

ENSURING ACCURATE COMET SHAPE RECONSTRUCTION TO PREPARE FOR THE ROSETTA LANDER DELIVERY

Michael L. Flegel⁽¹⁾, Jan Essert⁽²⁾, and Robin Raus⁽³⁾

⁽¹⁾ESA, ESOC, HSO-GFF, Robert-Bosch-Str.5, 64293 Darmstadt, Germany, +49 6151 903629, michael.flegel@esa.int

⁽²⁾CGI (Germany) GmbH & Co KG, Robert-Bosch-Str.5, 64293 Darmstadt, Germany, +49 6151 902096, jan.essert@esa.int

⁽³⁾Telespazio VEGA Deutschland GmbH, Robert-Bosch-Str.5, 64293 Darmstadt, Germany, +49 6151 903220, robin.raus@esa.int

Abstract:

When ESA's Rosetta mission arrives at the comet 67P/Churyumov-Gerasimenko, the shape of the comet will be estimated in order to select a suitable landing site for the Philae lander. An approach for validating the shape reconstruction process is presented. This is split into two major parts. Firstly, realistic images and other data have to be provided. Details on how this is performed are given. Secondly, methods are developed on how to use the available data in order to validate the shape estimation process and quantify the quality of the reconstructed shape. Emphasis is placed on methods that can be reused during real operations, i.e. without access to data that was used as input to generate the images in the first step. We conclude that the Flight Dynamics Team at ESOC is ready for real comet operations and lander delivery.

Keywords: Rosetta, Philae, 67P/Churyumov-Gerasimenko, comet shape reconstruction, lander, validation

1. Introduction

ESA's Rosetta mission was launched in 2004 with the ultimate purpose of orbiting the comet 67P/Churyumov-Gerasimenko and delivering the Philae lander to its surface. On its ten-year cruise, Rosetta has performed three Earth and one Mars swing-by. The trajectory chosen furthermore enabled close proximity fly-bys of the asteroids Steins and Lutetia. After two and a half years in deep-space hibernation, while it receded from the Sun as far as 5.2AU, Rosetta has successfully exited hibernation in January 2014. A series of maneuvers is scheduled starting in May, 2014 to reach orbit around the comet in August 2014. Lander delivery is planned for November 11, 2014.

Arguably the most critical aspect of the Rosetta mission, the lander delivery, requires the knowledge of a multitude of parameters about the comet, both absolute and relative to Rosetta. In particular, an accurate 3D shape model of the comet has to be created in order to select a suitable landing site. Shape reconstruction is performed in the three months between arrival at the comet and the lander delivery. The Flight Dynamics Team at ESOC in Darmstadt, Germany performs this employing an iterative process using images taken by navigation cameras on board Rosetta. This paper discusses the methodology employed in order to gain confidence in the methods used to reconstruct the shape as well as in the accuracy of the shape model. In the search for validation methods, emphasis is put on methods that lend themselves to also be used operationally when the shape of the real comet is reconstructed.

To this end, realistic comet kinematics, an arbitrary but representative comet shape, and relative Rosetta-comet trajectory are chosen. Images are rendered at realistic intervals, accurately modeling camera distortion of the navigation cameras. In total over 700 images were generated, simulating the images that are expected to be taken in the time between arrival at the comet and lander delivery.

These are then split into batches and used to exercise the two-pronged operational process. Before reconstructing the shape of the comet, the comet state is determined. For this purpose, landmarks on the comet surface are manually selected and correlated between images separated in time. These landmarks are then used to determine the comet kinematic state and relative Rosetta-comet trajectory as a function over time. A by-product of this reconstruction is a landmark database, giving landmark positions in a comet-fixed frame. This step is also performed and validated by the Flight Dynamics Team at ESOC, but is not subject of this paper.

Upon successful reconstruction of the state, the same images are reused to reconstruct the comet shape using the Rosetta attitude and the now-known comet attitude and relative Rosetta-comet trajectory. The validation of this process is the focus of this paper. Gaining confidence in the shape model and its accuracy is performed by several means. Emphasis is placed on methods which make no use of any a priori knowledge of the simulated comet, and as such can also be used operationally. Such methods augment other validation approaches, which do make use of the a priori knowledge.

The paper is structured as follows. In section 2., the operational process is briefly described in order to convey where and how validation may be performed. In section 3. we follow this by a description of the validation approach. The majority of the section focuses on how images of a simulated comet are generated to serve as input to the operational process. Finally, section 4. describes the validation methods employed and the thought processes that led to their inception. Two major approaches are identified. One concentrates on consistency checks of the shape model, the other on using the available images and comparing them to the shape model. We conclude with a short outlook on how the validation may be applied when Rosetta arrives at the comet 67P/Churyumov-Gerasimenko.

Notation. In the following, we use certain conventions and notations, which we describe here. We commonly abbreviate spacecraft as “S/C” and shall refer to Rosetta simply as “spacecraft” or “S/C.” Similarly, 67P/Churyumov-Gerasimenko is abbreviated to “CG” and referred to simply as “the comet.”

