AAS 22-137

AUTONOMOUS NAVIGATION AND DENSE SHAPE RECONSTRUCTION USING STEREOPHOTOGRAMMETRY AT SMALL CELESTIAL BODIES

Jacopo Villa[*], Jay McMahon[†], Benjamin Hockman[‡], Issa Nesnas[§]

Navigating spacecraft and mapping the surface in the vicinity of small Solar System bodies is historically challenging. While surface landmarks and related optical measurements are key to enhance navigational performance in such an environment, processing close-proximity images relies on ground operations and requires demanding, human-led coordination between orbit determination, shape modeling, and image processing tasks. In this paper, we present a vision-only autonomous pipeline for terrain-relative positioning, landmark matching, and dense mapping to perform such optical-navigation tasks onboard. We leverage stereophotogrammetric techniques, in particular structure-from-motion for camera pose estimation and dense stereo matching for shape reconstruction. The pipeline requires no prior knowledge of the environment or the observer's state. Further, we propose a novel feature matching algorithm based on geometric invariant properties between landmarks, rather than local image data, for feature description; we show its robustness to lighting and viewpoint variations. We assess performance for each pipeline phase using real imagery from the OSIRIS-REx mission, comparing our results with the mission's reconstructed estimates. We demonstrate good performance of the proposed approach using close-proximity images of the rubble-pile asteroid Bennu, where surface shadowing and viewpoint changes are prominent.

INTRODUCTION

Small body missions remain an active and lively area in astronautics. A deeper knowledge of these celestial objects would not only provide great scientific value to study the formation and evolution of our Solar System,[1,2] but it would also open technological opportunities such as exploiting small-body resources during future space missions and refining current planetary-defense techniques for asteroid deflection.[3,4] On top of the tremendous achievements in small-body exploration missions over the last decade, such as NASA's OSIRIS-REx,[5] JAXA's Hayabusa2[6] and ESA's Rosetta,[7] more and more missions are planned for the upcoming years, demonstrating that there is still much to learn from these targets. NASA's Psyche,[8] Lucy,[9] and Janus,[10] as well as ESA's Comet

[*] Graduate Student, Aerospace Engineering Sciences, University of Colorado Boulder, 3775 Discovery Drive, Boulder, CO 80303

[†] Assistant Professor, Aerospace Engineering Sciences, University of Colorado Boulder, 3775 Discovery Drive, Boulder, CO 80303

[‡] Robotics Technologist, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA

[§] Group Supervisor, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA

Interceptor,[11] are just some examples of upcoming missions reaching a diverse and largely unknown set of targets.

Today, operating a spacecraft in the vicinity of a small body is challenging, chiefly due to strong perturbations in the dynamical environment, as well as the lack of knowledge about most target bodies and their related physical properties, when these are firstly reached. In this context, precise navigation plays a crucial role to ensure mission safety, GNC performance, and ultimately the fulfillment of operational requirements. An established technique to achieve this is using imagery of the small body to track reference surface points, a process known as landmark-based navigation. Tracking surface landmarks using images is known as an arduous task: the appearance of small-body surface topography, as seen from the spacecraft camera, is subject to continuous evolution, due to (1) the rapid spin of small bodies about the axis of rotation, affecting surface lighting conditions and causing shadow patterns to change, and (2) the relative motion between the surface and the orbiting spacecraft, which in turn affects the viewpoint the surface is observed from. In current small-body navigation, detecting and extracting landmarks from images is a ground-based process requiring experienced specialists. Images are inspected and surface landmarks are selected and tracked manually, based on the operator's expertise. This process is highly coupled with shape-estimation tasks: the appearance of the topography surrounding landmarks is predicted using simulated data, which is then used to update both the local topographical data and the state of the observing spacecraft. This process is known as Stereophotoclinometry (SPC) and is performed in closed-loop with optical navigation and orbit determination.[12] These techniques represent the state of the art in terms of navigation accuracy and shape modeling. However, performance trade with operational complexity and reliance on human expertise, which in turn affect the scalability of small body missions and put constrains on in-situ operations. A greater level of onboard autonomous navigation would increase our capabilities in the Solar System, accelerating the exploration of the diverse and abundant population of small bodies and potentially increasing the scientific return.[13]

Recently, the OSIRIS-REx mission pioneered autonomous vision-based navigation at the target asteroid Bennu during the Touch-and-Go maneuver to acquire a surface sample. This operation was successfully performed using Natural Feature Tracking (NFT), a technique leveraging *a-priori* knowledge of surface topography to predict the appearance of optical data around landmarks and match predictions with observed data during the descent.[14] While this technique achieved mission objectives with outstanding performance and in an autonomous fashion, it still relied on prior knowledge collected via human-based operations, especially the high-resolution shape model of the asteroid Bennu and the higher-resolution topography map of the Touch-and-Go site, as well as the set of surface landmarks selected for navigation. Another example of autonomy applied to small-body missions is the upcoming vision-based navigation system aboard the ESA's HERA mission. The system makes use of frame-to-frame feature tracking techniques for proximity operations, but is not provided with any feature matching or loop closure capabilities.[15] Additionally, descriptor-less feature tracking may only work for some types of surface topography (e.g. sparse rocks on top of a relatively smooth surface background) and may suffer from drift due to moving shadows, as previous studies have shown.[16]

Previous work has advanced autonomous navigation capabilities for several small-body scenarios, such as optical navigation during far and close approach, as well as silhuette-based shape reconstruction.[17] Studies have also shown that modern Simultaneous Localization and Mapping (SLAM) frameworks can be used for navigation.[18] However, most of these works focused on real or simulated small body images capturing a relatively smooth topography, where the effects of evolving lighting conditions are less prominent and those of viewpoint are easier to compensate with properties of traditional visual features, such as SURF.[19]

One key challenge for autonomous navigation at small bodies is enabling robust, vision-based navigation for proximity operations. This is the scenario where local shadowing effects – usually caused by rocks and boulders – become dominant and, given the close-range perspective, viewpoint changes are harder to model without an accurate topographical map of the surface, such as the those produced via SPC. In this work, we propose an algorithmic pipeline to perform both navigation and mapping in the proximity-operations scenario, where lighting and viewpoint are subject to rapid evolution. Our pipeline is solely based on images as inputs and makes use of Stereophotogrammetry (SPG) techniques to firstly estimate the camera poses and a sparse cloud of surface landmarks, and secondly perform dense shape reconstruction of the observed surface. In addition to estimating landmarks and camera poses for navigation purposes, onboard shape reconstruction can be valuable to enhance situational awareness, such as performing hazard detection and avoidance, as well as selecting landing and operational sites during the mission.

NAVIGATION AND MAPPING PIPELINE

In this paper, we define *landmarks* as the 3D locations on the small body surface corresponding to an optical observation, and *keypoints* as the projections of landmarks into the camera plane, i.e. the actual pixel coordinates of optical observations. *Features* are the data describing the traits of a given keypoints, e.g. data related to the image region surrounding a keypoint. Camera *poses* are the joint representation of camera position and orientation in 3D space. Finally, *loop closure* is the task of detecting previously-estimated landmarks in order to refine the current camera pose and back-propagate corrections in previous landmarks and camera poses. The latter is a common practice within SLAM algorithms.

The proposed process is illustrated in Figure 1. The input data is imagery of the target body. Camera intrinsic parameters can either be inputted or estimated in the process, depending on their availability and accuracy. Input images can either be consecutive frames, or frames sampled under considerably different lighting and viewpoint conditions. In the former case, frame-to-frame feature matching is executed to link optical observations throughout the image set. In the latter case, an image pre-processing step is applied to detect keypoints ad extract robust feature descriptors that are invariant to lighting and viewpoint; then, feature matching is performed using such robust descriptors. Once feature matches are computed, a Structure-from-Motion (SfM) algorithm is run to reconstruct both the observer's camera poses and the observed surface landmarks. Lastly, using such estimates of the observed scene, stereo matching is employed to estimate a dense representation of the small body surface. Stereo matching provides a dense point cloud as the output, which can eventually be converted into a mesh representation of the surface.

This pipeline is solely based on optical data as inputs, and hence represents a purely vision-based, surface-relative navigation system. As such, it requires no *a-priori* knowledge of the observed scenario or the observer's state. All estimated parameters are relative to the coordinate system corresponding to the first camera pose. These can be translated and rotated to the small body-fixed reference frame if an estimate for this transformation is available. Further, camera poses and 3D landmarks can be fed into a navigation filter in order to link such geometric estimates to the dynamical environment and inertial parameters. One should note that, since the process is based on monocular-camera images, the poses and landmark outputs lack an estimate of absolute scale. In actual operations, this can be compensated by estimating the scale factor in the navigation filter, e.g.

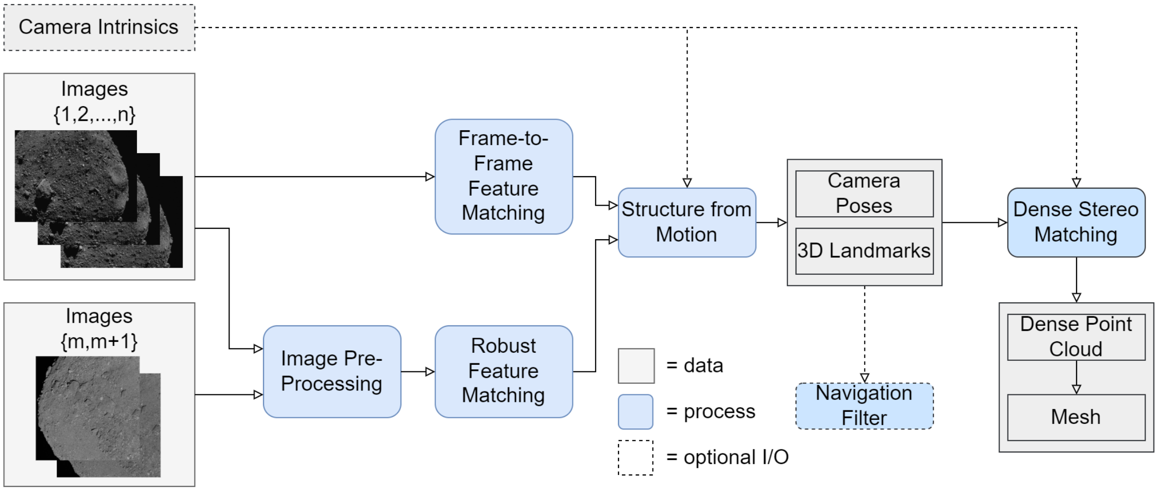


Figure 1: Proposed pipeline for navigation and dense mapping.

estimating the size of the observed small body.

