
Autonomous visual recognition of known surface landmarks for optical navigation around asteroids

N. Rowell

nr@roe.ac.uk

M. N. Dunstan and S. M. Parkes

Space Technology Centre

University of Dundee

Dundee

UK

J. Gil-Fernández

GMV

Tres Cantos, Madrid

Spain

I. Huertas (Serco Nederland BV) and S. Salehi (Rhea System BV)

ESA/ESTEC

Noordwijk

The Netherlands

ABSTRACT

We present an autonomous visual landmark recognition and pose estimation algorithm designed for use in navigation of spacecraft around small asteroids. Landmarks are selected as generic points on the asteroid surface that produce strong Harris corners in an image under a wide range in viewing and illumination conditions; no particular type of morphological feature is required. The set of landmarks is triangulated to obtain a tightly fitting mesh representing an optimal low resolution model of the natural asteroid shape, which is used onboard to determine the visibility of each landmark and enables the algorithm to work with highly concave bodies. The shape model is also used to estimate the centre of brightness of the asteroid and eliminate large translation errors prior to the main landmark recognition stage. The algorithm works by refining an initial estimate of the spacecraft position and orientation. Tests with real and synthetic images show good performance under realistic noise conditions. Using simulated images, the median landmark recognition error

is 2m, and the error on the spacecraft position in the asteroid body frame is reduced from 45m to 21m at a range of 2km from the surface. With real images the translation error at 8km to the surface increases from 107m to 119m, due mainly to the larger range and lack of sensitivity to translations along the camera boresight. The median number of landmarks detected in the simulated and real images is 59 and 44 respectively. This algorithm was partly developed and tested during industrial studies for the European Space Agency's Marco Polo-R asteroid sample return mission.

NOMENCLATURE

x^{AST}	superscript denotes quantities in the body frame
x^{CAM}	superscript denotes quantities in the camera frame
x^{IM}	superscript denotes quantities in the image plane
T	denotes matrix transpose
*	denotes quaternion multiplication
K	bold indicates tensor quantities and matrices
<u>L</u>	underline indicates vector quantities

AMICA	Asteroid Multi-band Imaging Camera
BRDF	Bidirectional Reflectance Distribution Function
FOV	Field of View
GNC	Guidance, Navigation and Control
NEA	Near Earth Asteroid
PANGU	Planet and Asteroid Natural Scene Generation Utility
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events
SURF	Speeded Up Robust Features

1.0 INTRODUCTION

Small Near-Earth Asteroids (NEAs) are particularly interesting targets for space exploration for a number of reasons. There are roughly 2,000 known NEAs that are energetically easier to reach than the surface of the Moon⁽¹⁾ and the weak gravity and lack of atmosphere makes sample return missions far more feasible than for larger bodies. Asteroids represent the rubble left over from the formation of the rocky planets, and small asteroids in particular are thought to exist in a pristine state having undergone minimal processing since formation. As such, analysis of the composition and structure of their surface material can provide great insight into the conditions and processes at work in the early solar nebula, and retrieval of samples for analysis in Earth laboratories offers a much wider range of analysis techniques than is available by *in-situ* measurements. Often rich in rare metals, asteroids also represent a potentially vast reserve of natural resources.

However, the small size and weak gravity also pose significant challenges to the proximity navigation and landing⁽²⁾. The low mass results in an often highly irregular shape, producing a weak, irregular gravitational field with orbits strongly perturbed by the Sun third body effect and radiation pressure. In addition, the surface normal can have large deviations from the local gravity direction, making it difficult to determine which way is 'down' when close to the surface. The small size leads to stringent requirements on the landing precision, and the large distances from Earth render teleoperation infeasible over the landing timescale. The use of onboard navigation cameras, coupled with intelligent machine vision algorithms, is the crucial enabling technology for successful operation in this challenging domain.

1.1 Vision-based navigation

The development of machine vision algorithms for navigation purposes is a large and active area of research, and a wide variety of systems of varying complexity have been developed for terrestrial applications. Major examples include visual odometry for cars⁽³⁾, state estimation for unmanned aerial vehicles⁽⁴⁾ and localisation of indoor mobile robots⁽⁵⁾. However, the limited processing power available onboard spacecraft, combined with the need for highly robust yet autonomous algorithms, and the very different environment geometry and dynamics render many terrestrial approaches invalid in space. We avoid a thorough review of vision based guidance here, and focus mainly on the development of techniques for spacecraft navigation.

The classical method of vision based navigation for interplanetary spacecraft involves observing the target while at a large distance and the target is only a few pixels in extent. Measurement of the position of the target against the background stars gives a precise estimate of the direction towards the target in the inertial frame, a technique known as beacon-based optical navigation⁽⁶⁾. At closer distances, observations of the limb of the target body can be used in a similar way to find the approximate direction to the body centre of mass. The first use of optical navigation in the demanding area of planetary entry, descent and landing was the 2003 landing of the Mars Exploration Rovers. These used a downward-pointing camera to detect and track surface features between pairs of images taken during descent⁽⁷⁾. The displacement of features between consecutive frames was the only means to detect lateral motion of the landing module induced by winds, which was then cancelled by transverse pointing thrusters prior to touchdown. The optical detection and tracking of unknown surface features is a major branch of vision based navigation, and work is underway on the development of combined camera and dedicated image processing boards to support high-framerate feature tracking⁽⁸⁾. These algorithms are based ultimately on corner or blob detection filters that select distinctive but transient features by looking for maxima in the first or second spatial derivatives of the intensity. Common choices are the Harris corner detector⁽⁹⁾ and the speeded up robust features (SURF) algorithm⁽¹⁰⁾. Tracking features between frames is performed by correlation of intensity over small regions surrounding a feature, or more robustly by building a rotation and scale invariant description of the local intensity distribution in the case of SURF. The displacement of features in the image plane encodes the change in orientation and position of the camera from one image to the next, which are ultimately recovered from the feature tracks by analysis of the epipolar geometry and essential matrix⁽¹¹⁾. By tracking transient optical features over many frames the trajectory of the camera can be calculated relative to the starting point, a mode known as relative navigation. A key shortcoming of relative navigation is the gradual accumulation of error: the estimated trajectory takes a random walk away from the true trajectory as the navigation filter integrates noisy sensor measurements. The solution to this is to periodically reset the filter by directly solving for the instantaneous position and orientation of the camera. This is performed by observing landmarks of known 3D position in the target body frame. Simultaneous observation of four or more landmarks provides a unique solution for the camera position and orientation, or 'pose', using photogrammetry techniques (e.g. Ref. 12).

The selection, definition and autonomous recognition of known landmarks is a much more challenging task than that of tracking transient optical features. This is because the viewing conditions, including the perspective, illumination, rotation and scale of features, may change arbitrarily between when the landmarks are first selected and when they are observed again later. The descriptor used to identify each landmark thus needs to be invariant to the viewing conditions, or at least a known function of them in order that it can be updated for the current imagery, should the viewing conditions be known. This is in stark contrast to tracking transient unknown features, which is only concerned with the displacement from one frame to the next so is subject

to much smaller changes in the viewing conditions, and which only needs to identify landmarks uniquely within their immediate surroundings so can tolerate a simpler description of the landmark.

Several variants of known landmark recognition have been used in spacecraft navigation over the years. The earliest implementation was during the NEAR-Shoemaker mission, where craters on the surface of asteroid Eros were used as landmarks. Orbital corrections were computed on-ground by matching observed craters against a database of craters compiled during the initial mapping phase of the mission⁽¹³⁾. Craters are good landmarks because they are prominent features that can be described by a simple geometric model and matched on the basis of both their size and orientation.

In recent missions, a more sophisticated technique has been developed that uses small 3D digital models of arbitrary patches on the target body as landmarks: these 'L-maps' are rendered into the navigation imagery and aligned with the real image by correlation of intensity. This approach was developed during the Hayabusa mission⁽¹⁴⁾ and has been used recently during the Dawn mission encounter with asteroid Vesta⁽¹⁵⁾. The benefit of this approach is that it does not rely on a particular type of morphological feature, such as craters, which may not be present. It is also an intrinsic part of the target characterisation and mapping process, which uses a technique called Stereophotoclinometry to estimate the global shape model for the target⁽¹⁶⁾.

1.2 Marco Polo-R scenario

The known landmark recognition algorithm presented in this paper was initially developed during an industrial assessment study for the Marco Polo-R mission. Marco Polo-R was a mission concept to land a spacecraft on a small NEA, collect a sample of material from the surface and return it to Earth⁽¹⁷⁾. It was a candidate for the M3 launch opportunity within the Cosmic Vision programme⁽¹⁸⁾ although was not eventually selected. The algorithm as described here is a refined version based on the outcome of tests performed during the study, which ranged from software-only tests using fully simulated imagery, to high fidelity robotic simulations using 3D models, scaled dynamics and real optical sensors.

The vision segment of the navigation system consists of a combination of feature tracking and landmark recognition. The feature tracking algorithm employs the Harris corner detector to select points and intensity correlation to track them through later frames, and is described in Ref 19. The landmark recognition algorithm was used at the start of the descent and landing phase to initialise the navigation filter. It had to satisfy a range of explicit and implicit constraints imposed by the mission scenario and the space environment, which had a major impact on the design of the system. These constraints are as follows:

No colour Navigation cameras built for spacecraft take monochrome images, so landmark recognition cannot use colour to discriminate between landmarks. Note that even if colour images were used, asteroids tend to have very small colour variation anyway so this would be of limited use.

Highly concave body Small asteroids are not large enough for gravity to pull them into a spherical shape, and they can exhibit very concave forms with the asteroid Itokawa being a classic example. This is important because landmarks may be wholly or partially obscured by intervening terrain, and determining the visibility of landmarks on a concave body is a much more difficult task than for roughly-spherical convex bodies where a landmark is only obscured if it lies over the horizon.

Self-similar surface with no craters Unlike nearly all larger bodies in the solar system, small asteroids do not necessarily have a cratered surface. In fact it is impossible to guarantee the existence of any particular type of geological feature as not much is known about the morphology of these

objects. It is likely that much of the surface is highly self-similar and lacks any type of regular feature for which a simple geometric description would suffice. In particular, the algorithm should not be based on crater recognition.

1.2.1 Prior information

The mission scenario and the design of the rest of the guidance, navigation and control (GNC) system mean that several pieces of prior information are available to aid in landmark recognition. During an initial phase of global characterisation, a digital 3D shape model of the asteroid is computed from which a database of landmarks can be extracted on ground and uploaded to the spacecraft. The shape model is assumed to be computed by applying stereophotoclinometry techniques to images of the target, a process which can produce a shape model with a resolution matching that of the highest resolution images. A major component of the landmark recognition algorithm involves the landmark definition and the selection and construction of the database from the digital shape model.

An estimate of the spacecraft position and orientation in the asteroid body frame at the time the landmark recognition algorithm is triggered is also available. The landmark recognition algorithm uses this in conjunction with the real time navigation imagery and the landmarks database in order to identify landmarks and refine the pose estimate, which is then used to reset the navigation filter at the start of the descent and landing. The accuracy and error distribution of the initial pose estimate is of prime interest: the orientation of the spacecraft in the asteroid body frame is relatively well known, as the orientation in the inertial frame is known accurately from star tracker measurements and the inertial-frame orientation of the asteroid is reasonably well known from measurement of the rotational pole and rate during earlier characterisation phases. We have assumed a conservative orientation error distribution that is characterised by an angular error distributed uniformly in the range $[-2:2]$ degrees, about an axis distributed isotropically on the unit sphere. The position error is somewhat less certain, due to the relative instability of orbits about small, irregular bodies. The position in the inertial frame is estimated on ground using Doppler measurements or delta-DOR⁽²⁰⁾, which provides an accuracy of around 50m at the nominal range to the target from Earth. This is fused with the onboard position estimate, which due to the use of a laser altimeter results in an estimate of the spacecraft position which is more certain along the line of sight to the asteroid than in the transverse directions. We have therefore adopted a model for the spacecraft position error that has components distributed uniformly in the range $[-50:50]$ metres in directions perpendicular to the direction towards the asteroid, and in the range $[-5:5]$ metres along the line of sight to the asteroid. Note that these are expressed in the inertial frame: the position error in the asteroid frame is inflated due to the uncertainty on the orientation, although the component along the line of sight remains unchanged.