With \mathbb{N} , \mathbb{R} , and \mathbb{H} , we denote natural numbers, real numbers, and quaternions, respectively. For vectors $x \in \mathbb{R}^n$ no special care is taken of whether they are row or column vectors. This can always be inferred from context. The Euclidean norm of a vector $x \in \mathbb{R}^n$ is denoted by $\|x\|$. The scalar product of two vectors $x, y \in \mathbb{R}^n$ is denoted by $\langle x, y \rangle$.

When referring to the “inertial frame,” we mean Earth mean equator at J2000. Solar system object states are given with respect to the solar system barycenter. Where applicable, the JPL DE405 ephemerides (see [1]) are used. The position of an object is denoted by

$$r^{\text{object name}} : \mathbb{R} \mapsto \mathbb{R}^3,$$

e.g. r^{Sun} , $r^{S/C}$, or r^{CG} .

An attitude is described by a quaternion $q \in \mathbb{H}$ of unit length. By convention, the scalar is given last, i.e. the basis of the quaternion space \mathbb{H} is $(i, j, k, 1)$.

Let a vector $x \in \mathbb{R}^3$ be given, then it is rotated by the quaternion $q \in \mathbb{H}$ by the following parameterized function, $\mathbf{R} : \mathbb{H} \times \mathbb{R}^3 \mapsto \mathbb{R}^3$

$$\mathbf{R}(q; x) := \bar{q} \cdot (x, 0) \cdot q \Big|_{1,2,3} \quad (1)$$

where the operator $|_{1,2,3}$ selects the first three components of the result. The multiplication used in Eq. 1 is the quaternion multiplication. The line over q indicates the quaternion conjugate. Since we only use unit quaternions, this is equal to the inverse, $\bar{q} = q^{-1}$.

In general, a quaternion will be imbued with a suffix which denotes from which frame to which frame it transforms, e.g., $q_{S/C \rightarrow \text{inert}}$. The inverse transformation is then implicitly given by $q_{\text{inert} \rightarrow S/C} = \bar{q}_{S/C \rightarrow \text{inert}}$.

2. Operational Approach

Since one of the main goals of the validation approach is for it to also be usable in real operations, it is important to understand the process that yields a shape model of the comet. This will be discussed in the current section.

In general, the shape estimation is part of the operational process. As such, the shape model is iteratively refined from iteration to iteration.

We assume that in each iteration of the operational process, a nominal relative trajectory of the spacecraft with respect to the comet is selected. A spacecraft attitude is chosen, nominally pointing some camera boresight at the comet. Note that no closed-loop tracking of the comet is performed. Therefore, any such attitude is preselected and is necessarily based on the nominal trajectory.

Then images are acquired at a suitable frequency. Which images to acquire at which frequency is discussed elsewhere [2, 3]. Here, we assume that a suitable number of images is available in any iteration.

Procedure 1. *The following steps are performed iteratively to construct an estimated comet shape.*

1. *Select landmarks as (x,y) pixel coordinates in pairs of images of the comet for a time series of images.*
2. *Translate all pixel coordinates to inertial directions, using knowledge of the spacecraft attitude and camera properties.*
3. *Estimate relative state of spacecraft and comet as a function of time. In particular, the following is estimated:*
 - *relative position and velocity of the comet/spacecraft system,*
 - *comet attitude,*
 - *landmarks in comet-fixed frame.*
4. *Using the now-known relative state of spacecraft and comet, estimate or refine the shape of the comet.*

How to perform the steps described in Procedure 1 is not subject of this paper. See [4] for step 1, [2, 3] for steps 1 and 4, and [5, 6] for step 3. Step 2 above involves combining spacecraft attitude, alignment of the camera on the spacecraft, and camera distortion models, see [7, 8].

We have several cameras on board Rosetta at our disposal. There are two dedicated navigation cameras, NAVCAM-A and NAVCAM-B. Additionally, we may also use the cameras of the OSIRIS payload (see [9]), NAC (narrow angle camera) for high-resolution, narrow-angle images, and WAC (wide angle camera) with a larger field of view. Note that NAVCAM-B is of the same type as NAVCAM-A, and is included on the spacecraft for redundancy purposes. All of these cameras have roughly the same boresight direction in spacecraft frame, so it is possible to interleave images from different sources without having to slew the spacecraft significantly, if at all.

3. Validation Approach

As part of our preparation for operations near the comet, we have gone through several validation steps. At the lowest level, each step of Procedure 1 is individually validated. Some dummy data is provided as input, is processed, and interface files are created. These are checked for syntax, completeness, and content. Validating the content can vary greatly in difficulty. Step 2, for example, is fairly simply to validate, since it “only” consists of translating pixel coordinates to inertial directions.

As a more involved example, we have validation of step 3, which requires a consistent set of landmarks, comet and spacecraft kinematics, and the ability to simulate all input data in the format that the real processes would yield. Note that this step does not require actual images of the (simulated) comet, but only inertial directions of landmarks.

