Active Terrain Relative Navigation with Surface Mapping Sensors

Davis Adams, Ph.D.*
NASA Johnson Space Center, Houston, TX 77058

Vikas Patel†
Stanford University, Stanford, CA 94305

Isaac Rowe‡
Aerodyne Industries – Amentum JETS II Contract, Houston, TX 77058

This work presents a three-dimensional surface map-informed Active Terrain Relative Navigation (ATRN) system for spacecraft precision landing and extra-planetary surface mobility applications. Rotationally and scale-invariant 3D feature extraction and identification methods are introduced, facilitating both global and relative localization in real-time. Numerical spatial gradient and Laplacian methods exploit the surface statistics of the local terrain to identify points which exhibit high degrees of slope and curvature. In contrast to these derivative-based approaches, the proposed Zernike Invariant Gradient (ZIG) 3D feature extraction method is fundamentally built on an integral technique and is inherently robust to noise. Reliable detection and description of 3D features enables correspondence between successive digital elevation map (DEMs) and onboard global reference maps. Resultant fused high-resolution maps inform hazard detection/avoidance and safe-site selection activities for precision landing applications and route planning for surface mobility applications. The architecture of a standalone C++ flight software library that implements the proposed methodologies is also presented. Performance of the proposed ATRN system is evaluated via extensive testing with synthetic digital elevation maps in real-time hardware-in-the-loop (HITL) simulations as well as data obtained during helicopter flight testing of the NASA Goddard Hazard Detection LiDAR (HDL).

I. Introduction

Active sensing systems such as LiDAR enable robust navigation in environments characterized by hazardous terrain, variable lighting, or limited map availability by producing three-dimensional surface digital elevation maps (DEMs) independent of illumination. In these scenarios, passive terrain relative navigation (TRN) systems informed by standard two-dimensional (2D) imagery are inadequate due to illumination variance and difficulty matching high-resolution real-time images to low-resolution reference imagery. To robustly leverage terrain maps for localization and hazard detection/avoidance activities, novel approaches for three-dimensional (3D) feature extraction and correspondence must be developed.

Passive TRN methodologies have been ubiquitously employed for the majority of historical human and robotic space exploration missions during both descent, landing and surface mobility operations. Over time, state-of-the-art passive TRN approaches have continuously evolved delivering impressive navigation accuracies with position errors less than 10m for Lunar precision landing [1] and 1m [2] for ground mobility operations when subjected to favorable lighting conditions. However, these systems break down quickly in situations where 1) *insitu* lighting conditions are not consistent with conditions captured by orbital imagery, 2) orbital imagery resolution severely exceeds *in situ* imagery ground sample distance, or 3) no *a-priori* information of the underlying terrain exists [3]. Ongoing work is attempting to produce illumination invariant 2D feature identification algorithms to enable template matching between two images exhibiting varying lighting conditions [4], however this approach requires significant pre-mission analysis and is not practical for long-duration autonomous missions with consistently changing lighting conditions. Real-time synthetic image generation can be employed to produce reference imagery for comparison between *in situ* imagery [5], but this

*Navigation Engineer, Aeroscience and Flight Mechanics Division, davis.w.adams@nasa.gov
†GN&C Graduate Research Assistant,patelva@stanford.edu
‡Software Engineer,isaac.s.rowe@nasa.gov

Trade names are used in this presentation for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

approach requires accurate knowledge of the spacecraft state or sufficiently close adherence to the reference trajectory, and is computationally demanding.

Active 3D surface mapping sensors and associated algorithms which leverage the structured elevation data can be employed to circumvent the aforementioned shortcomings of traditional passive TRN systems. Given this massive value proposition, there are numerous on-going NASA-funded activities to develop a surface mapping sensor capable of creating high-resolution DEMs at ranges and within timelines required for precision landing operations including the Hazard Detection LiDAR (HDL) being developed at NASA Goddard Space Flight Center and the Europa LiDAR Sensor Assembly (ELSA) being developed at NASA Jet Propulsion Library with Hexagon US Federal [6, 7]. While specific mission concept of operations (ConOps) typically define the detailed performance requirements, many of these active surface mapping sensors are targeting the ability to generate a 100m $\times$ 100m DEM with a ground sample distance of 10 cm or better within 4 seconds or less from at least 500 m range to target to accomodate both Lunar and Martian descent and landing operations. These performance criteria are required to ensure the robotic or manned lander has enough time and information to perform hazard detection/avoidance activities.

Although significant attention is being paid to the development of the surface mapping sensors, spacecraft navigation systems which natively leverage digital elevation maps to robustly perform localization and mapping with sufficient accuracy do not exist. While extensive work has been done in the fields of 3D feature/object recognition [8–13] and simultaneous localization and mapping (SLAM) [14–17] for applications including robotics, autonomy, and machine learning, these methods succumb to the unique challenges produced when trying to navigate with respect to natural terrain. Iterative approaches to correspond 3D point clouds yielding a relative position and attitude between two or more clouds, such as Iterative Closest Point [18], are widely used in robotics. However, these techniques are computationally burdensome, often rely on a highly accurate initial guess, and are not robust when comparing point clouds encompassing vastly different sizes and resolutions. Template matching approaches such as normalized cross correlation (NCC) [19] can be used to match digital elevation maps without an initial guess, but exhibit the same shortcomings as iterative methods with regards to computational requirements and increased complexity when comparing maps of different sizes and resolutions. Sparse 3D feature-based methods such as global point signatures (GPS) [20], intrinsic shape signatures (ISS) [21], heat kernel signatures [22], and wavelet signatures [23] offer an advantage in terms of processing time and are more analogous to passive TRN approaches, but results acquired during this work indicated that existing methods are insufficient when attempting to extract and correspond features from natural terrain.

To address these shortfalls, a holistic sparse 3D keypoint-based active terrain relative navigation system has been developed and is presented in this work. Primarily, this work details multiple novel rotationally and scale invariant 3D feature extraction and description methods along with robust correspondence algorithms. These developments enable accurate global and relative localization as well as producing a high-resolution DEM constructed from successively observed terrain maps. A standalone C++ flight software library has also been developed with plans for release in the NASA software catalog. Performance of the proposed ATRN system in terms of navigation accuracy, robustness to common sensor induced artifacts, and computation aboard representative space flight hardware is evaluated on both synthetic and flight test DEM data.

II. Feature Identification from 3D Surface Maps

Extracting salient features from digital elevation maps representing natural terrain is inherently challenging due to the general lack of unique high-contrast regions (topographically speaking). Once reliable features have been detected, they must be described in a rotationally and scale-invariant manner such that they can be corresponded with features from other surface maps which may exhibit different ground sample distances, extents, and local reference frames.

A. Surface Map Pre-Processing

Before attempting to extract features from the relevant DEMs, it was imperative that the DEMs are pre-processed to ensure that the extraction methods did not falsely identify certain data as features to track when they might not be persistent across different DEMs of the same terrain. The first step in this process was to mitigate a common issue in DEMs stemming from LiDAR data. Due to various hardware, and potentially some post-processing related, shortcomings, it was found that some DEMs generated from real data contained elevation outliers with a significantly high deviation from surrounding data. Specifically, after manually identifying a candidate outlier cell in the DEM, comparative statistical analysis of surrounding elevation and slope values clearly indicated errant sensor returns. To mitigate the effects of outliers, which are persisent in LiDAR sensing technology [24], an elevation outlier filter was implemented. The outlier filter sweeps through each elevation point and finds the standard deviation of elevation

of all neighboring points. A minimum required neighbor threshold ensures that not too many edge elevations are removed when there are not enough data points surrounding a given point due to empty pixels or edges. Once the standard deviation is found, the data point in question is then compared to the mean surrounding points. If this elevation difference is greater than some multiple of the surrounding standard deviation, it can be characterized as an outlier and smoothed over to be the median of the surrounding elevations. Although a median pre-filter could further reduce the effects of elevation outliers and noise, empirical data indicated the pre-filter tended to excessively smooth over the data, removing potentially valuable information about crater rims or sudden terrain changes.

Subsequently, DEM pre-conditioning was employed in the form of a 2D Fourier-based low pass filter to mitigate the effects of sensor noise. Notably, it was found that when comparing DEMs of different ground sample distances, applying any of the proposed 3D feature extraction methods resulted in differing information maps (details to follow). For example, when looking at a large rock placed in an otherwise level hazard field, the slope, Laplacian, and Zernike invariant gradient (ZIG) information maps would look different at different resolutions. More specifically, if the goal is to investigate locations exhibiting of high degrees of slope, the DEM with a smaller ground sample distance (finer resolution) would be able to discern more data between the level terrain and the edge of the protruding rock. This would mean that the DEMs with a smaller ground sample distance would consistently find higher slopes around areas where the terrain changes are on the order of magnitude of the resolution. The 2D Fourier filter solves this by first completing a 2D Fourier transform of the underlying DEM, removing data corresponding with higher frequencies, and then completing a 2D inverse Fourier transform. This cutoff frequency is tailored to the physical distance of wavelengths which should be removed, and the resulting processed, inverted Fourier map eliminates differences in ground sample distance. It must be noted that due to the edge effects along the edges of the DEM and null-values, a buffer must be applied to ensure that the extracted features were not a result of edge effects but rather the underlying terrain. In addition, a Tukey window effectively mitigated the stronger effects on the edges of the DEMs.

Finally, it must be noted that these pre-processing steps contribute to a significant portion of the total computation time, and especially due to the nature of the 2D Fourier transform, an excessively large DEM results can drastically increase computation time. Fortunately, this pre-processing step and subsequent feature extraction steps only need to be computed once for a DEM that is likely to be large, such as an onboard reference DEM.

B. 3D Feature Extraction

The first step to 3D feature identification between disparate surface maps is extracting a set of candidate feature locations. It is important to note that high-quality 3D features must exhibit three key qualities: uniqueness, repeatability, and invariance. Previous work presented spatial gradient methods to find points at which the underlying surface exhibits high/low-degrees of slope, denoted the high/low-gradient methods, respectively [25]. For both the high- and low-gradient methods, the first step is to produce a two-dimensional off-vertical angle map using numerical differentiation. For a function $f(x, y)$ represented by a digital elevation map, the off-vertical angle, θ, at each point (i, j) is calculated as:

$$\theta(i, j) = \arctan\left(\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}\right) \tag{1}$$

where $\frac{\partial f}{\partial x}$ is the partial derivative in the x-direction and $\frac{\partial f}{\partial y}$ is the partial derivative in the y-direction. High-gradient features are located by finding maximum values of the off-vertical map, and low-gradient features are located by finding minimum values of the off-vertical map. Notably, both gradient methods are inherently not rotationally invariant due to the dependence on the local reference frame definition. However, empirical testing revealed the high-gradient methodology paired with either feature matching technique to be discussed below produces sufficiently accurate pose estimates when the ground plane definitions are within $\sim 10^\circ$ of one another. To address this theoretical lack of rotational invariance, a new derivative-based approach has been developed using the Laplace-Beltrami operator, denoted the Laplacian method. Fundamentally, Laplacian is the sum of second partial derivatives and is a measure of how a function differs from the average of its neighboring values and is known to be a rotationally invariant operator. The Laplacian of a function $f(x, y)$ is defined as:

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{2}$$

Laplacian feature points are located at maxima of the Laplacian map. Since the gradient and Laplacian methods are fundamentally derivative-based methods, they exacerbate noise exhibited by the underlying digital elevation maps.

In practice, effects of noise can be mitigated by pre-processing techniques, such as low-pass filter masking, which sufficiently filter the noise while retaining the true signal.

In an effort to increase the algorithm's robustness, a windowing technique is optionally applied for the gradient and Laplacian methods. In a normal feature point extraction process, the best N feature points are extracted globally across the map as described above. However, in a windowed technique, the resulting off-vertical or Laplacian map can be divided into sub-sections, and a certain number of feature points are found in each quadrant such that a total of N feature points are still found. This ensures that the feature points found are relatively well distributed throughout the map. Take the case where a given map has one area that is particularly feature rich. Finding the global best N features results in many regions which may not have many, if any, keypoints found. This is especially noticeable with extracting keypoints in a larger DEM. If the goal is to seek a transformation with a DEM that overlaps in this region of relatively low topographic variation, there may not be a sufficient overlap in extracted keypoints. Therefore, this technique results in the ability to guarantee a certain number of overlapping keypoints with some relationship of overlap in the DEMs to compare.

To avoid the effects of noise and maintain rotational invariance, a mathematically rigorous keypoint extraction method has been developed which leverages the functional moments of 3D Zernike polynomials. Zernike polynomials are a special class of orthogonal basis functions, fundamentally valid within the unit disk in 2D space, which can be hyper-dimensionalized to be valid throughout the unit-sphere. The mathematical structure of the set of 3D Zernike polynomials is very similar to spherical harmonics; however, they exhibit a radius modulation function so they are not just valid on the surface of the unit sphere. Extensive details on 3D Zernike moments can be found in work by Novotini et al. [26, 27], but the relevant developments are included here for completeness.