SIMULATION SETUP

To assess robustness to the appearance and rapid evolution of optical data in proximity operations, we test our algorithms using real imagery from the OSIRIS-REx mission. The rich and rocky surface of the mission target, asteroid Bennu, provides challenging surface conditions such as the very dense field of rocks and boulders with self-similar traits, as well as the rapidly evolving surface shadows. We use images acquired by the Natural-Feature-Tracking Camera (NFTCAM)[20] to assess frame-to-frame matching and SfM performance, and an imageset from the OSIRIS-REx's mapping camera (MAPCAM)[21] to test robust feature matching, where the same scene is observed with enough overlapping surface, but under different lighting and viewpoint. More details on each imageset are provided in the corresponding section below. Further, we use reconstructed mission data from the OSIRIS-REx's SPICE kernels as references to compare our estimates with.[22]

POSE ESTIMATION

The first task executed by the autonomous pipeline is camera pose estimation. Not only is this step essential for navigation, but it also provides reference geometric data for dense shape reconstruction. In particular, the parameters of interest are the camera poses and the 3D landmarks, in this case observed on the surface of the target body. Numerous approaches exist to tackle the simultaneous estimation of camera poses and the environment 3D geometry. In this work, we apply a standard Structure-from-Motion algorithm: multiple observations of surface landmarks, from different viewpoints, are leveraged to estimate the structure of the observed scene. We choose an image set from the OSIRIS-REx mission, immediately before the descent for the Touch-and-Go maneuver, using the imagery from the NFTCAM*. This is the image set processed by the onboard autonomous system to guide the spacecraft to the surface and collect the sample. This scenario is particularly challenging from the lighting standpoint, as shadows and shading effects rapidly evolve

*https://sbnarchive.psi.edu/pds4/orex/

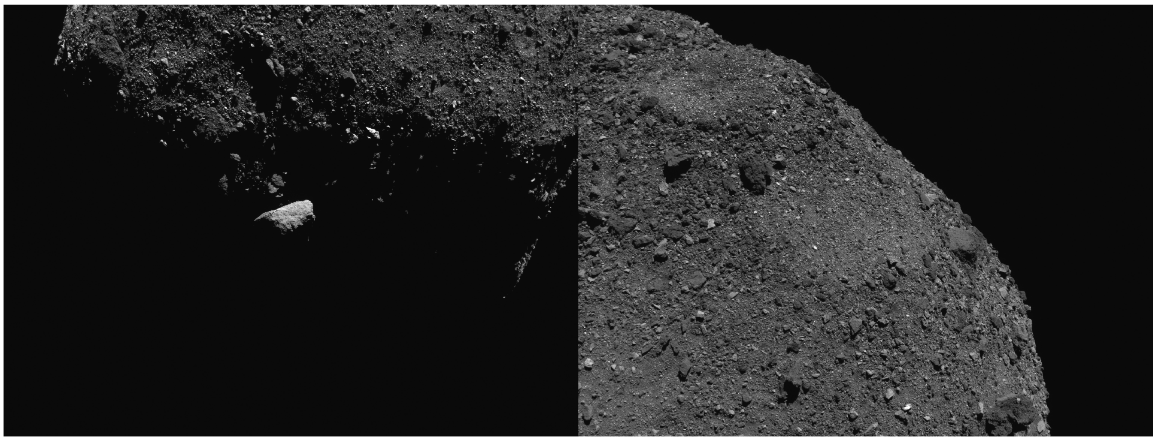


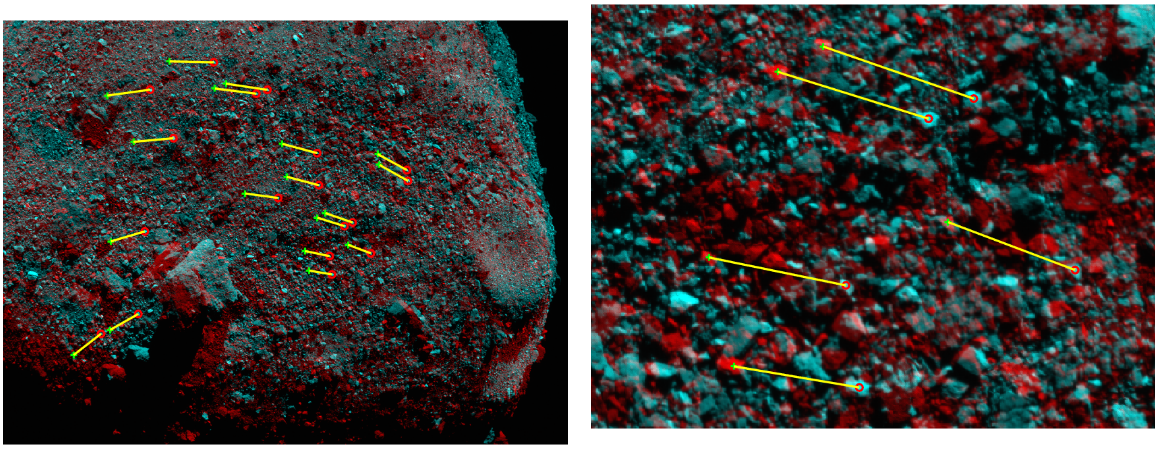
Figure 2: First (left) and last (right) images of the set used for the Structure-from-Motion simulation. The imagery is from the OSIRIS-REx's NFTCAM, and has been collected before the Touch-and-Go descent.

form image to image. The first and last image of this set are shown in Figure 2. The time stamps of the first and last images, in Coordinated Universal Time (UTC), are “2020-10-20 18:58:51.696” and “2020-10-20 20:11:12.782”, respectively.

Frame-to-Frame Feature Matching

Firstly, keypoints are collected in each image. When keypoints related to the same surface landmark are sampled from multiple views, these can be used to estimate the 3D position of both the camera and the landmark. Each keypoint corresponds to a visual feature, i.e. a representation of the image neighborhood around the interest point. Features are commonly used to link keypoint observations between pairs of images – a process known as feature matching. The first step is feature detection, where interest points are identified in each image. We use KAZE features because of their boundary-preserving property and the use of a nonlinear scale space, which improve detection accuracy.[23] In the context of asteroid-surface imagery, rocks, and in particular their centroids, can be used as keypoints, which can be detected using region-based feature detectors such as SIFT, SURF, or KAZE.[24] In the considered dataset, KAZE features provide the highest number of detections and good matches when compared to SURF or SIFT, two other state-of-the-art, region-based feature types. For example, the total number of detections for the first image is 33959 for KAZE, 23100 for SIFT, and 6506 for SURF. Further, KAZE features are well-suited to precisely detect and track surface rocks with highly irregular shape, commonly found in small body imagery, since it does not leverage Gaussian blurring to detect features (whereas SIFT and SURF do).

In each image, we detect KAZE keypoints and keep the 100 strongest detections, extracting a descriptor for each of them. Then, pairwise feature matching is performed between consecutive images. An example of matched features between the 7th and the 8th image in the set is shown in Figure 3. Using frame-to-frame matching, a given feature is tracked over more than two frames. The mean track length is 2.3 frames, with 72% of features tracked for 2 frames, 19% for 3 frames, 5% for 4 frames, 3% for 5 frame, and 1% for 6 frames. These numbers are highly case-specific and depend on multiple factors, such as the frame-to-frame pixel displacement, small body's spin rate, and number of keypoint appearances within the camera field of view. However, it is interesting to



(a) Full-size image.

(b) Image magnification around a set of features close to the image center.

Figure 3: Example feature matches using KAZE, between the 7th and 8th image (shown in false colors), in the NFTCAM dataset. 100 features are extracted in each frame for matching. Feature detections are commonly placed close to the centroid of surface rocks.

note that most features are tracked for a relatively short number of consecutive observations, likely due to the increasing changes in viewpoint and lighting, affecting the local feature data.

Structure-from-Motion

Pairs of point correspondences provided by feature matching are the input for the actual SfM process, which is summarized in the following. For each pair of images, the camera essential matrix is estimated, as a first step, using the M-estimator sample consensus (MSAC) algorithm.[25] We set the Sampson distance threshold to 0.05, used to reject outliers (such as feature matches with low detection accuracy). We assume that camera intrinsic parameters are available and accurate. (We do not compensate for off-nominal camera geometric distortions affecting the NFTCAM.) Note that if this is not the case, the camera fundamental matrix could be estimated, instead. Second, the camera essential matrix is used to estimate the camera pose relative to the previous frame; this process is repeated until at least 95% of keypoints related to the camera pose are inliers. Third, matched keypoints and the corresponding camera poses are stored in the set of views which links all keypoint observations with their observing geometry. The global coordinate system used for the process is the camera frame at the first image. Additionally, feature tracks are created or updated so that pairwise observations of a given keypoint are linked to the broader set spanning all views where that keypoint appears. As a next step, for each feature track, a corresponding 3D landmark is estimated by triangulating all views and observations related to such a keypoint.[26] This process provides initial guesses for all observed 3D landmarks. Finally, bundle adjustment is used to refine both the 3D landmark estimates (for all views) and camera poses using all the previous observations. After this optimization step, all landmarks whose reprojection error into the camera plane is larger than 0.5 pixels are discarded, while the others are processed in the next update (i.e. where new images are sampled).