After each step has been successfully validated individually, everything is combined in so-called flight dynamics system tests, which have a system-level focus, and involve all experts in the Flight Dynamics Team. These system tests try to mimic real operations as closely as possible, and involve many more steps than detailed in Procedure 1.

In order to perform this run-down of real operations, inputs that are usually obtained from external sources have to be provided by some other means. First off, a realistic comet attitude is selected (we know, e.g., that the rotational period is approximately 12 hours, see [10]). On top of a nominal rotation, nutation is added. For our tests we opted not to include any dynamic effects on the comet attitude (due to, e.g., jets which apply a torque to the comet).

Then, compatible comet and spacecraft states as functions of time are chosen. This relative state is similar but not identical to our planned nominal kinematics.

3.1. Image Generation

At this stage in real operations, we would wait for the spacecraft to fly the commanded attitude, and acquire any planned images from the various navigation cameras. After the retrieval of these images from telemetry, the steps detailed in Procedure 1 are kicked off.

In the absence of real images, we have to simulate them. For this purpose, we need a 3D model with appropriate detail, and a way to render it realistically, taking camera alignment and characteristics into account.

As mentioned in section 2., the spacecraft attitude is selected and commanded based on the nominal state. Thus, it may occur that images only contain part of the comet, or even not at all. The state and shape estimation process must be robust against this kind of behavior.

3.1.1. The Renderer

For our renderer, we chose the freely available Persistence of Vision Raytracer (POV-Ray, see [11]), using the `mesh_camera` camera type (see [12]) to emulate the distortion of the cameras. For this purpose, the distortion models for the various cameras (see [7, 8]) are translated to the format required by the `mesh_camera` camera type.

Any valid pixel (x, y) has a direction d_{xy} in camera frame associated with it. An equilateral triangle t_{xy} with surface normal equal to d_{xy} is constructed. These triangles can be used to define the `mesh_camera` used in [12].

Note that we perform this construction only for integer x and y . In particular, this means that we cannot anti-alias images, as can be seen by the graininess evident in several of the images shown in this paper.

No attempt was made to simulate exposure times, which would theoretically yield darker or lighter (or even over-exposed) images. However, integration times will be chosen such that we have “good” exposure, yielding usable images. Should it occur in real operations that an image is not usable, it will be discarded, effectively simply reducing the set of available images.

3.1.2. Scene Setup

Let the spacecraft attitude be given by a function of time, $q_{\text{inert} \rightarrow \text{S/C}} : \mathbb{R} \mapsto \mathbb{H}$, as described above. Furthermore, let the camera alignment be given by the fixed unit quaternion $q_{\text{S/C} \rightarrow c} \in \mathbb{H}$, where $c \in \{\text{NAVCAM-A, NAVCAM-B, NAC, WAC}\}$ denotes any of the cameras we use. Spacecraft and comet position are given according to the selected relative state described at the beginning of this section and denoted by $r^{\text{S/C}}$ and r^{CG} respectively.

Note that we assume the spacecraft position to be equal to the camera position. Though in general the camera focal point is not placed at the center of mass of the spacecraft, this approximation is more than sufficient given the scales involved: spacecraft size is on the order of 2m, distance to the comet on the order of 10km to 100km.

In our implementation, we place the camera (and hence the spacecraft) at the origin $(0, 0, 0)$ and set up the scene to be in camera frame.

A scene is then defined for a specific instance in time, denoted by t_j , where $j = 1, \dots, n$. Additionally, a scene is defined for one of the possible cameras only, which depends on the image taken, i.e. $c = c(j)$.

A sole light source is placed at $l_{\text{scene},j}$, given by

$$l_{\text{scene},j} := \mathbf{R}(q_{\text{inert} \rightarrow \text{S/C}}(t_j) \cdot q_{\text{S/C} \rightarrow c(j)}; r^{\text{Sun}}(t_j) - r^{\text{S/C}}(t_j)). \quad (2)$$

For simplicity's sake, the light source is chosen as a point light source. At 3AU, the approximate distance to the Sun when the lander is delivered to the comet surface, the angular diameter of the Sun is approximately 0.18° so this is a reasonable approximation.

Similar to the Sun transformation, the comet position $r_{\text{scene},j}^{\text{CG}}$ is given by

$$r_{\text{scene},j}^{\text{CG}} := \mathbf{R}(q_{\text{inert} \rightarrow \text{S/C}}(t_j) \cdot q_{\text{S/C} \rightarrow c(j)}; r^{\text{CG}}(t_j) - r^{\text{S/C}}(t_j)).$$

Finally, if the comet attitude is given by $q_{\text{inert} \rightarrow \text{CG}} : \mathbb{R} \mapsto \mathbb{H}$, its attitude in the scene is then given by

$$q_{\text{CG} \rightarrow \text{scene},j} := q_{\text{CG} \rightarrow \text{inert}} \cdot q_{\text{inert} \rightarrow \text{S/C}}(t_j) \cdot q_{\text{S/C} \rightarrow c(j)}.$$

3.1.3. Shape Model Selection

The final piece missing to fully define the scene and render an image of the comet is a representative model of the comet.

