
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Knuuttila, Olli; Kestilä, Antti; Kallio, Esa

Synthetic photometric landmarks used for absolute navigation near an asteroid

Published in:
Aeronautical Journal

DOI:
[10.1017/aer.2020.41](https://doi.org/10.1017/aer.2020.41)

Published: 01/09/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Knuuttila, O., Kestilä, A., & Kallio, E. (2020). Synthetic photometric landmarks used for absolute navigation near an asteroid. *Aeronautical Journal*, 124(1279), 1281-1300. Article 000192402000041.
<https://doi.org/10.1017/aer.2020.41>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Synthetic photometric landmarks used for absolute navigation near an asteroid

Olli Knuuttila
olli.knuuttila@aalto.fi

Department of Electronics and Nanoengineering, Aalto University School of Electrical Engineering, Espoo, Finland

Antti Kestilä
antti.kestila@fmi.fi

Finnish Meteorological Institute, Helsinki, Finland

Esa Kallio
esa.kallio@aalto.fi

Department of Electronics and Nanoengineering, Aalto University School of Electrical Engineering, Espoo, Finland

ABSTRACT

The need for autonomous location estimation in the form of optical navigation is an essential requirement for forthcoming deep space missions. While crater based navigation might work well with larger bodies littered with craters, small sub-kilometer bodies do not necessarily have them. We have developed a new pose estimation method for absolute navigation based on photometric local feature extraction techniques thus making it suitable for missions that cannot rely on craters. The algorithm can be used by a navigation filter in conjunction with relative pose estimation such as visual odometry for additional robustness and accuracy. To estimate the position and orientation of the spacecraft in the asteroid-fixed coordinate frame, it uses navigation camera images in combination with other readily available information such as orientation relative to the stars and the current time for an initial estimate of the asteroid rotation state. Evaluation of the algorithm when using different feature extractors is performed, on one hand, using Monte Carlo simulations and, on the other hand, using actual images taken by the Rosetta spacecraft orbiting the comet 67P/Churyumov–Gerasimenko. Our analysis, where four different feature extraction methods (AKAZE, ORB, SIFT, SURF) were compared, showed that AKAZE is most promising in terms of stability and accuracy.

NOMENCLATURE

ATON	Autonomous Terrain-based Optical Navigation
BSM	Binary Shadow Matching
DLR	The German Aerospace Center
FoV	Field of View
GTM	Global Topography Model
ICP	Iterative Closest Point algorithm
IMU	Inertial Measurement Unit
LDB	Landmark Database
MSE	Mean Squared Error
NavCam	Navigation Camera
P3P	Perspective-3-point Problem
SfM	Structure from Motion
SHOE	Shadow-Hiding Opposition Effect
SLAM	Simultaneous Localization and Mapping
SPL	Synthetic Photometric Landmark
UKF	Unscented Kalman Filter
VO	Visual Odometry

Greek Symbol

α	Phase Angle
μ	Emittance Angle
μ_0	Incidence Angle
ρ	Scaling Constant
ω	Single Scattering Albedo

1.0 Introduction

Visual navigation, also called vision-based or optical navigation, is an important research topic and of particular interest for future CubeSat missions that will operate close to a distant solar system small body, as they will need to independently position themselves relative to their target.

Navigation methods, in general, can be divided into two classes: absolute and relative. Absolute navigation uses measures (line-of-sight, position, orientation, pose) in either the target- or spacecraft-fixed coordinate frame. Examples of absolute navigation use various landmark- and global map-based methods. In turn, relative navigation exclusively uses measures of change relative to an earlier point in time, e.g. the change in pose since a starting time. Typically, however, they return absolute pose estimates in an arbitrary internal frame or in a frame initialized to coincide with a real-world coordinate frame. Measurement errors will build up over time leading to a growing bias in the absolute pose estimates returned. Relative pose estimation in the visual navigation context is called visual odometry (VO). When integrated with a navigation filter, VO can increase the navigation accuracy significantly by providing constraints from the current state to past states. At least one absolute pose-estimation method is typically needed to limit the error due to the drift inherent in VO.

The work presented here was started as an investigation to determine the visual navigation architecture suitable for a CubeSat mission called APEX⁽¹⁾ (previously ASPECT⁽²⁾), which is scheduled to be deployed in 2027 near the binary asteroid 65803 Didymos by ESA's Hera mission⁽³⁾. APEX will be deployed at a distance of 10 kilometers from Didymos and then navigate to a 4-kilometer circular orbit, where it will stay for several weeks. Next, APEX will navigate to the L_4 and L_5 points in the Didymos system, roughly 1200 meters from the centers of the primary and secondary body, whose mean diameters are around 780 meters and 135 meters, respectively⁽⁴⁾. The extended mission plan also includes periods where APEX will stay near the L_1 and L_2 points at a distance of around 180 meters from the center of mass of the slightly elongated secondary body. The altitude is expected to be from 70 to 90 meters during these periods.

1.1 Reference mission

Even though our interest is with the APEX mission, the analysis presented in this article uses the Rosetta mission scenario for the comet 67P/Churyumov–Gerasimenko. The reason for this is that no detailed 3D models or images from Didymos were available when this research was performed, while an accurate 3D model⁽⁵⁾ and NavCam images⁽⁶⁾ are now available from the Rosetta mission. These images also have metadata which give the comet position and Rosetta pose. Although our reference mission is to a comet, we will continue to use the term asteroid, as the use of “small body” would be somewhat unwieldy. Other asteroid parameters such as orbital elements and the rotation period are taken from this comet. The Rosetta NavCam parameters of $5^\circ \times 5^\circ$ FoV and 1024 x 1024 pixel resolution were used in order to allow testing with actual Rosetta NavCam images.

1.2 Past missions and related work

The first asteroid mission which used a form of optical navigation was NASA's NEAR mission⁽⁷⁾. Earth-based human analysts had a global topography model (GTM or shape model) of asteroid 433 Eros, which was projected on top of a target NavCam image. The position and orientation of the model were adjusted manually so that the limbs would match. When the probe was too close to the asteroid to have its limbs visible, 3D-ellipses were fitted manually to craters and used as landmarks.

The Rosetta comet mission⁽⁸⁾ and the Dawn asteroid mission⁽⁹⁾ used human analysts for shape model estimation and the creation of landmarks. The landmarks consisted of a detailed, textured 3D-model of a small region around a physical landmark like a boulder or crater and were called “L-maps”. Phase correlation was used to match the rendered L-maps in the database to NavCam images. To be able to render the L-maps, a rough relative pose of the spacecraft and asteroid was needed in addition to the direction of sunlight. The previous pose estimate and possibly a phase correlation pass using the rendered asteroid shape model, were used. The same method is currently used by the OSIRIS-REx asteroid mission⁽¹⁰⁾ to 101955 Bennu, with the difference that landmark matching can be done on-board.

Hayabusa 1⁽¹¹⁾⁽¹²⁾ used a combination of asteroid centroids estimated from NavCam images and distance measurements from a laser range finder to position itself in relation to the asteroid 25143 Itokawa while the asteroid could fit in the FoV. When closer than that, the original plan had been to use a template extraction and matching scheme and to measure the distance to the matched areas using the laser range finder. However, this method was

never used. Navigation was done manually from the ground instead because one of the reaction wheels had broken down, thus preventing targetted laser measurements. For landing, Hayabusa 1 used a combination of the laser range finder and tracked artificial target markers that it deployed on Itokawa.

Some scale and rotation invariant automatic local feature detectors like SIFT⁽¹³⁾, SURF⁽¹⁴⁾, BRISK⁽¹⁵⁾ and ORB⁽¹⁶⁾ were evaluated by Takeishi et al.⁽¹⁷⁾ for the Hayabusa 2 mission that arrived in June 2018 at the asteroid 162173 Ryugu. However, at the time of writing, no public information seemed to be available about the navigation algorithms used by this mission.