The range at which the landmark recognition algorithm will be used is known in advance, which is useful as this defines the overall scale of asteroid surface features in the images and can be used to tailor the landmark database generation. Finally, the direction towards the Sun is known accurately onboard, which will turn out to be important later on.

1.2.2 Landmark definition, extraction and recognition

Landmarks are defined as sites on the asteroid surface that consistently produce strong, stable Harris corners⁽⁹⁾ when imaged over wide changes in the viewing and illumination angles. The shape model of the asteroid is used to identify and select these features by re-imaging it under varying conditions. Such sites include prominent boulders on smooth plains, pits containing dark shadows, outcrops of rock, peaks and ridges. Selecting landmarks in this manner places no requirements

on the types of morphological features present on the surface. The minimum required scale for features on the asteroid surface is set by the resolution of the camera and not by the algorithm itself; as long as features are present at scales that can be resolved by the camera, then landmarks can be extracted. In the tests presented in Sections 4.2 and 4.3, the pixel scale is roughly one metre on the asteroid surface, resulting in a minimum feature scale of several metres (allowing for an imperfect optical transfer function).

Landmarks are characterised only according to their position in the asteroid body-frame and associated uncertainty. The landmark identification algorithm relies on prior knowledge of the spacecraft position and orientation, which is used to project the landmark positions into the current navigation image. The landmark positions are then compared with the positions of strong Harris corners extracted from the current image, and positive identifications are made on the basis of proximity alone. The set of identified landmarks is then used to improve the knowledge of the spacecraft pose by updating the parameters to bring the projected landmarks into alignment with the observed Harris corners.

The full set of 3D landmarks is also triangulated to obtain a low resolution mesh that roughly fits the surface of the asteroid. This serves two purposes. First, this allows landmarks lying on the far side of the asteroid and those obscured by intervening terrain to be purged from the identification step. This is important for highly concave bodies as there is no straightforward way to determine the set of visible landmarks, unlike for a roughly spherical body. Second, the low-resolution mesh can be used in conjunction with the known sun direction to render a rough image of the asteroid for comparison with the observed image. By aligning the simulated and real images, large translation errors in the image plane can be greatly reduced prior to the landmark identification step. This is important as the proximity-based landmark identification method is highly sensitive to these misalignments, which can cause false-positive identifications.

Note that the Harris corner detector has been selected over a range of other point-feature detectors^(21,22), with the well-established and characterised SURF algorithm^(10,23) being the most obvious alternative. SURF is a sophisticated feature matching algorithm: it builds a rotation and scale invariant description of the features it detects, so that they can be matched in a second image under arbitrary rotation and scale changes. However, it is sensitive to variations in the illumination angle and rotations not around the camera boresight, both of which occur in the present application, so it could not be used for example to match landmarks in the database to features in the images. In addition, the scale invariance is not required as the range to the asteroid is known in advance. However, the feature detector element of SURF could be used in place of the Harris detector. This uses the second spatial derivatives of the image intensity and thus finds maxima and minima, or 'blobs', of dark and light features. In contrast Harris uses the first spatial derivatives to find features where the intensity changes sharply, which have the appearance of corners. The two are functionally equivalent in the present context, and the Harris detector was selected due to the fact that the optical feature tracking element of the navigation system also used Harris corners, so this enabled reuse of the onboard resources; in the complete system, features currently being tracked were matched against the landmarks database.

2.0 LANDMARK DATABASE CONSTRUCTION

This section explains how the asteroid shape model is processed to extract a database of visual surface landmarks. The example used is that of asteroid 25143 Itokawa, for which a high resolution shape model is available from the Hayabusa Science Archive⁽²⁴⁾. The model has a resolution of ~60cm per vertex, with about 1.57m vertices comprising the full set. The model itself is provided

in implicitly connected quadrilateral format⁽¹⁶⁾, an efficient format that is used by major space agencies for publishing small body shape data. It is particularly suited to complex shaped bodies for which the radial vector may in some cases pierce the surface several times. However, the particular format of the shape model is irrelevant.

2.1 Image simulation

The first stage in database construction is to generate a large number of simulated images of the asteroid under a wide range of viewing and illumination angles. In this work we use the Planet and Asteroid Natural scene Generation Utility (PANGU) tool⁽²⁵⁾, which is a high-fidelity image simulation package tailored to space environments, and is used to test and validate spacecraft vision-based guidance systems in software.

The position and orientation of the spacecraft relative to the asteroid, and the direction towards the sun are varied from image to image in order to ensure that the whole surface is viewed at a range of angles and lighting conditions. The distance to the asteroid is kept fixed at the expected operational range, and the image size and field of view is selected to match the parameters of the navigation camera. This ensures that surface features in the simulated images are seen at a similar scale and resolution as when the landmark recognition algorithm is used.

2.2 Candidate landmark extraction

The next stage in the landmark extraction process is to apply the Harris corner detector to all the simulated images in order to identify candidate landmark sites. PANGU allows points in the 2D image to be de-projected in order to find the corresponding 3D point on the surface of the asteroid shape model, which enables recovery of the 3D body-frame position of the surface feature lying along the line-of-sight towards each Harris corner. These points are considered to be candidate landmarks at this stage: the full set of candidate landmarks obtained over all the simulated imagery undergoes further processing to extract the final set of landmarks.

Note that not all Harris corners are suitable: in particular, strong corners occurring on the asteroid limb generally don't correspond to fixed surface features but instead mark the position of the horizon against the background sky. In order to eliminate these cases, each feature is correlated with its immediate surroundings within a 31×31 pixel window. Features with one or more strong peaks in the correlation (indicating a part of the image similar in appearance) are purged, which successfully removes features lying along the limb.

2.3 Landmark selection and characterisation

The landmark candidates consists of a set of 3D points on the asteroid surface corresponding to locations that produced a strong Harris corner in one simulated image. Embedded within this point cloud are dense clusters corresponding to prominent, localised surface features that produced Harris corners in many simulated images. These indicate the locations of good known landmark sites, and they are extracted from the point cloud by applying a 3D spatial clustering algorithm. Traditional clustering algorithms such as K-means and expectation maximisation are not applicable in this context, because not all points are to be assigned to a cluster. Also, both of these algorithms distribute the points among a fixed number of clusters that must be specified in advance, but in the present application we want the number of clusters to emerge naturally from the data.

The clustering algorithm designed for this study involves two stages: a preliminary stage of initial cluster extraction, followed by a refinement stage of cluster merging and pruning. The initial cluster extraction stage works as follows. Each point in the set is selected in turn. The remaining

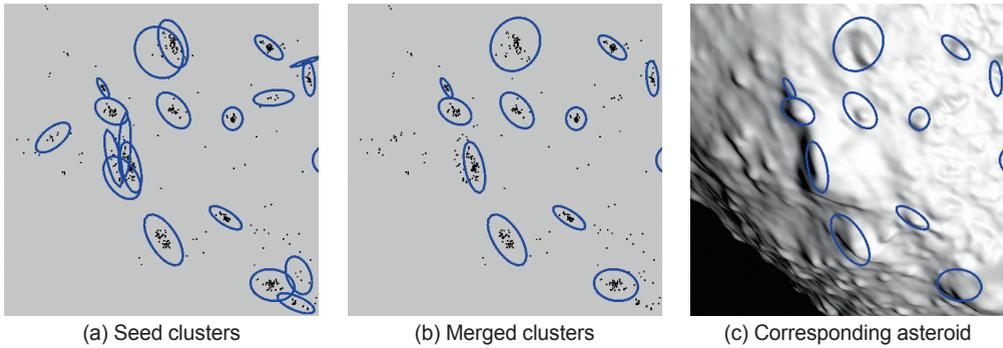


Figure 1. Extracting landmarks from initial cloud of points.

points are then iterated over to search for any that lie within a fixed distance of the starting point. These points form a seed cluster, and further points are added to the seed cluster if they lie within a fixed radius of the cluster centroid, which is updated as more points are added. Due to the updating of the cluster mean position, this stage is repeated until no more points are added, as a previously excluded point may move within the cluster radius. Once the points are exhausted, the search is repeated but this time instead of a fixed distance threshold for cluster membership, the membership criterion is based on the separation distance weighted by the cluster dispersion tensor. This allows extraction of clusters with an intrinsic size larger than the seed radius. Diffuse clusters that have been truncated by the fixed seed length tend to have a 1σ confidence region roughly half that of the seed length at this stage, so a confidence threshold of at least 2σ should be used. Empirically, a 3σ threshold allows a good trade off between extracting diffuse clusters and artificially extending compact clusters, and this is the value adopted in this study.

The cluster refinement stage involves merging and pruning sets of nearby clusters that correspond to the same surface feature seen from different angles. This avoids the possibility of many landmarks appearing at roughly the same position in an image, which would make it difficult to identify the correct one and lead to underestimated uncertainty on the landmark position. Each cluster is selected in turn, and the remaining clusters are iterated over to search for any lying within a covariance-weighted distance of 9σ (the covariance on both cluster positions is considered). Clusters lying within this threshold are considered possible observations of the same surface feature. These clusters are tested in pairs to determine if a merger of the two is appropriate. A merger is considered appropriate if the sum of the traces of the individual cluster dispersion tensors is greater than the trace of the dispersion tensor of the single cluster formed by their merger. Note that the trace provides an orientation-independent estimate of the total cluster dispersion, and this criterion allows clusters with approximately similar shapes to be merged. For cluster pairs that do not satisfy the merger criterion, the less populated of the two clusters is removed from the set. These correspond to distinct surface features lying sufficiently close together that they present a risk of false positives during the landmark identification stage. Removing the less populated of the two reduces the risk of this happening.

Figure 1 depicts the various stages in the clustering algorithm. Each cluster in the final merged set corresponds to one known landmark. The position of each landmark is obtained from the mean position of the cluster, and the landmark position uncertainty is obtained from the cluster dispersion tensor. The landmark position uncertainty is thus modelled as a multivariate Gaussian in the asteroid body-frame coordinates. The covariance terms are important because the covariance ellipsoid is usually highly non-spherical: due to the landmarks being confined to the asteroid surface the covariance ellipsoid will generally have one principal axis smaller than the others, corresponding

approximately to the local surface normal direction. Note that although each landmark corresponds roughly to one distinct surface feature, the landmark position covariance does not represent the uncertainty in the position of the surface feature. Instead, this essentially quantifies the variation in the projected location of the Harris corner produced in an image of the feature, under variations in the view and illumination angle.

2.4 Surface triangulation

The landmarks are connected with their neighbours to obtain a polygon mesh that roughly fits the surface of the asteroid, and consists of a set of triangular facets with a landmark lying at each vertex. The mesh is obtained by first applying a 3D Delaunay triangulation to the landmarks. An incremental insertion algorithm is used for efficiency (see Ref 26), though speed is not a prime concern during the landmark database construction phase. Note that triangulation methods designed for laser range scan data, such as the ball pivoting algorithm⁽²⁷⁾, are not suitable because they are tailored for point clouds that have a characteristic separation between points. In contrast, our landmarks are spaced according to the natural distribution of asteroid surface features and tend to vary in density across the surface. The Delaunay triangulation of a set of points has the property that each simplex (in 3D, a tetrahedron formed by four points) that is a member of the triangulation has a circumscribing sphere (in 3D) that does not contain any other points in the set. This triangulation tends to avoid long and narrow triangles, and provides outer triangular faces that connect nearby landmarks in a manner that closely approximates the natural shape of the asteroid. The outer faces can be identified easily by using the constraint that they form part of only one tetrahedron in the triangulation. In contrast, all internal triangular faces form part of two adjacent tetrahedra.