Functional moments are fundamentally defined as projections of a square-integrable function $f \in L^2$ onto a set of functions $\Psi = \{\psi_i\}$ over a domain Ω:

$$\mu_i = \langle f, \psi_i \rangle = \int_\Omega f(x)\psi_i(x)\, dx \tag{3}$$

Using functional moment-based values as feature descriptors is attractive due to the inherent mathematical rigor as long as the basis functions exhibit invariance under similarity transformations, orthonormality, and completeness. Recalling the definition of the set of spherical harmonics, $Y_l^m(\theta, \phi)$, which are defined by:

$$Y_l^m(\theta, \phi) = N_l^m P_l^m(\cos\theta) e^{im\phi} \tag{4}$$

where P_l^m are associated Legendre polynomials and

$$N_l^m = \sqrt{\frac{2l+1}{4\pi} \cdot \frac{(l-m)!}{(l+m)!}} \tag{5}$$

the 3D Zernike polynomials can be defined as:

$$Z_{nl}^m(x) = R_{nl}(r) Y_l^m(\theta, \phi) \tag{6}$$

The preceding equation can be rewritten in Cartesian coordinates as:

$$Z_{nl}^m(\mathbf{x}) = \sum_{\nu=0}^{k} q_{kl}^\nu |\mathbf{x}|^{2\nu} e_l^m(\mathbf{x}) \tag{7}$$

where

$$e_l^m(\mathbf{x}) = r^l Y_l^m(\theta, \phi) = c_l^m r^l \left(\frac{ix - y}{2}\right)^m z^{l-m} \sum_{\mu=0}^{\lfloor \frac{l-m}{2} \rfloor} \binom{l}{\mu} \binom{l-\mu}{m+\mu} \left(-\frac{x^2+y^2}{4z^2}\right)^\mu \tag{8}$$

with normalization:

$$c_l^m = c_l^{-m} = \sqrt{\frac{(2l+1)(l+m)!(l-m)!}{l!}} \tag{9}$$

and where the $\lfloor \cdot \rfloor$ operator denotes the floor operation and

$$q_{kl}^\nu = (-1)^k \cdot 2^{-2k} \sqrt{\frac{2l+4k+3}{3}} \binom{2k}{k} (-1)^\nu \binom{k}{\nu} \binom{2(k+l+\nu)+1}{2k} \binom{k+l+\nu}{k} \tag{10}$$

This allows the definition of the 3D Zernike functional moments to be written as:

$$\Omega_{nl}^{m} := \frac{3}{4\pi} \int_{|\mathbf{x}| \leq 1} f(\mathbf{x}) Z_{nl}^{m}(\mathbf{x})\, d\mathbf{x} \tag{11}$$

leading to the definition of a rotationally invariant descriptor vector given by:

$$F_{nl} := \|\Omega_{nl}\| = \sqrt{\sum_{m=-l}^{l} |\Omega_{nl}^{m}|^2} \tag{12}$$

Notably, using the orthonormality of Z_{nl}^{m}, the underlying function, f, can be reconstructed from the Zernike functional moments with:

$$\hat{f}(x) = \sum_{n=0}^{N_M} \sum_{l=0}^{n} \sum_{m=-l}^{l} \Omega_{nl}^{m} Z_{nl}^{m}(\mathbf{x}) \tag{13}$$

In similar spirit to the previous methodologies which use a gradient to expose areas of the surface exhibiting high degrees of terrain variation, a ZIG map can be constructed to expose points at which the mathematical definition of the surface significantly varies. This is done by implementing a sliding window across the entirety of the DEM and computing the Zernike invariant vector on the terrain within the window at that point. Subsequently, the ZIG map can be constructed by evaluating the expression:

$$\|\nabla f(\mathbf{x})\| = \|\nabla f(\Omega(F(\mathbf{x})))\| = \left\| \frac{\partial f}{\partial \Omega} \frac{\partial \Omega}{\partial F} \begin{bmatrix} \frac{\partial \mathbf{F}}{\partial x} & \frac{\partial \mathbf{F}}{\partial y} \end{bmatrix} \right\| \tag{14}$$

$$= \left\| \left[\sum_{n=0}^{N_M} \sum_{l=0}^{n} \sum_{m=-l}^{l} Z_{nl}^{m}(\mathbf{x}) \right] \begin{bmatrix} \hat{\Omega}_{0,0}^{0} & 0 & \dots & \dots & 0 \\ 0 & \hat{\Omega}_{1,1}^{0} & 0 & \dots & 0 \\ 0 & \hat{\Omega}_{1,1}^{1} & 0 & \dots & 0 \\ 0 & 0 & \hat{\mathbf{\Omega}}_{n,l} & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \hat{\mathbf{\Omega}}_{N,l} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{F}}{\partial x} & \frac{\partial \mathbf{F}}{\partial y} \end{bmatrix} \right\| \tag{15}$$

where $\frac{\partial f}{\partial \Omega}$ is a constant value and can be arbitrarily set to take the form

$$\frac{\partial f}{\partial \Omega} = \eta * \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}_{1 \times N_F} \tag{16}$$

such that N_F is the number of invariants produced using a Zernike moment order of N_M according to the following expression:

$$N_F = \sum_{n=0}^{N_M} \left\lfloor \frac{n - (n \bmod 2)}{2} \right\rfloor + 1 \tag{17}$$

and where the $(n, l)^{th}$ order 3D Zernike moment unit vectors are defined as

$$\hat{\mathbf{\Omega}}_{n,l} = \left[\frac{\Omega_{n,l}^{m}}{\|\mathbf{\Omega}_{n,l}\|} \right] \tag{18}$$

The spatial gradient for each invariant, $\begin{bmatrix} \frac{\partial F_i}{\partial x} & \frac{\partial F_i}{\partial y} \end{bmatrix}$, can be computed numerically to facilitate the generation of the ZIG map. It is important to note that the resultant ZIG maps and processing time are extremely dependent on the values selected for the Zernike moment order, voxel size, sliding window size, and sliding window step size. Fig. 1 displays the variance of the ZIG map given varying algorithmic parameter values. Qualitatively, these four parameters have a significant impact on the resultant ZIG map which largely affect the ability to extract consistently salient features. Empirical testing determined that computation time is slightly increased with a higher window size and moment order and significantly decreased with a smaller window step size and voxel size. Further, it was found that rotational invariance and robustness is improved slightly with a decreased moment order, moderately with a smaller window step size, and significantly with a smaller voxel size.

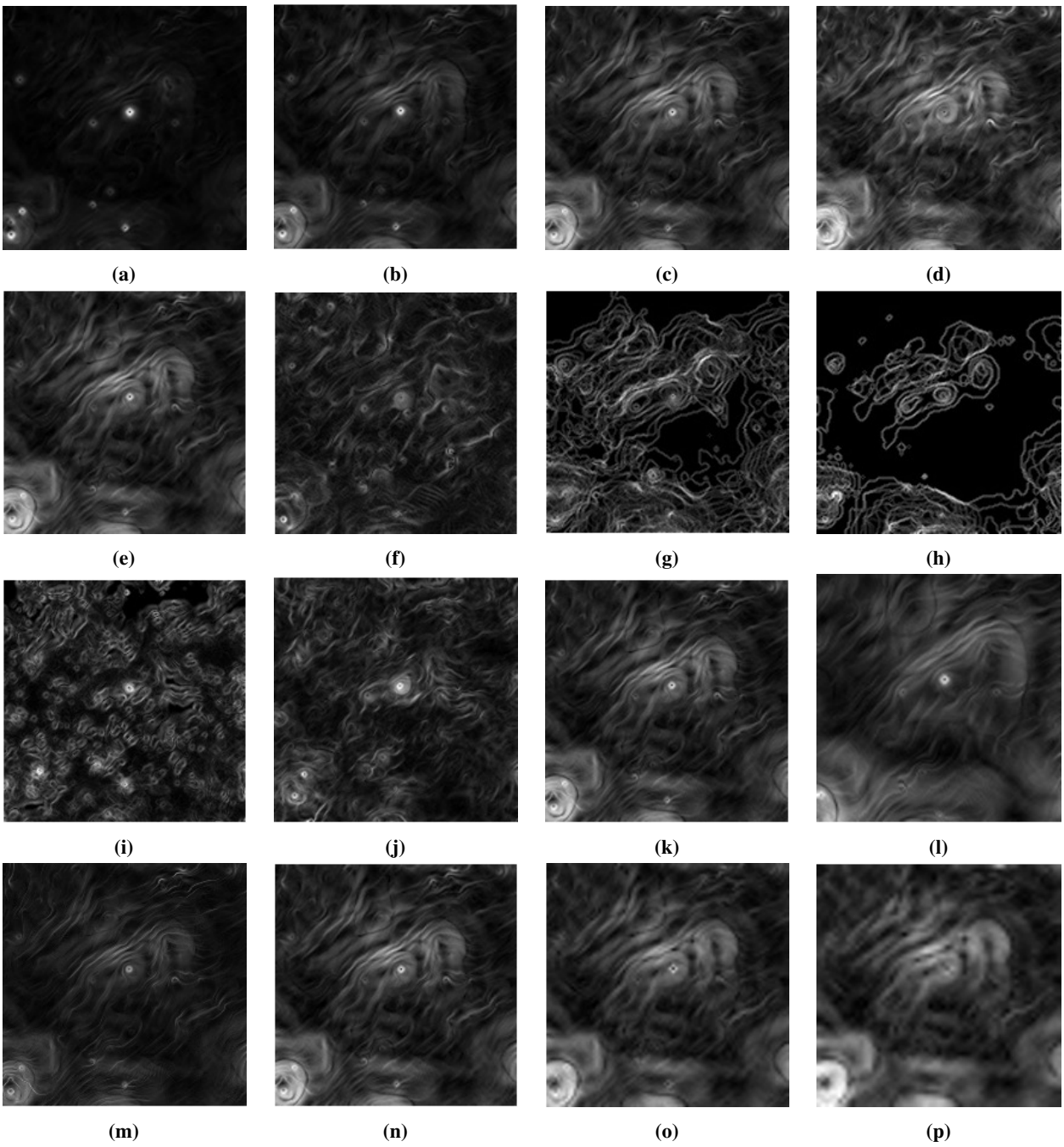
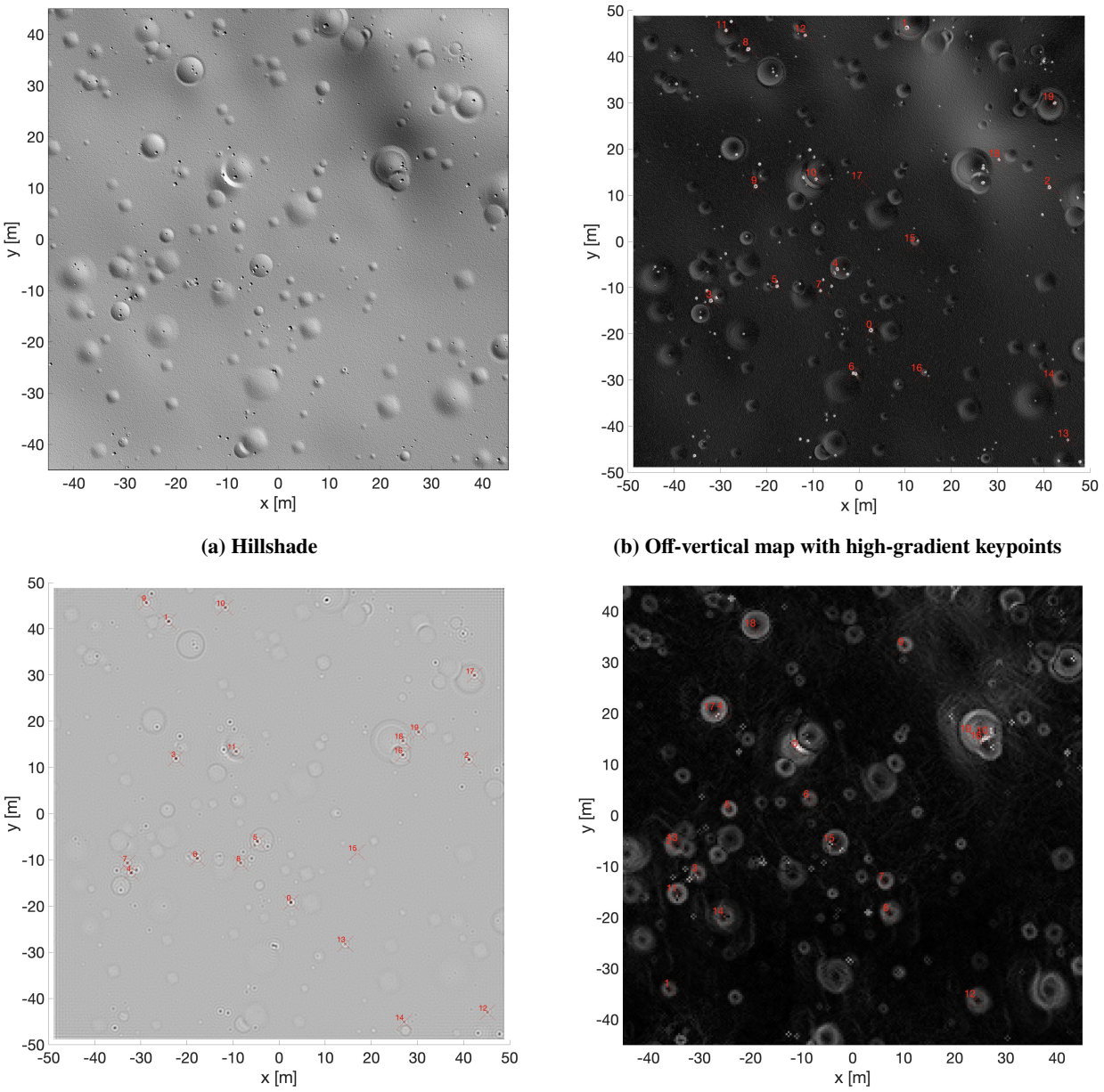


Fig. 1 Effects of varying Zernike Invariant Gradient (ZIG) map parameters: (a-d) increasing moment order, (e-h) decreasing voxel size, (i-l) increasing window size, (m-p) increasing window step size

Following the synthesis of the rotationally invariant 2D ZIG map, ZIG features are extracted using classical 2D feature detection algorithms such as SIFT [28], ORB [29], KAZE [30], etc.. Experimental testing suggested that KAZE features exhibited the most reliable detections with a relatively even spatial distribution; however, SIFT also yielded acceptable detection consistency. The 3D position of the detected 2D feature is obtained by corresponding the 2D pixel of the ZIG map with the original DEM.