The German Aerospace Center (DLR) has been working since 2010 on a comprehensive optical navigation system called Autonomous Terrain-based Optical Navigation (ATON)⁽¹⁸⁾. Their reference mission is to land a probe on the Moon using a sensor set composed of an inertial measurement unit (IMU), star tracker, navigation camera, a laser altimeter, and a LIDAR. Measurements of these sensors are used by a wide range of different navigation algorithms and the results are fused in a navigation filter. Different algorithms are used at different distances from the lunar surface. For example, crater navigation is used while still relatively far, at a distance of 10–100 km. This algorithm is based on detecting craters modeled by adjacent areas of above and below average brightnesses. The detected craters are matched with a global crater database for a position estimate. When starting the powered descent stage at an 10 km altitude, navigation starts to rely on a filter based visual odometry (VO) algorithm⁽¹⁹⁾ that tracks Shi–Tomasi features⁽²⁰⁾, which are similar to Harris corners and vary based on scale and orientation.

It is worth noting, that there is a large variety of other visual odometry—and by extension, simultaneous localization and mapping (SLAM)—algorithms in existence. Visual SLAM differs from VO in that it can achieve limited absolute navigation by constructing a global map⁽²¹⁾⁽²²⁾. The global map is constructed by detecting the arrival back into the previously explored territory (loop closing). SLAM can only perform absolute navigation when revisiting previously mapped areas. One challenge is to maintain the map small enough for the target hardware.

ATON handles the final, almost vertical descent phase—starting at an altitude of 1 km—by using two absolute navigation algorithms: the binary shadow matching algorithm (BSM)⁽²³⁾ and the iterative closest point (ICP) algorithm. The BSM uses groups of shadows as local features and matches them to a database of pre-calculated reference features. The ICP algorithm matches a point cloud extracted by a LIDAR or by a structure from motion (SfM) algorithm to a reference point cloud.

1.3 Preliminary choices and contribution

By leaving out the final landing phase, the overall architecture of ATON seems relevant to the case analyzed in this paper. There is, however, a concern that there might not be enough craters for crater-based absolute navigation on the target binary asteroid Didymos, as this seems to be the case with sub-kilometer asteroids, e.g. Itokawa, Bennu and Ryugu. Even the comet 67P/Churyumov–Gerasimenko—which is ten times larger than Ryugu—has sides with no obvious craters. Moreover, the diameters of the Didymos primary and secondary target bodies are only around 780 meters and 135 meters, respectively⁽⁴⁾. The two other options for absolute navigation—BSM and ICP—have their challenges. ICP relies on VO and SfM to provide a point cloud measurement, which might fail when the appearance of the target changes fast compared to the relative pose. One such situation for APEX is during mission

phases where it observes the secondary body from the relatively stable L_4 and L_5 points. Also, tracking is lost from time to time due to eclipses and periods of high phase angle. SfM needs to track features for long enough to produce a relatively accurate point cloud, reducing the usefulness of ICP. Compared to generic local photometric features, BSM seems overly restricted as it only focuses on shadows, of which there might be too few, depending on the phase angle and distance.

This study is dedicated to developing a new algorithm that can provide medium-range navigation with absolute pose estimates for a wide variety of asteroid and comet targets. Medium-range is here contrasted with far-range when the asteroid is too far for individual features to be well resolved, and with near-range, when only a small portion of the asteroid fits in the FoV. An implementation of the algorithm should be expected to run on CubeSat hardware and be able to update the pose estimate every couple of minutes. For the purposes of this study, this is kept in mind but not validated.

Local photometric features have been studied for visual navigation near asteroids⁽²⁴⁾ or for a lunar landing^{(25) (26) (27)}. These features are typically rotation- and scale-invariant and are robust to affine transformations. However, they cope poorly with the stark change of appearance of an asteroid when the direction of light changes. To get an absolute pose estimate we need to be able to match features extracted from the current NavCam image with landmark features whose 3D coordinates are known. For the matching to work, the landmark features need to be from a scene with a similar viewpoint and direction of light. We see two ways of achieving this: 1) having a landmark database (LDB) spanning a four-dimensional scene parameter space composed of two asteroid coordinates and two direction-of-light coordinates populated offline and/or online by using SLAM methodology, and; 2) using a detailed global topography model that is rendered on-board for a synthetic reference image, from which local photometric features are extracted to serve as landmarks. We call these synthetic photometric landmarks (SPLs). Although the LDB approach seems feasible and it is used by the above-mentioned studies using photometric features, we decided to explore the SPL approach as it seemed conceptually simpler and it avoids the risk of a prohibitively large LDB. The main contribution of this work is to

1. Develop an SPL-based pose estimation algorithm for absolute navigation near asteroids. To the authors' knowledge, there are no previous studies exploring asteroid pose estimation using photometric features that are extracted from on-board rendered synthetic images. Methods that have been studied before typically compare photometric features across different camera frames.
2. Identify, explore and analyze different design options and configurations.
3. Analyze the algorithm performance under varying environmental conditions.
4. Select a reasonable algorithm configuration.
5. Define bounds on the environmental conditions inside which the algorithm is expected to perform well.
6. Investigate the expected performance of the algorithm when inside this operational zone.

The article is organized as follows. First, we present the proposed algorithm. Then, we go through its analysis and results. Finally, we discuss the lessons learned and the applicability of the proposed approach for a deep space mission.

2.0 The proposed pose estimation method

The algorithm studied here works by extracting features and their corresponding image coordinates from a target image. The features consist of a descriptor vector, which can be used to match the features from the target image with previously extracted landmark features. These SPLs have associated with them real-world 3D coordinates in the asteroid's frame of reference. With multiple good matches, the projection matrix that transforms each matched landmark's 3D coordinate to the corresponding feature's 2D image coordinates can be solved, giving an estimate of the relative pose between the asteroid and the spacecraft (Fig. 1).

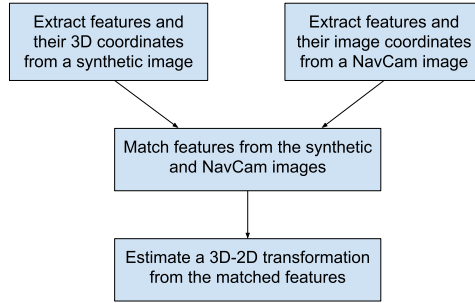


Figure 1. Overview of the algorithm analyzed in this work with its main steps visible.

For local feature extraction we consider SIFT⁽¹³⁾, SURF⁽¹⁴⁾, ORB⁽¹⁶⁾ and AKAZE⁽²⁸⁾. All have implementations in OpenCV⁽²⁹⁾. More details and a comparison of them can be found in Chien et al.⁽³⁰⁾. In short, for each feature point found, these methods return a descriptor vector, descriptor strength, orientation, size, and 2D image coordinates (see Fig. 2). Descriptor strength is used when deciding which features to retain for further processing as computational resources are limited. Some general properties of the algorithms are presented in Table 1.

Table 1
Feature extractors/descriptors investigated

Algorithm	Descriptor	Expected speed	Year, license
ORB	binary, 32 bytes	very fast	2011, open source
AKAZE	binary, 61 bytes	fast	2011, open source
SIFT	float, 128 bytes	slow	2004, US patents
SURF	float, 64 bytes	fast	2008, US patents

We will next discuss the initial state estimate, synthetic image generation from the GTM, feature extraction, details about feature matching and, finally, solving the projection matrix.

2.1 System state model and the initial state estimate

The system state consists of the asteroid position relative to the Sun r_{ast} , asteroid orientation relative to the stars q_{ast} , spacecraft position relative to the asteroid r_{sc} , and spacecraft orientation relative to the stars q_{sc} . The initial state estimate is arrived at by combining different sources of information:

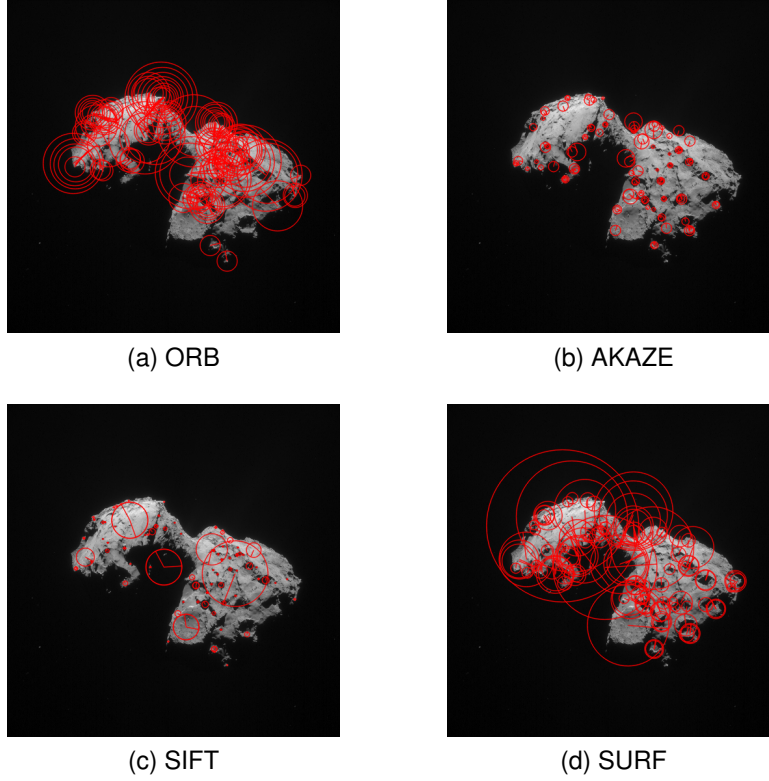


Figure 2. An example of Rosetta NavCam image⁽³¹⁾ with features extracted by different algorithms. Feature image coordinates (circle centers), orientation (direction of the line within a circle), and size (circle radius) are shown.

