Towards Onboard Multi-View Analysis of Satellite Imagery: Monocular 2D Motion Estimation and 3D Reconstruction

Anonymous CVPR submission

Paper ID *****

Abstract

Driven by high-performance processors, modern Low Earth Orbit (LEO) satellites can now execute AI and deep learning workloads in orbit. This shift enables spacecraft to process sensor data into actionable insights rather than serving as mere conduits for downlinking. However, current onboard AI predominantly focuses on single-image analysis tasks like segmentation and detection. These methods fail to leverage the motion and geometric information inherent in multi-view imagery, which is crucial for characterizing dynamic volumetric phenomena—such as clouds and plumes—and for enabling autonomous, real-time monitoring without the latency of ground-station downlink. Multi-view analysis and photogrammetry remain largely constrained to ground-based processing due to their high computational demands or reliance on large external datasets. To address this gap, we present a framework for onboard monocular multi-view analysis designed for edge deployment; it performs 2D motion estimation—exploring both classical and learning-based methods—and subsequent geometric 3D scene reconstruction. This work represents a pioneering effort in transitioning multi-view analysis directly onboard Earth Observation (EO) satellites. We validate our framework using real-world data from the [Anonymous] mission, a satellite testbed for AI flight software, paving the way for upcoming deployment on this and other platforms with advanced compute.

1. Introduction

Recent developments in the space sector are bringing new capabilities to orbit. This shift is characterized by the proliferation of LEO platforms, which have improved temporal and spatial coverage while lowering the cost of observation services. Alongside this expansion in infrastructure, advanced satellite communications links now enable 24/7 low-data-rate connectivity, facilitating the rapid dissemination of observed events. Central to this transformation is the shift from traditional downlink paradigms, where raw data is stored and forwarded for ground-side processing, toward onboard edge computing. Driven by powerful Commercial Off-The-Shelf (COTS) "AI processors", modern LEO satellites can now execute deep learning and other intensive workloads directly in orbit [13, 19, 25]. This evolution, further supported by Linux environments, allows spacecraft to deploy complex applications and immediately convert raw data into actionable knowledge.

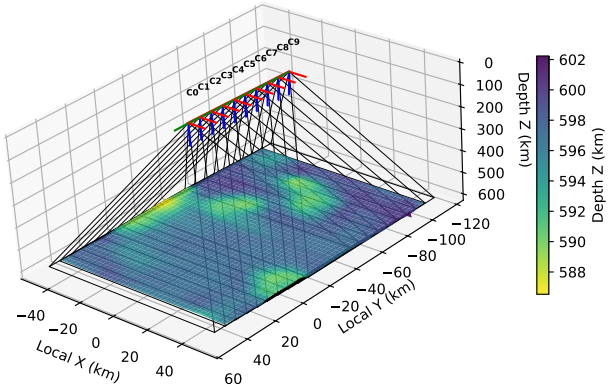


Figure 1. Monocular 3D reconstruction from an orbital overpass. This example shows our framework's ability to resolve cloud heights by leveraging orbital parallax. The pipeline is amenable to the resource-constrained compute on modern LEO satellites, enabling real-time onboard analysis.

Despite these satellite hardware advancements, current work on onboard AI has predominantly focused on single-image analysis. State-of-the-art implementations typically leverage deep learning to perform 2D tasks such as scene classification, semantic segmentation, target detection, and anomaly detection. While effective for interpreting individual frames, these methods do not exploit the information available from multiple images.

Multi-view image analysis and photogrammetry, essential for motion and geometric understanding, remain largely bound to ground-based processing. Traditional approaches are typically computationally intensive or reliant on large external datasets, such as landmarks and global Digital Elevation Models (DEMs), making them still difficult to port to spacecraft. Onboard multi-view analysis could unlock spatiotemporal information currently unavailable with standard 2D techniques. Satellites could now disambiguate scenes where spectral data alone is insufficient; for instance, distinguishing between high-altitude clouds, low-lying fog, and snow cover. By extracting motion fields and geometric structures directly at the edge, spacecraft could immediately characterize evolving volumetric events such as clouds and volcanic plumes. Ultimately, this multi-view intelligence could enable time-critical disaster response or real-time autonomous navigation, where decisions must be made in seconds rather than days.