Fig. 2 displays an example of keypoint detections using the high-gradient, Laplacian, and ZIG methodologies on a synthetically-augmented version of the Lunar South Pole DEM Site01, which generated using data from the Lunar Reconnaissance Orbiter (LRO) Lunar Orbiter Laser Altimeter (LOLA) [31]. The original DEMs were augmented by

the Simulation and Graphics branch at NASA JSC to enhance resolution and include high-fidelity rocks, boulders, and craters.



(a) Hillshade

(b) Off-vertical map with high-gradient keypoints

(c) Laplacian map with extracted keypoints

(d) Zernike Invariant Gradient map with extracted keypoints

Fig. 2 Keypoint extraction results on synthetically augmented Lunar South Pole DEM using (a) high-gradient, (b) Laplacian, and (c) ZIG map algorithms

C. 3D Feature Correspondence

Once salient 3D features have been extracted, descriptor vectors must be assigned to enable feature correspondence between another DEM facilitating simultaneous localization and mapping. There are many existing approaches for 3D feature descriptors including histogram-based methods, geometric surface statistics, spin images, and spectral techniques including Intrinsic Shape Signatures (ISS), Global Point Signatures (GPS), wavelet signatures, and heat kernel signatures. In addition to these approaches, the Zernike invariants can serve as a high-quality, unique, and

rotationally invariant descriptor vector. The highest performing descriptor vector was found to be a custom descriptor formed of data that described the off-vertical or laplacian maps (depending on the chosen method) surrounding the extracted 3D feature. This consisted of compiling the data in 2 concentric circles that surround the point and recording the mean, standard deviation of the data, as well as data distribution characteristics such as kurtosis and multiple percentile values. For the Zernike method, the descriptor vectors corresponding to the image keypoint detection method could also be used. To find correspondences between descriptor vectors in separate DEMs, the most successful method proved to be a Mahalanobis distance based approach where the covariance matrix was built based on the spread of the descriptor vectors.

Consider the set of keypoints for reference and active DEMs to be $\mathcal{K}_{\text{ref}} = \{k_i^{\text{ref}}\}_{i=1}^{N_{\text{ref}}}$ and $\mathcal{K}_{\text{new}} = \{k_j^{\text{new}}\}_{j=1}^{N_{\text{new}}}$, respectively. Each k^{th} keypoint is described by a 3D position, $\mathbf{p} \in \mathbb{R}^3$, and a descriptor vector, $\mathbf{d} \in \mathbb{R}^D$, where D is the descriptor vector length. The consolidated set of keypoints is expressed by:

$$\mathbf{F} = \{\mathbf{f}_i\} = [\mathbf{d}_1^{\text{ref}}, \ldots, \mathbf{d}_{N_{\text{ref}}}^{\text{ref}}, \mathbf{d}_1^{\text{new}}, \ldots, \mathbf{d}_{N_{\text{new}}}^{\text{new}}] \in \mathbb{R}^{D \times N_{\text{total}}}, i \in \{0, N_{\text{total}}\} \tag{19}$$

where $N_{\text{total}} = N_{\text{ref}} + N_{\text{new}}$. The mean descriptor vector can be computed via

$$\boldsymbol{\mu} = \frac{1}{N_{\text{total}}} \sum_{i=1}^{N_{\text{total}}} \mathbf{f}_i \tag{20}$$

To facilitate the computation of a Mahalanobis distance, the covariance matrix is constructed according to:

$$\mathbf{P} = \frac{1}{\max(1, N_{\text{total}} - 1)} \sum_{i=1}^{N_{\text{total}}} (\mathbf{f}_i - \boldsymbol{\mu})(\mathbf{f}_i - \boldsymbol{\mu})^T \tag{21}$$

To ensure invertibility, a regularization factor is applied to the covariance matrix as:

$$\mathbf{P}_{\text{reg}} = \mathbf{P} + \lambda \mathbf{I} \tag{22}$$

where $\lambda = 10^{-5}$ is the regularization parameter and $\mathbf{I}$ is the identity matrix. For each pair of keypoints $(k_i^{\text{ref}}, k_j^{\text{new}})$, the the squared Mahalanobis distance is computed via:

$$d_M^2(i, j) = (\mathbf{d}_i^{\text{ref}} - \mathbf{d}_j^{\text{new}})^T \mathbf{P}_{\text{reg}}^{-1} (\mathbf{d}_i^{\text{ref}} - \mathbf{d}_j^{\text{new}}) \tag{23}$$

To aid in the identification of the best match for each reference keypoint, first, the two closest candidate matches are identified by

$$j_1^* = \arg\min_j d_M^2(i, j) \tag{24}$$

$$j_2^* = \arg\min_{j \neq j_1^*} d_M^2(i, j) \tag{25}$$

To prevent outliers and ensure uniqueness, the following absolute and relative thresholds are applied. If both threshold checks pass, the candidate keypoint pair is considered to be a valid match.

$$d_M^2(i, j_1^*) \leq \tau_{\text{abs}} \tag{26}$$

$$\frac{d_M^2(i, j_1^*)}{d_M^2(i, j_2^*)} \leq \tau_{\text{rel}} \tag{27}$$

Alternatively, instead of corresponding individual features, geometric invariants of sets of points offer a rotationally and scale-invariant method to describe the relationship between 2 or more points in a set. Specifically, a triad geometric invariant matching algorithm is proposed by first generating triangles between each feature found, and combinations of 2 neighbors, ignoring degenerate triangles or excessively collinear point sets. Given a triad of 3D keypoints, $\mathbf{t}_i = \{\mathbf{p}_{i_0}, \mathbf{p}_{i_1}, \mathbf{p}_{i_2}\}$, a 7-dimensional descriptor vector can be constructed consisting of the three edge lengths, three interior angles, and area of the triangle formed by the three points. It is important to note that although only three independent geometric invariants exist for a set of 3 points, using this proposed 7D vector enhances uniqueness and facilitates more robust matching.

The first step in the triad geometric invariant matching method is building triplets of points for each keypoint, k_i, using its K nearest neighbors, where K is a pre-defined parameter. First, the distances from every keypoint, k_i, with position $\mathbf{p}_i$, to every other keypoint, k_j is computed with:

$$d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|, \quad j \neq i \tag{28}$$

The K nearest neighbors are selected to form:

$$\mathcal{N}_i = \{j_1, j_2, \ldots, j_K\} \tag{29}$$

where $d_{ij_1} \leq d_{ij_2} \leq \cdots \leq d_{ij_K}$. Then, for each keypoint, triads are constructed using pairs of its neighbors according to:

$$\mathcal{T}_i = \{(\mathbf{p}_i, \mathbf{p}_{j_m}, \mathbf{p}_{j_n}) : j_m, j_n \in \mathcal{N}_i, m < n, j_m > i, j_n > i\} \tag{30}$$

where the constraint $j_m > i, j_n > i$ prevents duplicate triangle generation. Once the triads are constructed, the three edge lengths are first computed with:

$$l_0 = \|\mathbf{e}_0\| = \|\mathbf{p}_1 - \mathbf{p}_0\| \tag{31}$$

$$l_1 = \|\mathbf{e}_1\| = \|\mathbf{p}_2 - \mathbf{p}_1\| \tag{32}$$

$$l_2 = \|\mathbf{e}_2\| = \|\mathbf{p}_0 - \mathbf{p}_2\| \tag{33}$$

To ensure rotation invariance of the length descriptors, the lengths are sorted such that $\mathbf{l}_{\text{sorted}} = \text{sort}([l_0, l_1, l_2])$. The area, A, and three interior angles are then computed via:

$$A = \frac{1}{2}\|\mathbf{e}_0 \times \mathbf{e}_1\| \tag{34}$$

$$\alpha_0 = \arccos\left(\frac{l_1^2 + l_2^2 - l_0^2}{2l_1 l_2}\right) \tag{35}$$

$$\alpha_1 = \arccos\left(\frac{l_0^2 + l_2^2 - l_1^2}{2l_0 l_2}\right) \tag{36}$$

$$\alpha_2 = \arccos\left(\frac{l_0^2 + l_1^2 - l_2^2}{2l_0 l_1}\right) \tag{37}$$

Similar to the side lengths, the interior angles are also sorted to ensure rotation invariance such that $\boldsymbol{\alpha}_{\text{sorted}} = \text{sort}([\alpha_0, \alpha_1, \alpha_2])$. The resultant triangle feature vector is then defined as:

$$\mathbf{f}_{\text{tri}} = [\mathbf{l}_{\text{sorted}}, \boldsymbol{\alpha}_{\text{sorted}}, A]^T \tag{38}$$

A triad feature is deemed to be valid if the area and smallest interior angle exceed predefined thresholds. The Mahalanobis distance matching method can subsequently be used using the point triad geometric invariant vector. Alternatively, a voting based method can be employed which leverages the following normalized relative distance metric which is computed for two arbitrary triad feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$:

$$s(\mathbf{f}_1, \mathbf{f}_2) = \sqrt{\sum_{k=1}^{7} \frac{(f_{1,k} - f_{2,k})^2}{\max(\epsilon, f_{1,k}^2)}} \tag{39}$$

where $\epsilon = 10^{-6}$ prevents division by zero. The iterative voting process starts by comparing all triads containing the reference keypoint, k_i^{ref}, to all triads containing each new keypoint, k_j^{new}. If $s(\mathbf{f}_{T_i^{\text{ref}}}, \mathbf{f}_{T_j^{\text{new}}}) < \tau_{\text{tri}}$, the pair of triads is recorded as a potential match, i.e. the match count, c_j is iterated. During this iterative process, the mean similarity score for every j^{th} new keypoint, k_j^{new}, is updated incrementally according to:

$$\bar{s}_j^{(n+1)} = \frac{c_j \cdot \bar{s}_j^{(n)} + s_{\text{new}}}{c_j + 1} \tag{40}$$

The j^{th} new keypoint, k_j^{new}, is considered a match if it maximizes the match count. Voting ties are broken by minimum mean similarity score. The candidate match (k_i^{ref}, k_j^{new}), is validated if:

$$c_j \geq c_{\min} \tag{41}$$

$$\text{keypoint } k_i^{\text{ref}} \text{ has at least } T_{\min} \text{ valid triangles} \tag{42}$$

$$\text{keypoint } k_{j^*}^{\text{new}} \text{ has at least } T_{\min} \text{ valid triangles} \tag{43}$$

where c_{min}, $T_{\min}$, and $T_{\min}$ are pre-defined parameters.

In both the descriptor-based and triangle-based matching methods, most of the true positives are usually found, but some false positives may also be included alongside some false negatives which are not included in the initial set of feature correspondences. As a result, a "reverse search" option is included in the feature matching process. The reverse search goes through each combination of 4 initial correspondence guesses, completes a calculation of the set's resulting geometric transformation, and keeps the feature match set and transformation with the lowest mean reprojection error, given it is of reasonable magnitude. This set is assumed to be certified true positives. Subsequently, the point correspondences are first tested from the list of initial guesses, where any set that fits the assumed true transform is kept in addition to the 4 assumed true matches. For a given point in one DEM, if a match is not found using the initial guesses, a more exhaustive manual search of combinations of correspondences is tested. This provides a good way to ensure that as many false negatives are found and it also has the consequence of removing the vast majority of false positives.

