RAFT’s zero-shot performance remains a robust alternative whose accuracy could be further improved through domain-specific fine-tuning and hardware-level GPU optimization.

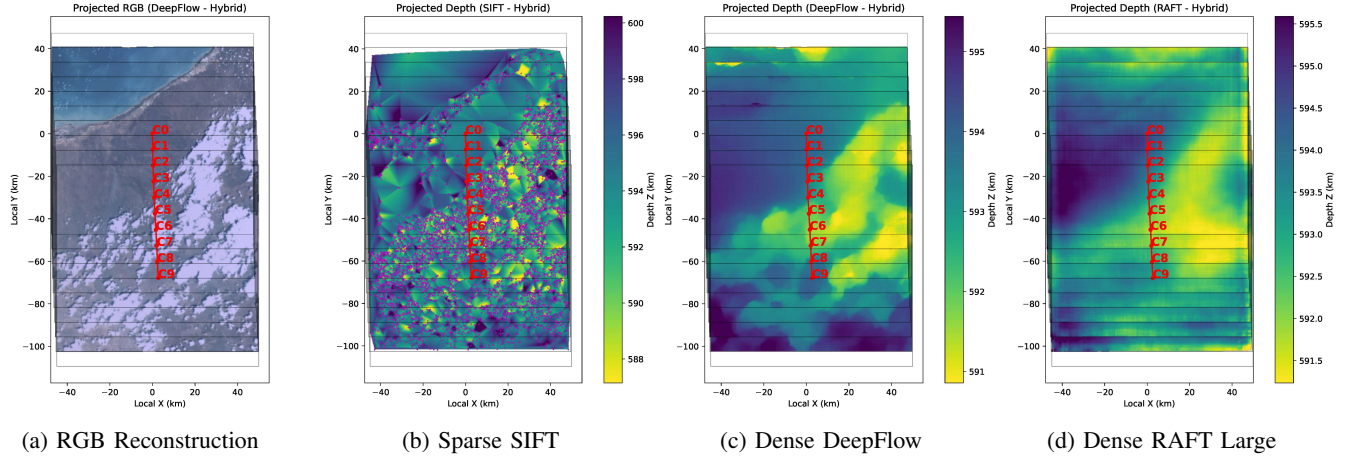


Fig. 4: Comparison of 3D scene reconstruction results for a scene in Northeast Australia (07 December 2025). These projections illustrate the framework’s capacity to recover visual and volumetric structures. Note that while sparse methods (i.e., SIFT) leave geometric gaps requiring interpolation, dense approaches maintain structural continuity. DeepFlow generates sharper feature boundaries but introduces more outliers, whereas RAFT produces a smoother surface.

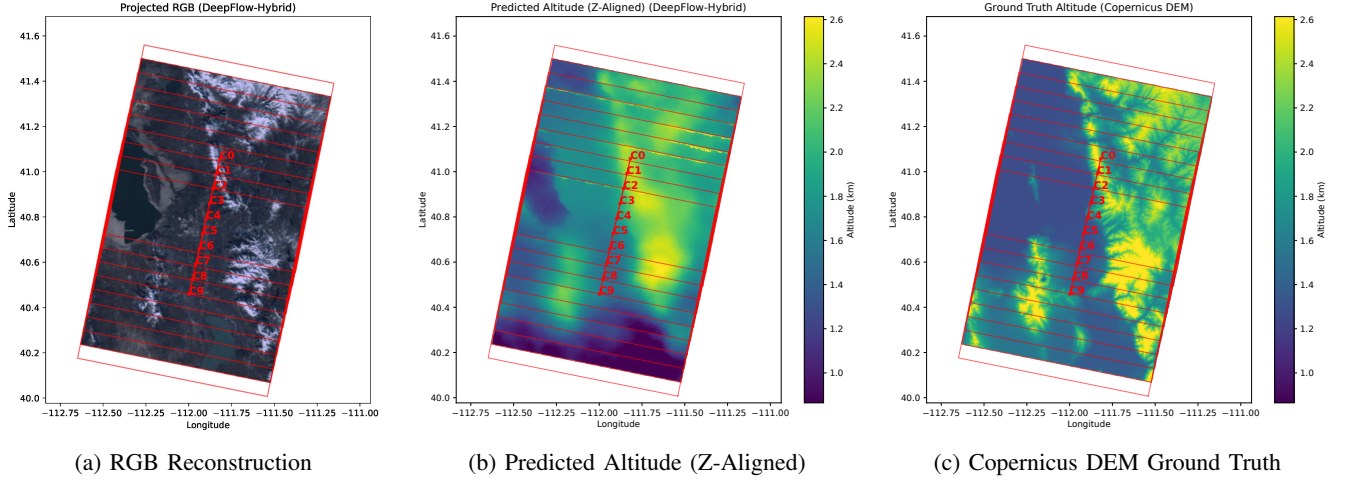


Fig. 5: Topographic validation over the Wasatch Range, Utah (18 January 2026). The DeepFlow-Hybrid reconstruction balances a smooth, lower-resolution profile with strong structural alignment against the Copernicus DEM ground truth.