- r_{ast} is modeled using the Keplerian orbital elements of the asteroid and solving the Kepler problem⁽³²⁾. Apart from having the orbital elements available, the spacecraft also needs to maintain time.
- q_{ast} is given by the asteroid rotation model. Somewhat accurate timekeeping is needed as with a fast spinning asteroid—for NEAs, around 10 rotations per day⁽³³⁾—an error of 24s corresponds to an error in the orientation of 1° .
- r_{sc} is estimated by either using a recent, previous result from this algorithm or from a far-range algorithm in use, which could be based on e.g. the center of brightness when taking into account the asteroid shape, like in section 3.2 in Rowell et al.⁽³⁴⁾, or in section 5 in Pellacani et al.⁽³⁵⁾.
- q_{sc} is measured by a star tracker.

The asteroid rotation model used here is a simple rotation around an axis. This approximation is quite reasonable as most asteroids rotate around their maximum inertia principal-axis^{(36) (37)}. Modeling non-principal-axis rotation states is left for future work.

To increase the robustness of the complete navigation system, it would be beneficial to have some algorithm that could estimate a rough pose even without an initial system state

in case the spacecraft fails to keep the current time. Such an algorithm could be based on a deep learning system or a system based on clustering the feature space to a limited number of classes and matching the class distribution in the current scene with the class distributions in different scenes saved into a database. There is a survey done by Zheng et al.⁽³⁸⁾ of these methods, but the exploration of them is beyond the scope of this study.

2.2 Getting synthetic photometric landmarks using a GTM

In this work, we evaluate a system where a global topography model (GTM) is rendered on board the spacecraft for a synthetic image of the asteroid from where the landmark features (SPLs) are then extracted. This approach is attractive because it can handle all the different view angles and lighting conditions. The GTM is estimated offline based on images taken from a parking orbit or by the mothership before deployment. For the algorithm to work, the GTM must be detailed enough. “How detailed?” is one of the questions we try to answer in this study.

The images are rendered on-board using a lunar–Lambertian reflectance model⁽³⁹⁾, as the same model was used by Grumpe et al.⁽⁴⁰⁾ related to machine vision and 67P/Churyumov–Gerasimenko. The lunar–Lambertian reflectance model as used by Grumpe is

$$I(\mu, \mu_0, \alpha) = \rho[2L(\alpha)\mu_0/(\mu + \mu_0) + (1 - L(\alpha))\mu_0], \quad \dots(1)$$

where μ_0 is the incidence angle, μ is the emittance angle, α is the phase angle, ρ is a scaling constant, and $L(\alpha)$ is a weight function depending on the phase angle, which could be read from figures from McEwen⁽³⁹⁾, where different functions corresponding to different Hapke model parameters have been presented⁽⁴¹⁾. For this paper, we parameterized the weight function $L(\alpha)$ as a 5th-order polynomial and fit the polynomial coefficients to match our synthetic NavCam images, which were rendered using a Hapke reflectance model.

We decided to use a Hapke model for the synthetic NavCam images to avoid a situation where the onboard rendering would use a perfectly matching reflectance model. We used single scattering albedo $\omega = 0.034$, shadow-hiding opposition effect (SHOE) amplitude $B_0 = 2.25$, and SHOE angular half-width $h_s = 0.061^\circ$ corresponding to the values previously estimated for the comet 67P/Churyumov–Gerasimenko by Fornasier et al.⁽⁴²⁾. We estimated the single-term Henyey–Greenstein function parameter to be $b = 0.3463$ and the effective large-scale roughness parameter to be $\theta = 27.1^\circ$. The reason for not estimating opposition surge parameters was that the Rosetta NavCam batches we considered did not contain samples with low enough phase angles. All the images have phase angles between 20° – 120° . For the same reason, we did not estimate the single scattering albedo ω , as it became part of a general scaling parameter that made the image brightnesses match.

The estimation of ω , B_0 and h_s was done by using 20 Rosetta NavCam images taken at roughly 5° phase angle intervals thus spanning the range 20° – 120° uniformly. We settled on using 20 images, as the results are very similar when using only ten images. The parameter values were optimized using the Nelder–Mead algorithm, which minimized a cost function that was calculated by rendering a synthetic image corresponding to each Rosetta NavCam image in the training set and calculating a pixel-wise mean squared error (MSE) between them. The correct comet pose for rendering was set based on the corresponding Rosetta NavCam image metadata.

Efficient, shadow mapping based self-shadowing was used⁽⁴³⁾. The system initial state is

used for positioning the camera and the direction of sunlight. Examples of real and synthetic images are shown in Fig. 3. Gamma correction of 1.8 is applied to all rendered images to match the gamma correction used for the Rosetta NavCam images during preprocessing. As a side product of a standard image rendering algorithm, a depth map is produced. This is used to determine the 3D coordinates of the local features extracted from the 2D image. Some of the extracted features are located outside the asteroid, as they describe some part of the limb contour. We could have discarded these features like in Rowell et al.⁽³⁴⁾. However, based on initial testing we obtained more accurate results when including them and setting their depth component to the depth of the nearest pixel belonging to the asteroid.

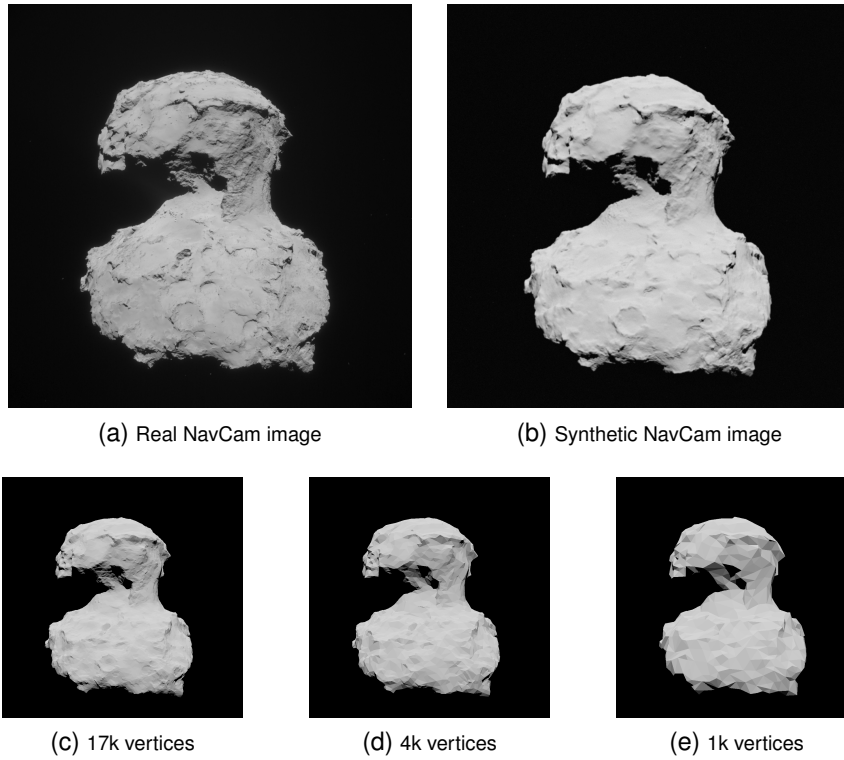


Figure 3. Five different image types were used in the study. Top left is an actual preprocessed Rosetta NavCam image, top right is an image rendered using a fitted Hapke reflectance model and a shape model with 1.9M vertices. Bottom row is rendered using a fitted lunar-Lambert reflectance model and shape models with varying vertex count: 17,000 (left), 4,000 (middle) and 1,000 (right).