Following the generation of a candidate feature match set, it is critical to implement an outlier rejection technique to ensure the point sets that are passed along to a navigation filter or used to solve for a relative position and attitude are correctly matched. Fusing incorrectly matched points into a traditional navigation system can quickly cause filter divergence. The first step is to get sort any evident outliers. This is done by using all points to calculate a transform. This results in a transform that is more influenced by the true positives, which all push the resulting calculated transform towards the true transform, while false positives have transforms with contribute to the calculation in random directions. Once this rough transform is found, the subsequent reprojection errors for points below a certain standard deviation from the error distribution are kept as true matches. This removes the majority of false positives, making subsequent outlier rejection more effective. The next step is to utilize Random Sample Consensus (RANSAC). This ubiquitous technique to filter out outliers is imperative prior to solving for a pose or passing point sets to a navigation filter.

III. Localization and Mapping

Simultaneous localization and mapping (SLAM) is a well-studied area of research in the context of passive imaging systems; however, literature and programmatic implementations are sparse regarding reliable approaches to compute an accurate relative pose between DEMs using 3D feature sets and how to robustly fuse DEMs in the presence of uncertainty and differing ground sample distances or extents.

Given two sets of m corresponded 3D points, one set expressed in reference frame $\mathcal{A}$ and one set expressed in reference frame $\mathcal{B}$, which are related through the expression:

$${}^{\mathcal{B}}\mathbf{p}_i = T_{\mathcal{A}}^{\mathcal{B}\mathcal{A}}\,{}^{\mathcal{A}}\mathbf{p}_i + {}^{\mathcal{B}}\mathbf{t}_{\mathcal{B}/\mathcal{A}} \tag{44}$$

the following linear system can be solved to produce an estimate of the relative transformation between the $\mathcal{A}$ and the $\mathcal{B}$ frame and the associated uncertainty associated with the relative pose estimate [32]:

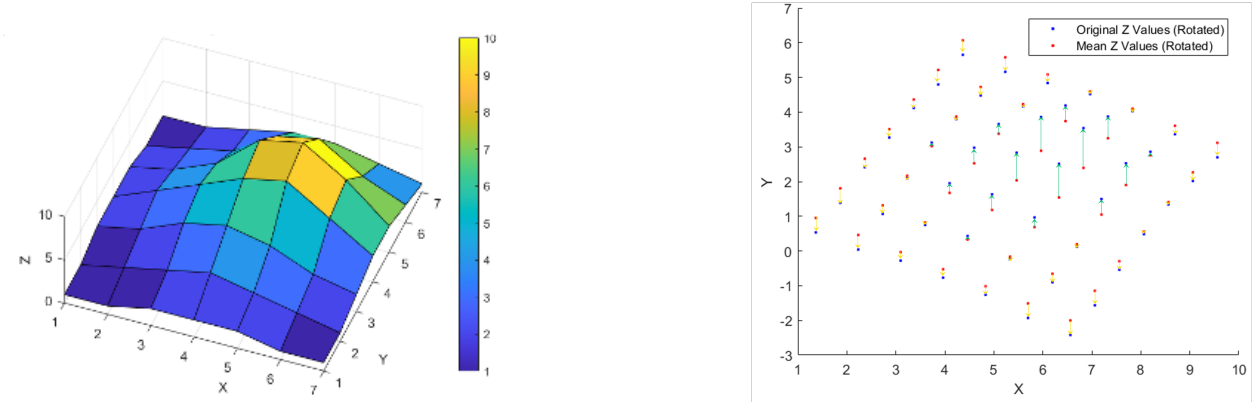
$$\begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_m \end{bmatrix} = \begin{bmatrix} \left[\mathbf{s}_1^{\times}\right] & I_{3\times3} \\ \vdots & \vdots \\ \left[\mathbf{s}_m^{\times}\right] & I_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{t}^* \end{bmatrix} \tag{45}$$

where $\mathbf{g}$ is the vector of classical Rodrigues parameters which maps the $\mathcal{A}$ frame to the $\mathcal{B}$ frame, the notation $[\cdot^{\times}]$ defines the skew-symmetric matrix formed by a 3-element vector, and

$$\mathbf{d}_i = {}^{\mathcal{B}}\mathbf{p}_i - {}^{\mathcal{A}}\mathbf{p}_i\,, \quad \mathbf{s}_i = {}^{\mathcal{B}}\mathbf{p}_i + {}^{\mathcal{A}}\mathbf{p}_i\,, \quad {}^{\mathcal{B}}\hat{\mathbf{t}}_{\mathcal{B}/\mathcal{A}} = \left(I_{3\times3} + \left[\hat{\mathbf{g}}^{\times}\right]\right)^{-1}\hat{\mathbf{t}}^* \tag{46}$$

Using this derived transformation, the DEMs can then be fused. Take the case where an incoming DEM is found to have some transformation with respect to a base DEM. The fusing process is completed by turning the incoming DEM into a point cloud, rotating the point cloud using the derived transformation, and interpolating within this point cloud based on the base DEMs' geometry.

It is important to note that this fusing process is straightforward when fusing two DEMs where the reference planes are parallel, but less so for non-coplanar DEMs. In such cases, any deviation in the incoming DEMs' elevation direction result in deviations when projected in the base DEMs' reference plane. This results in the points not forming some square or parallelogram basis, making bilinear interpolation inappropriate. Fig. 3 shows the result of a DEM with an out-of-plane rotation projected onto a new arbitrary reference plane. A mean plane which is parallel to that DEM plane set to a constant elevation value is then subject to the same out-of-plane rotation. As shown, the points representing the mean plane form a consistent parallelogram basis. However, the points which comprise the rotated DEM, when projected, do not. The elevation deviations in the DEM's out-of-plane axis result in in-plane deviations when rotated. This means that a square or parallelogram basis is not kept. Additionally, in places of high terrain change, corresponding points can translate rather drastically compared to the projection of the mean plane. Ultimately, this precludes the correct interpolation points from being easily identified.

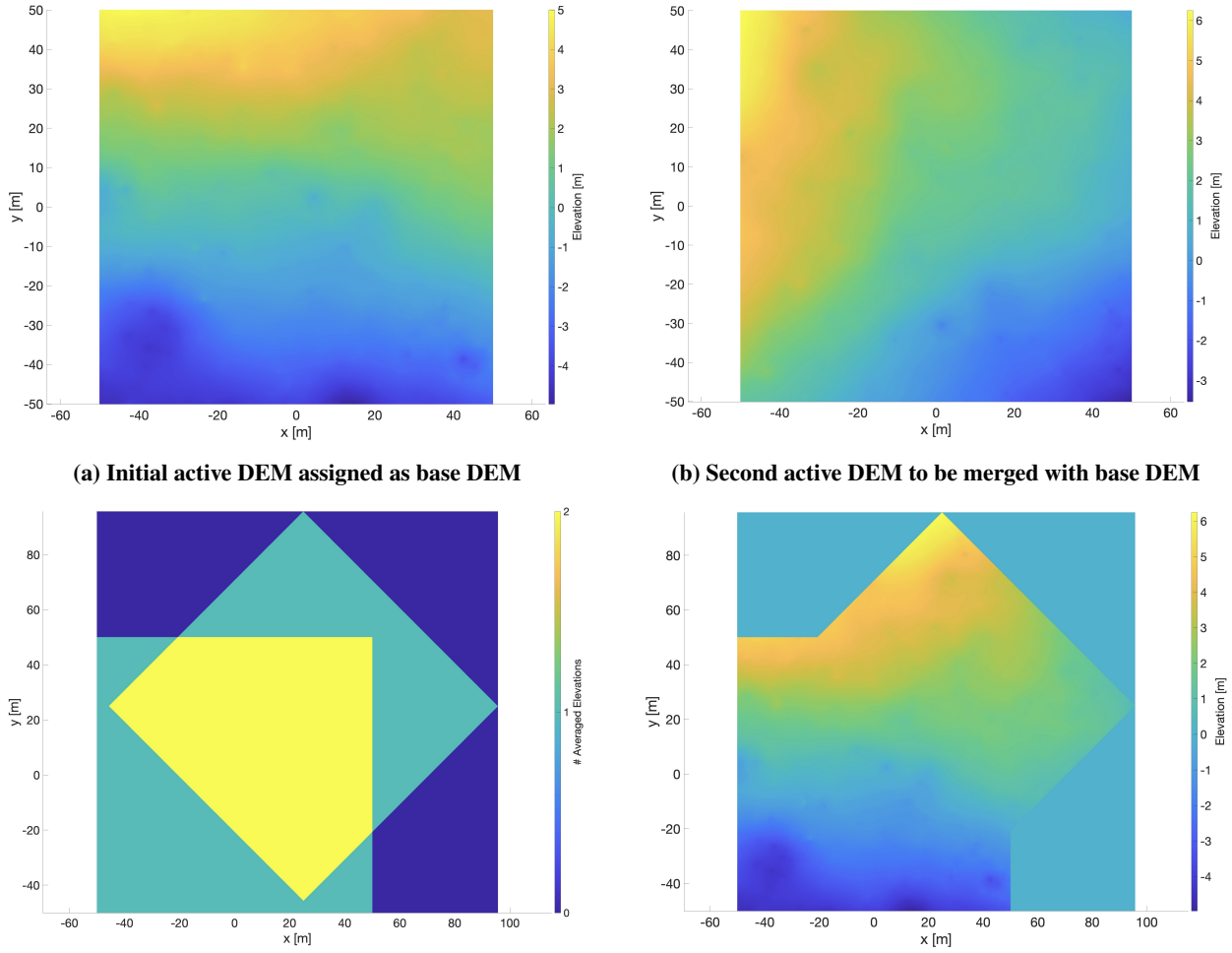


(a) Example digital elevation map with large slope

(b) In plane difference between rotated mean elevation plane and true values expressed in rotated basis' lateral plane

Fig. 3 Lateral distortion with respect to new base plane induced by out-of-plane rotation applied to DEM with largely varying elevations

A solution was developed to the problem of finding the closest set of base DEM points to leverage for the interpolation. With the known desired interpolation point (not pictured in Fig. 3 but would be gridded along the new basis), the nearest mean grid point (red) can be found easily. Then, the error vector is found between that mean grid point and its corresponding rotated point (in blue). By subtracting that error vector from the nearest mean grid point, the approximate mean grid point whose corresponding rotated point is most likely nearest to the desired interpolation point can be found, along with the 4 closest neighbors to interpolate upon. This interpolation can be completed using methods such as Radial Basis Function (RBF) interpolation or Shepard's Interpolation. The resulting product would be similar to the DEM shown in Fig. 4. The merged DEM would then become the new base DEM for comparison with subsequent incoming active DEMs.



(a) Initial active DEM assigned as base DEM

(b) Second active DEM to be merged with base DEM

(c) Resultant base DEM merging weights

(d) Resultant merged DEM without interpolation artifacts

Fig. 4 DEM merging process flow instituting overlapping elevation averaging and optimal interpolation point search algorithm

IV. Flight Software Library Design and Implementation

There are two major components to the software system used to demonstrate how these techniques are utilized during integrated flight: the algorithm library and the flight software interface application that composes the library functions in a way that can be leveraged by an existing system. The architecture is shown in detail in Fig. 5.

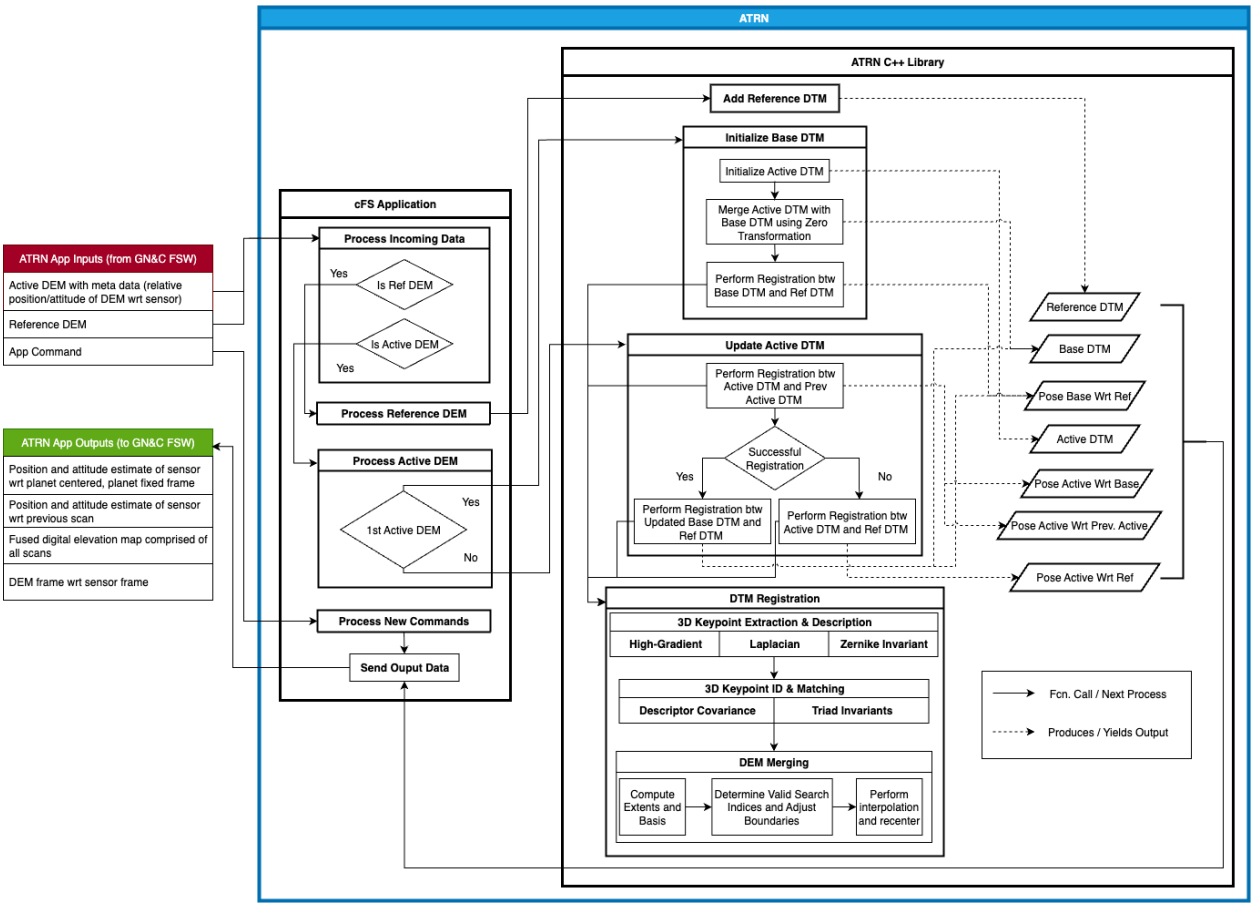


Fig. 5 High-level active terrain relative navigation C++ library and flight system interface software architecture