This work presents a framework for monocular multi-view EO analysis designed for edge deployment (Fig. 1). Our approach enables the extraction of motion and geometry information from a single-camera sequence. Specifically, the framework performs 2D motion estimation via dense optical flow—exploring both classical and learning-based methods—which serves as the foundation for subsequent 3D scene reconstruction. To the best of our knowledge, this work represents one of the earliest efforts to perform multi-view EO image analysis directly onboard satellites, and the first to specifically use deep learning. We validate our approach using real EO data from the [Anonymous] mission, a satellite testbed for AI flight software [2]. These results pave the way for our method's deployment on this and other satellites with advanced onboard compute.

2. RELATED WORK

2.1. 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 [8]. ASE featured a suite of flight software capable of detecting and responding to dynamic science events; however, it was limited by the EO-1 legacy hardware, which lacked the agility and edge compute capabilities of modern LEO spacecraft. ESA's Φ-sat-1 served as an initial experimental testbed for onboard deep learning, utilizing the Intel Movidius Myriad 2 Vision Processing Unit (VPU) [13]. The mission showcased several AI applications, most notably *CloudScout*, which employed a Convolutional Neural Network (CNN) to filter cloudy imagery and reduce data transmission. Φ-sat-1 was also used to validate wildfire detection and land cover classification. Similarly, the ESA OPS-SAT mission functioned as a testbed to host diverse experiments, including onboard TensorFlow Lite image classification [19].

Onboard satellite autonomy is evolving from passive data analysis toward active decision-making. Orbital Edge Computing (OEC) established a framework enabling deep learning to be parallelized across nanosatellite constellations to process data at the source [9]. The CogniSAT-6 mission [25] demonstrated the "dynamic targeting" concept [4]: by scanning its upcoming orbital path, the satellite used a CNN to identify cloud-free zones and autonomously reorient the spacecraft to capture clear imagery [28].

2.2. Ground-Based Multi-View Photogrammetry

While single-image analysis has been demonstrated at the edge, processing of multi-view photogrammetry remains on the ground. Standard pipelines rely on dense matching algorithms, such as Semi-Global Matching (SGM) or Multi-View Stereo (MVS) methods, which impose prohibitive memory requirements and computational costs at the edge, often further constrained by a reliance on high-resolution terrestrial DEMs. Historically, multi-view missions such as NASA's Multi-angle Imaging SpectroRadiometer (MISR) [10] have downlinked vast raw radiance datasets for terrestrial processing. Utilizing nine pushbroom cameras to capture imagery from distinct perspectives, the instrument enables the retrieval of geometric attributes such as cloud-top heights and the characterization of volcanic plume dynamics [11]. This reliance on ground-side processing persists in modern commercial constellations like Maxar's WorldView [5] and Airbus's Pléiades Neo [21], which utilize stereo images for high-resolution 3D mapping.

2.3. Learning-Based Multi-View Analysis

State-of-the-art approaches transition from traditional photogrammetry to learning-based methods. The Visual Geometry Grounded Transformer (VGGT) [33] leverages a large-scale transformer backbone to directly infer 3D attributes—including camera poses and depth maps—in a single forward pass. Parallel to explicit reconstruction, deep learning has redefined dense correspondence and motion estimation. This evolution spans from earlier CNN-based architectures like FlowNet2 [17], PWC-Net [29], LiteFlowNet2 [15], and IRR-PWC [16], to more recent models such as RAFT [30] and the transformer-based FlowFormer++ [27].

In the satellite domain, learning-based optical flow has been investigated for low-latency onboard tasks, such as autonomous lunar landing navigation [24] and visual-inertial odometry for navigation over featureless terrain [18]. Unlike classical trackers, these deep learning models offer superior robustness against the low-contrast and repetitive textures common in orbital imagery. Conversely, recent architectures such as SP-MVS [20] and DC-SatMVS [7] focus on estimating 3D structuring while preserving sharp vi-

sual boundaries, such as building outlines that are typically blurred in standard orbital depth estimation. However, these models remain largely incompatible with edge processors due to their large memory and power footprints.