SfM results for the pre-TaG scenario are shown in Figures 4 in 3D space and in Figure 5, as the position error in the Radial-Transverse-Normal (RTN) frame. Estimated positions are compared with the OSIRIS-REx's reconstructed trajectory. Positions are estimated sequentially, using all images acquired up to the epoch considered, without using future observations. Since the SfM process is by design independent of scale, we use the same scale factor for the estimated and ground-truth cases in order to compare results. As shown by the RTN plot, the positioning error is below 3 meters in the radial direction, 10 meters in the transverse, and 5 meters in the normal direction. For reference, the maximum transverse position error is below 2% of the radial distance at that epoch (520 m from the asteroid's center of mass). A total of 150 landmarks are accepted at the end of the process, i.e. after the last image poses are refined via bundle adjustment. Interestingly, a relatively low number of feature matches and related surface landmarks is sufficient to provide good positioning performance, which in turn decreases computational requirements for this process. In fact, we noticed decreasing position accuracy when processing a higher number of lower-quality detections. These results highlight the importance of accurate detections for small body autonomous navigation. Finally, note that we assume that no 3D landmarks were known *a priori* to perform loop closure. Using known 3D landmarks, e.g. extracted at previous orbits around the body, can further improve navigation accuracy and decrease the position drift that can appear for longer trajectory arcs, e.g. via loop closure.

While the proposed SfM algorithm needs to be tested on more small-body scenarios, with a broader spectrum of surface topographies, the presented results suggest that similar techniques can be employed for precise terrain-relative navigation at small bodies, even when the shadowing and viewpoint changes are prominent. Reported data also suggest that long tracks of optical data are not required for precise positioning, and that links between multiple feature tracks, processed as a bundle, can compensate for relatively short tracks. In future work, we will (1) test more imagesets related to multiple target bodies, (2) compare the SfM approach to more modern SLAM algorithms, (3) evaluate performance for camera attitude estimation, and (4) integrate this pipeline within a broader navigation-filter framework that accounts for the dynamical environment and related information and uncertainties (e.g., spin pole parameters, scale factors, accelerations).

ROBUST FEATURE MATCHING

Standard feature description and matching techniques (such as KAZE) applied to small body images appear to perform well for frame-to-frame matching, when the imaging frequency is relatively high. However, poor performance is obtained when the images are taken from substantially different viewpoints and Sun phase angles. The relative motion between the observing spacecraft, the small body, and the Sun yields to large changes in surface lighting conditions and perspective, which affect the surface appearance. Standard feature descriptors are computed locally, i.e. using the image data in the neighborhood of a given keypoint. While these can be robust to some amount of lighting and viewpoint changes, they are not robust to moving shadows and the rugged topography (i.e. far from resembling planar surfaces) typical of a small body surface. Additionally, many features on the surface of a small body are very similar with each other. For example, commonly detected features in close-proximity images are surface rocks and boulders that are similar to other rocks with the same or different size.

In landmark-based navigation, linking new observations with past ones is valuable to increase navigation performance and robustness, as well as to store a reasonably low number of landmarks. The simplest method to do that is detecting the same surface landmarks over multiple body rotations and from different orbital conditions. In this section, we propose a novel feature matching technique that is robust to lighting and viewpoint changes and hence can be used to recover features (and

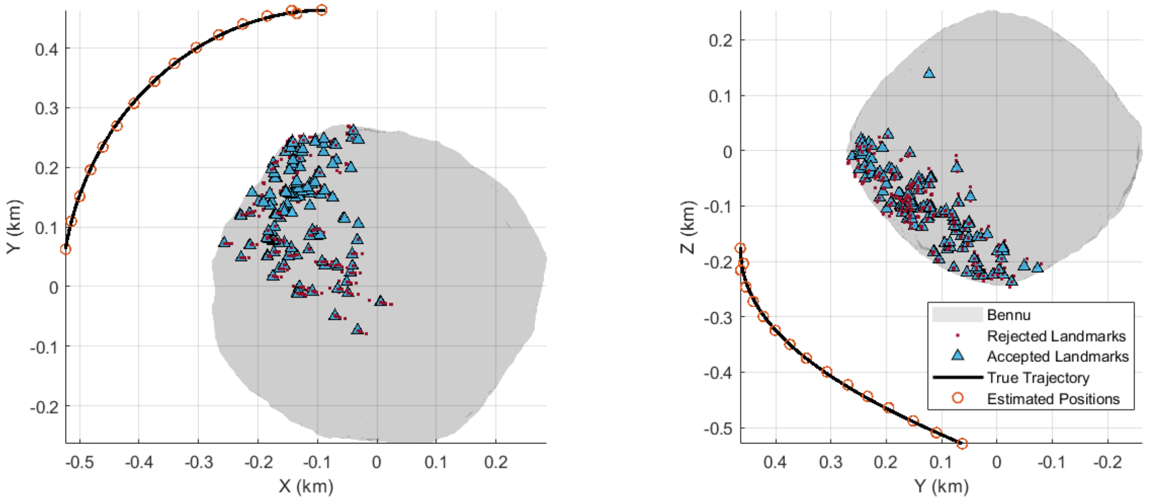


Figure 4: Structure-from-Motion results for the pre-TaG scenario, shown in XY (left) and YZ (right) planes of the asteroid-fixed frame. Sequentially estimated camera positions are plotted on top of the OSIRIS-REx's reconstructed trajectory, and the related landmarks are reported. Compared results are adjusted for scale.

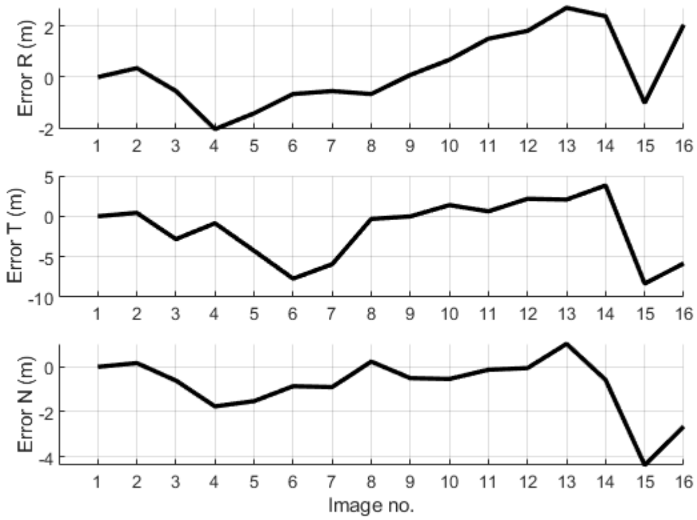


Figure 5: Structure-from-Motion sequential-position error for the pre-TaG scenario, reported in the RTN frame. The maximum error is below 2% of the radial distance. Compared results are adjusted for scale.

related landmarks) observed from very different Sun phase angles and perspectives. Instead of using local image data to construct a feature descriptor, we leverage the geometric constraints between the landmarks: considering the small body as a rigid body, we assume that the relative position between landmarks is fixed. Once landmark detections are available, the geometry between them can be used to progressively restrict the feature search space and identify the best feature match for any given feature. This technique does not require any *a-priori* knowledge of the scenario, such as spacecraft pose, Sun phase angle, and scale factors. As such, it can be used to solve a surface-relative "lost-in-space" problem, but could also be used for a loop-closure scenario. The proposed feature matching process is divided into four phases: feature detection, feature description, feature matching, and feature confirmation, which are described in the following.

We test the feature matching process on four images sampled during the OSIRIS-REx's Reconnaissance A phase, using the MAPCAM instrument. We perform feature matching between the first and the third image, with UTC time stamps "2019-09-26 17:37:36" and "2019-09-26 21:42:42", respectively. The second ("2019-09-26 17:38:09") and fourth ("2019-09-26 21:43:14") images, which are sampled about 30 seconds after the first and third respectively, are merely used for feature triangulation, as described in the following. We use this pair of images to test feature matching as they are characterized by different lighting conditions and viewpoint, as can be seen by the shadows cast on the surface by the visual appearance of the larger boulders and their shadows cast on the surface. Future work will extend these analyses to more cases and imagery scenarios.

Feature Detection

The first step consists in detecting the surface features of interest. Images sampled in close proximity are usually rich in details, provided by the dense population of rocks and boulders with great diversity in size, shape, roughness, and albedo. While most rocks can easily be used as manually-tracked landmarks, their diversity and density on the surface makes the detection task more challenging to be performed algorithmically. Nevertheless, rocks are still valuable visual features as they represent image regions which are detected by state-of-the-art features. One way to mitigate the abundance of rock-type features is to only detect those with certain user-defined traits, using image processing techniques. We propose two different detection algorithms to identify specific rock families within the population: low-variance rocks and high-albedo rocks. These detections can eventually be used as the input keypoints for the description and matching steps. Previous work has shown that rocks and boulders can be detected on the surface of small bodies using neural networks.[27] However, such techniques were applied for imagery with lower surface resolution and hence lower density of rock-type features. Other studies have shown that craters can also be robustly identified on the surface of and planetary bodies.[28] All such techniques could potentially replace the detection algorithms presented here for different types of planetary surfaces or different surface resolutions.