A. ATRN Algorithm Library

The pre-processing, extraction, and matching steps are encapsulated into functions and packaged into a software library. This library provides an entry point for adding a reference DEM for processing, then adding subsequent active DEMs as they are acquired. When the first active DEM is acquired, it is set as the base DEM, and registration is performed between this base DEM and the reference DEM, producing a pose estimation of the base DEM relative to the reference DEM. The method of keypoint extraction and matching used during this registration step are compile-time configuration properties of the library. As subsequent DEMs are acquired, registration is performed between the current and previous DEMs, resulting in a relative pose estimate between the two DEMs. This in turn is used to create an updated base DEM and relative pose estimate between the base DEM and reference DEM.

B. Flight Software Application

The application is intended to interface with the NASA core Flight System (cFS) [33]. In particular, it interfaces with an existing instance of cFS built to demonstrate precision landing technologies which includes a interface application for the HDL and a navigation application. The application initializes library with the reference DEM after loading it from an onboard file. It receives a notification via the cFS software bus when new LiDAR scans are available and retrieves them from memory. This data is formatted to the specifications of the library and the library state is updated using this data as the new active DEM. After each active DEM is processed, the resulting data, including the combined base DEM, relative pose between scans, and relative pose between the base DEM and reference DEM. In particular, this information can be used to produce a pose estimate of the sensor with respect to the planet-centered, planet-fixed frame, as well as the frame of the DEM with respect to the sensor frame. These estimates are then sent on the cFS software bus to be consumed by the navigation application, which can use them to improve its solution.

C. Resource Optimization

Several optimizations were necessary in order to keep runtime within a reasonable limit, as well as adhere to internal software standards meant to ensure reliability for flight use.

1. Memory Management

One salient detail of the internal software standard applied to this code that is that dynamic memory allocation after initialization is prohibited for safety-critical code, and discouraged in any real-time code. This presented a particular challenge when working with large DEMs. The Eigen linear algebra package was leveraged for many of the operations within the algorithms [34], and the DEMs were internally represented as Eigen matrix types. Both dynamically sized and fixed-sized matrices can be defined with Eigen, but without special care, dynamic matrices utilize heap allocations to expand to fit the data that is placed within them. Using fixed-sized matrices mitigates this problem, but in the case of this application, although the necessary sizes can be computed a priori, many of the DEMs and intermediate structures used in computation exceed stack size limits. To mitigate this, generic programming (templates in C++) were used to define appropriately-sized buffers at compile-time to appear in the block starting symbol (bss) segment of the program binary. The `Eigen::Map` type was used to place matrices of both the dynamic and fixed-sized types inside these pre-allocated memory areas and prevent hidden allocation. The `Eigen::Ref` type was used where necessary to allow functions to accept matrices of varying sizes as well as avoid temporary copies of matrices. The `EIGEN_RUNTIME_NO_MALLOC` macro was used to ensure the assumptions about memory allocation were correct. This macro causes assertions to be called and can halt program execution, so it is only appropriate to define for debugging builds of the software.

2. Vectorization

Modern computer processors, including those planned for the NASA High Performance Spaceflight Computer (HSPC) [35], of which the Descent and Landing computer used is intended to a surrogate platform for [36], utilize a technique called single instruction, multiple data (SIMD) execution to increase throughput. At various points in the algorithms, operations in loops were reordered to remove branches and iteration dependencies to enable auto-vectorization by the compiler, which will then emit SIMD instructions. Memory alignment was manually specified for the `Eigen::Map` buffers holding dynamic matrices so that groups of elements could be packed into a single word for SIMD compilation. These changes, paired with the appropriate compiler optimization settings, improved runtime performance by orders of magnitude.

D. Matching Redundancy

Finally, in the full loop implementation, it was found that redundancy may be increased by completing both triangle-based matching and descriptor-based matching. In different situations with smaller amounts of overlap, certain factors may cause one method to perform better than another. As a result, matching and a relative transformation can be found using both, where the best transform can be chosen based on the reprojection error of the points used and the number of inliers used to calculate the transformation.

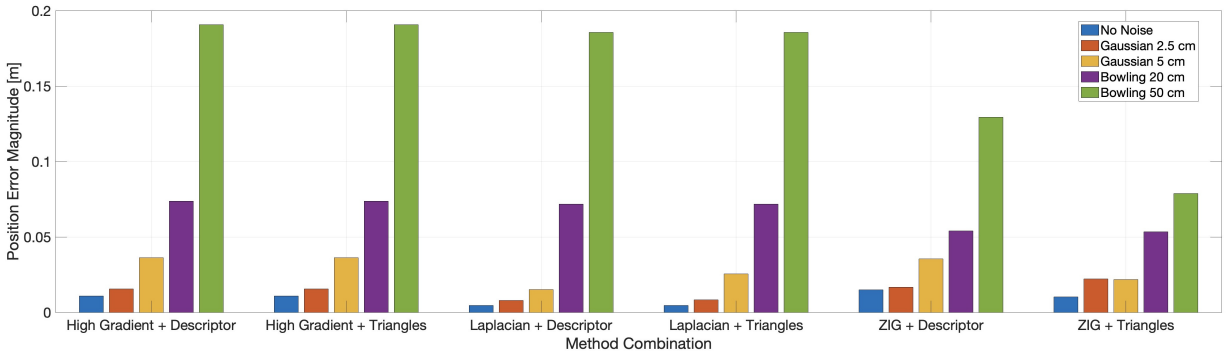
V. Performance Evaluation and Experimental Testing

An extensive verification and validation campaign was undertaken to evaluate the performance of the proposed ATRN methodologies. Following parameter sweep testing to identify optimal values for parameters which drive performance for each of the three methods (results not included in this work), the first primary objective of the campaign aimed to determine each method's sensitivity to common artifacts exhibited by terrain maps obtained by various terrain mapping sensor technologies including random Gaussian elevation noise, map bowling induced by spiral scan patterns or surface velocity assumptions, and lateral distortion induced by erroneous ground plane definitions. Furthermore, each method's ability to consistently match features between maps of varying overlap fractions was evaluated. Baseline performance for each keypoint identification approach was evaluated using high-resolution synthetic DEM data. Next, each of the proposed methods were applied to real flight data obtained by the NASA GSFC Hazard Detection LiDAR (HDL) during helicopter flight testing at Kennedy Space Center's hazard field. Finally, the ATRN system was deployed onto representative space flight hardware, specifically the Descent and Landing Computer [37], to evaluate runtime and memory performance.

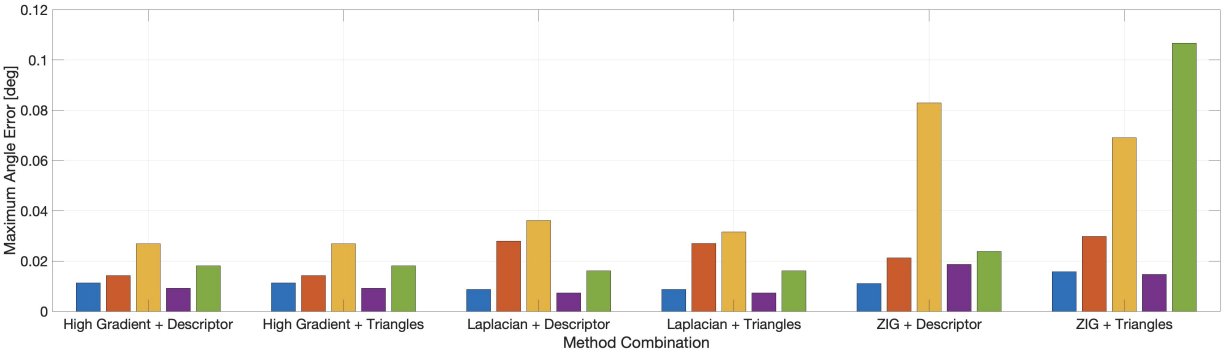
A. ATRN Sensitivity Study

There are several common artifacts exhibited by DEMs produced by terrain mapping sensors which can drastically impact the ability to leverage the DEM to perform navigation and hazard detection activities. These artifacts include, but are not limited to, elevation noise, lateral distortion, and radial distortion (bowling). To evaluate the impacts on the reliability and performance of the proposed methodologies, several high-resolution synthetic DEMs were constructed exhibiting varying levels of each of these artifacts using a subset of the LM1 and LM4 landing site DEMs from the LOLA Lunar South Pole data set. Elevation noise was varied from 0cm to 5cm (1σ), bowling was applied resulting in both 20 and 50 cm of total elevation difference between the center point of the DEM and an edge pixel, and assigned elevation planes were varied from 0-45°. Fig. 6 displays position, attitude, and 3D point reprojection errors of the high-gradient, Laplacian, and ZIG 3D feature identification and matching methods under varying noise and bowlings to an otherwise unaltered DEM. Point reprojection error is critical to characterize the uncertainty of a 3D point-pair prior to being processed by a navigation filter and is defined as the Euclidean distance of a matched feature projected into the base DEMs reference frame. It is important to note that the DEMs used for this test exhibited no translational difference.

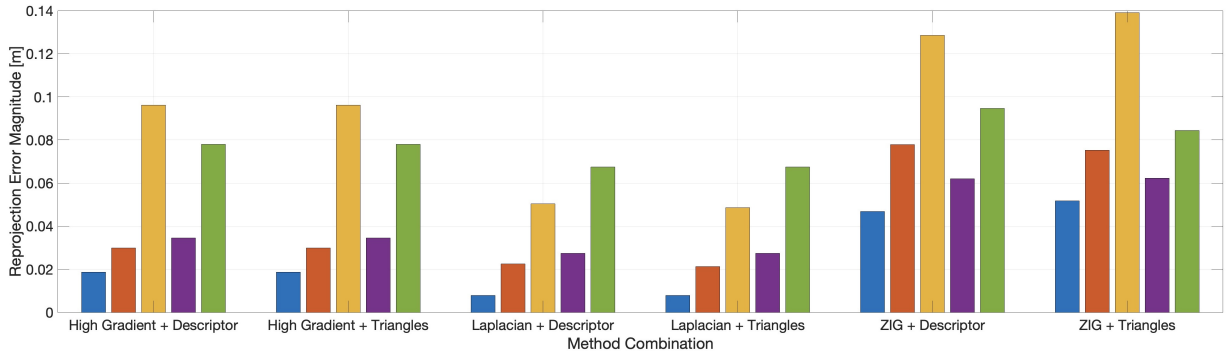
From this data, it can be seen that radial distortion resulted in the largest impact in the estimated relative position, although even in the largest cases, the highest error found was less than 20 centimeters. The Zernike method seems to minimize such errors in radial distortion somewhat effectively. When analyzing the estimated attitude errors, it was found that Gaussian noise has the largest influence, and the largest angular component was consistently the out-of-plane rotation. Finally, reprojection errors increase noticeably with gaussian noise for similar reasons. The Zernike method may inherently result in larger reprojection errors across the board due to information loss when creating the ZIG map, which will have a lower resolution compared to the original map. Ultimately, in the context of inertial navigation, these errors are found to be quite reasonable given the noise applied.