2.3 Matching and pose estimation

Matching of the features extracted from the current scene with a set of known features is done by calculating the difference of descriptors in a brute force manner and returning the closest two matches for each scene feature. Matching could also be done using FLANN⁽⁴⁴⁾, which is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. However, we decided to use the brute force approach for simplicity. Following Lowe⁽¹³⁾, we

reject most spurious matches by comparing the descriptor distance of the best match with the distance of the second best match. If the distance ratio between best and second best match is less than a certain limit, the best match is included in the set of matched features. Lowe uses a ratio limit of 0.8 but after testing different values, we decided to use 0.85 instead because it resulted in a higher number of correct matches for all the feature extraction algorithms tested.

Matched feature points define various 3D to 2D mappings, which are used to solve the relative translation and rotation between the known reference features and the current scene images using RANSAC⁽⁴⁵⁾, which rejects geometrically inconsistent matches. The position of the spacecraft and possibly asteroid rotation state can be updated based on the results. RANSAC uses an inner algorithm to solve the perspective-3-point (P3P) problem that gives the camera pose based on a minimal set of three known feature points. We considered various P3P solving algorithms available in OpenCV and settled on AP3P⁽⁴⁶⁾ during initial testing.

By feeding the geometrically consistent matches from RANSAC to a navigation filter⁽²⁴⁾⁽⁴⁷⁾, the accuracy could be substantially improved. The asteroid rotation axis and rotation phase shift at epoch could be included in the filter state. These are left for future work.

All the algorithms, Monte Carlo simulation, error calculations, and Rosetta NavCam image loading were done using Python and OpenCV. The source code can be found on GitHub⁽⁴⁸⁾.

3.0 Results

We used Monte Carlo simulations to measure the performance of different feature extraction algorithms. For each generated sample we measured: i) spacecraft–asteroid distance error; ii) lateral displacement error, and; iii) orientation error. It is practical to give the first two errors as relative to the actual distance, as this way we can remove the geometric dependency on distance for these errors. The algorithm will exit early if there are less than 12 geometrically consistent feature matches, resulting in a pose estimation failure. For each feature extraction algorithm, we generate 10,000 Monte Carlo samples, after which we calculate algorithm failure rates and percentiles of the three different error measures. Each sampling consists of the following steps:

1. Pick a random set of parameter values that defines the system state.
2. Generate noisy measurements including a NavCam image.
3. Based on measurements from the previous step, run the SPL algorithm that then gives an estimate for the relative spacecraft–asteroid pose.
4. Compare estimated pose with the real pose from step 1, calculate and log the errors.

The relevant system state can be defined by the spacecraft orientation, the asteroid rotation axis, the right ascension of asteroid zero longitude at epoch, time since epoch and the asteroid–spacecraft relative position. The relative position is sampled using spherical coordinates with uniform distribution and distance between 25–400 km. The spacecraft orientation is sampled so that the asteroid center of volume is always contained in the FoV. The asteroid rotation axis parameters are not sampled but set to the values corresponding to the comet 67P/Churyumov–Gerasimenko. Time, which is sampled to be between 2015-01-01 and 2016-01-01, affects asteroid orientation and also the Sun–asteroid position. If the phase angle is larger than 140°, the system state is sampled again. The phase angle is the angle between the lines connecting the asteroid to the spacecraft and the asteroid to the Sun, i.e. a phase angle of 0° would mean

that the whole asteroid is in the sunlight and an angle of 180° means that no sunlit areas of the asteroid can be seen.

Based on the sampled system state, the following measurements are generated for the pose estimation algorithm to use:

- Asteroid rotation axis, tilted randomly between 0° and 10°
- Asteroid zero longitude right ascension at epoch, Gaussian noise, $\sigma = 5^\circ$
- Time instant determining asteroid orientation and Sun–asteroid position, noiseless measurement is used as the effect of any reasonable noise would be negligible compared to other sources of noise
- Spacecraft orientation, each Euler angle with Gaussian noise, $\sigma = 1^\circ$
- A synthetic NavCam image rendered using a detailed shape model with 1.9M vertices⁽⁴⁹⁾, a Hapke reflectance model⁽⁴¹⁾ and self-shadowing (see Fig. 3). Foreground pixel noise (from a normal distribution) and a starry background (from a Pareto distribution) are generated to closely match real Rosetta NavCam images.

The asteroid and spacecraft orientation noises are selected to be as high as possible—while still seeming reasonable to the authors—to better measure the robustness of the different feature extraction algorithms. For comparison, the orientation error of a typical star tracker is less than 20 arcsec (1-sigma)⁽⁵⁰⁾ and the uncertainty of the asteroid rotation axis longitude after extensive lightcurve observations approach 7° (1-sigma)⁽⁵¹⁾. On-board asteroid rotation modeling is expected to be much more accurate than this.

The reference image rendered on-board by the pose estimation algorithm is based on a less detailed shape model and the lunar–Lambert reflectance model. To investigate the effect of the vertex count on the algorithm performance, we tested each feature extraction algorithm with on-board shape models having 17,000, 4,000 and 1,000 vertices (see Fig. 3). The results are summarized in Table 2 and Fig. 4.

The failure rates in Table 2 might seem high, but the situation is not so severe, because most of the failures are due to a very difficult spacecraft position where the phase angle is high, the initial orientation error is high, the asteroid is very far, or so near that only a part is visible. If an algorithm fails to give a pose estimate for a certain image, this failure does not contribute towards the error measures. For example, ORB has the highest failure rate of 28.5% when using the 4,000 vertex shape model but then has lower errors for distance and orientation than AKAZE, which has the lowest failure rate of 16.6%. What likely occurs is that ORB fails on difficult samples and thus avoids barely successful, inaccurate pose solutions from worsening its error statistics. Because of this, the failure rate takes precedence when comparing the different algorithms. Only when it is similar between algorithms, can we compare the error percentiles fairly. Based on this, it seems that SIFT performs best when a reasonably detailed shape model is used, as both its failure rate and position related error percentiles are better than the others. However, when a less detailed shape model is used, it appears that AKAZE is the best performer. For the crudest shape model, AKAZE has a failure rate of 21.4% compared to the next best option, SURF with a 36.6% failure rate. For the middling shape model with 4,000 vertices, the comparison between AKAZE and SIFT is more ambiguous than with the extremes. The AKAZE failure rate is 16.6% versus SIFT 20.0% while the error percentiles favor SIFT.

To investigate the operational space of the algorithms, we classified each Monte Carlo sample as a success or a failure. A sample was considered a success if the resulting orientation

Table 2
Monte Carlo simulation results with synthetic NavCam images

Method	Vertices ^a	Time ms ^b	Fail %	Dist. err (m/km)		Lat. err (m/km)		Orient. err (°)	
				p50 ^c	p84.1 ^c	p50	p84.1	p50	p84.1
ORB	17k	158	26.4	7.30	34.2	0.64	1.80	1.46	5.25
AKAZE	17k	353	16.2	8.24	32.2	0.65	1.62	1.72	5.00
SURF	17k	376	19.8	6.85	20.8	0.56	1.29	1.76	4.41
SIFT	17k	387	15.6	6.84	22.5	0.55	1.34	1.67	4.62
ORB	4k	171	28.5	8.22	38.3	0.65	1.80	1.60	5.59
AKAZE	4k	350	16.6	8.90	31.6	0.65	1.60	1.84	5.05
SURF	4k	385	22.6	7.97	24.1	0.56	1.31	1.91	4.71
SIFT	4k	401	20.0	7.70	23.9	0.55	1.33	1.86	4.84
ORB	1k	192	38.6	11.4	52.3	0.71	2.01	2.28	7.35
AKAZE	1k	360	21.4	12.1	39.8	0.70	1.81	2.36	5.96
SURF	1k	418	36.6	10.9	30.5	0.61	1.42	2.56	5.60
SIFT	1k	456	39.3	11.3	34.8	0.61	1.45	2.68	6.34

The cells with gray background correspond to the best values across methods that use the same shape model. Darker gray on Fail % column reflects the high importance of it.