Ideally, a model should exhibit different surface characteristics over different parts of its surface. This way, landmarks should be easier to identify and recognize in a series of images.

Rather than creating a model from scratch, we went looking on the Internet and found a suitable model, see [13].



Figure 1: The 3D comet model [13], scaled by 0.6, and rendered on NAVCAM-A with a 50%/100% bozo texture at a distance of approximately 104km. The image is cropped to show only the comet. No post-processing noise was added.

from image to image. Rather than being a hindrance, this furthered the validation, ensuring robustness against seemingly incongruous comet behavior.

As with the center of mass, no attempt was made to align the principal inertia axes with the model’s geometric axes.

3.2. Addition of Low-Level Detail

Our first tests were performed with images as seen in Fig. 1. These initial tests placed the spacecraft at distances between 95km and 120km from the comet. As can be seen in Fig. 1, the level of detail was conducive to these tests, giving a sufficient amount of identifiable landmarks.

As the tests progressed and we “approached” the comet to distances down to 10km, it became obvious that the level of detail was not sufficient at these close distances, making it very hard to find any recognizable landmarks.

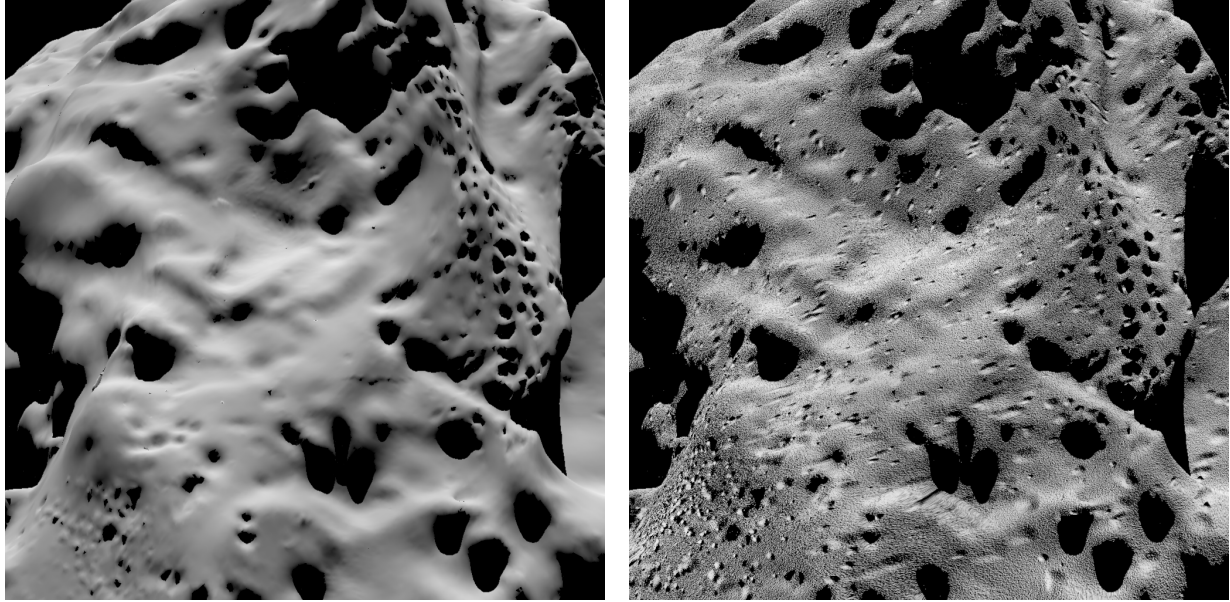
It was therefore decided to add high-resolution detail to the model in the form of a bump map. A bump map was created containing 10,000 randomly placed, procedurally-generated craters of varying sizes, shapes, and depths. Additionally, some low-amplitude noise was overlaid over the bump map, to give the comet a more “realistic” appearance. The effect can be seen in the side-by-side comparison in Fig. 2.

Though the model given by [13] is supplied with a texture map, we chose not to use this. Given that all our cameras are monochrome cameras, we chose to render the comet with a POV-Ray “bozo” texture (see [11]) with gray levels between 50% and 100%. Rather than rendering in pure white, the bozo texture provides some local random (but reproducible) variation.

The finish was chosen to be 100% pure diffuse reflection, i.e. no specular reflection and no ambient light.

The radius of the comet is estimated to be on the order of 2km, see, e.g., [10]. The model given by [13] is somewhat larger (assuming that the units used are km), so a scale of 0.6 was applied to all vertex coordinates.

No attempt was made to align the center of mass with the geometric barycenter of the model. In fact, the center of mass is off by a distance of approximately 1.2km, giving the comet a seemingly lop-sided appearance, when seen rotating



(a) Bozo texture only.

(b) Crater bump map added.

Figure 2: The exact same scene, rendered twice for NAVCAM-A from a distance of approximately 24km.

We also regenerated images used in earlier tests, where we had greater distances between spacecraft and comet. The scene of Fig. 1 was rendered again, this time using the bump map. The result can be seen in Fig. 3.