Low-Variance Features One way to only detect a subset of surface rocks is applying image processing to identify rocks that appear smooth within the image context. One metric for image smoothness is its standard deviation. As such, we extract smooth features using the following process:

1. Given the original image $\mathcal{I}_0$, a *standard deviation filter* is applied to $\mathcal{I}_0$. The output $\mathcal{I}_{sd}$ is an image whose pixels are the standard-deviation values of each pixel neighborhood around the corresponding input pixel. We use a neighborhood window with size $w \times w$, where $w = 3$ pixels;

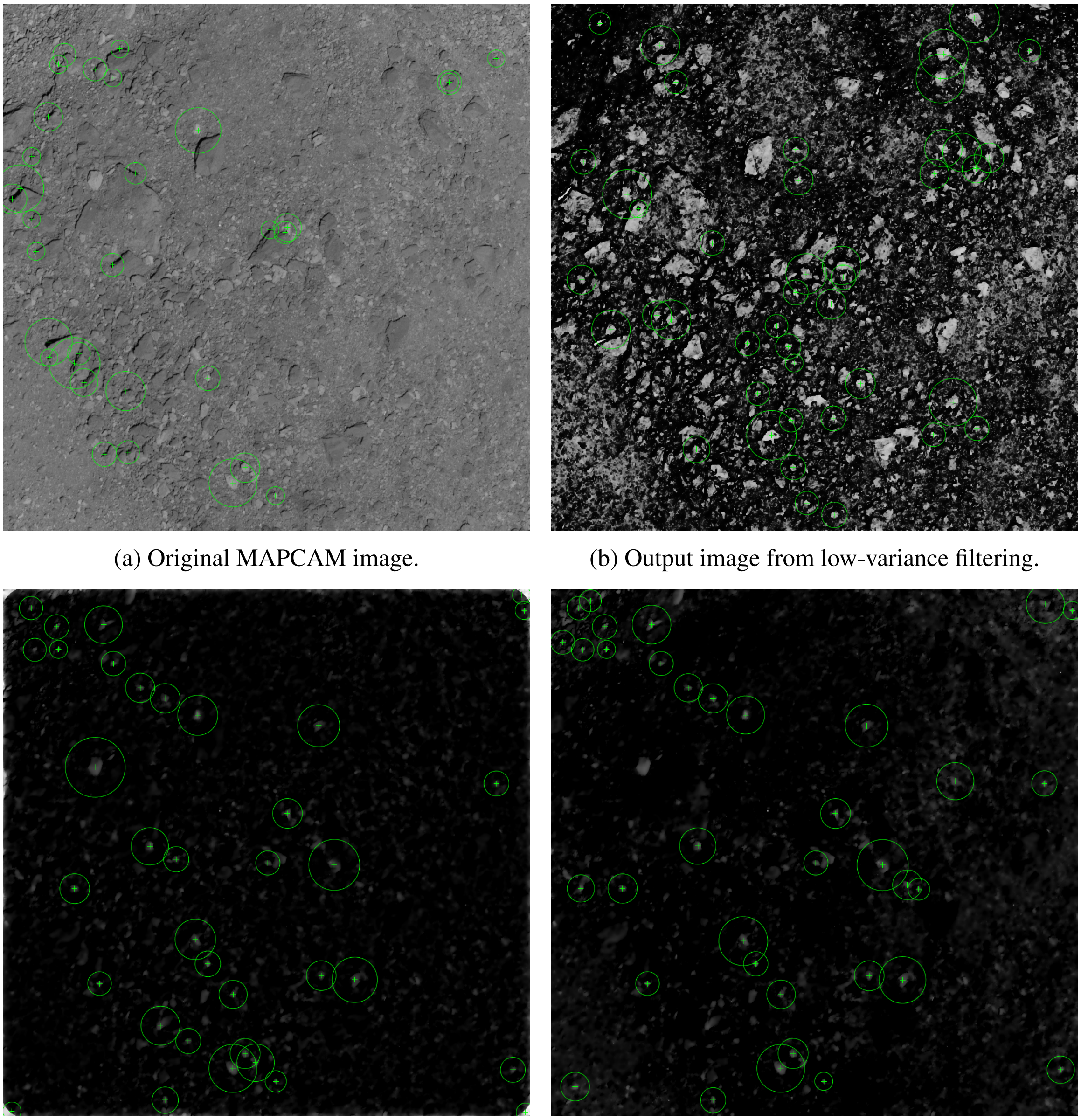
2. the negative of the image is computed as $\hat{\mathcal{I}}_n = \mathbb{1} - \hat{\mathcal{I}}_{sd}$, where $\mathbb{1}$ is the matrix whose elements are all equal to 1 and $\hat{\mathcal{I}}_{sd}$ is the image $\mathcal{I}_{sd}$ normalized, so that the pixel values span in between 0 and 1;
3. the median of $\hat{\mathcal{I}}_n$ is subtracted to the image, i.e.: $\hat{\mathcal{I}}_{ms} = \hat{\mathcal{I}}_n - \texttt{median}(\hat{\mathcal{I}}_n)$, where the `median` operator computes the median of all pixels in the image;
4. the positive-pixel image $\hat{\mathcal{I}}_p$ is obtained by setting all negative pixel values of $\hat{\mathcal{I}}_{ms}$ to zero and then by normalizing the image once again;
5. the final low-variance image $\hat{\mathcal{I}}_{ld}$ is obtained applying a *bilateral filter* to $\hat{\mathcal{I}}_p$ to reduce the noise while preserving the edges for feature detection.[29] We set the closeness-function's standard deviation $\sigma_d = 3$ pixel and the similarity-function's standard deviation $\sigma_r = 0.01$;
6. KAZE features are detected on $\hat{\mathcal{I}}_{ld}$. If two features are closer than 10 pixels with each other, then only the larger-scale feature is kept and the other one is deleted.

The values of w, σ_d and σ_r are chosen so to provide a good trade-off between noise and edge preservation. However, initial trials did not show high sensitivity to such tuning parameters. This process is also designed to remove shadows in the image, by subtracting the image median and zeroing all negative pixels, making it more robust to lighting changes. Since the shadows cast by rocks are usually projected on the surface background (which is generally subtracted by the median step), the foreground of detected smooth rocks is less affected by shadowing artifacts.

The output image from low-variance filtering is shown in Figure 6b and compared with the original MAPCAM image (Figure 6a). Relatively smooth regions appear as the brightest ones, despite their albedo and size in the original image. The 50 strongest KAZE feature detections for each image are also shown: while many strong features detected on the original image are centered on shadows (which do not represent a fixed object, and hence are not a reliable landmark), low-variance features are detected roughly at the center of a rock, making it a much more robust landmark to track. It should also be noted that most strong matches among low-variance features are those spanning 10-30 pixels across. Centroiding performance for such medium-sized features is superior to those for larger ones, whose appearance is generally more affected by viewpoint changes.

High-Albedo Features One alternative approach to detecting smooth surface regions consists in filtering the image so that only the locally-bright spots are emphasized. If such regions have higher albedo than the neighboring ones, then will remain relatively brighter even under varying lighting conditions. Small body imagery from past missions have shown that scattered higher-albedo rocks can be found on the surface of some targets. These features can be detected by subtracting the local median of the image. By doing so, both the lower-albedo background and the shadows are removed from the output image, which makes the resulting features more robust to lighting, as in the low-variance case. High-albedo features are filtered as follows:

1. Given the original image $\mathcal{I}_0$, a *median filter* is applied to $\mathcal{I}_0$. A median image $\mathcal{I}_m$ is obtained where each pixel is the median of the pixels in its $w \times w$ neighborhood window. We use $w = 50$ pixels;
2. the median image is subtracted to the original image, obtaining the median-subtracted image $\mathcal{I}_{ms} = \mathcal{I}_0 - \mathcal{I}_m$;
3. The final high-albedo image $\hat{\mathcal{I}}_{ha}$ is the normalized median-subtracted image.



(a) Original MAPCAM image.

(b) Output image from low-variance filtering.

(c) Output image from high-albedo filtering, using a sliding-window median filter.

(d) Output image from high-albedo filtering when subtracting the full-image median.

Figure 6: Comparison between image-processing techniques for rock detection. Green markers represent the 50 strongest KAZE keypoints detected in the image; green circles represent each feature's scale.

4. KAZE features are detected on $\hat{\mathcal{I}}_{ha}$. If two features are closer than 10 pixels with each other, then only the larger-scale feature is kept and the other one is deleted.

The high-albedo output image, as well as the detected features, are shown in Figure 6c. In the case considered, the features in $\hat{\mathcal{I}}_{ha}$ are smaller on average than those in $\hat{\mathcal{I}}_{ld}$, but the scale of the stronger detections does not substantially differ from the low-variance case. One should note that the features provided by $\hat{\mathcal{I}}_{ha}$ can either correspond to high-albedo regions, or to regions with high reflectance due to the Sun-surface-observer geometry, or both. This can potentially increase the number of false detections. One final consideration on this algorithm is that the computational cost of the median filter depends on the size of the neighborhood window w. A much faster implementation consists in computing $\mathcal{I}_{ms}$ by simply subtracting the median of the full image (Figure 6d), instead of using a window-based median filter. The downside of this approach is that some image gradient can remain unfiltered. Nevertheless, detection performance is not highly affected in the case considered.

Feature Descriptors

Once keypoint detections are available, the 3D geometry of the corresponding landmarks is used to generate feature descriptors. We leverage the triangles connecting a given landmark with two of its nearest neighbors, in 3D space. In particular, we use the angles related to each of those triangles as description parameters. Since the relative position between landmarks is fixed, such angles will not change under multiple observations. Angles are scale-invariant and, since the related triangles are computed using the 3D points, are also viewpoint-invariant. Finally, the property of being invariant to lighting conditions is acquired by the visual feature detection step, discussed in the previous section.