^a Shape model vertex count used for on-board rendering

^b Average algorithm runtime on an Intel i7 7700HQ 2.8GHz laptop – useful only for rough comparison between methods

^c p50 is the median and p84.1 refers to the 84.1th percentile of the error, which corresponds to one standard deviation away from the mean for normally distributed data.

error was below 7°. We represented the failures and successes with respect to the different situational parameters we identified before, namely phase angle, initial orientation error, asteroid distance and an estimated percentage of the asteroid in the FoV (see Fig. 4). To make the plots clearer, we excluded situations that were deemed “not easy” on a situational dimension that was not part of that particular plot, e.g. if plotting against distance and initial orientation error, we would exclude all samples that had a high phase angle or were only partially in the FoV. The range for inclusion for distance was 50–250 km, for phase angle 20–100°, for initial orientation error 0–10°, and for in-camera-view 80–100%. We also selected for further analysis only the data from using the 4,000 vertex shape model to further simplify and constrain the number of graphs.

Fig. 4 shows that SURF and SIFT degrade less than AKAZE and ORB when the distance increases. In particular, ORB practically stops working if the distance is more than 300 km while the others continue to function, provided that the situation otherwise is advantageous. ORB and AKAZE handle slightly better high phase angles than SURF and SIFT. ORB and AKAZE also seem to tolerate a high initial orientation error better. AKAZE seems most tolerant of situations when only a small part of the asteroid is in the FoV.

To understand better how the various error measures behave in different situations, we plotted the failure rate and the three different error measures as a function of distance, phase angle and initial orientation error, skipping the in-camera-view percent as relatively uninteresting

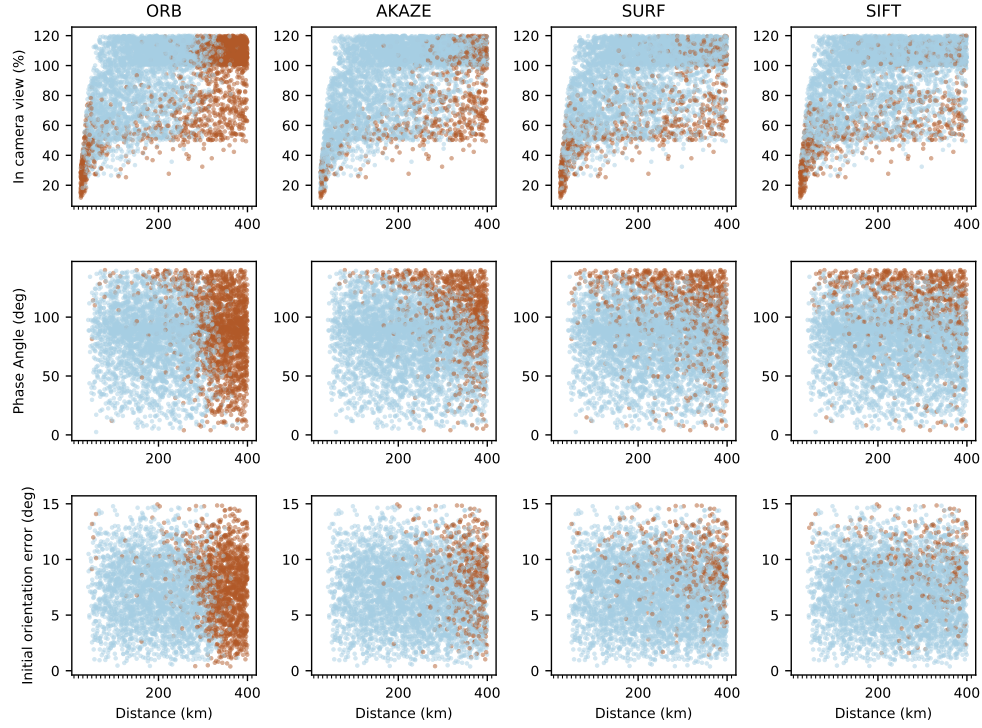


Figure 4. Successes (light blue) and failures (brown) of Monte Carlo samples with different feature extraction algorithms. For better sample separation when in-camera-view is exactly 100%, we add a 0–20% random component if the parameter value is 100%, so that values between 100–120% map to 100%.

(see Fig. 5). The plotted failure rate is calculated over a Gaussian sliding window with a standard deviation of one-tenth of the x-axis range. The error curves are plotted by taking the median on a sliding window with a width of one-tenth of the x-axis range. Here we also included only the “easy” situations, except for the chosen situational dimension on the x-axis. The same ranges for “easy” were used as for Fig. 4.

For Fig. 5, the same observations hold as was already noticed from the success-failure scatter plot. Looking at the failure rates, AKAZE seems to be most robust for most situations. It has the lowest failure rates with respect to phase angles, initial orientation errors, and distances below 250 km. However for distances above 250 km, SIFT seems to work best. In advantageous situations the failure rates of different algorithms are very similar, so the error medians should be comparable. In these regions, ORB seems to give the most accurate results.

We also tried the different algorithms with Rosetta NavCam images that are freely available from ESA. We used six sets of images⁽⁶⁾: Prelanding - MTP006, Prelanding - MTP007, Comet Escort 2 - MTP017, Comet Escort 4 - MTP024, Comet Extension 1 - MTP025, and Comet Extension 1 - MTP026. Some of the images were excluded from the sets if they were e.g. almost completely black or extremely overexposed. Of a total of 3504 images, 3296 were used. All the images have metadata stored in a separate file, which has the asteroid–spacecraft system state as estimated by the Rosetta system. We use that state as the ground truth. Noise with the same distributions as before is added to this state before passing it to the relative

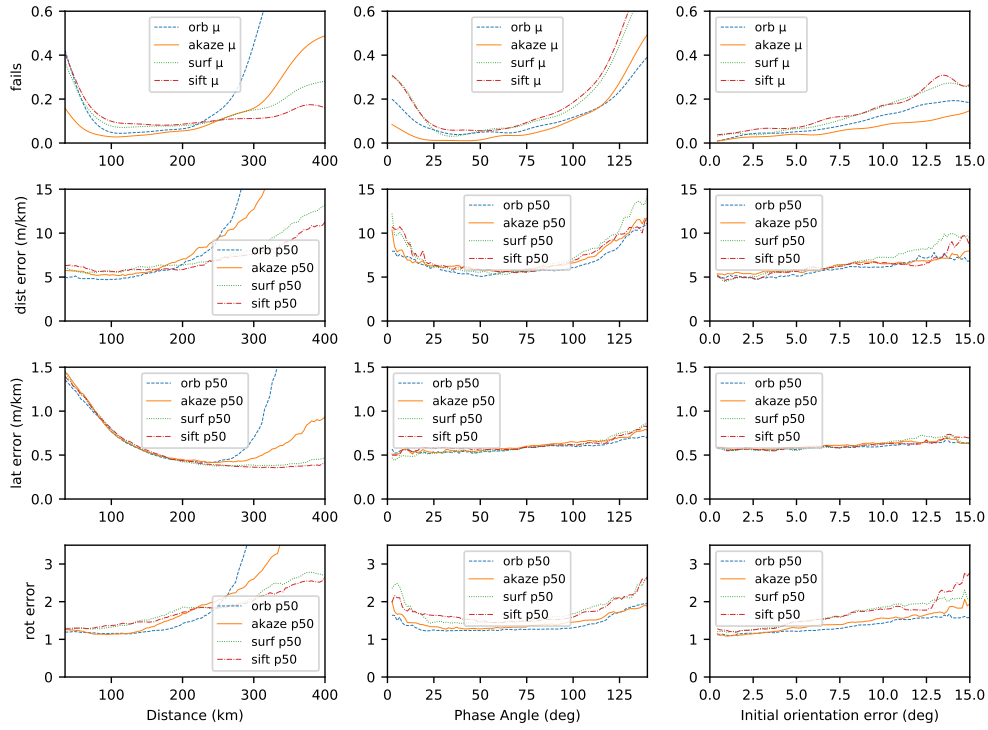


Figure 5. Failure rate and errors vs different situations.

pose estimation algorithms. We give the same table and figures for the results from the real NavCam images as we did for the results based on synthetic images (Table 3, Fig. 6, Fig. 7).