2.4. Onboard Multi-View Analysis

Onboard 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 geometry [31]. Similarly, onboard stereo vision for On-Orbit Servicing and Manufacturing (OSAM) uses CNNs for pose estimation of rigid bodies against high-contrast backgrounds [26], but it requires known target models and has been validated on ground-based testbeds for proximity operations rather than for EO. The StereoBit project demonstrated 3D cloud height retrieval on a SpaceCube v3.0 Mini testbed rather than in orbit [6]. While its FPGA-accelerated sparse matching is effective for atmospheric features, it fails to generate continuous data in textureless regions and lacks the flexibility of modern, software-defined frameworks. The Multi-view Onboard Computational Imager (MOCI) mission, slated for a 2026 launch [12], represents a key effort in orbital 3D reconstruction. Utilizing an NVIDIA Jetson TX2i, MOCI implements a classical Structure-from-Motion (SfM) pipeline via the custom SSRLCV library [1]. The yet-to-be-flown system performs feature matching and bundle adjustment to generate high-resolution 3D DEMs directly in orbit.

3. Methodology Overview

The proposed framework addresses the computational bottlenecks of traditional photogrammetry. Conventional SfM and Multi-View Stereo (MVS) pipelines rely on iterative stages—including sparse feature matching, global bundle adjustment, and intensive interpolation for mesh generation—that can often exceed the compute constraints of small satellites. To overcome this, we utilize a direct pipeline that operates on low-level data ($L0$ or basic $L1$). This approach is designed to process imagery without 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 [32]. This pipeline leverages dense correspondences for a semi-dense 3D point cloud, balancing geometric fidelity with computational efficiency. We adopt a hybrid methodology over a fully end-to-end learning-based approach due to the scarcity of high-resolution 3D ground truth for dynamic atmospheric phenomena in LEO imagery. Decoupling 2D flow from 3D reconstruction ensures the framework remains interpretable, enforces physical constraints, and allows for algorithm-agnostic flexibility depending on available onboard compute (Fig. 2).

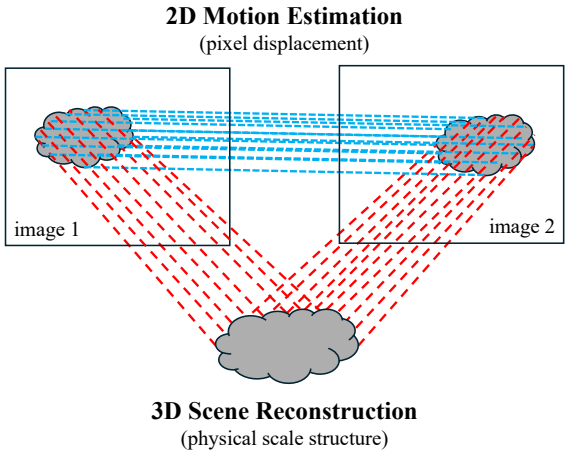


Figure 2. Conceptual overview of the approach. Pixel displacements between sequential images are tracked via 2D motion estimation. These dense correspondences are then triangulated to perform 3D scene reconstruction, yielding physical scale structure.

The framework is a modular, sequential pipeline that consists of these steps: a) image preprocessing, b) 2D motion estimation (via dense optical flow), c) camera parameter estimation (intrinsics and extrinsics), d) correspondence outlier filtering, e) point cloud triangulation, and f) point cloud postprocessing. The technical description is divided into two sections: Section 4 details the 2D motion estimation phase (steps a–b), while Section 5 delineates the 3D scene reconstruction process (steps c–f). Following this methodology, Section 6 introduces the [Anonymous] mission, datasets, and experiments used to evaluate the framework. The results are presented and analyzed in Section 7.

4. 2D Motion Estimation

4.1. 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.