The outer surface of the Delaunay triangulation provides the convex hull for the landmarks. This is not appropriate for the asteroid surface mesh approximation as any landmarks on concave regions of the surface lie inside the convex hull. The triangulation thus requires some post-processing to extract the concave hull, which has the property that all of the landmarks lie on the outer surface.

The concave hull can be obtained by removing tetrahedra from the triangulation to raise any internal points to the surface. In general, the concave hull is not unique and there are a variety of ways to remove tetrahedra to accomplish this. However, when landmarks are sufficiently densely placed on the surface, one solution is clearly superior. A good heuristic method for obtaining the concave hull is to eliminate tetrahedra for which the centre of the circumscribing sphere lies outside of the convex hull. This obtains concave regions that are shallow compared to other possible solutions (for example those that expose an internal point by piercing a narrow hole through the triangulation from the opposite side), resulting in a surface approximation that has a low gravitational potential. This method also has the advantage of being easily automated. This results in a surface mesh as shown in Fig. 2.

The final step is to obtain the outwards-pointing surface normal for each triangular facet. Each triangle has two possible surface normals pointing in opposite directions perpendicular to the plane of the triangle. The direction corresponding to the outward normal, i.e. the direction out of the asteroid surface, can be recovered by reference to the full triangulation: the outward-pointing normal points away from the tetrahedron that a given triangular facet is part of.

By placing the mesh vertices at the locations of Harris corners and, by proxy, prominent surface features, we end up with a rough shape model that optimally captures the natural variation in surface texture, that has greater resolution in regions where the surface terrain is more variable, and lower resolution in smooth, featureless regions.

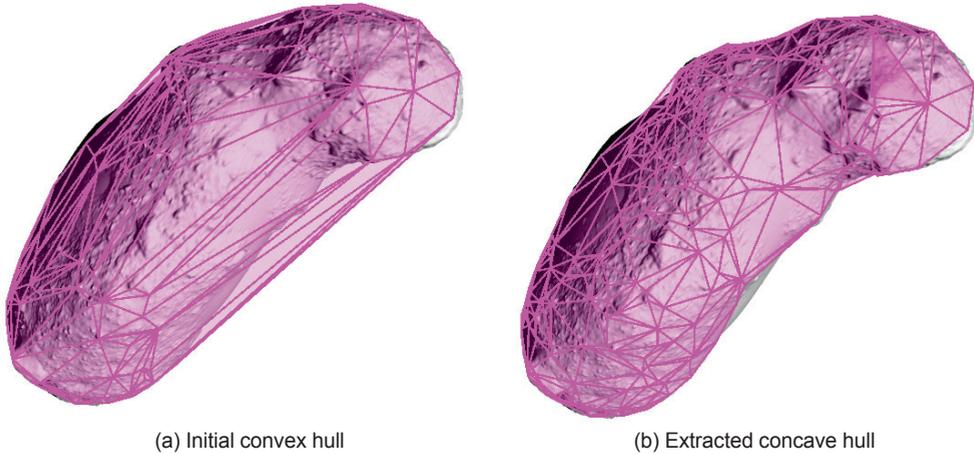


Figure 2. Extracting a surface mesh using the Delaunay triangulation.

2.5 Database format

The full database consists of a list of all the landmarks, followed by a list of all the triangular facets. Each landmark entry is composed of nine floating-point numbers, three corresponding to the landmark mean position and six corresponding to the unique elements of the landmark position covariance matrix (which is 3×3 and symmetric). Each triangle entry consists of three integers that give the indices of the three landmarks in the landmark list that lie at the triangle vertices. The outward-pointing normal for each triangle is embedded in the order that the landmark indices are listed: the normal \underline{n} is recovered from the three landmarks $\underline{L}_0, \underline{L}_1, \underline{L}_2$ using the rule

$$\underline{n} = (\underline{L}_1 - \underline{L}_0) \times (\underline{L}_2 - \underline{L}_0), \quad \dots (1)$$

where the cross indicates the vector product.

3.0 LANDMARK RECOGNITION

The landmark recognition algorithm operates on images taken by the navigation camera, and aims to identify any known landmarks that are presently visible. The basic technique is to project the landmarks database into the navigation image, coarsely align it with the observed asteroid image based on the brightness centroids, then look for matches between observed Harris corners and projected landmarks.

The algorithm requires the camera calibration matrix \mathbf{K} , which contains the camera focal lengths along each image axis (f_i and f_j) and the co-ordinates of the principal point (p_i and p_j). Note that the conventional definition of the image axes is used, where i and j are aligned with the X and Y axes of the camera frame. \mathbf{K} has the form

$$\mathbf{K} = \begin{bmatrix} f_i & 0 & p_i \\ 0 & f_j & p_j \\ 0 & 0 & 1 \end{bmatrix}, \quad \dots (2)$$

with all variables in units of pixels. These quantities are assumed constant during the flight. Also required is an initial estimate of the spacecraft position and orientation with respect to the asteroid body-fixed frame, in terms of the vector $\underline{T} = (T_x \ T_y \ T_z)$ and unit quaternion $q = (q_0 \ q_1 \ q_2 \ q_3)$ that define the transformation from the body-fixed frame to the camera frame (note that strictly $|q| = 1$ in order to represent orientation). These quantities change dynamically throughout flight as the spacecraft moves around the asteroid. They are estimated continually by the spacecraft navigation filter as part of the spacecraft state vector, and they contain noise. The main goal of the landmark recognition algorithm is to use observations of known landmarks to reduce the error on \underline{T} and q .

3.1 Projection of the landmarks database into the image

Each landmark is defined by the mean \underline{L}^{AST} and covariance \mathbf{S}_L^{AST} of its position in the body-fixed frame, where

$$\underline{L}^{ast} = (X, Y, Z) \quad \mathbf{S}_L^{ast} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{bmatrix} \quad \dots (3)$$

The position of each landmark is transformed to the camera frame using standard transformation rules, specifically

$$(0, \underline{L}^{cam}) = q * (0, \underline{L}^{ast}) * q^{-1} + (0, \underline{T}) \quad \dots (4)$$

where the asterisk denotes quaternion multiplication. Note that an additional zero term must be added to the vectors temporarily to allow quaternion multiplication. The landmark position is then projected into the image plane using the camera matrix \mathbf{K} according to

$$\begin{bmatrix} i \\ j \\ w \end{bmatrix} = \mathbf{K} \times \underline{L}^{cam} \quad \dots (5)$$

The image co-ordinates in pixel units are recovered on division by the perspective co-ordinate w

$$\underline{L}^{im} = \begin{bmatrix} i \\ j \end{bmatrix} \leftarrow \begin{bmatrix} i/w \\ j/w \end{bmatrix} \quad \dots (6)$$

The landmark position covariance matrix is transformed to the camera frame according to the standard tensor transformation law

$$\mathbf{S}_L^{Cam} = \mathbf{R}(q) \times \mathbf{S}_L^{ast} \times \mathbf{R}(q)^T \quad \dots (7)$$

where $\mathbf{R}(q)$ is the orthonormal rotation matrix corresponding to the quaternion q , and is calculated using

$$\mathbf{R}(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad \dots (8)$$

Note that $\mathbf{R}(q^{-1}) = \mathbf{R}(q)^T$. The covariance matrix for the landmark is then projected into the image plane by first order propagation, i.e.

$$\mathbf{S}_L^{im} = \begin{bmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{bmatrix} = \mathbf{J}^T \times \mathbf{S}_L^{cam} \times \mathbf{J} \quad \dots (9)$$

where the Jacobian matrix \mathbf{J} contains the partial derivatives of the image plane co-ordinates with respect to the camera frame co-ordinates,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial i}{\partial X} & \frac{\partial j}{\partial X} \\ \frac{\partial i}{\partial Y} & \frac{\partial j}{\partial Y} \\ \frac{\partial i}{\partial Z} & \frac{\partial j}{\partial Z} \end{bmatrix} = \begin{bmatrix} \frac{f_i}{Z} & 0 \\ 0 & \frac{f_j}{Z} \\ -\frac{f_i X}{Z^2} & -\frac{f_j Y}{Z^2} \end{bmatrix} \quad \dots (10)$$

3.1.1 Landmark visibility

Not all landmarks can be seen from a given viewpoint, and it is important to identify any that are not visible at this stage in order to exclude them from the recognition step. Landmarks lying outside of the field of view or behind the camera are trivial to identify based on their image and camera frame coordinates, however those lying on the far side of the asteroid or obscured by intervening terrain are more difficult. These cases are solved using the surface mesh, as follows:

Far side test

For each landmark, the surface normals of all triangular facets that it is connected to are computed using Equation (1). If all of the surface normals point away from the direction towards the camera, then the landmark is judged to lie over the asteroid horizon and is not visible in the current image.

Intervening terrain test

All landmarks that pass the first test are subject to a more rigorous test designed to identify those lying on terrain that faces towards the camera, but which are still obscured due to additional terrain lying closer along the line of sight. This test is particularly important for concave bodies. For each landmark, all the triangular facets that the landmark is not connected to are tested to check if the line of sight vector towards the landmark intersects it, and at what distance. If any triangle intersects the line of sight at a closer distance than the landmark, then the landmark is considered to be obscured by that section of the mesh and is not visible in the image. Note that this stage can be computed in an efficient manner by fitting a minimum bounding sphere to each triangle when the database is first loaded, which allows rapid culling of triangles that lie far from the line of sight.

3.2 Database image rendering and centroiding

The component of the translation error vector lying in the image plane leads to a misalignment between the asteroid image and the projection of the landmarks database, which can lead to large errors during landmark identification as the main criterion for matching is proximity in the image plane. Fortunately, it is possible to significantly reduce the translation error using a pre-processing stage that updates the translation estimate so as to align the landmarks database with the asteroid in the image plane. This is achieved by using ray tracing techniques to render an image of the landmarks database that serves as an approximation to the real image of the asteroid. The update to the translation vector is then computed to bring the brightness centroids of the real and synthetic images into alignment. This stage is described in more detail in the following sections.

3.2.1 Image computation

The image of the landmarks database is computed by first casting a ray out through each pixel and finding the intersection between the ray and the landmark surface mesh. The ray \underline{r} for pixel (i, j) is computed by

$$\underline{r}^{cam} = \mathbf{K}^{-1} \times \begin{bmatrix} i + 0.5 \\ j + 0.5 \\ 1 \end{bmatrix} \quad \dots (11)$$

The point of intersection, if any, between this ray and the landmark surface mesh is found by iterating over the mesh and testing each triangle in turn for intersection with the ray. The full computation for each triangle is quite costly, so a fast rejection method is used where each triangle is fitted on initialisation with a minimum bounding sphere to allow rapid detection and rejection of widely non-intersecting rays. Only triangles for which the ray passes within the bounding sphere need to be subjected to the full intersection calculation. The full calculation uses standard 3D geometry methods and is not repeated here. In general, rays that intersect the landmark surface mesh will have two or more intersections corresponding to the near and far sides. The point of intersection closest to the camera is the desired one.

Once the intersection point is found, the brightness of the pixel is computed. This is done by using the triangle surface normal, the direction towards the camera and the known sun direction to estimate the intensity of reflected light. This requires knowledge of the bidirectional reflectance distribution function (BRDF) for the asteroid surface material, which defines the ratio between the irradiance incident from the light source and the radiance reflected into the view direction. Usually, this is estimated during extraction of the initial shape model, so it is assumed that this function is known reasonably well. In the case of Lambertian reflectance, the BRDF is simply the cosine of the angle between the light source direction and the surface normal.