Surprisingly, Table 3 shows that all other feature extraction methods perform quite poorly—as shown by the failure rate—compared to AKAZE, which has a comparable failure rate for the Rosetta images as it did for the synthetic images. This could be due to the dataset having more situations where AKAZE is good at. Such situations would be e.g. when the spacecraft is very close to the asteroid.

By looking at Fig. 6, we can see that AKAZE performs even better compared to the others when evaluated using real images. Especially, AKAZE shines when 67P/Churyumov–Gerasimenko is near and only partly visible or when the initial orientation error is high.

In Fig. 7 we present the failure rate and different errors when using the real images in the same way as we did for the results based on the synthetic ones. The samples covered only poorly the distances between 110–160 km and farther than 230 km. These areas in the plots (with 10 km margins, shown in pink) should be considered highly inaccurate. Also, plots with distance or phase angle on the x-axis show discontinuities around a region where there were many samples centered on 200 km distance and 90° phase angle.

As can be seen in Fig. 7, AKAZE performs best by a large margin. The phase angle does not seem to have a significant impact, at least in the range 20–140°. The higher errors seen between 80–100° can be explained by the sample distribution as all distant samples were in that phase angle range. The initial orientation error seems to have a much higher impact on

Table 3
Full results with real NavCam images from Rosetta

Method ^a	Vertices	Time ms	Fail %	Dist. err (m/km)		Lat. err (m/km)		Orient. err (°)	
				p50	p84.1	p50	p84.1	p50	p84.1
ORB	17k	307	43.9	6.43	18.1	0.34	1.09	1.23	3.52
AKAZE	17k	449	11.3	6.29	16.0	0.35	0.98	0.91	2.46
SURF	17k	646	37.6	7.18	18.0	0.37	1.00	1.16	3.20
SIFT	17k	616	36.2	6.27	15.8	0.36	0.95	1.14	3.02
ORB	4k	319	50.6	6.92	20.7	0.36	1.06	1.46	3.86
AKAZE	4k	477	15.9	6.89	19.2	0.38	1.22	1.11	2.9
SURF	4k	689	50.6	6.96	19.1	0.38	0.85	1.33	3.28
SIFT	4k	675	53.1	6.65	17.9	0.36	0.85	1.34	3.37
ORB	1k	382	67.4	10.8	32.4	0.45	1.19	2.21	5.35
AKAZE	1k	561	34.3	8.51	26.6	0.42	1.33	1.64	4.21
SURF	1k	746	68.3	8.12	22.8	0.40	0.88	1.84	4.17
SIFT	1k	744	72.2	8.23	22.1	0.39	0.87	1.93	4.52

^a Row and column meanings are the same as in Table 2.

performance than what was expected based on the experiment with synthetic images. For AKAZE, a 10° initial orientation error seems to correspond roughly to a 7% failure rate.

In order to report some useful single values for the expected errors for the AKAZE-based algorithm, we define—based on Fig. 6 and Fig. 7—an operational zone for it. A sample belongs to the zone if:

- Distance to target is less than 250 km
- At least 50% of the target is in the FoV
- Phase angle is less than 140°
- Initial orientation error is less than 10°

This limited our data set to 1,253 samples. The results for AKAZE with different shape model vertex counts are presented in Table 4.

Table 4
Operational zone results using AKAZE and tested with Rosetta images

Vert.	Fail %	Dist err (m/km)				Lat err (m/km)				Orient err (°)			
		μ	$\mu + \sigma$	p50	p84.1	μ	$\mu + \sigma$	p50	p84.1	μ	$\mu + \sigma$	p50	p84.1
17k	0.40	6.02	11.1	4.64	10.4	0.30	0.48	0.26	0.42	1.04	2.06	0.77	1.85
4k	0.64	6.33	12.1	4.47	11.2	0.32	0.53	0.27	0.47	1.13	2.20	0.86	2.02
1k	4.47	8.71	17.3	6.03	15.5	0.41	0.74	0.32	0.63	1.62	3.65	1.30	2.99

By comparing the mean and standard deviation shown in Table 4 with the corresponding

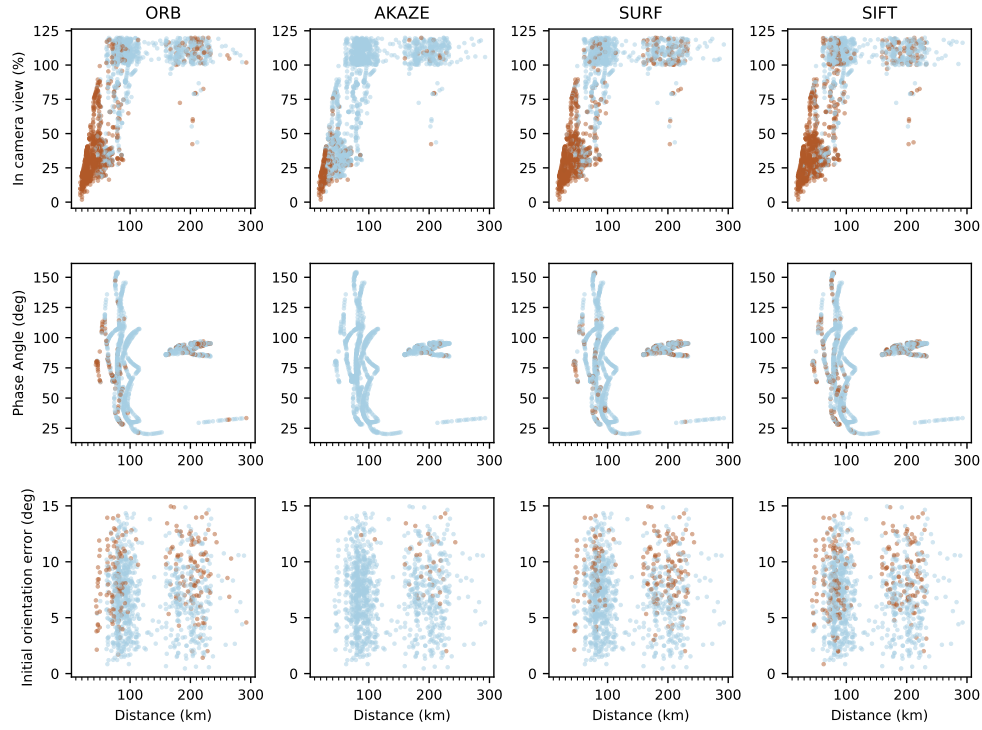


Figure 6. Successes (light blue) and failures (brown) in different situations based on real Rosetta NavCam images. This figure is produced the same way as 4, please refer to the image caption there.

percentiles for a normal distribution, it is possible to estimate how close the error distribution is to it. Finally, we give some AKAZE example solutions in Fig. 8 and Fig. 9 to illustrate the algorithm performance when presented with some non-ideal NavCam images. On the left, there is a Rosetta NavCam image next to a reference image rendered on-board from a shape model with 4,000 vertices at a distance of 70 km and with an inaccurate initial orientation. On the right, there is the same pair of images joined by lines representing those matches that are retained (inlier matches) after RANSAC has eliminated all geometrically inconsistent ones. Red circles show all features points that were not matched successfully. Images are not normalized in any way as AKAZE does not require it.

4.0 Discussion

In the operational zone defined based on our experiments, a conservatively sized shape model with 4,000 vertices gives an average distance error of 6.3 m/km and a lateral error of 0.32 m/km. The average orientation error was 1.1. The accuracy can be improved slightly by using a more detailed shape model. Also, it is worth bearing in mind that this is not the final navigation accuracy as the estimates given by our algorithm should be fed into a navigation filter, which then outputs the final pose estimate.