(a) Position error magnitude



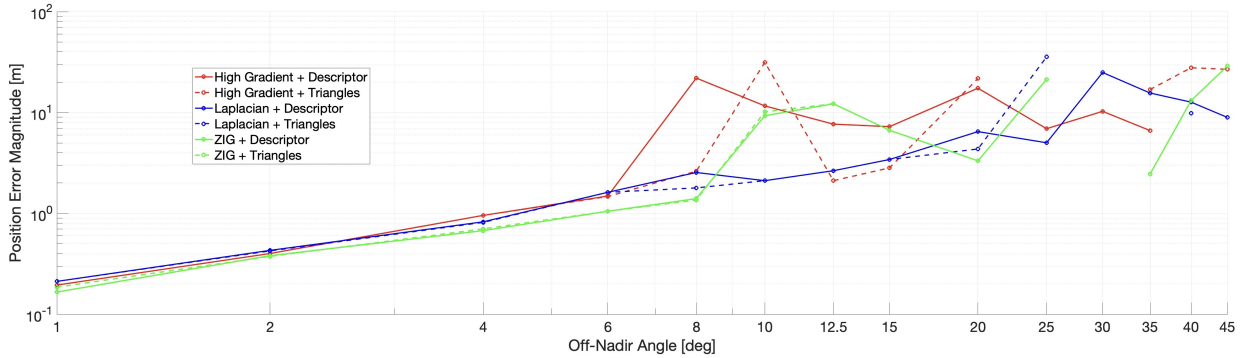
(b) Maximum angle error



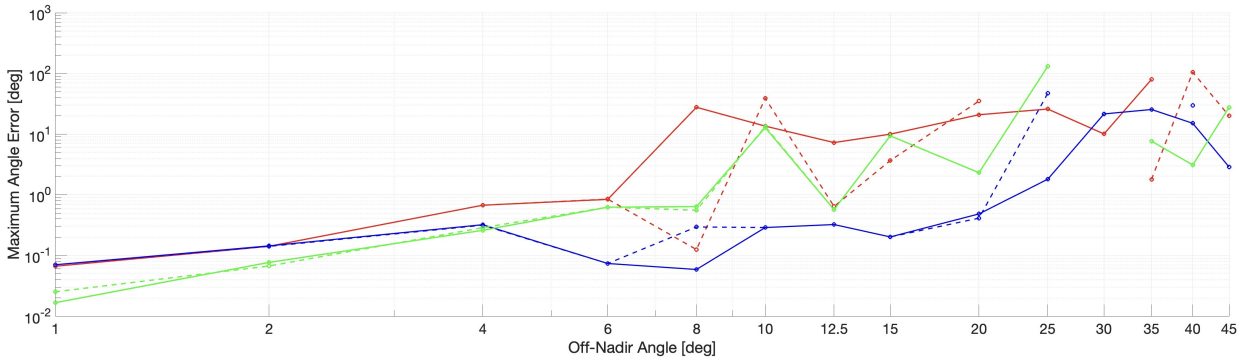
(c) RMS reprojection error for all keypoints

Fig. 6 Pose estimation errors of the high-gradient, Laplacian, and ZIG 3D feature ID methods under varying noise and bowling

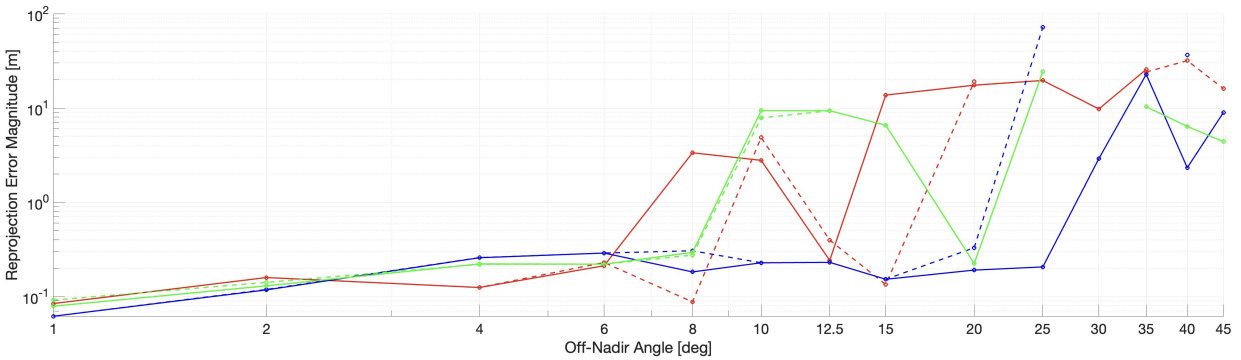
Fig. 7 displays the estimation and reprojection errors given differing ground plane definitions between the DEMs under test: an assumption which induces lateral distortion in the gradient and Laplacian maps. It is important to note that missing lines/data points correspond to an instance when a valid solution was not found. The errors are found to scale linearly until 6°-8° of ground plane difference. After this point, the errors become more sporadic, indicating a higher level of sensitivity to extremely non-coplanar ground plane definitions. The high-gradient method seems to perform the worst, followed by the Zernike method, with the Laplacian method proving quite robust. This is due to the fact that the Laplacian method inherently analyzes radii of curvature, which are more consistent across DEMs with varying ground-plane definitions.



(a) Position error magnitude



(b) Maximum angle error

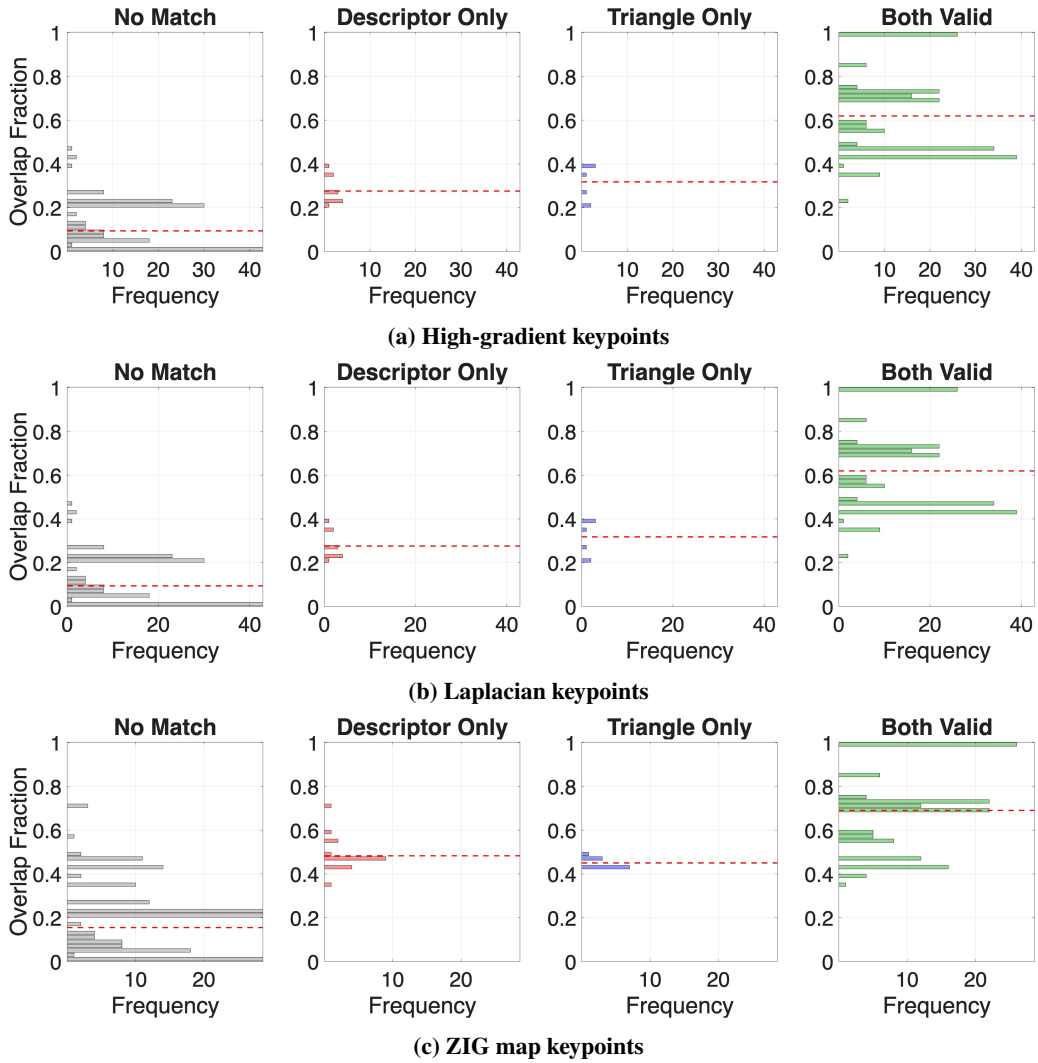


(c) RMS reprojection error for all keypoints

Fig. 7 Pose estimation errors of the high-gradient, Laplacian, and ZIG 3D feature ID methods under varying lateral distortion induced by incorrect ground plane definitions

In the scenario where the proposed ATRN system will be employed on succsesively-acquired terrain maps, it is important to understand the amount of DEM overlap required to produce reliable and accurate 3D feature matches which result in an accurate pose estimate to drive sensor pointing constraints. To assess the overlap cutoff threshold of each 3D feature ID method, synthetic DEMs with overlap fractions ranging from 10%-100% with varying relative in-plane orientations were sampled and processed. A 1.0 cm (1σ) Gaussian noise without distortion was applied to the DEMs for this test in an attempt to emulate a realistic scenario [38]. Fig. 8 displays the frequency of finding successful transformations using the high-gradient, Laplacian, and ZIG 3D feature identification and matching methods given varying levels of DEM overlap. All three methods prove to be relatively successful with high overlap in DEMs. In the case of the high gradient and Laplacian methods, largely due to the implementation of windowed keypoints mentioned in Section II.B, the true transformation was shown to be found in all cases where the DEM overlap was greater than 50%. Additionally, these methods proved to be quite successful at lower overlap amounts too, demonstrating consistently

successful performance up to about 30% of overlap. In some edge cases between 40% and 50% of overlap, it was found that the DEMs used were sampled from quite flawed regions of the data source. Nonetheless, these same DEMs were still successful in the majority of cases. From the data, it is clear that some cases were only successful when using either the descriptor based matching or the triangle based matching. This was the motivation behind completing these methods in parallel and finding the solution with the best indication of a successful transform, as mentioned in Section IV.



(a) High-gradient keypoints

(b) Laplacian keypoints

(c) ZIG map keypoints

Fig. 8 Frequency of succesful pose estimates (RMS error < 2m) of keypoint ID methods given varying DEM overlap fractions

In addition to analyzing performance, it is imperative to be able to determine the quality of a transform when not given a truth estimate to compare to. As a result, for the testing above, plots were made to compare the reprojection error and overlap fraction against the computed transform's actual position error, as seen in Figure 9. Found transforms with a successful found transformation, determined roughly for analysis to be a true position error of less than 1 meter, were colored in green. Similar to the above analysis, all three methods are capable of robustly finding a successful transform, but the high gradient and Laplacian methods are much more consistent, especially when the performance of the Zernike method starts to degrade. More importantly from this data, from the graphs on the right, it is clear that a threshold for reprojection error can be found to separate successful and unsuccessful transforms. This threshold is more distinct and defined for the high gradient and Laplacian methods. Being able to discern a good transformation from bad one is crucial to prevent sending a potentially incorrect transformation to a navigation filter.