Finally, self-shadowing is important at all but very low solar phase angles, especially for concave bodies, and must be considered when rendering the image of the landmarks surface mesh. Once the point of intersection between a given ray and the surface mesh is found, a second ray is projected from the intersection point towards the sun. Any intersection between this ray and another section of the surface mesh indicates that the original intersection point is in shadow. These pixels are set to black in the simulated image. Figure 3 shows a simulated image of a landmarks database at a significant solar phase angle, with evident self-shadowing.

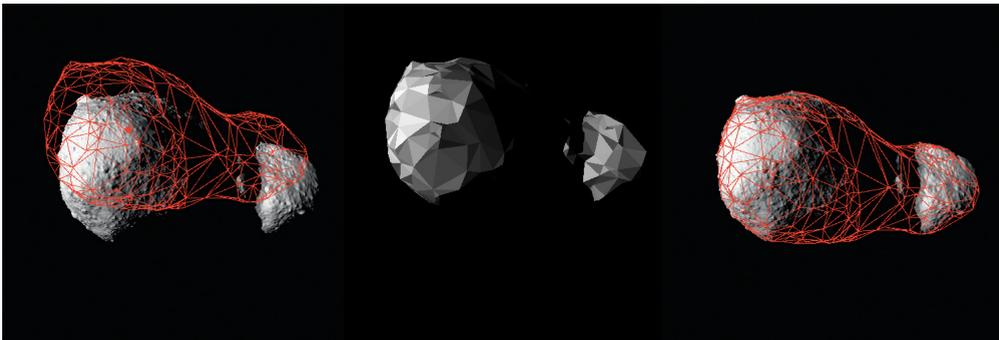
3.2.2 Translation update

The correction to the translation vector is computed from the centres of brightness of the observed and simulated images, which are calculated according to

$$\begin{bmatrix} i \\ j \end{bmatrix}_{COB} = \frac{1}{\sum I(i, j)} \begin{bmatrix} \sum I(i, j) \times i \\ \sum I(i, j) \times j \end{bmatrix} \quad \dots (12)$$

Unit vectors in the camera frame pointing towards each of these centroids, denoted \hat{r}_{obs} and \hat{r}_{sim} , are then found by deprojecting the centroid image co-ordinates using Equation (11). The initial estimate of the spacecraft position in the asteroid body frame, \underline{T} , is used to resolve these directions into a correction to the translation vector $\Delta \underline{T}$ according to

$$\Delta \underline{T} = |\underline{T}| \text{Cos}(\theta) \hat{r}_{obs} - |\underline{T}| \hat{r}_{sim} \quad \dots (13)$$



(a) Initial database projection (b) Rendered image of database mesh (c) Fit following centroid alignment surface

Figure 3. Landmarks database and asteroid fitting based on brightness centroid alignment. The mesh resolution is on average much lower but is tailored to optimally match the surface terrain, with lower resolution in featureless regions and higher resolution in regions of greater surface variation.

where $\text{Cos}(\theta) = \hat{r}_{obs} \cdot \hat{r}_{sim}$. The estimated spacecraft translation is then updated to $\underline{T}' = \underline{T} + \Delta\underline{T}$.

The process of simulating an image of the landmarks database and updating the translation vector to align the brightness centroid with the real image is repeated until the centroid position varies by fewer than five pixels from one iteration to the next. This is necessary mainly to handle cases where either the real or simulated asteroid image lies partially outside the FOV, although even if both are fully contained within the FOV a significant $\Delta\underline{T}$ can bring new parts of the database into view and perturb the brightness centroid. Figure 3 depicts one iteration of the algorithm, showing the initial fit (left), the rendered database image (centre) and the fit following the alignment of the brightness centroids (right).

3.2.3 Brightness centroid error and surface reflectance function

The success of the brightness centroid alignment approach relies on the assumption that the real asteroid image and the low-resolution database image have similar brightness centroids when seen from the same viewpoint and under the same illumination conditions. Additionally, in order to render an image of the database a particular surface reflectance function has to be adopted, and while this can be reasonably well constrained during derivation of the shape model, any remaining uncertainty will propagate to an error in the brightness centroid.

In order to test the similarity in the brightness centroids and the effect of uncertainty in the surface reflectance function, we conducted a Monte Carlo simulation that involved rendering 1,000 images of the database along with a corresponding image of the high resolution asteroid shape model. The orientation of the asteroid was randomly drawn for each sample, and a random solar phase angle of up to 45° was used in order to test the effect of self-shadowing on the centroid deviation. In each case, the brightness centroid of the database image and the simulated asteroid image were measured.

The results are plotted in Fig. 4. Each graph plots the results of 1,000 tests, with the ellipses indicating the 1σ sample covariance. The three cases correspond to three different choices for the real asteroid surface reflectance function. On the left, the Lambertian reflectance function was used, which is a standard model of diffuse reflectance and is a common default in computer graphics. In the centre, the Oren-Nayar reflectance function was used⁽²⁸⁾, which models the reflectance of rough diffuse surfaces such as plaster and cloth. On the right, the Hapke reflectance function

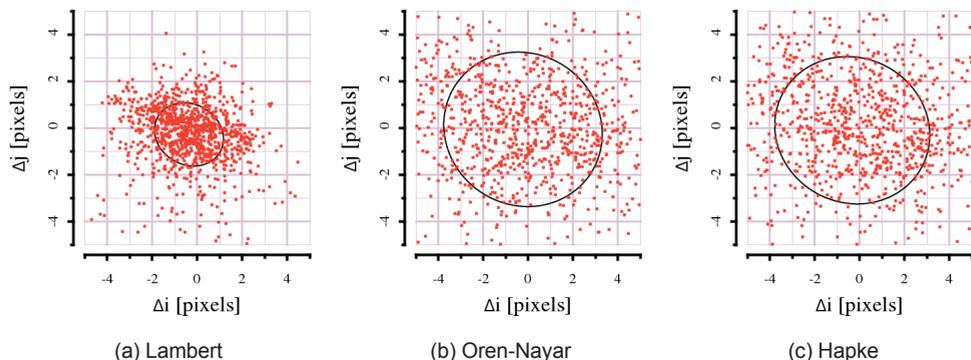


Figure 4. Deviation in the brightness centroid position between a simulated high-resolution image of the asteroid and a corresponding image of the (low-resolution) database surface mesh from the same viewpoint, under different assumptions about the surface reflectance function type.

was used⁽²⁹⁾, which models the reflectance of particulate surfaces including Lunar regolith. This exhibits an unusual reflectance with a strong backscattering component towards the light source. This selection covers a wide range of possible reflectance functions. In all three cases the database image was rendered using the Lambertian reflectance function, so that the latter two cases test the effect of an incorrect surface reflectance function on the brightness centroid.

The results in Fig. 4(a) indicate that the brightness centroids of the high-resolution asteroid image and the low-resolution database image match to within 1-2 pixels when the reflectance function is consistent between the two. The results in Figs 4(b) and 4(c) show that when the incorrect reflectance function is used, the centroids may deviate by up to ~4 pixels. Note that in the test images, which are similar to Fig. 3, the asteroid subtends around 200-300 pixels depending on the configuration so an offset of 1-2 pixels is approximately a one percent deviation.

3.3 Landmark detection and matching

Landmark identification is performed on the basis of proximity in the image plane between observed Harris corners (i, j) and projected landmarks (\hat{i}, \hat{j}) . Strong Harris corners are extracted from the image using the same algorithm used during database construction. These are matched with nearby known landmarks using a covariance weighted distance measure calculated according to

$$d^2 = [i - \hat{i} \quad j - \hat{j}] \times \begin{bmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{bmatrix}^{-1} \times \begin{bmatrix} i - \hat{i} \\ j - \hat{j} \end{bmatrix} \quad \dots (14)$$

The covariance matrix quantifies the uncertainty on the known landmark position, and is calculated according to Equation (9). The closest Harris corner to each known landmark is considered a positive match if it lies within 6σ of the landmark ($d^2 < 36$) and the pairing is reciprocal, i.e the Harris corner does not lie closer to another landmark. This threshold is configurable and was selected in these tests based on the deviation of observed landmarks from their predicted positions following the database rendering and centroiding step.

3.4 Pose estimation

Once the set of recognised known landmarks has been established, the landmark observations are used to refine the estimated spacecraft position and orientation. This is achieved by adjusting the position and orientation parameters to bring the projected landmarks into better alignment with the observed Harris corners to which they are matched. A two stage pose estimation algorithm is adopted. In the first stage, a closed-form algorithm is applied to the observed landmarks to solve for the spacecraft position and orientation directly, without recourse to the initial estimated pose. This amounts to solving the Perspective- n -Point problem of computer vision, and we use the efficient algorithm of Ref. 12 to obtain a solution in a single step. This provides a good starting point for the second stage, which involves performing a non-linear optimisation. Note that the original estimated spacecraft pose is accurate enough to perform known landmark identification, but usually too coarse to optimise directly as the optimisation tends to get stuck in local minima. This motivates the need for the additional direct pose estimation stage.

The optimisation stage uses a classic non-linear parameter estimation algorithm where the parameters of the spacecraft pose are iteratively updated in order to reduce the total reprojection error, which is the displacement in the image plane between the observed landmarks position and the predicted location given the current parameters solution. Because the various quantities involved in the optimisation are somewhat complex, the algorithm is described in greater detail here.

3.4.1 Relevant quantities

The parameter vector \underline{x} contains the free parameters of the pose estimation problem, namely the quaternion and vector components that describe the orientation and position of the camera frame relative to the landmarks database frame:

$$\underline{x} = [q_0 \ q_1 \ q_2 \ q_3 \ T_x \ T_y \ T_z]^T \quad \dots (15)$$

Given a set of N identified landmarks, the data vector \underline{D} has a size of $2N \times 1$ and contains the observed image co-ordinates for each landmark:

$$\underline{D} = [i_1 \ j_1 \ i_2 \ j_2 \ i_3 \ j_3 \ \dots \ i_N \ j_N]^T \quad \dots (16)$$

Each landmark has a corresponding entry in the $2N \times 1$ model vector \underline{M} , which contains the image co-ordinates computed by projecting the landmarks 3D position into the image plane given the current solution for the spacecraft pose:

$$\underline{M} = [\hat{i}_1 \ \hat{j}_1 \ \hat{i}_2 \ \hat{j}_2 \ \hat{i}_3 \ \hat{j}_3 \ \dots \ \hat{i}_N \ \hat{j}_N]^T \quad \dots (17)$$

The basic assumption is that these are coincident ($i = \hat{i}, j = \hat{j}$) given the correct pose solution, up to an error quantified by the covariance matrix for (\hat{i}, \hat{j}) . Note that uncertainty on the observed position is considered to be zero. In addition, the covariance between landmarks is assumed to be uncorrelated. The covariance matrix for the complete problem thus has the following block diagonal form:

$$\mathbf{S}_M = \begin{bmatrix} \sigma_{i_1}^2 & \sigma_{i_1 j_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_{i_1 j_1} & \sigma_{j_1}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{i_2}^2 & \sigma_{i_2 j_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{i_2 j_2} & \sigma_{j_2}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{i_3}^2 & \sigma_{i_3 j_3} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{i_3 j_3} & \sigma_{j_3}^2 & 0 & 0 \\ \vdots & & & & & & \ddots & \\ 0 & 0 & & & & & & \sigma_{i_N}^2 & \sigma_{i_N j_N} \\ 0 & 0 & & & & & & \sigma_{i_N j_N} & \sigma_{j_N}^2 \end{bmatrix} \dots (18)$$

where each block consists of the image plane covariance matrix for one landmark. The inverse of \mathbf{S}_M has the same block diagonal structure, and is obtained by replacing each block with its inverse which is simple due to the symmetry, e.g.:

$$\begin{bmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{bmatrix}^{-1} = \frac{1}{\sigma_i^2 \sigma_j^2 - (\sigma_{ij})^2} \begin{bmatrix} \sigma_j^2 & -\sigma_{ij} \\ -\sigma_{ij} & \sigma_i^2 \end{bmatrix} \dots (19)$$