The prerequisite for feature description is estimating the 3D positions (i.e. the landmarks) corresponding to each detected keypoint. We do so by observing the same features in two consecutive frames, i.e. from two slightly different viewpoints, and then triangulating the views as previously described. To track features from the first to the second frame, we use the Kanade-Lucas-Tomasi (KLT) tracking algorithm,[30,31] which performs well when features in consecutive frames are similar with each other and the pixel displacement is low (which is the case given the small baseline we use for triangulation). For reference, the mean displacement of the features tracked from the first to the second frame is 14.8 pixels (0.058°). The output of the triangulation is an estimate of the landmarks' 3D positions as well as the camera pose in the second view relative to the first view.

Feature descriptors are computed as follows. Let $\mathbf{p}_0 = [x_0, y_0, z_0]$ be the 3D point (the landmark) whose descriptor is to be computed. We call this the center point. First, the k-nearest neighbors of $\mathbf{p}_0$ are searched in 3D space;[32] we use k=7. We define $\mathcal{N} = \{\mathbf{p}_1, ..., \mathbf{p}_7\}$ as the set of the 7-nearest neighbors, sorted from the closest to the furthest from $\mathbf{p}_0$. Then, let $\mathcal{T}$ be the set of the IDs representing all the 2-combinations within $\mathcal{N}$, i.e. $\mathcal{T}$ is the set representing all possible pairs of elements in $\mathcal{T}$, such as $\tau_{12} = [1, 2]$, $\tau_{13} = [1, 3]$, etc. For each pair τ_{ij}, we set the i-th neighbor to be the closest point to $\mathbf{p}_0$ and the j-th neighbor to be the furthest point. So for example, τ_{13} does exist, but τ_{31} does not. Note that $\mathcal{T}$ represents the set of all possible triangles formed between $\mathbf{p}_0$ and two of its nearest neighbors. Further, let $\mathbf{v}_{ij,a}$, $\mathbf{v}_{ij,b}$, and $\mathbf{v}_{ij,c}$ be the vectors from $\mathbf{p}_0$ to $\mathbf{p}_i$, from $\mathbf{p}_0$ to $\mathbf{p}_j$, and from $\mathbf{p}_i$ to $\mathbf{p}_j$, respectively. The parameters in the feature descriptor are the ordered set of angles arranged in the vector $\mathbf{\Theta}_{ij} = [\theta_{ij,a}, \theta_{ij,b}, \theta_{ij,c}]$ subtended by $\mathbf{v}_{ij,a}$, $\mathbf{v}_{ij,b}$, and $\mathbf{v}_{ij,c}$, for all combinations in $\mathcal{T}$, as shown in Equation 1. Figure 7 illustrates the geometric relationships between all such parameters. Finally, the feature descriptor f is constructed using the

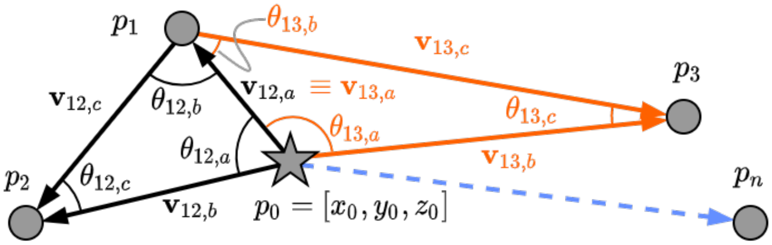


Figure 7: Schematic of the parameters used to construct the feature descriptor. The star represents the center point and the dots the neighboring points.

angles computed for all the triangles in $\mathcal{T}$, as shown in Equation 2. The resulting descriptor, for each feature, is a 63-element vector given by 3 angles for each of the $\binom{7}{2} = 21$ combinations.

The reason for storing all possible combinations of triangles formed by the neighbors rather than, say, only storing triangles formed by neighboring points in $\mathcal{N}$, is that some keypoints might miss in one image and be present in another, and vice versa. Hence, it is likely that only a subset of the triangles associated with a given feature will appear in two distinct images. Oftentimes, most of the keypoints are detected in different images, but are assigned different detection scores, hence might not be part of the "strongest keypoints" set in one of the images. The number of non-replicated keypoints depends on the feature detection logic, as well as its robustness to varying lighting and viewpoint. Future work will enhance detection performance so to increase the number of replicable keypoints.

The uniqueness of such features descriptors could be enhanced by adding more geometrical properties of the 3D point cloud and the related triangles. Future work will incorporate more of such parameters and compare matching performance with the descriptors presented here.

$$\begin{cases} \theta_{ij,a} = \cos^{-1}\left(\hat{v_{ij,a}}^{T} \cdot \hat{v_{ij,b}}\right) \\ \theta_{ij,b} = \cos^{-1}\left(-\hat{v_{ij,a}}^{T} \cdot \hat{v_{ij,c}}\right) \\ \theta_{ij,c} = \cos^{-1}\left(\hat{v_{ij,b}}^{T} \cdot \hat{v_{ij,c}}\right) \end{cases} \tag{1}$$

$$f = \{\mathbf{\Theta}_{12}, \mathbf{\Theta}_{13}, ..., \mathbf{\Theta}_{ij}\} \;\; \forall \tau_{ij} \in \mathcal{T} \tag{2}$$

Feature Matching Using the Five-Triangle Algorithm

The idea behind triangle-based feature matching is the following. Let f_α and f_β be features to be matched, with $\Theta_\alpha \in f_\alpha$ and $\Theta_\beta \in f_\beta$ being the angle vectors composing each descriptor. Note that, for example, $\tau_{\alpha,ij} \in \mathcal{T}_\alpha$ is the triangle associated with the angles vector $\Theta_{\alpha,ij}$. Then, if a geometrical configuration of *adjacent* triangles in f_α matches a configuration of adjacent triangles in f_β, then the two features are considered a candidate match. For example, if $\tau_{\alpha,ij}$ and $\tau_{\alpha,jk}$ are two adjacent triangles in f_α, then the goal is to find at least one matching triangle in f_β for each of the two τ_α. This will give the following pairs of matches: $\{\tau_{\alpha,ij}, \tau_{\beta,mn}\}$ and $\{\tau_{\alpha,jk}, \tau_{\beta,no}\}$. The higher the number of matched adjacent triangles, the higher is the likelihood of good match (i.e., the probability that such feature match is an "inlier"). Increasing the number of adjacent triangles used

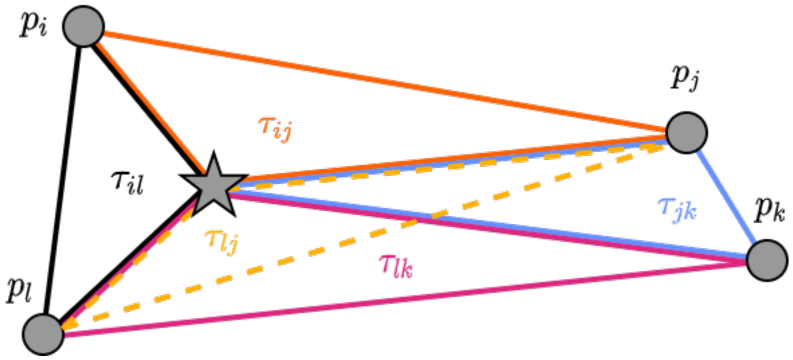


Figure 8: Schematic of the five triangles used to infer candidate feature matches. The star represents the center point and the dots the neighboring points.

to infer the match increases the good-match likelihood, but it also increases the sensitivity to missing keypoints (discussed in the previous section). In this work, we use 5 adjacent triangles connecting 5 different points (including the center point) to infer feature matching, as shown in Figure 8. Thus, if 5 pairs of adjacent triangles are matched, then the features f_α and f_β are considered a candidate match. Note that 2 out of 5 triangles are redundant in constraining the relative positions between the 5 points. However, these are used to increase robustness and avoid false matches given by error propagation from one adjacent triangle to the next.

The feature matching process is presented in Algorithm 1. The inputs to the algorithm are (1) the two features to be matched, the corresponding set of triangle IDs, and (3) a user-set error threshold. The outputs are (1) a boolean flag indicating whether the two features are considered a candidate match, (2) an error metric $E_{\alpha\beta,tot}$ used to accept or reject the candidate (described in the following section), and (3) a set of ID pairs $\Lambda_{\alpha\beta}$ corresponding to the 4 vertices used to construct the 5 matching triangles related to the candidate match. Firstly, the algorithm finds the best matches between triangles in f_α and those in f_β. Second, it verifies whether such best-matching pairs are also part of a configuration of adjacent triangles. If a 5-point, 5-triangle configuration is found, then the feature is considered as a candidate match. If the error metric associated to such a candidate match is below the user-set error threshold E_{MAX}, then the candidate match is returned. This error metric represents the total angular error accumulated by the 5-triangle configuration. We set $E_{MAX} = 30°$.

One advantage of this implementation is that, once the feature descriptors are generated and after the best-matches step is performed, triangle matches $\{\tau_\alpha, \tau_\beta\}$ and related errors $\mathcal{E}$ can be queried using triangle IDs and look-up tables, requiring relatively low computation cost. While the current implementation is not optimized for performance, we report for reference that the average execution time for the function `isCandidateMatch` (Algorithm 1) is 0.0020 s on a MATLAB implementation, using a i7-8750H CPU @ 2.20GHz. Future work will optimize the algorithm and provide more analyses on computation requirements.

When executing the feature matching algorithm for the whole set of features in the first and second image, this returns a set μ of ID pairs for all candidate matches, e.g. $\mu = \{[\alpha, \beta], [\alpha, \gamma], [\delta, \zeta], ...\}$.