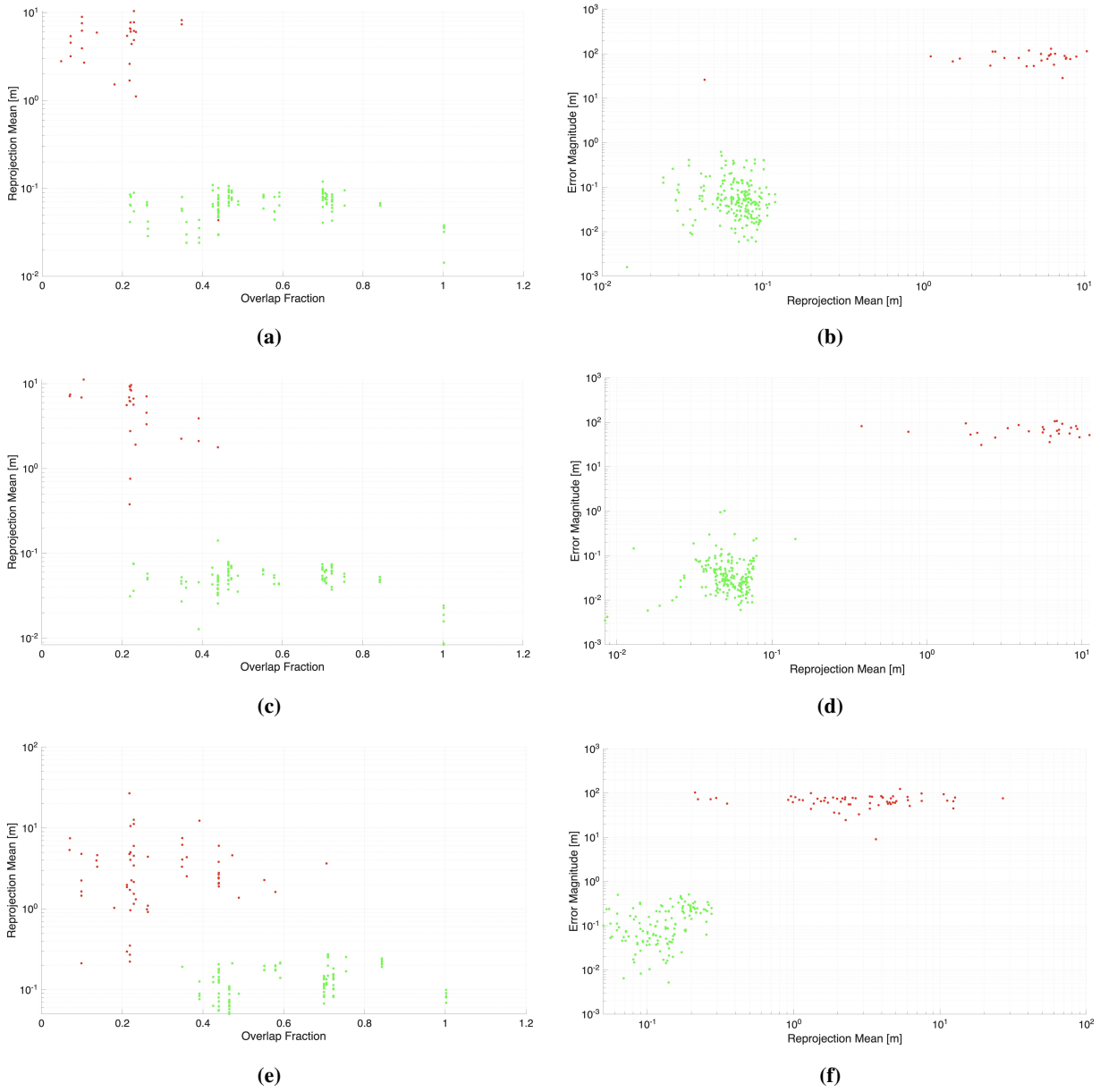
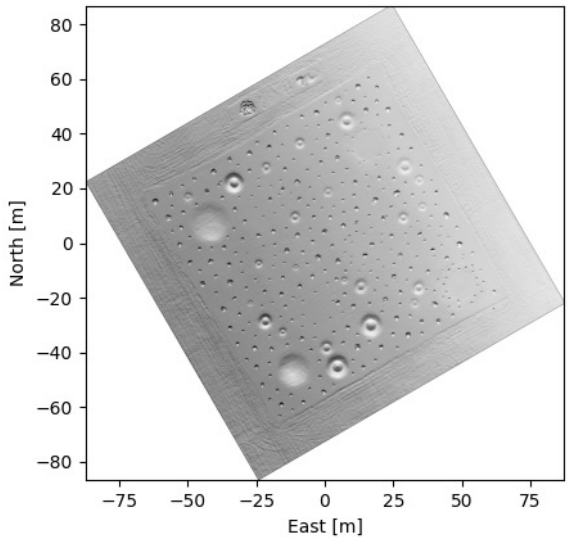


Fig. 9 (Left) RMS point reprojection errors as a function of overlap fraction and (right) position error magnitude as a function of RMS point reprojection errors for (a-b) high-gradient, (c-d) Laplacian and (e-f) ZIG map methodologies

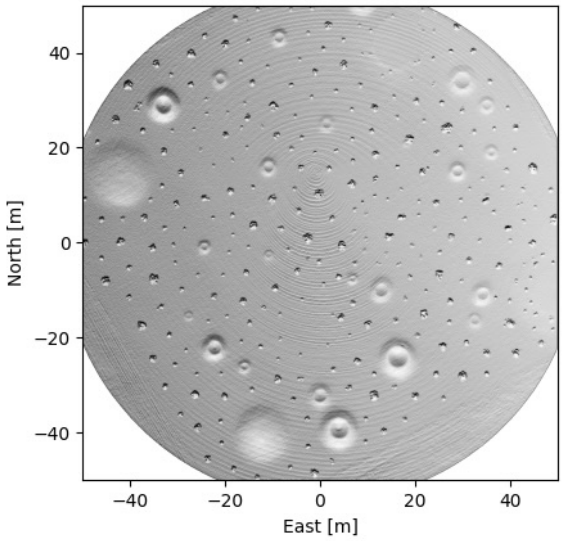
B. ATRN System Analysis with Flight Data

Evaluating the ATRN algorithms with flight data from the GSFC HDL revealed the criticality of several components of the proposed system. For example, the resulting DEMs occasionally had out-of-place elevation values, which would get exaggerated in an off-vertical or Laplacian calculation. In addition, the HDL's circular scanning pattern became increasingly evident on off-vertical and Laplacian maps, though most of this would become fixed with the Fourier Transform low pass filter discussed previously. However, the largest benefit of the low pass filter is evening out small-scale terrain changes. In particular, the Kennedy Space Center's hazard field is created with rock piles in an otherwise flat terrain area. This results in sharp terrain changes at the edges of the rock piles. Depending on the resolution of the sampled DEM, these terrain changes then appear as different radii of curvature, having a particularly

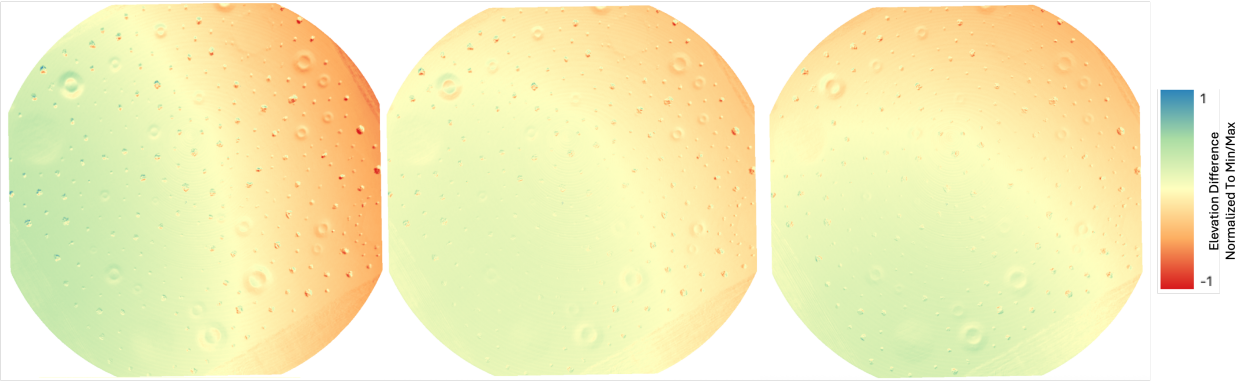
large impact on the Laplacian map. Thus, without DEM pre-conditioning, performance results were extremely degraded. Fig. 10 displays the difference maps between the high-resolution survey DEM and the transformed HDL DEM given the results yielded by the high-gradient, Laplacian, and ZIG methods. The survey DEM exhibits a 7mm ground sample distance, and a 1.6/2.2 cm 1σ lateral/elevation uncertainty, respectively. It is important to note that the difference maps have been normalized by the high-gradient's minimum and maximum, which yielded the largest magnitude errors. The majority of the high-gradient difference map stemmed from an inaccurate relative attitude estimate, as indicated by the left-to-right gradient. Whereas the Laplacian and ZIG difference maps seem to exhibit a more radial error distribution. This could be attributed to slight bowling of the HDL DEM induced by linearizations employed in the firmware algorithms or parameters used during this first field test campaign [39]. Due to the lack of a 'truth 'pose, transformation errors are unable to be computed. All three difference maps had a mean error of less than 1cm, and the maximum absolute value of the high-gradient's difference map is 3.22σ.



(a) High-resolution survey DEM of NASA KSC hazard field



(b) Real-time DEM of NASA KSC hazard field acquired by GSFC HDL



(c) Normalized difference maps using transformations produced by (left) high-gradient, (middle) Laplacian and (right) ZIG map methodologies

Fig. 10 ATRN system analysis using HDL helicopter flight data

C. Computational Peformance

Two of the most challenging aspects of the problem of DEM based localization is memory management and computation time. Handling and processing digital elevation maps are inherently computationally expensive. The underlying size of the DEM drastically increases the amount of time and memory required for processing. Further,

the parameters chosen for the ZIG method have a significant impact on performance; therefore it was important to understand the computational implications of varying parameter sets. Table 1 presents the end-to-end keypoint extraction, identification, matching, and DEM transformation processing time for the high-gradient, Laplacian, and multiple configurations of the ZIG methodologies when run on an desktop workstation when processing two 1000×1000 DEMs. The workstation utilizes an Intel Xeon CPU E5-2687W v4 processor set to run at the maximum 3.5 GHz clock speed. The high-gradient, Laplacian and best performing ZIG (in terms of computation time, robustness, and accuracy) methods require all require roughly the same amount of time. The data in Table 1 also suggests that the ZIG parameter values do not have a large impact to runtimes; however, empirical testing did indicate pose accuracy and robustness are significantly effected by the choice of the configuration parameter values. It is important to note that all the results presented in this section for the ZIG algorithm were generated using the parameter set listed in the last row in Table 1.

Table 1 3D keypoint ID and DEM transformation desktop runtimes on 1000×1000 DEMs

Method	ZIG Window Size	ZIG Window Step Size	ZIG Voxel Size	Zernike Order	Run Time [s]
High-gradient	-	-	-	-	0.190
Laplacian	-	-	-	-	0.176
ZIG	8	0.6	0.3	3	0.520
ZIG	10	0.6	0.3	3	0.499
ZIG	15	0.6	0.3	3	0.497
ZIG	15	0.8	0.3	3	0.490
ZIG	10	0.8	1.0	3	0.479
ZIG	10	1.0	0.5	3	0.191

Table 2 presents end-to-end processing times for the high-gradient, Laplacian, and ZIG approaches on the same desktop machine but processing two 3500×3500 DEMs. It is clear that the processing times do not simply scale linearly with the size of the DEM. The ZIG method scales with DEM size better than the high-gradient and Laplacian methods. This important characteristic can be attitbuted to the inherent data dimensionality reduction employed by the ZIG map methodology. In contrast to the high-gradient and Laplacian methods which operate on the entire DEM, the ZIG map is computed over a sliding windowed version of the underlying DEM wihtout loss of information. It is important to note that similar to the HDL, many descent and landing concepts of operations are considering DEMs with 1000×1000 rows/columns to inform localization and hazard detection/avoidance for planetary landing; therefore, it is important to be able to processed these sized DEMs in real-time. Whereas reference DEMs, typically with sizes >3000×3000, can be processed pre-flight.

Table 2 3D keypoint ID and DEM transformation desktop runtimes on 3500×3500 DEMs

Method	ZIG Window Size	ZIG Window Step Size	ZIG Voxel Size	Zernike Order	Run Time [s]
High-gradient	-	-	-	-	15.7
Laplacian	-	-	-	-	15.6
ZIG	10	1	0.5	3	3.29

To ensure the proposed ATRN system was suitable for embedded computers that could be used in spaceflight, the FSW library was optimized and deployed onto the ARM-based SPLICE DLC [36] and an Intel NUC mini-PC for static testing. These systems are meant to be representative of potential future spaceflight computing platforms, depending on size, weight, and power constraints. The DLC utilizes an AMD Zynq Ultrascale system-on-a-chip with an ARM Cortex A53 1.2 GHz processor, while the NUC utilizes an 11th-gen Intel Core i7 2.8 GHz processor with dynamic frequency scaling up to 4.7 GHz. Tables 3 and 4 display the end-to-end computation times for all three methods for various sized DEMs deployed onto the DLC and Intel NUC, respectively. It is important to note that the ZIG parameter set for these tests were the same as those used in Table 2. This data reveals the same relative computational performance trends between the high-gradient, Laplacian, and ZIG methods. For smaller DEMs, all three methods take around the same amount of time. However, the ZIG method scales much more efficiently for larger DEMs than the high-gradient and Laplacian methods. Run times on the Intel NUC were very similar to the high-powered desktop machine due to the

fact that this program does not currently avail itself of multiple cores. Future work will implement identified software updates to take advantage of multi-core processing and evaluate the impact of balancing it with computational demands from other flight software components.

Table 3 3D keypoint ID and DEM transformation runtimes on the DLC

Method	DEM Sizes (Rows×Columns)	Run Time [s]
High-gradient	1000×1000	2.94
Laplacian	1000×1000	2.94
ZIG	1000×1000	3.28
High-gradient	3500×3500	545
Laplacian	3500×3500	545
ZIG	3500×3500	46.9

Table 4 3D keypoint ID and DEM transformation runtimes on an Intel NUC

Method	DEM Sizes (Rows×Columns)	Run Time [s]
High-gradient	1000×1000	0.125
Laplacian	1000×1000	0.118
ZIG	1000×1000	0.140
High-gradient	3500×3500	20.3
Laplacian	3500×3500	20.0
ZIG	3500×3500	2.99

VI. Conclusion

Three novel 3D feature extraction and description methods, two robust matching algorithms, and an end-to-end flight software library was developed and presented in this work. The high-gradient keypoint extraction and description method proved to be effective at finding the true transformation between two DEMs across a range of noise conditions, overlap amounts, and with real-world data. To guarantee rotational invariance, the Laplacian method was developed which proved to be similarly as successful and robust as the high-gradient method. It was found that for DEMs with differing ground-plane definitions, the Laplacian method was more successful at minimizing performance effects for DEMs with larger angle differences. To avoid the exacerbation of sensor noise and guarantee rotational invariance, ZIG features were developed which can be described by any of the 3D feature description methods or traditional 2D feature vectors. Although it was found that this method was not as successful as the other methods in cases such as smaller amounts of overlap between DEMs and inherently may introduce more errors with the lower resolution of the ZIG maps, it was still found to be reasonably successful The descriptor vector and 3D point triad geometric invariant matching methods paired with classical outlier rejection algorithms are run in parallel to maximize the number of matched features yielding a higher confidence relative position and attitude between two DEMs. The resulting transformations can then be evaluated, giving a high confidence in the ability to detect a bad transformation in-flight.