The objective function for the problem is the total covariance weighted reprojection error, which is the sum of the geometric distance between observed and projected landmarks weighted by \mathbf{S}_M^{-1} , and coincides with the chi-square statistic:

$$\chi^2 = [\underline{D} - \underline{M}]^T \mathbf{S}_M^{-1} [\underline{D} - \underline{M}] \dots (20)$$

The Jacobian matrix contains the partial derivatives of the model with respect to the free parameters, and has the form

$$\mathbf{J} = \frac{\partial \underline{M}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \hat{i}_1}{\partial q_0} & \frac{\partial \hat{i}_1}{\partial q_1} & \frac{\partial \hat{i}_1}{\partial q_2} & \frac{\partial \hat{i}_1}{\partial q_3} & \frac{\partial \hat{i}_1}{\partial T_x} & \frac{\partial \hat{i}_1}{\partial T_y} & \frac{\partial \hat{i}_1}{\partial T_z} \\ \frac{\partial \hat{j}_1}{\partial q_0} & \frac{\partial \hat{j}_1}{\partial q_1} & \frac{\partial \hat{j}_1}{\partial q_2} & \frac{\partial \hat{j}_1}{\partial q_3} & \frac{\partial \hat{j}_1}{\partial T_x} & \frac{\partial \hat{j}_1}{\partial T_y} & \frac{\partial \hat{j}_1}{\partial T_z} \\ \frac{\partial \hat{i}_2}{\partial q_0} & \frac{\partial \hat{i}_2}{\partial q_1} & \frac{\partial \hat{i}_2}{\partial q_2} & \frac{\partial \hat{i}_2}{\partial q_3} & \frac{\partial \hat{i}_2}{\partial T_x} & \frac{\partial \hat{i}_2}{\partial T_y} & \frac{\partial \hat{i}_2}{\partial T_z} \\ \frac{\partial \hat{j}_2}{\partial q_0} & \frac{\partial \hat{j}_2}{\partial q_1} & \frac{\partial \hat{j}_2}{\partial q_2} & \frac{\partial \hat{j}_2}{\partial q_3} & \frac{\partial \hat{j}_2}{\partial T_x} & \frac{\partial \hat{j}_2}{\partial T_y} & \frac{\partial \hat{j}_2}{\partial T_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \hat{i}_N}{\partial q_0} & \frac{\partial \hat{i}_N}{\partial q_1} & \frac{\partial \hat{i}_N}{\partial q_2} & \frac{\partial \hat{i}_N}{\partial q_3} & \frac{\partial \hat{i}_N}{\partial T_x} & \frac{\partial \hat{i}_N}{\partial T_y} & \frac{\partial \hat{i}_N}{\partial T_z} \\ \frac{\partial \hat{j}_N}{\partial q_0} & \frac{\partial \hat{j}_N}{\partial q_1} & \frac{\partial \hat{j}_N}{\partial q_2} & \frac{\partial \hat{j}_N}{\partial q_3} & \frac{\partial \hat{j}_N}{\partial T_x} & \frac{\partial \hat{j}_N}{\partial T_y} & \frac{\partial \hat{j}_N}{\partial T_z} \end{bmatrix} \dots (21)$$

The Jacobian elements can be computed analytically by considering the transformation from the landmarks frame to the camera frame followed by the projection into the image plane. For example, the first elements of rows one and two are, from Equations (5) and (6),

$$\begin{aligned}\frac{\partial \hat{i}}{\partial q_0} &= f_i \times (Z_{cam} \times \frac{\partial X_{cam}}{\partial q_0} - X_{cam} \times \frac{\partial Z_{cam}}{\partial q_0}) \times Z_{cam}^{-2} \\ \frac{\partial \hat{j}}{\partial q_0} &= f_j \times (Z_{cam} \times \frac{\partial Y_{cam}}{\partial q_0} - Y_{cam} \times \frac{\partial Z_{cam}}{\partial q_0}) \times Z_{cam}^{-2}\end{aligned}\quad \dots (22)$$

where f_i and f_j are the camera focal lengths along the i and j directions in the image plane. The remaining column entries of \mathbf{J} (the partial derivatives with respect to the parameters $q_1, q_2, q_3, T_x, T_y, T_z$) can be computed by replacing q_0 in Equation (22). The partial derivatives of the landmark position vector in the camera frame with respect to the free parameters are

$$\begin{aligned}\frac{\partial X_{cam}}{\partial q_0} &= 2q_0 X_{ast} - 2q_3 Y_{ast} + 2q_2 Z_{ast} & \frac{\partial X_{cam}}{\partial T_x} &= 1 \\ \frac{\partial Y_{cam}}{\partial q_0} &= 2q_3 X_{ast} + 2q_0 Y_{ast} - 2q_1 Z_{ast} & \frac{\partial Y_{cam}}{\partial T_x} &= 0 \\ \frac{\partial Z_{cam}}{\partial q_0} &= -2q_2 X_{ast} + 2q_1 Y_{ast} + 2q_0 Z_{ast} & \frac{\partial Z_{cam}}{\partial T_x} &= 0 \\ \\ \frac{\partial X_{cam}}{\partial q_1} &= 2q_1 X_{ast} + 2q_2 Y_{ast} + 2q_3 Z_{ast} & \frac{\partial X_{cam}}{\partial T_y} &= 0 \\ \frac{\partial Y_{cam}}{\partial q_1} &= 2q_2 X_{ast} - 2q_1 Y_{ast} - 2q_0 Z_{ast} & \frac{\partial Y_{cam}}{\partial T_y} &= 1 \\ \frac{\partial Z_{cam}}{\partial q_1} &= 2q_3 X_{ast} + 2q_0 Y_{ast} - 2q_1 Z_{ast} & \frac{\partial Z_{cam}}{\partial T_y} &= 0 \\ \\ \frac{\partial X_{cam}}{\partial q_2} &= -2q_2 X_{ast} + 2q_1 Y_{ast} + 2q_0 Z_{ast} & \frac{\partial X_{cam}}{\partial T_z} &= 0 \\ \frac{\partial Y_{cam}}{\partial q_2} &= 2q_1 X_{ast} + 2q_2 Y_{ast} + 2q_3 Z_{ast} & \frac{\partial Y_{cam}}{\partial T_z} &= 0 \\ \frac{\partial Z_{cam}}{\partial q_2} &= -2q_0 X_{ast} + 2q_3 Y_{ast} - 2q_2 Z_{ast} & \frac{\partial Z_{cam}}{\partial T_z} &= 1 \\ \\ \frac{\partial X_{cam}}{\partial q_3} &= -2q_3 X_{ast} - 2q_0 Y_{ast} + 2q_1 Z_{ast} \\ \frac{\partial Y_{cam}}{\partial q_3} &= 2q_0 X_{ast} - 2q_3 Y_{ast} + 2q_2 Z_{ast} \\ \frac{\partial Z_{cam}}{\partial q_3} &= 2q_1 X_{ast} + 2q_2 Y_{ast} + 2q_3 Z_{ast}\end{aligned}\quad \dots (23)$$

3.4.2 Parameter updates

The iterative pose estimation procedure involves calculating updates to the parameters vector \underline{x} that drive the weighted total reprojection error χ^2 in Equation (20) towards zero. The parameter update vector

$$\underline{\delta} = [\Delta q_0 \ \Delta q_1 \ \Delta q_2 \ \Delta q_3 \ \Delta T_x \ \Delta T_y \ \Delta T_z]^T$$

for each iteration is calculated according to the Levenberg-Marquardt algorithm, which has the form

$$(\mathbf{J}^T \mathbf{S}_M^{-1} \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{S}_M^{-1} \mathbf{J})) \underline{\delta} = \mathbf{J}^T \mathbf{S}_M^{-1} (\underline{D} - \underline{M}) \quad \dots (24)$$

The parameter λ is the damping factor. Equation (24) is solved for $\underline{\delta}$ in a numerically stable manner by using the LUP decomposition to avoid explicit inversion of the 7×7 matrix $\mathbf{J}^T \mathbf{S}_M^{-1} \mathbf{J} + \text{diag}(\mathbf{J}^T \mathbf{S}_M^{-1} \mathbf{J})$. The update to the parameters is then applied according to

$$(\mathbf{J}^T \mathbf{S}_M^{-1} \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{S}_M^{-1} \mathbf{J})) \underline{\delta} = \mathbf{J}^T \mathbf{S}_M^{-1} (\underline{D} - \underline{M})$$

Note that the update to the parameters results in an attitude quaternion that in general no longer has $|q| = 1$, and it is important to renormalise the attitude quaternion elements after each parameter update.

3.4.3 Outlier rejection

Pose estimation is somewhat sensitive to incorrect landmark matches, so an iterative outlier rejection scheme is used in order to purge these. Once the pose estimation algorithm has converged, the set of identified landmarks is tested for consistency between the updated model and observed position; any model landmarks found to now lie greater than 6σ from their observed counterpart are removed, and any newly identified landmarks, given the updated model, are added to the set. This scheme is repeated until a stable set of landmarks is found, which typically occurs after 2-3 iterations.

3.4.4 Convergence

The convergence of the pose estimation algorithm and the ability for noisy known landmark observations to constrain the spacecraft pose were tested in a Monte Carlo fashion using many sets of simulated landmark data generated from the landmarks database described in Section 4.1. Each realisation was produced by randomly positioning the camera at a fixed range of 2km from the asteroid pointing towards the asteroid centre, and projecting the database into the image plane. Ten landmarks were selected at random and observed image plane positions were drawn from the landmark covariance matrices, producing a set of ten known landmark observations for input to the pose estimation algorithm. The histograms in Fig. 5 show the results of 100,000 trials, in terms of the distribution of errors on the camera/asteroid relative pose following algorithm convergence. The attitude error is computed as the error in the relative orientation of the camera and asteroid body frame, and the translation error is the error in the position of the camera in the asteroid body frame[†]. These metrics will be used later to determine the algorithm performance in more realistic scenarios. Note that in the non-linear-only case, the initial pose estimate was obtained by disturbing the ground truth spacecraft pose by adding random translation and orientation errors drawn from Gaussian distributions of standard deviation 50m and 2°, respectively. This margin is representative of the expected errors in a real scenario.

[†] Which is distinct from the position of the asteroid in the camera frame, which is considerably less uncertain due to errors in the asteroid orientation inflating the former translation error. The position of the spacecraft in the asteroid body frame is the quantity more relevant to navigation.

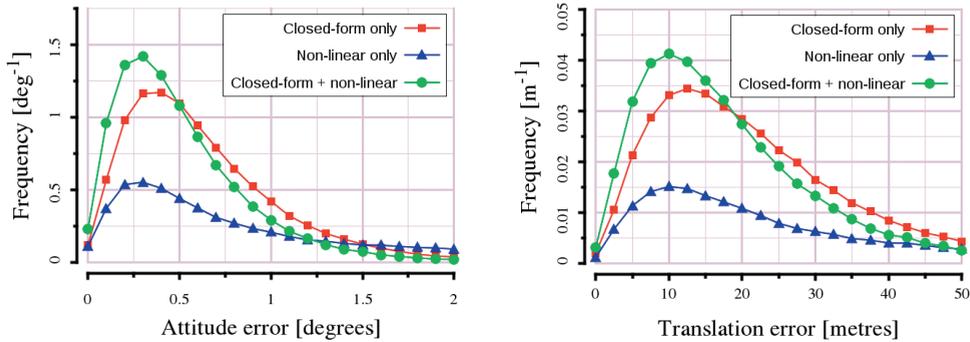


Figure 5. Comparison of pose estimation methods applied to simulated data. Note that the histograms are normalised to units of frequency per degree and metre respectively.