At this distance, it can be seen that landmarks that were previously clearly visible in the bozo-textured model have all but vanished in the “graininess” of the bump-mapped model. This is a direct consequence of the way the camera distortion is modeled (see section 3.1.1.), which does not allow anti-aliasing.

Rather than choose a different bump map (with, say, less noise added), this serves as an additional way to prove the robustness of the approach.

3.2.1. Image Post-Processing

After generation of an image, we perform some post-processing as described in the following.

First off, noise is added to the image. For this purpose, we make use of the commonly available ImageMagick [14] toolset. We generate three types of noise images:

- variation of the original image by multiplying pixel values with random noise levels in the range $[0.9, 1]$,
- a noise floor in the form of random pixel values in the range $[0, 0.05]$,
- specks (by thresholding random noise at 0.98) to simulate hot pixels and single event upsets.

Noise floor and specks are only applied to levels of the rendered image with gray levels below 0.1. Everything is composited to yield the final image.

Note that we do not add coma or stars to the images. With the integration times involved, we do not expect to see either on the real images.

Images are produced by POV-Ray and ImageMagick in .png format. Within Flight Dynamics at ESOC, we store images in .fits image files, see [15]. Therefore, as a final step, the generated .png image is converted to .fits format, rotating and mirroring images in the convention used for a particular camera. Also, header data is added to the final .fits image. In particular, this meta data includes the camera used and the epoch at which the image was taken.

4. Validation of the Shape Model

After the images are generated, they serve as input to Procedure 1. They are therefore made available to the experts within the Flight Dynamics Team who process them as described in section 2.

In order to robust our system against possible non-availability of images from the OSIRIS cameras WAC and NAC, it was decided to run the validation campaign with one of the navigation cameras only. Therefore, all examples given in the following make use of NAVCAM-A.

The validation of the shape model (step 4 of Procedure 1) is split into several self-contained checks which vary in nature. For each of these checks, software was developed in-house by the Flight Dynamics Team. Furthermore, this software was developed independently of any software used to produce the shape model or any intermediate product. This holds true also for areas where sharing of functionalities would have been possible, simply to keep the validation independent of the shape estimation proper. This way errors in the common functionality are spotted more easily.

As mentioned in the introduction, we focused on validation methods which can be reused during operations. In particular, this means that we largely ignore any information that was used to generate the images in section 3. Just like the operational flight dynamics system, we “pretend” that the images come from the spacecraft.

In the following sections, we describe each individual step we perform in the validation process.

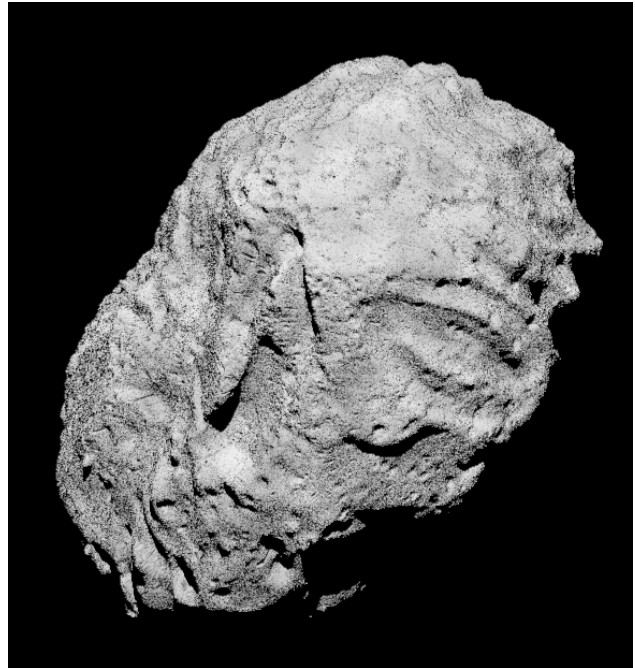


Figure 3: The same scene as in Fig. 1, but using the crater bump map.

4.1. Syntax Checks

First off, it is checked that the file containing the shape model conforms to the syntax agreed between Flight Dynamics and the external users of this file, see [16]. This is checked each time the file is read and parsed in any of the checks described in the following.

4.2. Consistency Checks

The shape of the comet is described as a collection of vertices given as 3D points, and triangular faces defined using these vertices. Note that the vertices are given in the comet-fixed reference frame estimated in step 3 of Procedure 1.

Without any knowledge of the images, or what a rendering of the shape model looks like, we can already check whether the shape is consistent with respect to itself, or with other output of Procedure 1. This is described in the following.

4.2.1. Internal Consistency

By the nature of its representation, the shape model describes the *surface* of the estimated shape. Therefore, it must be ensured that this surface does not contain any “holes” which would yield a non-physical object. For this purpose, we check that each edge is contained in exactly two triangular faces.

The algorithm starts at an arbitrary vertex, finds neighboring vertices (in the sense that they form the two ends of an edge), and checks that the edges defined in this way are contained in exactly two faces. In case any of the neighboring vertices have not been visited by the algorithm yet, the same check is iterated for each of these neighbors.