4.2. 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

Table 1. Comparison of Deep Learning Optical Flow Models

Model	Year	Parameters	File Size
RAFT Small	2020	~1.0 Million	~4.5 MB
RAFT Large	2020	~5.3 Million	~21 MB
LiteFlowNet2	2020	~6.4 Million	~25 MB
IRR-PWC	2019	~6.4 Million	~26 MB
PWC-Net	2018	~8.8 Million	~35 MB
FlowNet2	2017	~162 Million	~620 MB

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.

4.2.1. Classical Dense Methods

We evaluate the following suite of dense optical flow methods available in the OpenCV library [3]: DIS (Fast), Farneback, SimpleFlow, DeepFlow, SparseToDense, RLOF, DualTVL1, and PCAFlow. These algorithms vary significantly in runtime and estimation quality.

4.2.2. Learning-Based Models

We evaluate six deep learning architectures: FlowNet2 [17], PWC-Net [29], LiteFlowNet2 [15], IRR-PWC [16], and both the large and small variants of RAFT [30] (see Table 1 for model details). To assess their out-of-the-box utility for orbital imagery, all models are used with off-the-shelf pre-trained weights without additional fine-tuning. All architectures are leveraged from existing PyTorch implementations.

5. 3D Scene Reconstruction

The geometric formulations and mathematical notation utilized throughout this section build on the standard principles of multi-view geometry [14]. Our method is formulated for framing cameras, which capture a 2D image simultaneously and can be described using the standard pinhole camera model. In our implementation, 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.

5.1. 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:

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

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:

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

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:

$$P = K[R \mid \mathbf{t}]. \tag{4}$$

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:

$$\lambda \mathbf{u} = P\mathbf{X}, \tag{5}$$

where λ is the scalar projective depth.

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

5.1.1. Optical Flow-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.

5.1.2. 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 an ellipsoid 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.

5.2. 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:

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

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:

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

Matches where d exceeds a pixel threshold τ are discarded.

5.3. Point Cloud Triangulation

Using pixel disparities from Section 4, 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:

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

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

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

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:

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

5.4. 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.

6. Experiments

6.1. Mission and Dataset

The [Anonymous] mission [2] is a 6U CubeSat built on [Anonymous]'s bus. It launched on [Anonymous date], into a sun-synchronous orbit at an altitude of approximately 600 km. The mission is designed as an orbital testbed for hosting and validating machine learning payloads on embedded hardware. Its [Anonymous] computing module uses an AMD Zynq UltraScale+ Multi-Processor System-on-Chip (MPSoC), specifically the [Anonymous] model.

The satellite carries the [Anonymous] 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 [Anonymous] imager specifications.

6.2. 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 3. Flow accuracy directly impacts 3D reconstruction fidelity. Performance is quantified—across all dense methods and the Lucas-Kanade baseline [23]—using three metrics:

- **SIFT Correspondence Error:** Flow accuracy against high-confidence sparse SIFT keypoints [22], 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 and linear interpolation during point cloud postprocessing. To maintain consistency with the resized image resolution (48 m GSD $\times$ 4), we bin the data to 192 m/pixel. For geometric filtering, we apply a moderate epipolar constraint threshold τ of 3 pixels. We compare a selection of high-performing methods: the sparse SIFT baseline, the best classical dense algorithm, and the top deep learning model. We assess the two extrinsic estimation methods detailed in Section 5.1: Optical Flow-Only and Hybrid. Direct 3D validation for dynamic phenomena is exceptionally challenging; while geostationary platforms offer relatively high temporal cadence (e.g., 10–15 minute global updates), they lack the fine spatial resolution of LEO assets. Furthermore, while higher-resolution products exist, their infrequent revisit rates—often spanning several days—make them impractical for matching transient atmospheric events coincident with our dataset. Additionally, the [Anonymous] satellite does not carry active sensors (e.g., LiDAR) for generating ground-truth depth data. Therefore, we evaluate the framework using four validation metrics, where the first three isolate geometric modeling inaccuracies:

- **Epipolar Outlier Rate:** Percentage of feature matches rejected for violating epipolar constraints.
- **Reprojection Error:** Pixel-wise distance between original and reprojected features.
- **Geodetic Altitude Error:** Altitude mean absolute error (MAE) relative to the WGS84 ellipsoid surface.
- **Processing Time:** Divided between main processing and optional postprocessing of a ten-image sequence.