Table 1
Execution times for various stages in the landmark recognition algorithm

Stage	Execution time
Database image rendering	32.7s
Brightness centroid measurement	6.70ms
Feature extraction	758ms
Landmark recognition and pose estimation	520ms

For the assumed initial pose estimation dispersion, the non-linear optimisation scheme does not achieve good results on its own, due to getting stuck in local minima. The closed-form pose estimation algorithm gives better performance, however the full algorithm that uses the closed-form followed by non-linear optimisation gives the best results. Note that the absolute accuracy of the pose solution is affected significantly by the number of landmarks observed, their distribution in the image plane and the camera field of view; the performance in more realistic scenarios is demonstrated in Section 4.

3.5 Profiling

The execution times for various stages in the complete landmark recognition algorithm are listed in Table 1. These were measured during the tests with simulated images presented in Section 4.2, and were performed on a Toshiba Satellite Pro laptop with an Intel i5 CPU running at 2.40GHz. The database rendering stage is by far the most intensive, taking around 30 seconds for a 512×512 pixel image. However, the implementation presented in this paper is based on ray-tracing for simplicity, while in a real application the considerably faster polygon shading technique could be used to reduce execution time. In addition, if the algorithms are implemented in a dedicated image processing FPGA there may be opportunity to use a GPU-style architecture to further reduce execution time. In any case, this paper is concerned with the performance of the algorithms rather than their implementation in real hardware.

4.0 TESTING AND RESULTS

In order to test the performance of the full visual landmark recognition algorithm in a realistic scenario, images of asteroid Itokawa were obtained under known ground-truth spacecraft pose and illumination conditions. These were then processed by the known landmark recognition algorithm using a degraded pose in order to simulate realistic uncertainty on the spacecraft position and orientation.

Two sets of images were used. The first was generated artificially using the PANGU tool to render images of the high resolution Itokawa shape model. This model was also used to derive the known landmarks database. This set provides a very large number of images under a wide range of viewing conditions for robust statistical analysis. It also allows us to de-project points in the image to recover their 3D position on the asteroid surface, which enables an important test of the landmark recognition error that is not possible with real images. The second set of images was selected from real Itokawa image data taken by the Asteroid Multi-Band Imaging Camera (AMICA) instrument on the Hayabusa spacecraft. These images are limited in number, and have a limited range of illumination conditions: the sun is mostly in opposition due to the sun-synchronous orbit used by Hayabusa. The ground-truth spacecraft pose is obtained from the mission SPICE kernels, which are reconstructions of the ephemeris and attitude obtained by reprocessing all of the available sensor data, including the imaging data and shape model. No formal estimate of the accuracy is available from the SPICE kernels, but inspection of the alignment between the asteroid body and the landmarks database indicates that errors are small enough for the trajectory data to be considered ground truth for the purposes of these tests. These images are of interest as they demonstrate performance of the algorithm on real mission data affected by image noise and other degradations. They also implicitly validate the approach of extracting the landmarks database from the derived shape model (rather than e.g. directly from the images). This is important because any systematic errors in the shape model or the database construction procedure would not be apparent in the tests using only simulated images.

4.1 Landmarks database

The landmark database used in these tests was generated from a set of 500 simulated images of the Itokawa shape model. The camera was placed at a distance of 2km from the asteroid at a random position in the asteroid body frame, oriented to point towards the asteroid centre. The Sun was randomly positioned such that the camera-asteroid-Sun phase angle was less than 60 degrees. Two databases were generated tailored to each of the two test scenarios: for the simulated images tests, the database was generated using the Lambert BRDF in order to match the reflectance function used to generate the test images. This database contains 409 landmarks which are connected by 814 triangles comprising the surface mesh. In the real images tests, the database was generated using the Oren-Nayar BRDF in order to more closely match the surface reflectance function seen in the real images of Itokawa, which is considerably flatter with less backscattering than the Lambert BRDF. This database consists of 337 landmarks and 670 triangles, slightly fewer than for the Lambert case because the flatter BRDF results in lower contrast surface features producing fewer distinctive landmarks.

4.2 Simulated images

A set of 6,000 simulated images of Itokawa was generated in a similar manner to the images used to generate the landmark database. The camera was placed at distance of 2km from the asteroid origin in a random direction, and oriented to point the boresight towards the origin. The camera

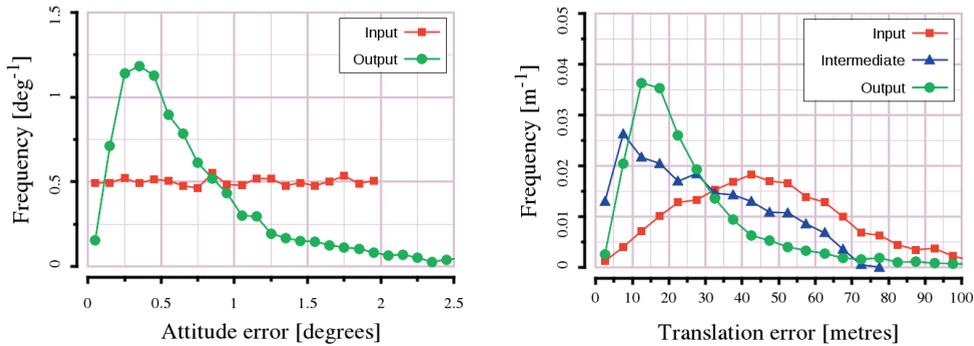


Figure 6. Spacecraft attitude and translation error distributions before and after the landmark recognition algorithm has converged. The attitude error is the rotation required to align the estimated and ground truth asteroid body frames. The translation error is the difference between the estimated and ground truth position of the spacecraft in the asteroid body frame. The improvement in position knowledge following the intermediate step of database rendering and centroiding is also shown on the right. Note that this step does not improve the knowledge of the orientation, so is not shown on the attitude plot.

field of view was set to 18.3° , a value identified as a good trade off between the ability to view the surface at high resolution while fitting the entire asteroid in the image. This was the value adopted for the navigation camera in the Marco Polo-R studies. The generated images were 512×512 pixels, which is smaller than normal for navigation cameras (which often have $1,024 \times 1,024$) but enables faster processing for Monte Carlo simulations. At the nominal range (2km) and asteroid radius (200m) this results in a pixel scale of roughly 1.13m on the asteroid surface, which corresponds to an asteroid rotation of ~ 0.324 (as viewed from the spacecraft) and a spacecraft translation in the asteroid frame of ~ 11.3 m. These set approximate limits on the accuracy of the asteroid position and attitude determination obtainable from imaging data.

The estimated position and attitude used as input to the landmark recognition algorithm was generated from the ground truth by adding random errors according to the following scheme. The main sources of error, according to Section 1.2, are in the spacecraft position and the asteroid orientation in the inertial frame. The spacecraft position knowledge was degraded by adding a random vector with components in the camera frame uniformly distributed within the ranges $[-50:50]$ metres in the X and Y axes (lying in the image plane), and $[-5:5]$ metres in the Z axis (along the boresight). The uncertainty in the spacecraft position along the boresight is lower due to the presence of a laser altimeter. The error distribution for components perpendicular to the boresight are determined by the accuracy of position estimation available by ground-based orbit determination. The asteroid orientation knowledge was degraded by adding a rotation about a random axis, with a magnitude uniformly distributed in the range $[-2:2]$ degrees. Once the degraded position and attitude of the spacecraft and asteroid have been set in the inertial frame, the relative position and attitude are computed for input to the landmark recognition algorithm.

4.2.1 Pose estimation error

The distribution of errors on the spacecraft position and orientation in the asteroid frame before and after the landmark recognition algorithm has operated are shown in Fig. 6. The median error on the attitude is reduced from 1.00° to 0.58° , and that on position is reduced from 45.3m to 20.9m. Note that the median is a more appropriate statistic than the mean due to a small number of high outliers. The position error distribution following convergence also changes shape, with

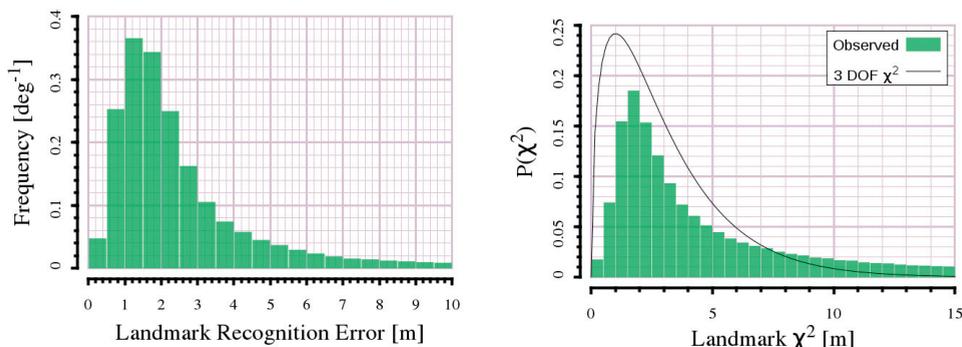


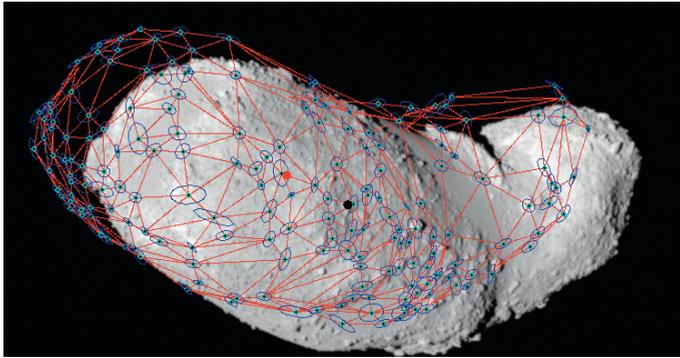
Figure 7. Deviation between the 3D position of the known landmark in the database, and the point on the asteroid surface identified with the landmark. The error is expressed in terms of the absolute deviation in metres (left) and the covariance-weighted squared deviation (the χ^2 statistic; right). The median absolute error is 1.99m. The fact that the observed χ^2 distribution is broader than expected indicates that the true landmark position error distribution is broader than Gaussian.

the error perpendicular to the boresight (the *XY* components) being reduced and the error along the boresight (*Z* component) inflated: the root-median-square error in each component changes from (27.1, 27.4, 2.45) to (10.2, 10.1, 8.11). This is due to the fact that images are less sensitive to the along-boresight position, especially when the field of view is small. In practice, fusion with an altimeter would reduce the along-boresight error. Interestingly, outlying points tend to have large errors on both the position and the attitude and arise due to incorrect landmark identifications; in these cases, the asteroid is incorrectly ‘rolled’ about an axis perpendicular to the line of sight, causing both the attitude and position to diverge. When incorporated into a navigation system, these cases would presumably be detected by the navigation filter and rejected.