TABLE II: Aggregated 3D Reconstruction Results (Mean $\pm$ Std. Dev.) on the Persistence Dataset

Method	Approach	Outliers (%)	Reprojection Error (px)	Geo. Alt. Error MAE (km)	DEM MAE (m)	Proc. Time (s)	Post. Time (s)
SIFT	Vision-Only	3.58 ± 2.71	0.235 ± 0.053	16.60 ± 0.21	2425.04 ± 154.72	0.63 ± 0.14	0.04 ± 0.01
	Hybrid	3.59 ± 2.72	0.235 ± 0.053	2.96 ± 0.99	2049.71 ± 160.78	0.62 ± 0.14	0.03 ± 0.01
DeepFlow	Vision-Only	4.60 ± 8.18	0.092 ± 0.071	16.64 ± 0.51	562.80 ± 238.25	6.85 ± 0.18	3.11 ± 0.31
	Hybrid	4.61 ± 8.18	0.092 ± 0.071	2.60 ± 1.28	460.27 ± 196.54	6.84 ± 0.18	2.95 ± 0.30
RAFT	Vision-Only	1.89 ± 6.53	0.112 ± 0.075	16.67 ± 0.34	576.44 ± 231.22	12.85 ± 0.50	3.18 ± 0.31
	Hybrid	1.89 ± 6.54	0.112 ± 0.075	2.44 ± 1.08	478.00 ± 193.40	12.83 ± 0.51	3.03 ± 0.27

VIII. CONCLUSIONS AND FUTURE WORK

This paper presents a lightweight modular framework for onboard monocular multi-view satellite imagery analysis, integrating 2D motion estimation with 3D scene reconstruction. Evaluation using Mission Persistence flight sequences confirms two key findings. First, top-performing classical and deep learning dense optical flow methods achieve sub-pixel accuracy without fine-tuning. Second, the generated

3D reconstructions recover volumetric structures with superior quality and density over sparse SIFT baselines, while achieving sub-kilometer topographic accuracy. Additionally, the hybrid mode leverages telemetry to refine metric scale while retaining geometric consistency. Executing in seconds without external landmarks or global DEMs, the framework demonstrates autonomy and suitability for onboard deployment. Consequently, these results are encouraging for the future development and flight integration of this framework.

Future work will adapt the framework for varied sensor geometries, such as pushbroom scanning, while extending capabilities to include joint 3D position and velocity estimation of scene targets. We also plan to conduct empirical evaluations of the pipeline on representative AI flight processors to profile hardware performance. Ultimately, these efforts aim to realize a technical demonstration on Mission Persistence and other next-generation edge-compute satellites.

ACKNOWLEDGMENT

This work was supported by the Earth Science Technology Office (ESTO), National Aeronautics and Space Administration (NASA). A portion of this work was conducted by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA (80NM0018D0004). Mission Control acknowledges the support of the Canadian Space Agency [SIMA03P2].