7. Results and Discussion

7.1. 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 2). 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. No-

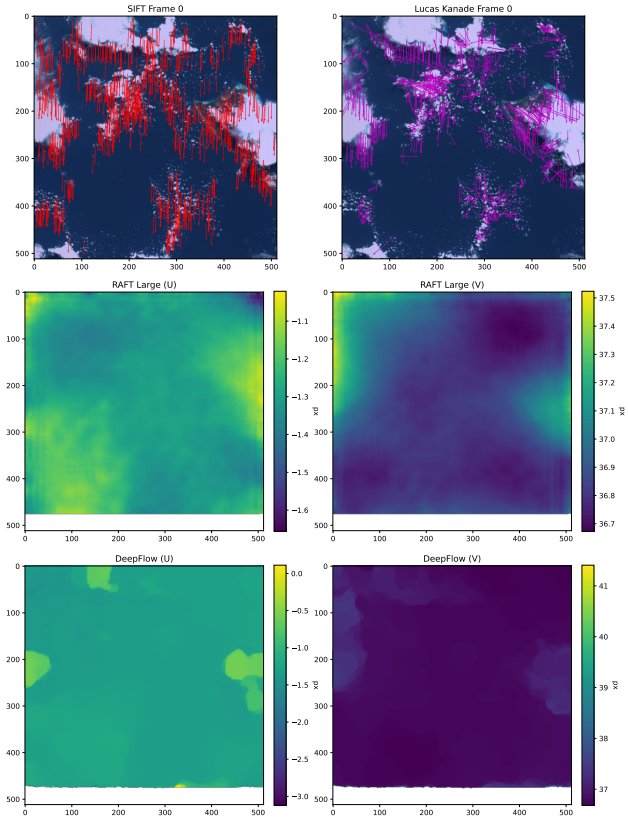


Figure 3. Optical flow estimation on an orbital image sequence. Top: SIFT and Lucas-Kanade correspondences. Middle: RAFT deep learning model. Bottom: DeepFlow classical algorithm.

tably, 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 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 reimplenting 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.

7.2. Results: 3D Scene Reconstruction

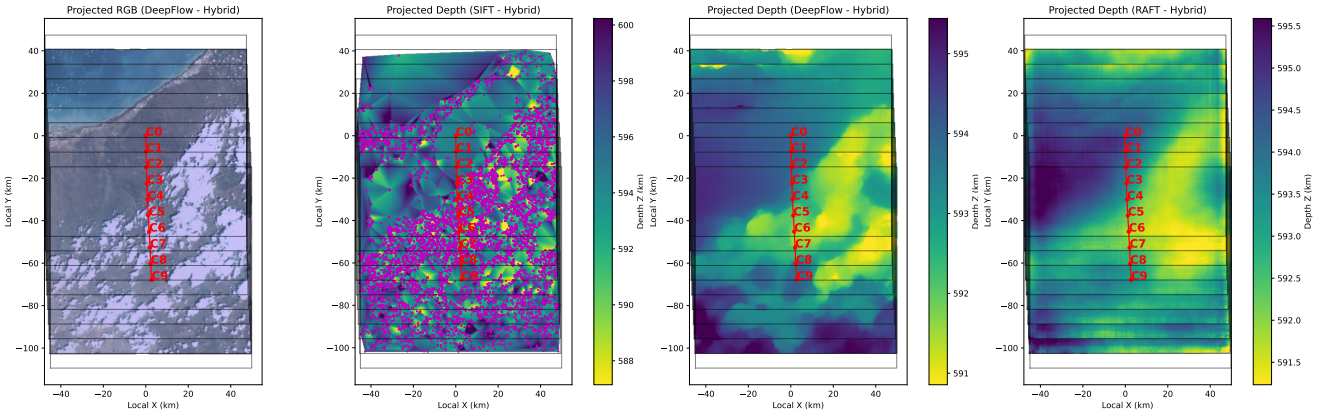
The 3D reconstruction results across the [Anonymous] dataset demonstrate high fidelity and geometric consistency (Fig. 4). While sparse methods like SIFT anchor the geometry, they leave significant spatial gaps requiring extensive interpolation. In contrast, dense approaches maintain superior structural 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.