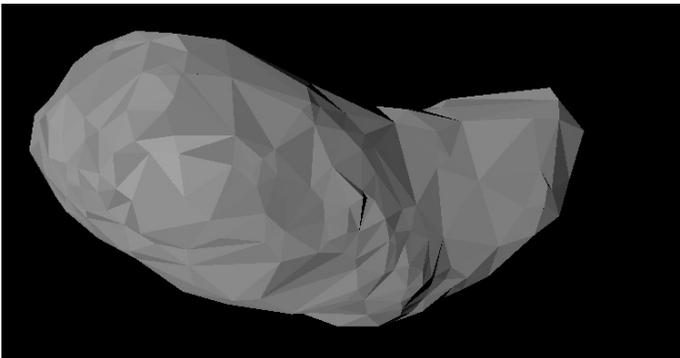
4.2.2 Landmark recognition error

The image simulation environment allows pixels containing the asteroid image to be de-projected to obtain the 3D position of the corresponding point on the asteroid surface. This enables an important test for the landmark recognition robustness, as it allows the 3D position of the known landmark in the database to be compared with the 3D position of the surface point that has been matched with the landmark. As landmarks are not individually recognisable, the only way to check that a correct identification has been made is to test the offset between the known landmark position in the asteroid body frame (recorded in the database) and the de-projected position of the surface point. The pose estimation algorithm requires that these points are coincident, up to an error characterised by the position covariance matrix for the known landmark. This can only be tested on a statistical basis by analysing a large enough set of landmarks.

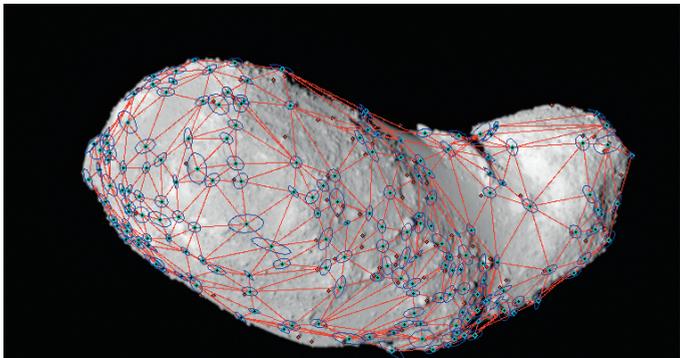
The error distributions for all recognised landmarks is shown in Fig. 7. These distributions consist of 337,854 individual landmark observations over 6,000 images in the test sequence, for an average of 56 landmarks per frame. The median absolute landmark recognition error is 1.99m. Folding in the landmark covariance, the χ^2 distribution is somewhat broader than expected from theory (a three degree-of-freedom χ^2 corresponding to the three spatial dimensions in which landmarks are localised); this would seem to indicate that the true error distribution on the landmark position is not truly Gaussian and is likely broader and flatter. However the assumption of a Gaussian distribution greatly simplifies matters by allowing straightforward error propagation, and the good overall accuracy vindicates this approach.



(a) Initial fit prior to landmark recognition



(b) Rendered landmarks database for iteration 1



(c) Converged solution

Figure 8. These figures depict various stages in the execution of one test using a real image of asteroid Itokawa. They have been cropped for clarity. Figure 8(a) presents the initial noisy projection of the landmarks database into the image; the black and red spots show the brightness centroids for the real asteroid and the rendered landmarks database, respectively. The blue ellipses indicate the 6σ matching radius for each of the model landmarks, which lie at the vertices of the mesh. Figure 8(b) shows the rendered landmarks database corresponding to 8(a): this is produced using the Oren-Nayar reflectance function in an attempt to match the flat, low contrast reflectance exhibited by the real asteroid. Figure 8(c) shows the final converged solution. The red diamonds mark Harris corners detected in the real image, a subset of which correspond to the observed locations of known landmarks.

4.3 Real images

A set of 132 real images of the asteroid Itokawa were selected for use in these tests. These were taken by the Hayabusa spacecraft on the 1st October 2005 from the ‘home’ position at a distance of 8km from the asteroid, using the AMICA instrument^(24,30). AMICA has a FOV of 5.83° and takes images of $1,024 \times 1,024$ pixels; at the nominal altitude of home position the entire asteroid is in the FOV. AMICA also has four 200×200 pixel polarizers mounted along one edge, which result in reduced intensity over roughly one fifth of the image. This effect would be difficult to simulate during the database rendering step, so the images were chosen with Itokawa in a configuration that leaves it clear of the masked area. The AMICA pixel scale of $\sim 0.00569^\circ \text{ pix}^{-1}$ provides a resolution of $\sim 0.78\text{m pix}^{-1}$ on the asteroid surface, which corresponds roughly to an asteroid rotation of $\sim 0.223^\circ$ and resulting spacecraft displacement in the asteroid body frame of $\sim 31\text{m}$. The smaller FOV and higher pixel resolution may therefore offer greater constraint on the orientation relative to the tests with simulated images, however the larger range to the asteroid will inflate the translation errors. In addition, the real images are slightly out of focus and it was necessary to increase the kernel size of the Harris corner detector in order to detect larger scale features that are not obliterated by the blurring. An image from the test sequence is shown in Figure 8(a), together with the initial noisy projection of the landmarks database. The same error distributions are used as in the simulated images case, and each real image was used ten times with ten noise realisations in order to increase the sample size to 1,320. Figure 8(b) shows the rendered database image for the first iteration, using the Oren-Nayar reflectance function to more closely match the observed reflectance (compare this to Fig. 3(b) that uses the Lambert BRDF). Figure 8(c) shows the final fit for this test following convergence of the landmark recognition algorithm.

4.3.1 Pose estimation error

The distribution of errors on the spacecraft orientation and position in the asteroid frame, before and after the landmark recognition algorithm has converged, are shown in Fig. 8. The median error on the orientation is reduced from 1.04° to 0.90° , whereas the median error on the position has increased from 107m to 119m. The position is not very well constrained due to the large distance to the asteroid: a small orientation error leads to a large change in the spacecraft position in the asteroid body frame. These errors are larger than expected based on a simple extrapolation from the simulated images tests to account for the smaller FOV and larger range, and presumably are increased due to the blur in the real images. The root-median-square position error in each component, expressed in the spacecraft frame, changes from (59.3, 58.0, 3.4) to (74.7, 46.7, 43.1). The behaviour is similar to the simulated images tests: the along-boresight position error is inflated due to the lack of sensitivity in this direction. Again, fusion with an altimeter would reduce this. Interestingly, the position error in the spacecraft *Y* axis (vertical in the images) is the only component to be reduced, presumably because in these particular images the asteroid is primarily aligned with and extended along the *X* axis (horizontal in the images — see Fig. 8), resulting in reduced localisation in that direction relative to the perpendicular direction.

4.3.2 Number of detected landmarks

The number of landmarks detected in each image, for both the simulated and real image tests, is shown in Fig. 10. Pose estimation formally requires a minimum of four landmarks in order to reach a unique solution; this is satisfied in all test cases. The median number of detected landmarks is 59 in the simulated images and 44 in the real images. The difference is consistent with the differing landmark database size for each scenario (409 versus 337), which is ultimately caused

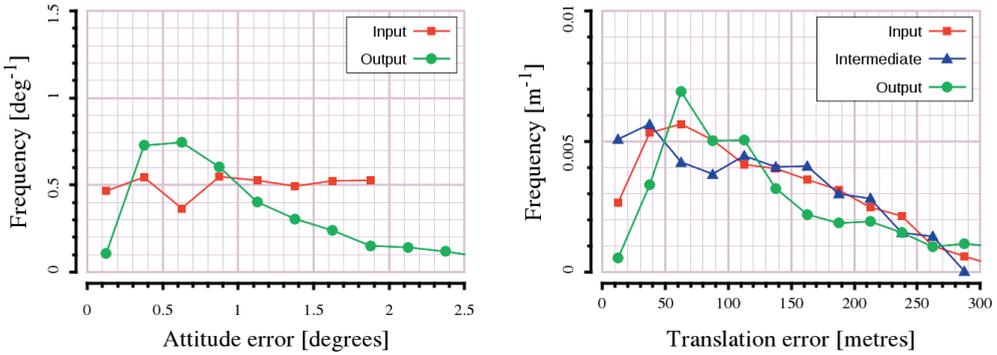


Figure 9. Error distributions on the spacecraft orientation and position in the asteroid body frame, before and after the landmark recognition algorithm has converged. The plots show the results of 1,320 tests performed on real images of the asteroid Itokawa, and are analogous to the results derived from simulated images that are presented in Fig. 6.

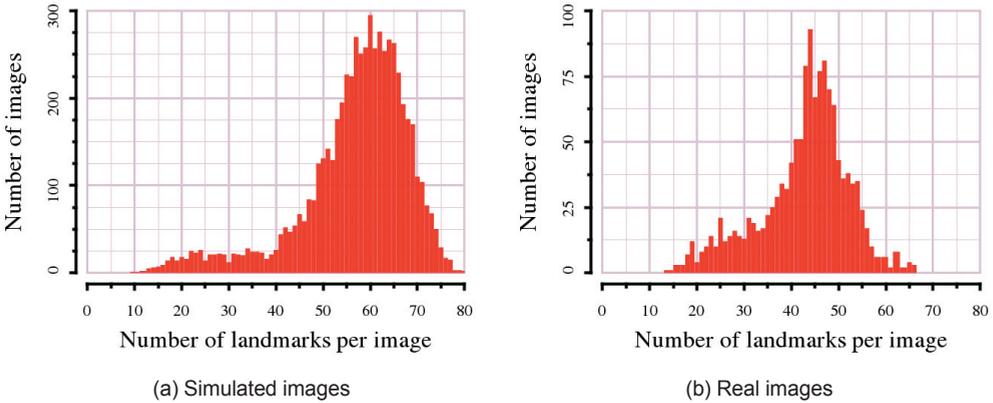


Figure 10. Number of landmarks detected per image, for both the simulated and real image test scenarios. The median in each case is 59 and 44, which is consistent with the different sizes of the landmark database.

by the different levels of contrast in surface detail caused by variations in the reflectance function. There does not appear to be a significant degradation in landmark recognition when moving to real images, which suggests the algorithm is robust to realistic image noise and that the method of defining the landmarks database is valid.

5.0 DISCUSSION

5.1 Realism

The tests presented in Section 4 have been designed to be as realistic as possible within the limitations imposed by computer simulation. Images from navigation cameras and other GNC sensory inputs from real mission data are not usually available as the scientific instruments are given downlink priority. On-ground tests with real cameras require complex and expensive equipment such as the robotic

facility⁽³¹⁾ that was used during the Marco Polo-R industrial studies that tested a previous version of the algorithm presented in this paper. (Note that as the algorithm is an extended and refined version of that used during the industrial studies, the results from those high-fidelity tests are likely not realistic for the algorithm in its current state.) Using a combination of computer simulated and real images from non-navigation cameras is the most feasible means of testing the algorithm for this paper. The simulation scenarios and parameters used in the tests derive from those adopted for the industrial studies, and were selected to be representative of a real asteroid optical navigation scenario. For example, the camera field of view was identified as a good trade off between high resolution and wide angle in order to optimise the surface pixel scale. The altitude at which the landmark recognition is used and the range of Sun elevations are selected considering that the spacecraft commences the descent and landing phase from either a self-stabilised terminator orbit or inertial hovering above the landing site⁽²⁾. The initial errors on the spacecraft/asteroid relative pose are derived considering the accuracy of star trackers, delta-DOR positioning, and the likely knowledge of the asteroid rotational state and ephemeris following the global characterisation phase and observations leading up to the encounter. The main limitation of simulated images is the lack of a realistic optical and electronic noise model, which was the motivation for including tests using real images from the Hayabusa mission. While AMICA is not a navigation camera, it is the same general type of instrument (a framing camera) with a detector size ($1,024 \times 1,024$ pixels) equal to that of the navigation camera considered during the Marco Polo-R studies⁽⁸⁾. Navigation cameras tend to use CMOS detectors whereas AMICA uses a CCD so the electrical noise will be a little different, but overall the effects of noise can likely be mitigated by judicious choice of filter sizes within the algorithm.