Algorithm 1 Compute candidate feature matches using 5 adjacent triangles.

```
function isCandidateMatch($f_\alpha$, $f_\beta$, $\mathcal{T}_\alpha$, $\mathcal{T}_\beta$)
    matchFlag ← false
    $\iota \leftarrow 0$
    for $\mathbf{\Theta}_\alpha \in f_\alpha$ do
        $\kappa \leftarrow 0$
        for $\mathbf{\Theta}_\beta \in f_\beta$ do
            $\epsilon_\kappa(\tau_\alpha, \tau_\beta) \leftarrow \|\mathbf{\Theta}_\alpha - \mathbf{\Theta}_\beta\|_2$    ▷ Compute the error between these two triangles
            $\kappa \leftarrow \kappa + 1$
        end for
        $\{\mathbf{\Theta}^*_\alpha, \mathbf{\Theta}^*_\beta\} \leftarrow \text{argmin}\,(\epsilon(\tau_\alpha, \tau_\beta))$    ▷ Find the best match for this triangle.
        $\mathcal{M}_\iota \leftarrow \{\tau^*_\alpha, \tau^*_\beta\}$    ▷ Save the IDs related to this triangle match.
        $\mathcal{E}_\iota \leftarrow \min\,(\epsilon(\tau_\alpha, \tau_\beta))$    ▷ Save the error for this triangle match.
        $\iota \leftarrow \iota + 1$
    end for
    for $\{\tau_{\alpha,ij}, \tau_{\beta,mn}\} \in \mathcal{M}$ do    ▷ Loop through all triangle matches.
        $E_1(\tau_{\alpha,ij}, \tau_{\beta,mn}) \leftarrow \mathcal{E}(\tau_{\alpha,ij}, \tau_{\beta,mn})$    ▷ Query error for the 1st triangle match.
        for $\{\tau_{\alpha,jk}, \tau_{\beta,no}\} \in \mathcal{M}$ $(k \neq i \text{ and } o \neq m)$ do    ▷ Find adjacent triangles.
            $E_2 \leftarrow \mathcal{E}(\tau_{\alpha,jk}, \tau_{\beta,no})$
            for $\{\tau_{\alpha,kl}, \tau_{\beta,op}\} \in \mathcal{M}$ $(l \neq i \text{ and } l \neq j \text{ and } p \neq m \text{ and } p \neq n)$ do
                $E_3 \leftarrow \mathcal{E}(\tau_{\alpha,kl}, \tau_{\beta,op})$
                if $\{\tau_{\alpha,il}, \tau_{\beta,mp}\} \in \mathcal{M}$ and $\{\tau_{\alpha,jl}, \tau_{\beta,np}\} \in \mathcal{M}$ then
                    $E_4 \leftarrow \mathcal{E}(\tau_{\alpha,il}, \tau_{\beta,mp})$
                    $E_5 \leftarrow \mathcal{E}(\tau_{\alpha,jl}, \tau_{\beta,np})$
                    $E_{\alpha\beta,tot} \leftarrow \sum_{y=1}^{5} E_y$
                    if $E_{\alpha\beta,tot} < E_{MAX}$ then
                        $\Lambda_{\alpha\beta} \leftarrow \{[i,m], [j,n], [k,o], [l,p]\}$
                        matchFlag ← true
                        return (matchFlag, $E_{\alpha\beta,tot}$, $\Lambda_{\alpha\beta}$)    ▷ Match found.
                    end if
                end if
            end for
        end for
    end for
    return matchFlag    ▷ Match not found.
end function
```

Match Confirmation

After obtaining a set of candidate matches μ, a match confirmation algorithm is employed to (1) accept or reject any given matches and (2) select the best match among the accepted ones, for any given feature. The first step in the process is as follows. Let μ_α be the set of matches corresponding to a generic feature with ID α. Among all matches in μ_α, let β and γ be the feature IDs associated with the two minimum values of matching error, i.e. $E_{\alpha\beta,tot}$ and $E_{\alpha\gamma,tot}$. Then, candidate feature matches β and γ are kept, while all other matches are rejected. To choose the best candidate match between β and γ, the number of ID pairs appearing both in $\Lambda_{\alpha\beta}$ and μ is counted. The result is an integer score $s_{\alpha\beta}$. The same process is performed for $\Lambda_{\alpha\gamma}$ and μ, obtaining the score $s_{\alpha\gamma}$. Finally, if $s_{\alpha\beta} > s_{\alpha\gamma}$, then the feature β is confirmed as the best match for feature α. Additionally, all the matches in $\Lambda_{\alpha\beta}$ are accepted as the best matches. Otherwise, if $s_{\alpha\beta} < s_{\alpha\gamma}$, then feature γ is confirmed as being the best match for feature α and all the matches in $\Lambda_{\alpha\gamma}$ are accepted as best matches. In case both $s_{\alpha\beta}$ and $s_{\alpha\gamma}$ are zero, feature α is rejected and not included in the set of confirmed feature matches. This process is repeated for all features in the two images.

The reasoning behind the above feature confirmation algorithm is that, if a feature match is correct, then most (if not all) of the matches for the neighboring features should also be correct as well, as they are often part of the same triangles used for matching. In other words, this process uses geometric constraints between neighboring features to cross-validate matches. This is true for sufficiently low numbers of missing keypoints.

Matching Results

We present here feature matching results using the low-variance feature detections (Figure 6b) previously described. The 50 strongest features are selected in both the first and the second image for matching. Feature matches are shown in Figure 9 together with the two images used for matching. Out of 50 detections in each frame, the algorithm provides 13 matches (i.e. 26%). The totality of such feature matches are inliers, i.e. the matched keypoints represent the same physical points on the surface (up to the detection error). Figure 10 shows a magnified version of the image patches around all matched features. One can visually verify that the appearance of matched features is substantially different in both viewpoint (i.e., geometric appearance) and lighting conditions, with the Sun phase angle almost symmetrical with respect to the surface normals, between the two images.

For reference, the number of good features matches (inliers) using KAZE feature detection and description, using the original images, with default tuning parameters, and selecting the strongest 1000 features in each frame, is zero (0%). Using the same setup but increasing the number of selected strongest features to 10000, the number of good matched rises to 23 (0.23%); however, note that 20000 KAZE descriptors require roughly 5 GB of storage. Finally, when using SURF features with default parameters for the matching task, only 678 and 744 features are detected in total in the first and second image, respectively. The number of good matches related to such features is also zero. Therefore, we conclude that the five-triangle algorithm substantially outperforms state-of-the-art techniques for the purpose of feature matching, in the tested scenario of an asteroid surface with substantially different lighting and viewpoint conditions. We showed that a relatively low number of detections (i.e. 50 features) is sufficient to match the image with the previously observed one, thus solving the loop closure problem. In this case, no outliers were produced by the matching task. However, note that outliers can effectively be rejected by estimating the fundamental matrix and hence leveraging pose estimates, as previously described. Future work will further increase both the accuracy and robustness of both feature detection and description.

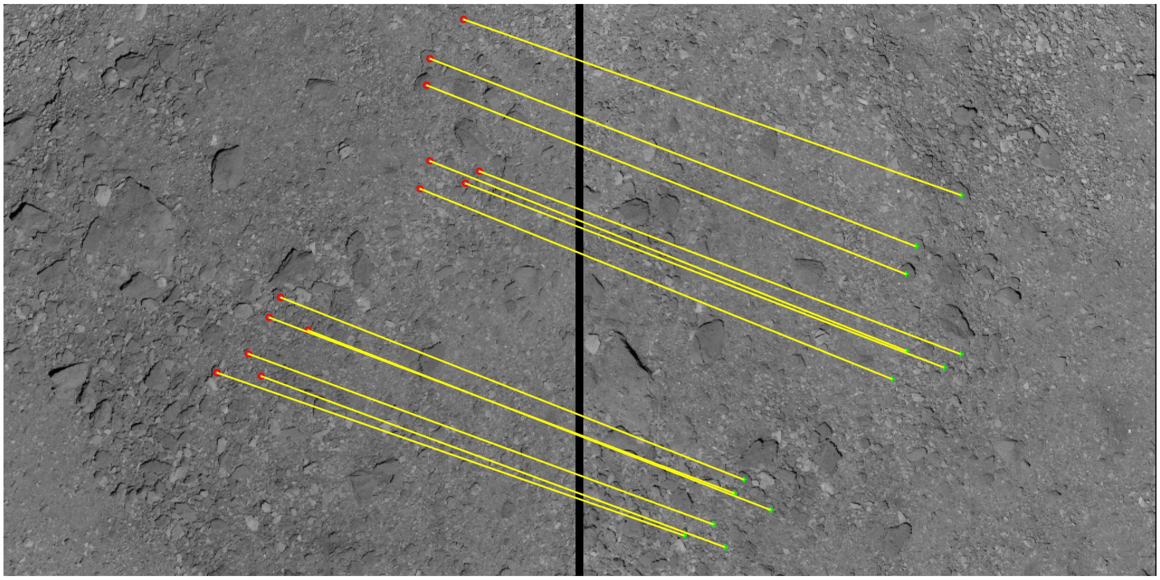


Figure 9: Matched features using the 5-triangle algorithm, under different lighting and viewpoint. 13 out of 50 (25%) features for each frame are matched. The totality of matches are inliers. Zero of out 1000 features (0%) were matched using KAZE detection and descriptors, and zero out of all detected (678) features (0%) were inlier matches using SURF detection and descriptors. Features are represented on top of the related OSIRIS REx's MAPCAM images.

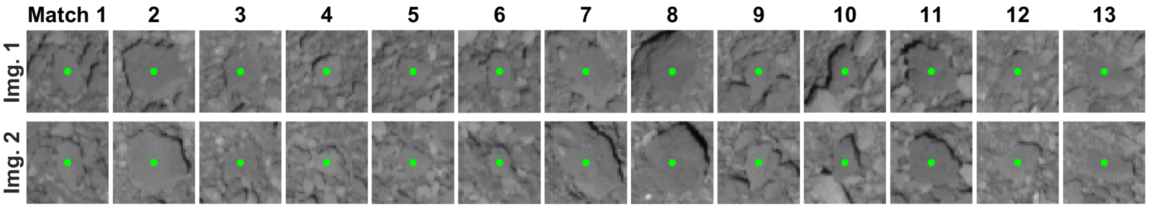


Figure 10: Matched features (green dots) shown on top of neighboring image regions. Rows represent features in the same image; columns represent pairs of feature matches.