Table 2. Aggregated Optical Flow Performance Results (Mean $\pm$ Std. Dev.) on the [Anonymous] Dataset

Method	**SIFT (px)**	**FB Err (px)**	**Runtime (s)**
Classical Methods			
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
Deep Learning			
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

As summarized in Table 3, the Hybrid approach resolves the scale ambiguity inherent in the purely vision-based OF-Only mode. While the OF-Only approach suffers from global altitude drift due to nominal spherical Earth assumptions, the Hybrid model leverages telemetry to reduce this error by nearly an order of magnitude. Crucially, this rescaling does not compromise relative camera geometry, as outlier rates and sub-pixel reprojection errors remain nearly identical. The residual error in the Hybrid approach is physically expected, as triangulated points capture clouds and terrain not modeled by the WGS84 reference ellipsoid.

While requiring more processing time, dense methods provide substantial accuracy gains; both RAFT and DeepFlow consistently achieve tighter reprojection errors than the sparse SIFT baseline. By extracting statistical patterns in motion fields, dense methods smooth localized noise to reduce geometric jitter. DeepFlow represents the best balance of accuracy and efficiency for this pipeline; despite a higher outlier rate than RAFT, it matches its stability while delivering the lowest reprojection error with lower latency. It should be noted that RAFT's performance could be further enhanced through domain-specific fine-tuning and inference optimization for GPU acceleration.



(a) RGB Reconstruction (b) Sparse SIFT (c) Dense DeepFlow (d) Dense RAFT Large

Figure 4. Comparison of 3D scene reconstruction results for a wildfire smoke plume in Northeast Australia (2025-XX-XX). 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.

Table 3. Aggregated 3D Reconstruction Results (Mean $\pm$ Std. Dev.) on the [Anonymous] Dataset

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

8. Conclusions and Future Work

This paper presents a modular framework for onboard analysis of monocular multi-view satellite imagery, integrating 2D motion estimation with 3D scene reconstruction. Evaluation using data from the [Anonymous] mission confirms two key findings. First, top-performing classical and deep learning dense optical flow methods achieve sub-pixel accuracy without fine-tuning. Second, the resulting 3D reconstructions recover volumetric scene structures with superior density and quality over sparse methods, namely SIFT. Additionally, the presented hybrid approach leverages telemetry, when available, to refine scale while retaining vision-derived geometric consistency. Notably, the framework executes in seconds without requiring external data (e.g., landmarks or DEMs), demonstrating 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 its capabilities to include joint 3D position and velocity estimation of scene targets. We also plan to conduct empirical evaluations of the full pipeline on representative AI flight processors and testbeds. Ultimately, these efforts aim to support a technical demonstration on the [Anonymous] mission and other next-generation satellites equipped with advanced onboard compute, enabling real-time monitoring of dynamic phenomena.

Acknowledgment

This work was supported by [Anonymous Sponsor]. This work was conducted by [Anonymous Organization], under a contract with [Anonymous Organization].