By design, the algorithm “walks” over the whole shape model, or at least parts that are connected to each other. Since we can reasonably assume that the comet is a single object, we finally check that there are no orphaned vertices not visited by the algorithm above.

Then, the so-called Euler characteristic χ for polyhedra is computed as follows (see, e.g., [17]):

$$\chi = n_v - n_e + n_f, \quad (3)$$

where n_v is the number of vertices, n_e the number of edges, and n_f the number of faces. Any polyhedron, which can be continuously mapped to a sphere, has the Euler characteristic $\chi = 2$. Since we expect the comet not to contain any holes (and hence be topologically equivalent to a torus or multi-torus), we check that this number is always $\chi = 2$.

In our tests, a Euler characteristic of $\chi \geq 3$, e.g., indicated that we had one or more isolated vertices in the shape model which was not part of any face. A Euler characteristic of $\chi = 0$ (the characteristic of a torus) indicated a hole in the model. In our test, it turned out to be a non-physical “tunnel” etched out of the shape model.

Finally, we obtain the volume and the geometric center of the model. Though these quantities cannot be directly used for any validation purposes, they are interesting to note, and provide a sense of the “plausibility” of the model. See also section 4.2.3. below.

4.2.2. Landmark-Shape Model Consistency

Recall that in step 3 of Procedure 1 we in particular estimate landmark positions in comet-fixed frame. This information can be used to check compatibility of landmarks and shape model.

As an initial check, the distance of each landmark to the shape model surface is computed. This involves the computation of the distance between the landmark and each triangle which makes up a face of the comet surface model. The smallest such distance is the distance between landmark and shape model. If $l_i \in \mathbb{R}^3$ is the landmark i in comet frame, and \mathcal{F} denotes the set of faces of the comet (also in comet frame), the distance d_i between landmark and comet surface can be expressed as

$$d_i := \min_{f \in \mathcal{F}} \left\{ \left\| l_i - \prod_f l_i \right\| \right\}, \quad (4)$$

where $\prod_f l_i$ denotes the projection of a vector $l_i \in \mathbb{R}^3$ onto the face $f \in \mathcal{F}$.

The resulting distance between comet and landmark is negated if the landmark is contained in the shape model to obtain a signed difference. Whether a landmark is inside the shape model or not is decided by counting the number of intersections a ray cast from the landmark to infinity makes. If the number of intersections is odd, the landmark is inside the shape model. Note that this is one use of the ray-tracing functionality discussed in section 4.3. below.

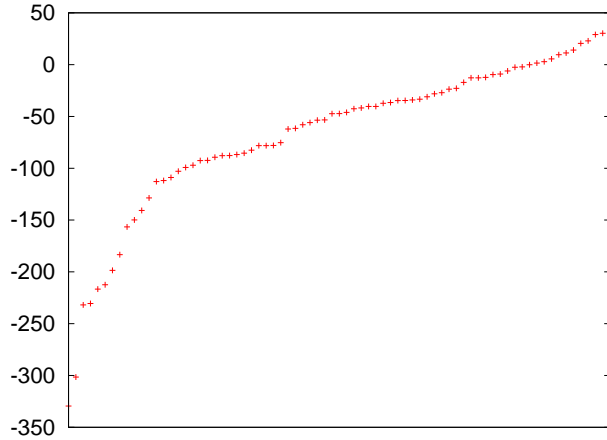


Figure 4: Landmark distance in [m] with respect to the shape model surface, ordered by distance. Negative numbers imply a landmark below the shape model surface.

In Fig. 4, we plot the distance of the landmark above or below the shape model surface, ordered by distance. As is evident from the graph, there is a systematic bias of the landmarks to be below the shape model surface. This can be explained by the way in which the surface is generated, see section 4.2.3.

Note that an inconsistency between landmark and shape model does not automatically imply a problem with the shape model. Something may have occurred in steps 1 through 3 of Procedure 1 to cause the inconsistency.

Therefore, in order to gain confidence in the landmarks, we perform a validation step aimed solely at checking landmark plausibility.

First, for any landmark $l_i \in \mathbb{R}^3$, we compute the vertex of the shape model which is the closest to

that landmark:

$$v_i^* := \arg \min_{v \in \mathcal{V}} \|l_i - v\|, \quad (5)$$

where \mathcal{V} is the set of all vertices of the shape model. Then, the “average” normal of that vertex is computed, i.e.

$$n_i^* := \frac{1}{|\mathcal{F}^{v_i^*}|} \sum_{f \in \mathcal{F}^{v_i^*}} n^f, \quad (6)$$

where $\mathcal{F}^{v_i^*}$ is the set of all faces which contain the vertex v_i^* , and $n^f \in \mathbb{R}^3$ is the normal of the face $f \in \mathcal{F}^{v_i^*}$.

For an image taken at time t_j , we now take the landmarks l_i selected in that image (an output of step 1 of Procedure 1), and compute the inertial direction of the closest average vertex normal of the shape model:

$$n_{\text{inert},i}^* := \mathbf{R}(q_{\text{CG,est} \rightarrow \text{inert}}(t_j); n_i^*) \quad \forall i, \quad (7)$$

where $q_{\text{CG,est} \rightarrow \text{inert}}$ is the comet attitude estimated in step 3 of Procedure 1.