Computational and navigation performance evaluations indicate that the proposed work is sufficient to address the requirements levied by upcoming precision landing and surface mobility missions and will directly enable repeatable precise and safe landing operations in any environment. The position errors produced by all methodologies when analyzing both synthetic and real flight data were consistently on the order of the ground sample distance of the DEMs under test. The angular errors were consistently less than 0.15° even when the DEMs were augmented with high-valued noise, bowling, and lateral distortion. With regards to computational performance, all three methods take essentially the same amount of time when processing 1000×1000 pixel DEMs. However, the ZIG algorithm is much more efficient than the high-gradient and Laplacian approaches when processing larger DEMs. Table 5 presents a comparitive analysis between the three proposed 3D keypoint identification algorithms with optimized configuration

parameters over the entire test set on the Intel NUC. The Laplacian method is considered the best overall given its low algorithmic complexity, minimum required overlap fraction, and error statistics. Based on run-times, if the DEMs to be processed are much larger than 1000×1000, the ZIG method is the most practical approach while still maintaining the same order of magnitude of errors offered by the high-gradient and Laplacian approaches. The proposed flight software library was deployed and tested on both an Intel NUC single board computer and the Descent, and Landing Computer developed by NASA JSC to validate its suitability on potential space flight hardware.

Table 5 3D keypoint ID method comparative analysis

	Algorithmic	Mean Runtime [s]		Min. Overlap	Mean Pos.	Mean Att.	Mean RMS
Method	Complexity	1k×1k	3.5k×3.5k	Fraction	Err. [m]	Err. [deg]	Reproj. Err. [m]
High-gradient	Low	0.125	20.2	21%	0.076	0.158	0.065
Laplacian	Low	0.118	20.0	23%	0.049	0.097	0.049
ZIG	High	0.140	2.99	35%	0.126	0.164	0.121

References

[1] Sakai, S., Kushiki, K., Sawai, S., Fukuda, S., Miyazawa, Y., Ishida, T., Kariya, K., Ito, T., Ueda, S., Yokota, K., Kawano, T., Ohtake, M., Saiki, K., Nakauchi, Y., Michigami, K., Furukawa, K., Akizuki, Y., Kanaya, S., Kinjo, T., Goto, K., Sawada, K., Sugimoto, Y., Takeuchi, H., Tomiki, A., Toyota, H., Nagata, T., Nakatsuka, J., Maki, K., Mizuno, T., Shiratori, H., Nishino, M. N., Usami, N., Kikuchi, J., Hamori, H., Hirasawa, R., Shibasaki, Y., and Saito, H., "Moon landing results of SLIM: A smart lander for investigating the Moon," *Acta Astronautica*, Vol. 235, 2025, pp. 47–54. https://doi.org/https://doi.org/10.1016/j.actaastro.2025.05.047, URL https://www.sciencedirect.com/science/article/pii/S0094576525003297.

[2] Verma, V., Nash, J., Saldyt, L., Dwight, Q., Wang, H., Myint, S., Biesiadecki, J., Maimone, M., Tumbar, A., Ansar, A., Kubiak, G., and Hogg, R., "Enabling Long Precise Drives for The Perseverance Mars Rover via Onboard Global Localization," *2024 IEEE Aerospace Conference*, 2024, pp. 1–18. https://doi.org/10.1109/AERO58975.2024.10521160.

[3] McCabe, J. S., and DeMars, K. J., *Anonymous Feature Processing for Efficient Onboard Navigation*, ???? https://doi.org/10.2514/6.2020-0598, URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-0598.

[4] Johnson, A. E., Trawny, N., Setterfield, T. P., Cheng, Y., Nash, J., Clouse, D., Massone, G., and Martin, A. S., *Challenges and Solutions for Early Morning Terrain Relative Navigation on Mars*, 2025. https://doi.org/10.2514/6.2025-2076, URL https://arc.aiaa.org/doi/abs/10.2514/6.2025-2076.

[5] ng Inc, S., "Space-ng Demonstrates AstroVision and Lunar Optical Navigation at the 40th Space Symposium," https://www.prnewswire.com/news-releases/space-ng-demonstrates-astrovision-and-lunar-optical-navigation-at-the-40th-space-symposium-302421087.html, 2025. Accessed: 2025-10-20.

[6] Restrepo, C. I., Chen, P.-T., Sostaric, R. R., and Carson, J. M., *Next-Generation NASA Hazard Detection System Development*, ???? https://doi.org/10.2514/6.2020-0368, URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-0368.

[7] Schindhelm, E., Rohrschneider, R., Roark, S., and Weimer, C., "A scanning LIDAR system for active hazard detection and avoidance during landing on Europa," *2018 IEEE Aerospace Conference*, 2018, pp. 1–7. https://doi.org/10.1109/AERO.2018.8396503.

[8] Guo, Y., Bennamoun, M., Sohel, F., Lu, M., and Wan, J., "3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, 2014, pp. 2270–2287. https://doi.org/10.1109/TPAMI.2014.2316828.

[9] Michaël, Alamin, M., Driss, M., Patrick, J., Ali, B., Jihad, M. P. H., and Roy, "Salient Spin Images: A Descriptor for 3D Object Recognition," *Image and Signal Processing*, edited by Abderrahim, N. Fathallah, M. D. M. Alamin, and E. Moataz, Springer International Publishing, 2018, pp. 233–242.

[10] Hadji, I., and DeSouza, G. N., "Local-to-Global Signature Descriptor for 3D Object Recognition," *ACCV Workshops*, 2014. URL https://api.semanticscholar.org/CorpusID:14053140.

[11] Johnson, A. E., and Hebert, M., "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, 1999, pp. 433–449. https://doi.org/10.1109/34.765655.

[12] Li, J., "LGEFE: Effective Local-Global-External Feature Extraction for 3D Point Cloud Classification," *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–8. https://doi.org/10.1109/IJCNN54540.2023.10191638.

[13] Savelonas, M. A., Pratikakis, I., and Sfikas, K., "Fisher encoding of differential fast point feature histograms for partial 3D object retrieval," *Pattern Recognition*, Vol. 55, 2016, pp. 114–124. https://doi.org/https://doi.org/10.1016/j.patcog.2016.02.003, URL https://www.sciencedirect.com/science/article/pii/S0031320316000595.

[14] Li, P., Wang, J., Zhao, Y., Wang, Y., and Yao, Y., "Improved algorithm for point cloud registration based on fast point feature histograms," *Journal of Applied Remote Sensing*, Vol. 10, 2016, p. 45024. https://doi.org/10.1117/1.JRS.10.045024, URL https://doi.org/10.1117/1.JRS.10.045024.

[15] Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M., "Aligning point cloud views using persistent feature histograms," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3384–3391. https://doi.org/10.1109/IROS.2008.4650967.

[16] Aulinas, J., Petillot, Y., Salvi, J., and Lladó, X., "The SLAM problem: a survey," *Artificial Intelligence Research and Development*, 2008, pp. 363–371.

[17] Dellaert, F., "Factor graphs and GTSAM: A hands-on introduction," *Georgia Institute of Technology, Tech. Rep*, Vol. 2, No. 4, 2012.

[18] shen Wu, L., lin Wang, G., and Hu, Y., "Iterative closest point registration for fast point feature histogram features of a volume density optimization algorithm," *Measurement and Control*, Vol. 53, 2020, pp. 29–39. https://doi.org/10.1177/0020294019878869, URL https://doi.org/10.1177/0020294019878869.

[19] Yoo, J.-C., and Han, T. H., "Fast normalized cross-correlation," *Circuits, systems and signal processing*, Vol. 28, No. 6, 2009, pp. 819–843.

[20] Chaudhari, A. J., Leahy, R. M., Wise, B. L., Lane, N. E., Badawi, R. D., and Joshi, A. A., "Global point signature for shape analysis of carpal bones," *Physics in Medicine & Biology*, Vol. 59, No. 4, 2014, p. 961.

[21] Zhong, Y., "Intrinsic shape signatures: A shape descriptor for 3D object recognition," *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 2009, pp. 689–696. https://doi.org/10.1109/ICCVW.2009.5457637.

[22] Bronstein, M. M., and Kokkinos, I., "Scale-invariant heat kernel signatures for non-rigid shape recognition," *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 2010, pp. 1704–1711.

[23] Aubry, M., Schlickewei, U., and Cremers, D., "The wave kernel signature: A quantum mechanical approach to shape analysis," *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, IEEE, 2011, pp. 1626–1633.

[24] Carrilho, A. C., Galo, M., and Santos, R. C., "STATISTICAL OUTLIER DETECTION METHOD FOR AIRBORNE LIDAR DATA," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-1, 2018, pp. 87–92. https://doi.org/10.5194/isprs-archives-XLII-1-87-2018, URL https://isprs-archives.copernicus.org/articles/XLII-1/87/2018/.

[25] Kobylka, K., Adams, D., and Shrotri, P., "Standalone Hazard Evaluation and Refinement From Instrument Findings (SHERIF)," *47th annual AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, 2025.

[26] Novotni, M., and Klein, R., "3D zernike descriptors for content based shape retrieval," *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, Association for Computing Machinery, New York, NY, USA, 2003, p. 216–225. https://doi.org/10.1145/781606.781639.

[27] Novotni, M., and Klein, R., "Shape retrieval using 3D Zernike descriptors," *Computer-Aided Design*, Vol. 36, 2004, pp. 1047–1062. https://doi.org/https://doi.org/10.1016/j.cad.2004.01.005, URL https://www.sciencedirect.com/science/article/pii/S0010448504000077, solid Modeling Theory and Applications.

[28] Wu, J., Cui, Z., Sheng, V. S., Zhao, P., Su, D., and Gong, S., "A Comparative Study of SIFT and its Variants," *Measurement science review*, Vol. 13, No. 3, 2013, p. 122.

[29] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: An efficient alternative to SIFT or SURF," *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[30] Alcantarilla, P. F., Bartoli, A., and Davison, A. J., "KAZE features," *European conference on computer vision*, Springer, 2012, pp. 214–227.

[31] Barker, M. K., Mazarico, E., Neumann, G. A., Smith, D. E., Zuber, M. T., and Head, J. W., "Improved LOLA elevation maps for south pole landing sites: Error estimates and their impact on illumination conditions," *Planetary and Space Science*, Vol. 203, 2021, p. 105119. https://doi.org/https://doi.org/10.1016/j.pss.2020.105119, URL https://www.sciencedirect.com/science/article/pii/S0032063320303329.

[32] Crassidis, J. L., and Junkins, J. L., *Optimal estimation of dynamic systems*, CRC press, 2011.

[33] Wilmot, J., "A core flight software system," *Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, Association for Computing Machinery, New York, NY, USA, 2005, p. 13–14. https://doi.org/10.1145/1084834.1084842, URL https://doi.org/10.1145/1084834.1084842.

[34] Guennebaud, G., Jacob, B., et al., "Eigen v3," http://eigen.tuxfamily.org, 2010.

[35] Cudmore, A., "High-Performance Spaceflight Computing (HPSC) Middleware Overview," , Dec. 2018. URL https://ntrs.nasa.gov/citations/20190001377.

[36] Rutishauser, D., Moore, R., Prothro, J., and Yim, H., "High Performance Computing for Precision Landing and Hazard Avoidance and Co-Design Approach," *2019 IEEE Aerospace Conference*, 2019, pp. 1–10. https://doi.org/10.1109/AERO.2019.8741888.

[37] Sostaric, R. R., Pedrotty, S., Carson, J. M., Estes, J. N., Amzajerdian, F., Dwyer-Cianciolo, A. M., and Blair, J. B., "The SPLICE project: safe and precise landing Technology development and testing," *AIAA scitech 2021 forum*, 2021, p. 0256.

[38] Chen, P.-T., Kaki, S., San Martin, A. M., Skulsky, D., Katake, A., and Trawny, N., "LIDAR-generated digital elevation models for hazard detection - requirements and analysis," , 2019. https://doi.org/2014/50428, URL https://hdl.handle.net/2014/50428.

[39] Ward, K. C., Smith, K. W., and Mendeck, G. F., "Motion compensation algorithm for online digital elevation map generation using scanning LIDAR data," *AIAA Scitech Forum*, , No. January, 2026.