Interesting and immediate directions for future work would be to i) assess how a filtering

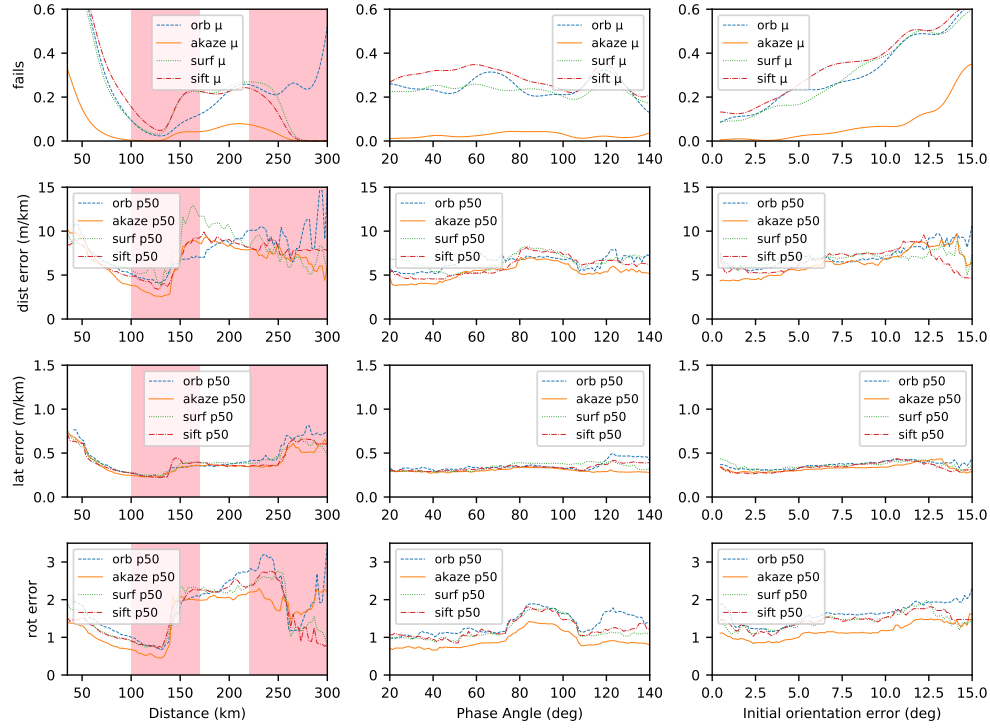


Figure 7. Failure rate and errors vs different situations based on real Rosetta NavCam images. Note how AKAZE (orange lines) has the lowest failure rate and lowest errors in almost all regions.

scheme affects the accuracy and robustness of the system, ii) what changes to the algorithm are needed so that it works reliably in a double-asteroid system, iii) try to implement the algorithm in suitable hardware and iv) try to improve synthetic NavCam image generation by using a texture map on the shape model or generating high-frequency features into it.

To increase accuracy and operational range, an integrated visual odometry scheme could be formulated, where new opportunistic features could be added on-the-fly to a feature database and later queried based on the initial pose estimate. These features could be mixed in with the SPL features detected from the synthetic reference image.

Another supplement could be to implement an approach based on deep learning to estimate the pose of the spacecraft relative to the asteroid without any initial pose estimate. This could be used in cases where we have lost the ability to generate a good enough initial pose estimate, e.g. due to an anomaly affecting the use of the star tracker or on-board clock.

5.0 Conclusion

We have shown that synthetic image generation together with local photometric features can be used for medium-range absolute navigation near an asteroid. The algorithm does not have limitations regarding the target size, axial tilt, and the existence of craters. We tested four feature-detection and -description algorithms using synthetic and real Rosetta NavCam im-

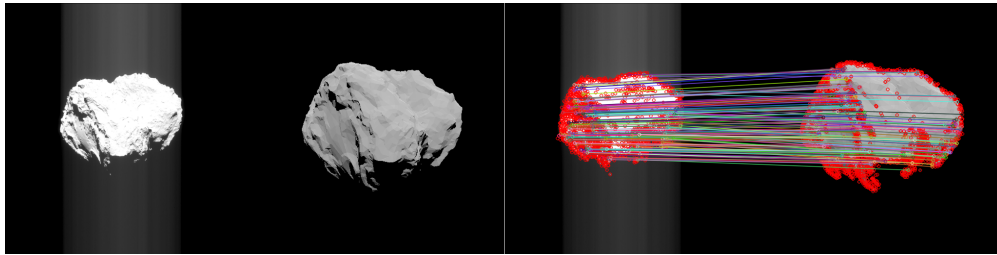


Figure 8. AKAZE features from overexposed NavCam image matched with features from onboard rendered 4,000 vertex shape model.

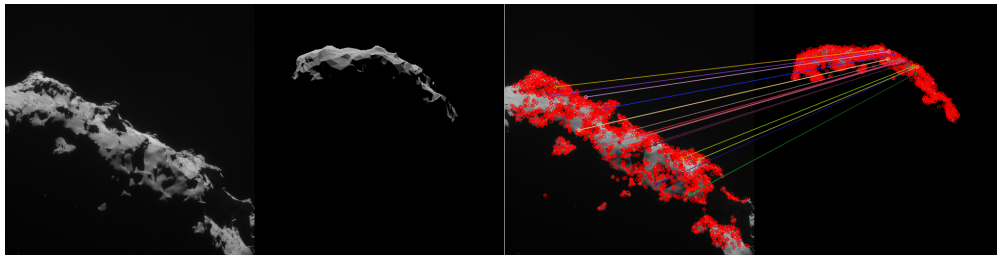


Figure 9. Illustration of feature matches when the comet is only partially visible. The algorithm is close to failing as there are so few inlier matches.

ages and compared the resulting performance under different environmental conditions. Related studies have typically included SIFT, SURF, and ORB but overlooked AKAZE, which proved to be most suitable to our application by a high margin. Using the results, we defined an operational zone for our AKAZE-based algorithm and reported its performance when using shape models of three different levels of detail. The analysis provides important information when considering vision-based navigation systems for future space missions.

Acknowledgment

The authors would like to thank Dr. Juho Kannala, Professor of Computer Vision at Aalto University for guidance in the early stages of this work, and Markku Alho, a doctoral candidate of space science and technology at Aalto University for assistance with the different shape models of the comet 67P/Churyumov–Gerasimenko.

REFERENCES

1. ESA. Cubesats joining hera mission to asteroid system. [Online]. Available: http://www.esa.int/Our_Activities/Space_Engineering_Technology/Hera/CubeSats_joining_Hera_mission_to_asteroid_system (Accessed on 2019-02-12).
2. T. Kohout, A. Näsilä, T. Tikka, M. Granvik, A. Kestilä, A. Penttilä, J. Kuhno, K. Muinonen, K. Viherkanto, and E. Kallio, Feasibility of asteroid exploration using CubeSats—ASPECT case study, *Advances in Space Research*, vol. 62, no. 8, pp. 2239–2244, 2018.
3. P. Michel, M. Kueppers, H. Sierks, I. Carnelli, A. F. Cheng, K. Mellab, M. Granvik, A. Kestilä, T. Kohout, K. Muinonen *et al.*, European component of the AIDA mission to a binary asteroid: Characterization and interpretation of the impact of the DART mission, *Advances in Space Research*, vol. 62, no. 8, pp. 2261–2272, 2018.
4. S. Naidu, L. Benner, M. Brozovic, S. Ostro, M. Nolan, J. Margot, J. Giorgini, C. Magri, P. Pravec, P. Scheirich *et al.*, Observations and Characterization of Binary Near-Earth Asteroid 65803 Didymos, the Target of the AIDA Mission, in *AGU Fall Meeting Abstracts*, 2016.
5. ESA. Esa planetary science archive. [Online]. Available: https://imagearchives.esac.esa.int/index.php?page/navcam_3d_models (Accessed on 2019-01-30).
6. ESA. Esa archive image browser. [Online]. Available: <http://imagearchives.esac.esa.int/index.php?category/63> (Accessed on 2019-01-30).
7. W. Owen Jr, T. Wang, A. Harch, M. Bell, and C. Peterson, NEAR optical navigation at Eros, Jet Propulsion Laboratory, California Institute of Technology, Tech. Rep., 2001.
8. R. P. de Santayana and M. Lauer, Optical measurements for rosetta navigation near the comet, in *Proceedings of the 25th International Symposium on Space Flight Dynamics (ISSFD)*, Munich, 2015.
9. N. Mastrodemos, B. Rush, D. Vaughan, and B. Owen, Optical navigation for Dawn at Vesta, in *21st AAS/AIAA Space Flight Mechanics Meetings*. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2011. [Online]. Available: <https://trs.jpl.nasa.gov/handle/2014/41960>
10. 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, in *2017 IEEE Aerospace Conference*. IEEE, 2017, pp. 1–12.
11. T. Miso, T. Hashimoto, and K. Ninomiya, Optical guidance for autonomous landing of spacecraft, *IEEE Transactions on aerospace and electronic systems*, vol. 35, no. 2, pp. 459–473, 1999.
12. T. Hashimoto, T. Kubota, J. Kawaguchi, M. Uo, K. Shirakawa, T. Kominato, and H. Morita, Vision-based guidance, navigation, and control of Hayabusa spacecraft—Lessons learned from real operation, *IFAC Proceedings Volumes*, vol. 43, no. 15, pp. 259–264, 2010.
13. D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
14. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, Speeded-up robust features (SURF), *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