Firstly, we check whether this landmark is illuminated by the Sun. For this purpose, we check that the following inequality holds true:

$$\arccos \left(\left\langle \frac{r^{\text{Sun}}(t_j) - r^{\text{CG,est}}(t_j)}{\|r^{\text{Sun}}(t_j) - r^{\text{CG,est}}(t_j)\|}, n_{\text{inert},i}^* \right\rangle \right) \stackrel{?}{\leq} 90^\circ + \delta^{\text{Sun}}. \quad (8)$$

Note that we add a margin of $\delta^{\text{Sun}} = 5^\circ$ in order to compensate for the fact that the comet shape will not match the real comet shape exactly. This number has served us well in our tests but may have to be fine-tuned once we arrive at the real comet.

Similar to Eq. 8, we can check whether a landmark is visible to the camera, and not on the “backside” as seen from the spacecraft:

$$\arccos \left(\left\langle \frac{r^{\text{S/C,est}}(t_j) - r^{\text{CG,est}}(t_j)}{\|r^{\text{S/C,est}}(t_j) - r^{\text{CG,est}}(t_j)\|}, n_{\text{inert},i}^* \right\rangle \right) \stackrel{?}{\leq} 90^\circ + \delta^{\text{camera}}. \quad (9)$$

Again, a margin of $\delta^{\text{camera}} = 7^\circ$ is added to counteract that the shape model cannot match the real comet shape exactly. Note that it is currently not checked whether the landmark falls in the field of view of the camera used. This may be added in the future.

Should the landmark/shape model surface consistency be called into doubt but the checks in Eq. 8 and Eq. 9 be successful, then the inconsistency cannot be unambiguously attributed to a problem with the shape model. In this case all steps of Procedure 1 have to be revisited. The exact cause has to be determined on a case-by-case basis.

4.2.3. Iterative Inclusion

As noted in section 2., the shape estimation is an iterative process. Procedure 1 is performed repeatedly, the shape updated in each of these iterations.

To initialize the shape for the first iteration, it is chosen to be a cube of sufficient size built up of cubes with an edge length of 150m. In each iteration, a subset of these cubelets are removed in order to fit the images according to a certain metric. Afterwards, a smoothing step is performed in order to remove the blockiness inherent in such a process.

Thus, the shape model generated in an iteration k has to be contained within the shape model generated in iteration $k - 1$ up to a certain error introduced by the smoothing step. In order to verify this, we check that each vertex of the new shape model is contained within the old shape model. To account for numerical inaccuracies as well as the behavior of the shape-smoothing for locally non-convex areas, we multiply the new vertex' coordinates by 0.99. On a comet with average radius 2km, this is equivalent to moving the vertex "inward" by 20m.

Note that we may be introducing problems rather than removing them in case there is a cave in the shape model. However, in our tests this was not a problem.

Comparison of Shape Model with Source Model. Note that by the above description of the shape model generation, the estimated shape model should theoretically always envelope the source model described in section 3.1.3. Therefore, the method described in section 4.2.3. can be applied to the source and shape models.

In general, the comet frame used to generate the images as input to the system test will not match the comet frame arbitrarily chosen in step 3 of Procedure 1. Therefore, one of the shapes first has to be transformed into the same frame as the other shape. Since we also have to scale the source model by a factor of 0.6 as mentioned in section 3.1.3., we prefer to transform the source model.

As in section 3.1.2., let $q_{\text{inert} \rightarrow \text{CG}} : \mathbb{R} \mapsto \mathbb{H}$ denote the attitude of the comet using the frame we chose to set up the scene. Furthermore, let $q_{\text{inert} \rightarrow \text{CG,est}} : \mathbb{R} \mapsto \mathbb{H}$ denote the attitude estimated in step 3 of Procedure 1. Then, for an arbitrary epoch t^* , we transform each vertex $v \in \mathcal{V}^{\text{source}}$ of our source model into the the estimated comet frame, performing the scaling in the same step:

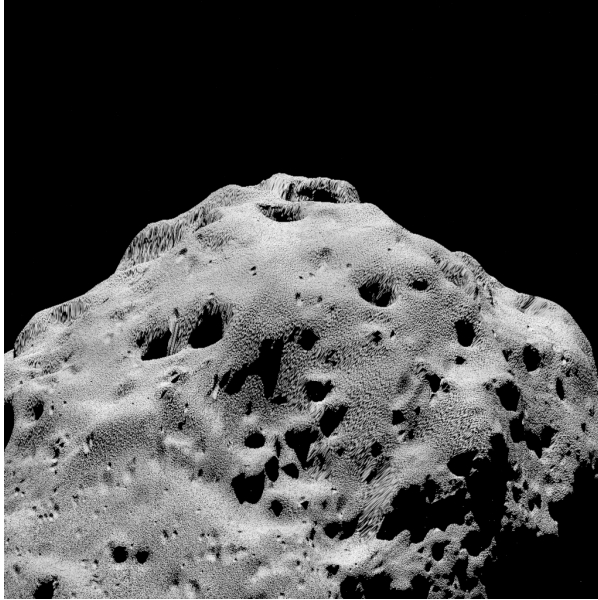
$$\mathcal{W} := \{w \in \mathbb{R}^3 \mid w = 0.6\mathbf{R}(q_{\text{CG} \rightarrow \text{inert}}(t^*) \cdot q_{\text{inert} \rightarrow \text{CG,est}}(t^*); v) \forall v \in \mathcal{V}^{\text{source}}\}. \quad (10)$$