DENSE MAPPING

The camera poses and landmarks estimated by the Structure-from-Motion process can be used to perform dense-shape reconstruction of the target body. If a portion of the surface is imaged from multiple viewpoints, geometric relationships between observations can be used to estimate the 3D profile of the observed surface. In particular, if the camera pose and some 3D surface landmarks are assumed to be known for each observation, then the photometric difference between observations can be used to infer the surface albedo and the 3D topography.

In this section, we assess the performance of stereophotogrammetric shape estimation for the case where reference camera poses and surface landmarks are autonomously estimated using SfM. We perform this analysis using *Agisoft Metashape*[*], a well-known software tool for photogrammetry applications based on digital imagery. The tool is based on the semiglobal matching method[33] for stereo processing, whereas the mesh generation from a dense point cloud is performed using the total generalized variation minimization technique.[34]

The accuracy of the estimated 3D data depends on the accuracy of the reference parameters, i.e. the camera poses and surface landmarks, as well as user-set tolerances defining the overall quality of the shape products. In turn, these trade with computational requirements and processing time. In this study, we set the quality requirements to their maximum so to assess the overall capabilities of using such techniques for small-body imagery. The impact of this choice on the SfM phase is that more features are processed than what stated in the above SfM-only simulations, to ensure a high density of surface landmarks for stereo matching. Future work will relax accuracy tolerances and provide more insights into computational requirements for onboard processing. We present SPG results for 3 different image sets: (1) a full revolution around the asteroid Bennu, observed from the OSIRIS-REx's MAPCAM (two of the images in this dataset are those shown in Figure 9), with a total of 180 images; (2) the pre-TaG scenario imaged using the NFTCAM, presented in the SfM section (Figure 2); (3) the TaG descent scenario from the NFTCAM. In the NFTCAM cases, the camera intrinsic parameters (focal length, lens distortion, and center pixel) are co-estimated in the process, with no initial guesses provided.

The MAPCAM image set is used to reconstruct a global shape model of the asteroid Bennu. The asteroid is observed from a distance of approximately 4 km from its center of mass, which corresponds to a surface resolution of about 24 cm/pixel. We perform dense stereo matching first, and we use the resulting point cloud to estimate a triangle mesh of the asteroid. The result is the 4.1-million-faces, which we compare against Bennu's 3.1-million-faces shape model, used as the ground truth. Figure 11 compares the estimated and ground-truth shapes in terms of their Hausdorff distance computed on the vertices.[35] Shapes are adjusted for scale for this comparison. The mean error is 0.892 m and the maximum error is 11.9 m. As shown by the colormap, the shapes differ by less than 1 meter for the majority of surface areas. Large errors are mostly caused by poor observability of the region of interest, because of surface occlusions, shadows, or both, bad geometry of the viewpoints, or any of these effects combined. Further, one should note that the reference shape model is not merely computed using the same 180-image data set, but was produced as part of a much broader mapping campaign.

SPG results for the NFTCAM image sets are shown in Figure 12, in terms of raw estimates of the dense point clouds. In both cases, the rich rubble-pile topography of Bennu is captured and rocks appear separate from each other. Each point in the cloud is also associated with an albedo estimate,

[*] `https://www.agisoft.com/`

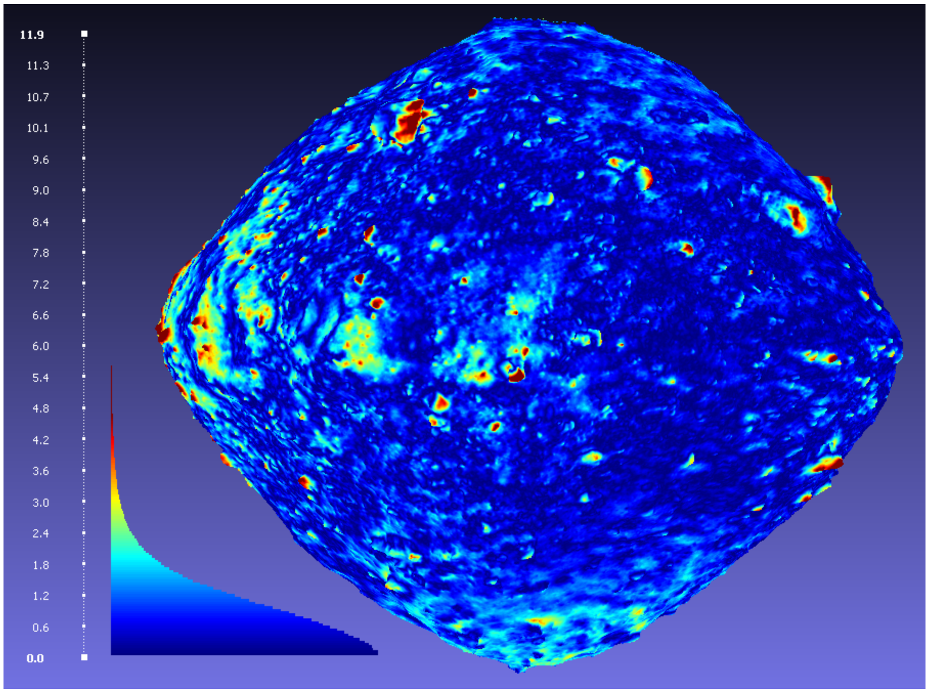
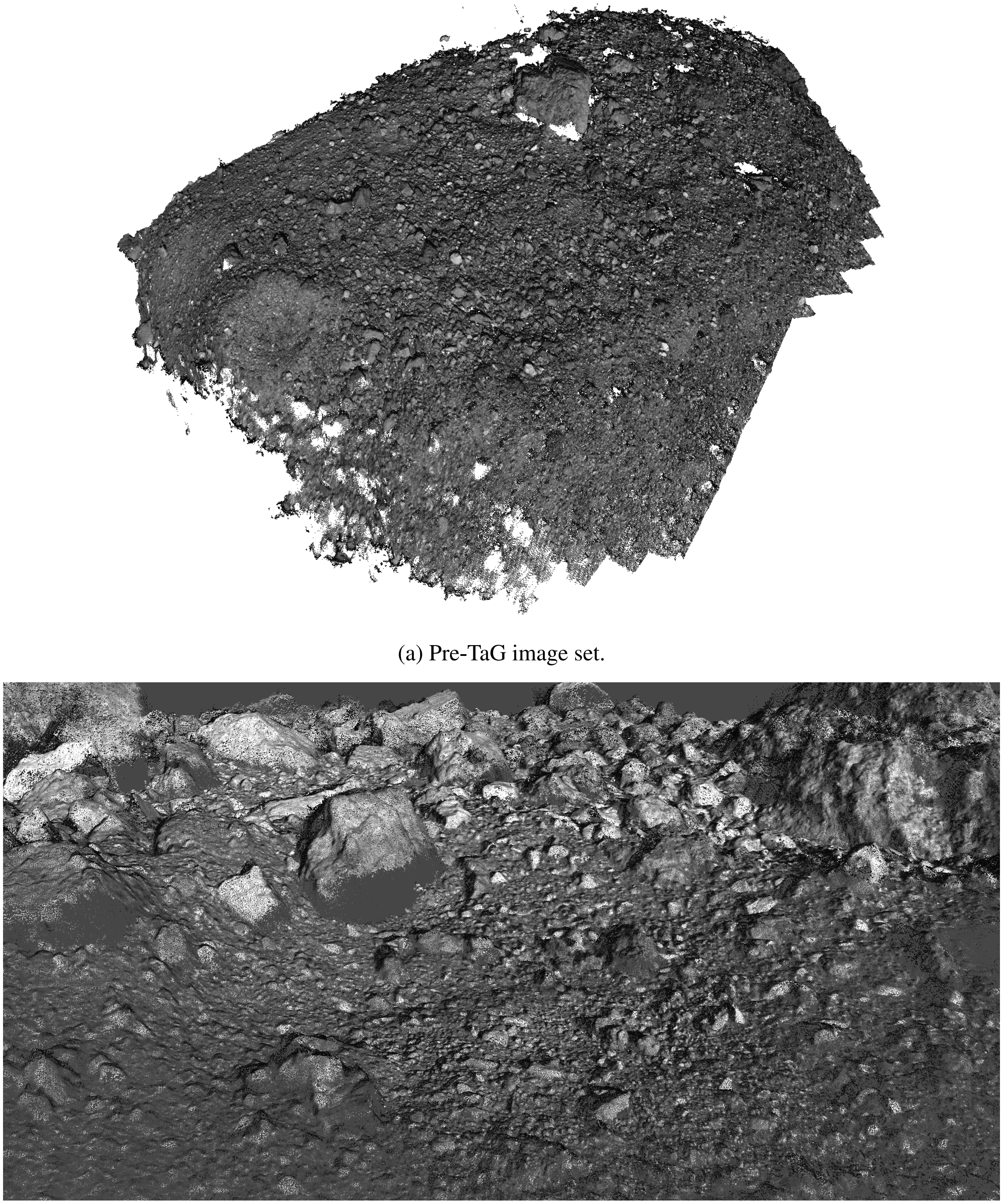


Figure 11: Estimated global shape model of asteroid Bennu, using Stereophotogrammetry. Vertex colors represent the Hausdorff distance error metric compared to the reference shape, adjusted for scale. The error is also reported as a histogram (bottom-left). The mean error is 0.892 m and the maximum error is 11.9 m.

which capture differences in the apparent brightness. Despite the lack of *a-priori* camera calibration, the algorithm is able to recover the 3D scene accurately. Some surface regions, such as those around some large boulders, are occluded for all observations and hence cannot be reconstructed in 3D. On the other hand, regions that are only occluded in some images but are visible in others (e.g. because of a moving shadow or changing viewpoint) are updated in the stereo matching process by merging information from all observations. As such, the stereo matching algorithm can compensate for moving shadows and accumulate information while the body rotates and the Sun phase evolves.