5.2 Relation to existing optical navigation strategies

An alternative optical navigation algorithm, developed recently and employed by the Dawn and Hayabusa missions⁽¹⁶⁾, uses landmarks that consist of small 3D shape models of isolated patches of the asteroid surface. These typically contain around 10,000 vertices, and tens to hundreds of them are pieced together over the whole asteroid in order to constrain the global shape model in an offline analysis. Only a small subset are used during optical navigation; these are projected into the image plane and rendered, then aligned with the observed asteroid features in order to constrain the spacecraft pose. Both the L-map method and the method presented in this paper use landmarks that are robust to variations in the view and illumination angles. In the L-map method this is achieved by rendering the landmarks using the (approximately) known values of these angles; our method achieves this by selecting landmarks that are stable under wide variations in these angles. Both methods can filter out landmarks that lie on the far side of the asteroid, by looking for landmarks for which the surface normal points away from the camera. However, in our method knowledge of the complete asteroid shape allows the identification of landmarks that lie on terrain facing towards the camera but that are obscured by intervening terrain, so it can potentially provide better performance on highly concave or irregular bodies. Our method uses a whole-asteroid alignment approach to reduce the lateral translation errors, by using an optimized low-resolution polygon mesh representation of the asteroid global shape. The L-map method does this using multiple individual landmarks. It is possible that the mesh allows comparable constraint on the lateral translation while being smaller than even a single 10,000-vertex L-map, although the sophistication and complexity of L-maps relative to our landmarks (Harris corners) likely gives better constraint on the position of individual landmarks, and thus better constraint on the final spacecraft pose. A full trade off of the performance and process burden of the two methods is outwith the scope of this paper; we feel that the two methods are complementary, and the use of a surface mesh could be combined with L-map based landmarks to exploit the best features of each algorithm.

6.0 CONCLUSION

In this paper we have presented an autonomous landmark recognition and pose estimation algorithm designed for use in vision-based navigation of spacecraft about small asteroids. The algorithm is based on the use of prominent Harris corners as known landmarks, and identifies them based on proximity in the image plane between predicted and observed Harris corners. In addition a rough surface mesh model is used to determine the visibility status of each landmark, and to simulate the brightness centroid in order to reduce large transverse position errors. The algorithm was tested on sets of simulated and real images. In the real images, the Harris corner detector filter size had to be enlarged in order to overcome blurring in the image and detect larger scale features. With that tuning, no major sensitivity to noise was detected in the landmark recognition process and the two test scenarios identified the same number of landmarks, once the differing size of their landmark databases was taken into account. The accuracy of pose estimation is dependent on the range to the target, as at larger ranges the orientation error causes the position error to be inflated. This, in addition to a slight blurring, led to poorer constraint on the spacecraft position in the real images tests. It was also found that the pose estimation solution was less constrained in the along-boresight direction, suggesting that fusion with an altimeter would be a natural extension to the system. The position error for landmarks based on Harris corners was found to be broader than Gaussian, and is induced by changes in the view and illumination angles. This non-Gaussianity may have an impact on GNC systems that assume Gaussian error models. In addition, the assumed surface reflectance function had an impact on the accuracy of the brightness centroid modelling, although at a relatively low level.

Possible future refinements include considering all observed Harris corners within the matching radius of a landmark as potential matches, and using a RANSAC scheme among all possible landmark matches to reject outliers and identify the most reliable set of landmark observations. In addition, the approach of using a surface mesh to model the brightness centroid and test the landmark visibility on a concave body is quite general and can be used with other more sophisticated types of landmark such as SURF features or L-maps. The process of reducing translation errors in the image plane by aligning the brightness centroids of the observed and simulated images could be enhanced by using whole-image alignment. Finally, the stages of database rendering and landmark recognition could be coupled to ensure that the pose estimation algorithm gives a final result that maintains a consistent model and observed brightness centroid, which could detect outlying pose estimation solutions.

ACKNOWLEDGEMENTS

This work was partially supported by funding from ESA under contract number 4000107209/12/NL/HB. The PANGU image simulation facility has been developed by the University of Dundee under contract for ESA. PANGU is available under license from ESA for use on ESA-funded projects, and is available from STAR-Dundee Ltd for other applications.

REFERENCES

1. NASA/JPL, Near-Earth Asteroid Delta-V for Spacecraft Rendezvous, http://echo.jpl.nasa.gov/~lance/delta_v/delta_v.rendezvous.html. Accessed: 21/08/2014.
2. GIL-FERNÁNDEZ, J. CADENAS, R., PRIETO-LLANOS, T., ROWELL, N., DUNSTAN, M., PARKES, S., HOMEISTER, M., SALEHI, S. and AGNOLON, D. Autonomous GNC system to enhance science of asteroid missions, 63rd International Astronautical Congress, October 2012, Naples, Italy.

3. LOVEGROVE, S., DAVISON, A. J. and IBANEZ-GUZMÁN, J. Accurate visual odometry from a rear parking camera, *Intelligent Vehicles Symposium (IV)*, 2011, Baden-Baden, Germany, *IEEE* June 2011, pp 788-793.
4. SCHWENDNER, J. Homography based state estimation for aerial robots, *KI 2008: Advances in artificial intelligence*, *Lecture Notes in Computer Science*, 2008, **5243**, pp 332-339, Springer Berlin, Heidelberg, Germany.
5. HAYET, J., LERASLE, F. and DEVY, M. A Visual landmark framework for indoor mobile robot navigation, *ICRA '02 IEEE International Conference, Robotics and Automation*, May 2002, **4**, pp 3942-3947.
6. OWEN, W.M. Methods of optical navigation, 21st AAS/AIAA Space Flight Mechanics Conference, February 2011, New Orleans, LA, USA.
7. CHENG, Y., GOGUEN, J., JOHNSON, A., LEGER, C., MATTHIES, L., MARTIN, M.S. and WILLSON, R. The Mars exploration Rovers descent image motion estimation system, *IEEE Intell Syst*, May 2004, **19**, (3), pp 13-21.
8. FLANDIN, G., POLLE, B., LHERITIER, J. and VIDAL, P. Vision based navigation for autonomous space exploration, *NASA/ESA Conference on Adaptive Hardware and Systems*, 15-18 June 2010, Anaheim, CA, USA, pp 9-16.
9. HARRIS, C. and STEPHENS, M. A combined corner and edge detector, *Fourth Alvey Vision Conference*, 1988, Manchester, UK, pp 147-151.
10. BAY, H., ESS, A., TUYTELAARS, T. and VAN GOOL, L. Speeded-up robust features (SURF), *Computer Vision and Image Understanding (CVIU)*, June 2008, **110**, pp 346-359.
11. HARTLEY, R.I. and ZISSERMAN, A. *Multiple View Geometry in Computer Vision*, Second Edition, 2004, Cambridge University Press, UK.
12. LEPETIT, V. MORENO-NOGUER, F. and FUA, P. EPnP: An accurate O(n) solution to the PnP problem, *Int J Computer Vision*, 2009, **81**, (2), pp 155-166.
13. OWEN, W.M., WANG, T.C.H., BELL, A.M. and PETERSON, C. NEAR optical navigation at Eros, *AAS/AIAA Astrodynamics Specialist Conference*, 30 July 2001, Quebec, Canada.
14. GASKELL, R., BARNOUN-JHA, O., SCHEERES, D., MUKAI, T., HIRATA, N., ABE, S., SAITO, J., ISHIGURO, M., KUBOTA, T., HASHIMOTO, T., KAWAGUCHI, J. YOSHIKAWA, M., SHIRAKAWA, K. and KOMINATO, T. Landmark navigation studies and target characterization in the Hayabusa encounter with Itokawa, *Astrodynamics Specialist Conference*, 21-24 August 2006, Keystone, CO, USA
15. MASTRODEMOS, N., RUSH, B., VAUGHAN, D. and OWEN, B. Optical navigation for Dawn at Vesta, 21st AAS/AIAA Space Flight Mechanics Conference, February 2011, New Orleans, LA, USA.
16. GASKELL, R., BARNOUN-JHA, O., SCHEERES, D.J., KONOPLIV, A.S., MUKAI, T., ABE, S., SAITO, J., ISHIG-URO, M., KUBOTA, T., HASHIMOTO, T., KAWAGUCHI, J., YOSHIKAWA, M., SHIRAKAWA, K., KOMINATO, T., HIRATA, N. and DEMURA, H. Characterizing and navigating small bodies with imaging data, *Meteoritics & Planetary Sci*, June 2008, **43**, pp 1049-1061.
17. MarcoPolo-R Science Study Team, Marco Polo-R assessment study report, Technical Report, December 2013, ESA/SRE(2013)4, European Space Agency.
18. BIGNAMI, G.F. CARGILL, P.J. SCHUTZ, B. and TURON, C. *Cosmic Vision: Space Science for Europe 2015-2025*, October 2005, BR-247, ESA Publications Division.
19. ROWELL, N., PARKES, S. and DUNSTAN, M. Image processing for near-Earth object vision based guidance systems, *IEEE Transactions on Aerospace and Electronic Systems*, April 2013, **49**, pp 1057-1072.
20. ESA, Delta-DOR: New techniques for ESA's deep space navigation, Technical Report, November 2006, Bulletin 128, European Space Agency.
21. TOMASI, C. and KANADE, T. Detection and tracking of point features, Technical Report, April 1991, CMU-CS-91-132, Carnegie Mellon University.
22. LOWE, D.G. Distinctive image features from scale invariant key-points, *Int J Computer Vision*, 2004, **60**, (2), pp 91-110.
23. COCAUD, C. and KUBOTA, T. SURF-based SLAM scheme using Octree occupancy grid for autonomous landing on asteroids, *Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 29 August-1 September 2010, Sapporo, Japan.
24. Hayabusa Project Science Data Archive, <http://darts.isas.jaxa.jp/planet/project/hayabusa/>. Accessed: 07/08/2014.
25. PARKES, S., MARTIN, I., DUNSTAN, M., ROWELL, N., DUBOIS-MATRA, O. and VOIRIN, T. A virtual test environment for validating spacecraft optical navigation, *Aeronaut J*, November 2013, **117**, pp 1075-1101.
26. NANJAPPA, A. Delaunay Triangulation in R3 on the GPU, PhD thesis, 2012, National University of Singapore.

27. BERNARDINI, F., MITTLEMAN, J., RUSHMEIER, H., SILVA, C. and TAUBIN, G. The ball-pivoting algorithm for surface reconstruction, *IEEE Transactions on Visualization and Computer Graphics*, October 1999, **5**, pp 349–359.
28. OREN, M. and NAYAR, S.K. Generalization of Lambert's reflectance model, 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH '94, Orlando, FL, USA, pp 239-246, 1994.
29. HAPKE, B. Bidirectional reflectance spectroscopy 1. Theory, *J Geophys Res*, 1981, **86**, (B4), pp 3039-3054.
30. ISHIGURO, M., NAKAMURA, R., THOLEN, D.J., HIRATA, N., DEMURA, H., NEMOTO, E., NAKAMURA, A.M., HIGUCHI, Y., SOGAME, A., YAMAMOTO, A., KITAZATO, K., YOKOTA, Y., KUBOTA, T. HASHIMOTO, T. and SAITO, J. The Hayabusa spacecraft asteroid multi-band imaging camera (AMICA), *Icarus*, June 2010, **207**, pp 714-731.
31. CABALLERO, F., MARTIN-ROMERO, J. R., MOLLINEDO, L. and SUATONI, M. Dynamic rendezvous and docking simulation facility: Use cases and status 2012, 5th International Conference on Astrodynamics Tools and Techniques, 29 May-1 June 2012, ESA/ESTEC, The Netherlands.