REFERENCES

- [1] G. Giuffrida *et al.*, “The phi-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.
- [2] G. Labrèche *et al.*, “Ops-sat spacecraft autonomy with tensorflow lite, unsupervised learning, and online machine learning,” in *Proc. IEEE Aerosp. Conf.* Big Sky, MT, USA: IEEE, 2022, pp. 1–17.
- [3] D. Rijlaarsdam *et al.*, “The next era for earth observation spacecraft: An overview of cognisat-6,” *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 18, pp. 2450–2463, 2025.
- [4] B. Bonham-Carter *et al.*, “Advancing ai reliability for space missions through machine learning operations,” in *18th International Conference on Space Operations (SpaceOps)*, Montreal, Canada, May 2025.
- [5] S. Chien *et al.*, “Using autonomy flight software to improve science return on earth observing one,” *Journal of Aerospace Computing, Information, and Communication (JACIC)*, pp. 196–216, April 2005.
- [6] B. Denby and B. Lucia, “Orbital edge computing: Nanosatellite constellations as a new class of computer system,” in *Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLoS)*, 2020, pp. 939–954.
- [7] A. Candela, J. Swope, and S. A. Chien, “Dynamic targeting to improve earth science missions,” *Journal of Aerospace Information Systems*, vol. 20, no. 11, pp. 679–689, 2023. [Online]. Available: <https://doi.org/10.2514/1.1011233>
- [8] SpaceNews Staff, “Using AI to scan the Earth,” SpaceNews, Dec. 2025, 2025 SpaceNews Icon Awards: Space AI Breakthrough. Accessed: February 13, 2026. [Online]. Available: <https://spacenews.com/using-ai-to-scan-the-earth/>
- [9] D. J. Diner *et al.*, “Multi-angle Imaging SpectroRadiometer (MISR) Instrument Description and Experiment Overview,” *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 4, pp. 1072–1087, 1998.
- [10] V. J. B. Flower and R. A. Kahn, “The evolution of icelandic volcano emissions, as observed from space in the era of nasa’s earth observing system (eos),” *Journal of Geophysical Research: Atmospheres*, vol. 125, no. 19, p. e2019JD031625, 2020.
- [11] S. J. Cantrell, J. B. Christopher, C. Anderson, G. L. Stensaas, S. N. Ramasari Chandra, M. Kim, and S. Park, “System characterization report on the WorldView-3 imager (ver. 1.1, october 2021),” U.S. Geological Survey, “Open-File Report 2021-1030, 2021.
- [12] P. Loudjani, S. Lemajic, V. Parage, and C. Rodrigues, “Report on new Pléiades Neo sensors benchmark,” Publications Office of the European Union, Luxembourg, Tech. Rep. JRC134704, EUR 31687 EN, 2023, joint Research Centre (JRC) Technical Report. [Online]. Available: <https://publications.jrc.ec.europa.eu/repository/handle/JRC134704>
- [13] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, “VGGT: Visual geometry grounded transformer,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. [Online]. Available: <https://arxiv.org/abs/2503.11651>
- [14] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of optical flow estimation with deep networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2462–2470.
- [15] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8934–8943.
- [16] T.-W. Hui, X. Tang, and C. Change Loy, “A lightweight optical flow CNN — revisiting data fidelity and regularization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2540–2551, 2020.
- [17] J. Hur and S. Roth, “Iterative residual refinement for joint optical flow and occlusion estimation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 260–12 269.
- [18] Z. Teed and J. Deng, “RAFT: Recurrent all-pairs field transforms for optical flow,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2020.
- [19] Z. Shi *et al.*, “FlowFormer++: Masked cost volume autoencoding for visual correspondence,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [20] A. Ostrogovich *et al.*, “A dual-mode approach for vision-based navigation in a lunar landing scenario,” in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2024, pp. 1222–2031.
- [21] J. M. Kang, Z. Sjanic, and G. Hendeby, “Visual-inertial odometry using optical flow from deep learning,” in *27th International Conference on Information Fusion (FUSION)*, 2024.
- [22] C. Liu, Y. Jiang, D. Wang, X. Shen, Y. Wang, and G. Zhang, “A structure-preserving multiview stereo network for edge-aware height estimation from satellite images,” *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 18, 2025.
- [23] Z. Chen *et al.*, “Surface depth estimation from multiview stereo satellite images with distribution contrast network,” *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 16 460–16 472, 2024.
- [24] G. Tsagakatakis *et al.*, “Onboard Sequence Compression through Multi-Frame Super-Resolution,” in *8th International Workshop on On-Board Payload Data Compression (OBPDC 2022)*, Athens, Greece, September 2022. [Online]. Available: <https://users.ics.forth.gr/~tsakalid/PAPERS/CNFRS/2022-OBPDC.pdf>
- [25] S. Sharma, C. Beierle, and S. D’Amico, “Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks,” in *Proc. IEEE Aerosp. Conf.*, 2018, pp. 1–10.
- [26] J. L. Carr, C. M. Wilson, D. T. Wilson, D. L. Wu, and H. Madani, “Stereobit: Onboard science data processing for stereo winds,” *Journal of Aerospace Information Systems*, vol. 22, no. 5, pp. 325–335, 2025. [Online]. Available: <https://doi.org/10.2514/1.1011289>
- [27] A. Flurry, “Mini satellites, major impact,” UGA Today, October 2025, accessed: February 13, 2026. [Online]. Available: <https://news.uga.edu/mini-satellites-major-impact/>
- [28] C. Adams, J. Parker, and D. Cotten, “A hardware accelerated computer vision library for 3d reconstruction onboard small satellites,” in *2021 IEEE Aerospace Conference*, 2021, pp. 1–14.
- [29] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt, “Joint estimation of motion, structure and geometry from stereo sequences,” in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 568–581.
- [30] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [31] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge University Press, 2004.
- [32] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *7th International Joint Conference on Artificial Intelligence (IJCAI ’81)*, Vancouver, Canada, 1981, pp. 674–679.
- [33] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, p. 91–110, Nov. 2004. [Online]. Available: <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- [34] G. Berton, A. Stoken, and C. Masone, “Astroloc: Robust space to ground image localizer,” 2025. [Online]. Available: <https://arxiv.org/abs/2502.07003>
- [35] A. Stoken and K. Fisher, “Find my astronaut photo: Automated localization and georectification of astronaut photography,” in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, June 2023, pp. 6196–6205.