Additionally, the stereo matching algorithm accounts for different image resolutions among observation, and retains the highest-resolution estimates. For example, Figure 12b shows the topography reconstructed using the NFTCAM while descending to the Touch-and-Go site, Nightingale. In this case, the surface in the immediate surrounding of the site is more resolved than the further regions, as multiple images were sampled up-close. This scenario suggests that photogrammetry data acquired at different altitudes and during different mission phases may be merged together automatically.

In conclusion, although a limited number of observations are used to produce the shape estimates presented, results suggest that an end-to-end autonomous shape estimation process can produce compelling results for a number of applications, such as onboard hazard assessment and avoidance.



(a) Pre-TaG image set.

(b) TaG image set capturing the Nightingale site (bottom center) and the surrounding region. The point cloud density increases towards the site, as a function of the surface resolution from descent imagery.

Figure 12: Raw estimates of dense point clouds and points albedo using the NFTCAM image sets.

CONCLUSIONS

In this work, we present an algorithmic pipeline for autonomous navigation and dense shape reconstruction for small body missions. The pipeline requires no prior knowledge of the observed scenario and only makes use of imaging data. While we plan to test different scenarios and extending the algorithmic capabilities on most fronts (feature detection, extraction, and matching, as well as pose estimation and integration with dynamics and a navigation filter), as described in the previous sections, the presented results reveal some findings. In particular, we conclude that, for a rubble-pile surface body observed using close-proximity imagery, and using real imaging data:

1. frame-to-frame feature matching is effective for the purpose of precise surface-relative position estimates and 3D-landmark estimation, with no *a-priori* landmark estimates and using a relatively low number of features;
2. surface photometric properties can be leveraged to detect morphological features – in particular rocks, such as low-variance and high-albedo features, previously described;
3. surface 3D landmarks and their geometric relationships provide feature descriptors that are robust to changes in lighting and viewpoint (including scale);
4. traditional dense stereo matching techniques can effectively reconstruct a dense representation of the observed small-body shape and related topography. This process is also robust to temporary occlusions due to moving shadows and can merge shape estimates with even large changes in local resolution.

ACKNOWLEDGEMENTS

A portion of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. ©2022 California Institute of Technology. Government sponsorship acknowledged.

REFERENCES

[1] E. Asphaug, E. V. Ryan, and M. T. Zuber, "Asteroid interiors," *Asteroids III*, Vol. 1, 2002, pp. 463–484.

[2] D. J. Scheeres, D. Britt, B. Carry, and K. A. Holsapple, "Asteroid interiors and morphology," *Asteroids IV*, Vol. 745766, 2015, pp. 745–766.

[3] M. C. Bazzocchi and M. R. Emami, "Comparative analysis of redirection methods for asteroid resource exploitation," *Acta Astronautica*, Vol. 120, 2016, pp. 1–19.

[4] "Defending Planet Earth: Near-Earth-Object Surveys and Hazard Mitigation Strategies," tech. rep., 2010.

[5] D. Lauretta, S. Balram-Knutson, E. Beshore, W. Boynton, C. Drouet d'Aubigny, D. DellaGiustina, H. Enos, D. Golish, C. Hergenrother, E. Howell, *et al.*, "OSIRIS-REx: sample return from asteroid (101955) Bennu," *Space Science Reviews*, Vol. 212, No. 1, 2017, pp. 925–984.

[6] S.-i. Watanabe, Y. Tsuda, M. Yoshikawa, S. Tanaka, T. Saiki, and S. Nakazawa, "Hayabusa2 mission overview," *Space Science Reviews*, Vol. 208, No. 1, 2017, pp. 3–16.

[7] K.-H. Glassmeier, H. Boehnhardt, D. Koschny, E. Kührt, and I. Richter, "The Rosetta mission: flying towards the origin of the solar system," *Space Science Reviews*, Vol. 128, No. 1, 2007, pp. 1–21.

[8] D. Y. Oh, S. Collins, T. Drain, W. Hart, T. Imken, K. Larson, D. Marsh, D. Muthulingam, J. S. Snyder, D. Trofimov, *et al.*, "Development of the psyche mission for NASA's discovery program," 2019.

[9] H. F. Levison, C. B. Olkin, K. S. Noll, S. Marchi, J. F. Bell III, E. Bierhaus, R. Binzel, W. Bottke, D. Britt, M. Brown, *et al.*, "Lucy mission to the Trojan asteroids: Science goals," *The Planetary Science Journal*, Vol. 2, No. 5, 2021, p. 171.

[10] D. J. Scheeres, J. McMahon, B. Bierhaus, J. Wood, L. Benner, C. M. Hartzell, P. O. Hayne, R. Jedicke, L. Le Corre, S. Naidu, *et al.*, "Janus: A NASA SIMPLEx mission to explore two NEO Binary Asteroids," *AGU Fall Meeting Abstracts*, Vol. 2020, 2020, pp. A252–01.

[11] C. Snodgrass and G. H. Jones, "The European Space Agency's comet interceptor lies in wait," *Nature communications*, Vol. 10, No. 1, 2019, pp. 1–4.

[12] R. Gaskell, O. Barnouin-Jha, D. J. Scheeres, A. Konopliv, T. Mukai, S. Abe, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, *et al.*, "Characterizing and navigating small bodies with imaging data," *Meteoritics & Planetary Science*, Vol. 43, No. 6, 2008, pp. 1049–1061.

[13] I. Nesnas, J. Castillo, T. Swindle, S. Bhaskaran, D. Gump, L. Maleki, J. McMahon, C. Mercer, H. Partridge, M. Pavone, *et al.*, "Autonomy Workshop Small Body Design Reference Mission," 2019.

[14] D. A. Lorenz, R. Olds, A. May, C. Mario, M. E. Perry, E. E. Palmer, and M. Daly, "Lessons learned from OSIRIS-Rex autonomous navigation using natural feature tracking," *2017 IEEE Aerospace Conference*, IEEE, 2017, pp. 1–12.

[15] A. Pellacani, M. Graziano, M. Fittock, J. Gil, and I. Carnelli, "HERA vision based GNC and autonomy," *Proceedings of the 8th European Conference for Aeronautics and Space Sciences (EUCASS), Madrid, Spain*, 2019, pp. 1–4.

[16] B. J. Morrell, J. Villa, S. Bandyopadhyay, D. Lubey, B. Hockman, S. Bhaskaran, D. Bayard, and I. Nesnas, "Autonomous feature tracking for autonomous approach to a small body," *ASCEND 2020*, p. 4151, 2020.

[17] I. A. Nesnas, B. J. Hockman, S. Bandopadhyay, B. J. Morrell, D. P. Lubey, J. Villa, D. S. Bayard, A. Osmundson, B. Jarvis, M. Bersani, *et al.*, "Autonomous Exploration of Small Bodies Toward Greater Autonomy for Deep Space Missions," *Frontiers in Robotics and AI*, Vol. 8, 2021.

[18] M. Dor, K. A. Skinner, T. Driver, and P. Tsiotras, "Visual SLAM for Asteroid Relative Navigation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2066–2075.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, Vol. 110, No. 3, 2008, pp. 346–359.

[20] B. Bos, M. Ravine, M. Caplinger, J. Schaffner, J. Ladewig, R. Olds, C. Norman, D. Huish, M. Hughes, S. Anderson, *et al.*, "Touch And Go Camera System (TAGCAMS) for the OSIRIS-REx asteroid sample return mission," *Space Science Reviews*, Vol. 214, No. 1, 2018, pp. 1–23.

[21] B. Rizk, C. Drouet d'Aubigny, D. Golish, C. Fellows, C. Merrill, P. Smith, M. Walker, J. Hendershot, J. Hancock, S. Bailey, *et al.*, "OCAMS: the OSIRIS-REx camera suite," *Space Science Reviews*, Vol. 214, No. 1, 2018, pp. 1–55.

[22] C. H. Acton Jr, "Ancillary data services of NASA's navigation and ancillary information facility," *Planetary and Space Science*, Vol. 44, No. 1, 1996, pp. 65–70.

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

[24] T. Lindeberg, "Scale invariant feature transform," 2012.

[25] R. Subbarao and P. Meer, "Beyond RANSAC: user independent robust regression," *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, IEEE, 2006, pp. 101–101.

[26] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[27] M. Pugliatti, M. Maestrini, P. Di Lizia, and F. Topputo, "On-Board Small-Body Semantic Segmentation Based on Morphological Features with U-Net," *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021, pp. 1–20.

[28] L. Downes, T. J. Steiner, and J. P. How, "Deep learning crater detection for lunar terrain relative navigation," *AIAA SciTech 2020 Forum*, 2020, p. 1838.

[29] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, IEEE, 1998, pp. 839–846.

[30] C. Tomasi and T. Kanade, "Detection and tracking of point," *Int J Comput Vis*, Vol. 9, 1991, pp. 137–154.

[31] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," Vancouver, 1981.

[32] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, Vol. 4, No. 2, 2009, p. 1883.

[33] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 30, No. 2, 2007, pp. 328–341.

[34] N. Poliarnyi, "Out-of-Core Surface Reconstruction via Global TGV Minimization," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5641–5650.

[35] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, “Mesh: Measuring errors between surfaces using the hausdorff distance,” *Proceedings. IEEE international conference on multimedia and expo*, Vol. 1, IEEE, 2002, pp. 705–708.