15. S. Leutenegger, M. Chli, and R. Siegwart, BRISK: Binary robust invariant scalable keypoints, in *2011 IEEE international conference on computer vision (ICCV)*. IEEE, 2011, pp. 2548–2555.
16. E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, ORB: An efficient alternative to SIFT or SURF., in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.
17. N. Takeishi, A. Tanimoto, T. Yairi, Y. Tsuda, F. Terui, N. Ogawa, and Y. Mimasu, Evaluation of interest-region detectors and descriptors for automatic landmark tracking on asteroids, *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 58, no. 1, pp. 45–53, 2015.
18. S. Theil, N. Ammann, F. Andert, T. Franz, H. Krüger, H. Lehner, M. Lingenauber, D. Lüdtkke, B. Maass, C. Paproth *et al.*, ATON (Autonomous Terrain-based Optical Navigation) for exploration missions: recent flight test results, *CEAS Space Journal*, vol. 10, no. 3, pp. 325–341, 2018.
19. B. D. Lucas, T. Kanade *et al.*, An iterative image registration technique with an application to stereo vision, in *Darpa Image Understanding Workshop*. Vancouver, British Columbia, April 1981, pp. 121–130.
20. J. Shi and C. Tomasi, Good features to track, Cornell University, Tech. Rep., 1993.
21. H. Strasdat, J. Montiel, and A. J. Davison, Real-time monocular SLAM: Why filter?, in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2657–2664.
22. G. Younes, D. Asmar, and E. Shammass, A survey on non-filter-based monocular visual SLAM systems, *arXiv preprint arXiv:1607.00470*, vol. 413, p. 414, 2016.
23. H. Kaufmann, M. Lingenauber, T. Bodenmueller, and M. Suppa, Shadow-based matching for precise and robust absolute self-localization during lunar landings, in *2015 IEEE Aerospace conference*. IEEE, 2015, pp. 1–13.
24. A. Ansar and Y. Cheng, Vision technologies for small body proximity operations, in *Proceedings of the International Symposium on Artificial Intelligence and Robotics & Automation in Space*, 2010.
25. Y. Wang, W. Zhang, and P. An, A survey of simultaneous localization and mapping on unstructured lunar complex environment, in *AIP Conference Proceedings*, vol. 1890, no. 1. AIP Publishing, 2017, p. 030010.
26. M. Ciarambino, P. Lunghi, L. Losi, and M. Lavagna, Development, validation and test of optical based algorithms for autonomous planetary landing, in *Proc. ICATT 2016*, 2016.
27. J. Delaune, G. Le Besnerais, T. Voirin, J.-L. Farges, and C. Bourdarias, Visual-inertial navigation for pinpoint planetary landing using scale-based landmark matching, *Robotics and Autonomous Systems*, vol. 78, pp. 63–82, 2016.
28. P. F. Alcantarilla and T. Solutions, Fast explicit diffusion for accelerated features in nonlinear scale spaces, *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.
29. O. Foundation. Open source computer vision library. [Online]. Available: <https://opencv.org/about/> (Accessed on 2019-05-20).
30. H.-J. Chien, C.-C. Chuang, C.-Y. Chen, and R. Klette, When to use what feature? SIFT,

- SURF, ORB, or A-KAZE features for monocular visual odometry, in *2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE, 2016, pp. 1–6.
31. ESA. Esa archive image browser. [Online]. Available: <http://imagearchives.esac.esa.int/index.php?/category/56> (Accessed on 2019-01-30).
 32. R. R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of astrodynamics*. Courier Corporation, 1971.
 33. P. Pravec, A. Harris, D. Vokrouhlický, B. Warner, P. Kušnirák, K. Hornoch, D. Pray, D. Higgins, J. Oey, A. Galád *et al.*, Spin rate distribution of small asteroids, *Icarus*, vol. 197, no. 2, pp. 497–504, 2008.
 34. N. Rowell, M. Dunstan, S. Parkes, J. Gil-Fernández, I. Huertas, and S. Salehi, Autonomous visual recognition of known surface landmarks for optical navigation around asteroids, *The Aeronautical Journal*, vol. 119, no. 1220, pp. 1193–1222, 2015.
 35. A. Pellacani, F. Cabral, A. Alcalde, P. Kicman, J. Lisowski, I. Gerth, and B. Burmann, Semi-autonomous attitude guidance using relative navigation based on line of sight measurements–AIM scenario, *Acta Astronautica*, vol. 152, pp. 496–508, 2018.
 36. A. W. Harris and P. Pravec, Rotational properties of asteroids, comets and TNOs, *Proceedings of the International Astronomical Union*, vol. 1, no. S229, pp. 439–447, 2005.
 37. D. Vokrouhlický, S. Breiter, D. Nesvorný, and W. Bottke, Generalized YORP evolution: Onset of tumbling and new asymptotic states, *Icarus*, vol. 191, no. 2, pp. 636–650, 2007.
 38. L. Zheng, Y. Yang, and Q. Tian, SIFT meets CNN: A decade survey of instance retrieval, *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1224–1244, 2018.
 39. A. S. McEwen, Photometric functions for photoclinometry and other applications, *Icarus*, vol. 92, no. 2, pp. 298–311, 1991.
 40. A. Grumpe, C. Schröer, S. Kauffmann, T. Fricke, C. Wöhler, and U. Mall, Refinement of stereo image analysis using photometric shape recovery as an alternative to bundle adjustment., *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 41, 2016.
 41. B. Hapke, *Theory of reflectance and emittance spectroscopy*. Cambridge university press, 2012.
 42. S. Fornasier, P. Hasselmann, M. Barucci, C. Feller, S. Besse, C. Leyrat, L. Lara, P. J. Gutierrez, N. Ockay, C. Tubiana *et al.*, Spectrophotometric properties of the nucleus of comet 67P/Churyumov-Gerasimenko from the OSIRIS instrument onboard the ROSETTA spacecraft, *Astronomy & Astrophysics*, vol. 583, p. A30, 2015.
 43. L. Williams, Casting curved shadows on curved surfaces, in *Seminal graphics*. ACM, 1998, pp. 51–55.
 44. M. Muja and D. G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration., *VISAPP (I)*, vol. 2, no. 331-340, p. 2, 2009.
 45. M. A. Fischler and R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
 46. T. Ke and S. I. Roumeliotis, An efficient algebraic solution to the perspective-three-

- point problem, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7225–7233.
47. D. S. Bayard and P. B. Brugarolas, On-board vision-based spacecraft estimation algorithm for small body exploration, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 243–260, 2008.
 48. O. J. M. Knuuttila. Visual navigation algorithms and test framework. [Online]. Available: <https://github.com/oknuutti/visnav-py/tree/55bee608648f377b5f4bfd8c241d893d6497d1d7> (Accessed on 2019-05-15).
 49. M. Malmer. 67p/c-g shape model. [Online]. Available: <http://mattias.malmer.nu/67pc-g-shapemodel/> (Accessed on 2019-05-13).
 50. M. A. Fialho and D. Mortari, Theoretical Limits of Star Sensor Accuracy, *Sensors*, vol. 19, no. 24, p. 5355, 2019.
 51. E. Bowell, D. Oszkiewicz, L. Wasserman, K. Muinonen, A. Penttilä, and D. E. Trilling, Asteroid spin-axis longitudes from the Lowell Observatory database, *Meteoritics & Planetary Science*, vol. 49, no. 1, pp. 95–102, 2014.