References

[1] Caleb Adams, Jackson Parker, and David Cotten. A hardware accelerated computer vision library for 3d reconstruction onboard small satellites. In *2021 IEEE Aerospace Conference*, pages 1–14, 2021. 3

[2] Anonymous Authors. Anonymous title. In *Anonymous Venue*, 2025. 2, 6

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 4

[4] Alberto Candela, Jason Swope, and Steve A. Chien. Dynamic targeting to improve earth science missions. *Journal of Aerospace Information Systems*, 20(11):679–689, 2023. 2

[5] S. J. Cantrell, J. B. Christopherson, C. Anderson, G. L. Stensaas, S. N. Ramaseri Chandra, M. Kim, and S. Park. System characterization report on the WorldView-3 imager (ver. 1.1, october 2021). Open-file report 2021-1030, U.S. Geological Survey, 2021. 2

[6] James L. Carr, Christopher M. Wilson, David T. Wilson, Dong L. Wu, and Houria Madani. Stereobit: Onboard science data processing for stereo winds. *Journal of Aerospace Information Systems*, 22(5):325–335, 2025. 3

[7] Ziyang Chen et al. Surface depth estimation from multi-view stereo satellite images with distribution contrast network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:16460–16472, 2024. 2

[8] S. Chien et al. Using autonomy flight software to improve science return on earth observing one. *Journal of Aerospace Computing, Information, and Communication (JACIC)*, pages 196–216, 2005. 2

[9] Bradley Denby and Brandon Lucia. Orbital edge computing: Nanosatellite constellations as a new class of computer system. In *Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 939–954, 2020. 2

[10] David J. Diner et al. Multi-angle Imaging SpectroRadiometer (MISR) Instrument Description and Experiment Overview. *IEEE Transactions on Geoscience and Remote Sensing*, 36(4):1072–1087, 1998. 2

[11] Verity J. B. Flower and Ralph 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*, 125(19):e2019JD031625, 2020. 2

[12] Alan Flurry. Mini satellites, major impact. UGA Today, 2025. Accessed: February 13, 2026. 3

[13] Gianluca Giuffrida et al. The phi-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022. 1, 2

[14] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2nd edition, 2004. 4

[15] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow CNN — revisiting data fidelity and regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2540–2551, 2020. 2, 4

[16] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12260–12269, 2019. 2, 4

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017. 2, 4

[18] 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. 2

[19] Georges Labrèche et al. Ops-sat spacecraft autonomy with tensorflow lite, unsupervised learning, and online machine learning. In *2022 IEEE Aerospace Conference (AERO)*, pages 1–17, Big Sky, MT, USA, 2022. IEEE. 1, 2

[20] Chen Liu, Yonghua Jiang, Dong Wang, Xin Shen, Yunming Wang, and Guangbin Zhang. A structure-preserving multi-view stereo network for edge-aware height estimation from satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18, 2025. 2

[21] Philippe Loudjani, Slavko Lemajic, Vincent Parage, and César Rodrigues. Report on new Pléiades Neo sensors benchmark. Technical Report JRC134704, EUR 31687 EN, Publications Office of the European Union, Luxembourg, 2023. Joint Research Centre (JRC) Technical Report. 2

[22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 6

[23] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, Vancouver, Canada, 1981. 6

[24] A. Ostrogovich et al. A dual-mode approach for vision-based navigation in a lunar landing scenario. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2022–2031, 2024. 2

[25] David Rijlaarsdam et al. The next era for earth observation spacecraft: An overview of cognisat-6. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18:2450–2463, 2025. 1, 2

[26] Sumant Sharma, Connor Beierle, and Simone D'Amico. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In *2018 IEEE Aerospace Conference*, pages 1–10, 2018. 3

[27] Zuoqiang Shi et al. FlowFormer++: Masked cost volume autoencoding for visual correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[28] SpaceNews Staff. Using AI to scan the Earth. SpaceNews, 2025. 2025 SpaceNews Icon Awards: Space AI Breakthrough. Accessed: February 13, 2026. 2

[29] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018. 2, 4

[30] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 4

[31] Grigorios Tsagkatakis et al. Onboard Sequence Compression through Multi-Frame Super-Resolution. In *8th International Workshop on On-Board Payload Data Compression (OBPDC 2022)*, Athens, Greece, 2022. 3

[32] Levi Valgaerts, Andrés Bruhn, Henning Zimmer, Joachim Weickert, Carsten Stoll, and Christian Theobalt. Joint estimation of motion, structure and geometry from stereo sequences. In *Computer Vision – ECCV 2010*, pages 568–581, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 3

[33] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2