We found that that the source comet model exceeded the estimated shape by a maximum of 106m. Half the spatial diagonal of a 150m-cube is $\frac{1}{2}150\sqrt{3} \approx 130\text{m}$. The observed error is therefore within limits.

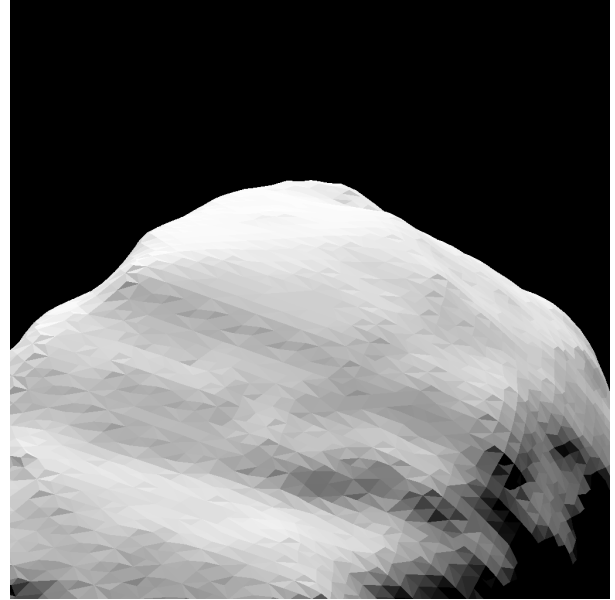
This step, though invaluable during the preparation phase, can obviously not be performed during real operations.

4.3. Real vs Ray-Traced Image Comparisons

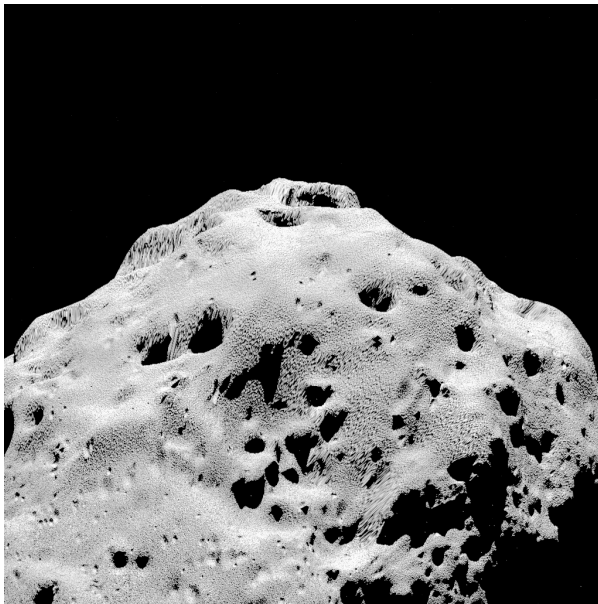
In the previous section we have focused on formal consistency checks of the shape model. Now we want to verify compatibility of the shape model with the images used to estimate the shape model.



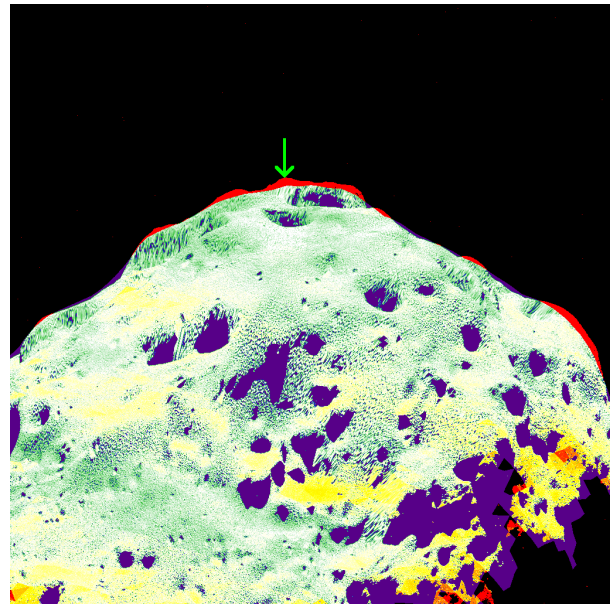
(a) The “real” image.



(b) The ray-traced shape model.



(c) The “real” image, its histogram modified to match the ray-traced image 5b.



(d) A false-color superimposition of images 5b and 5c. The green arrow designates the worst overhang of the “real” comet with respect to the shape model.

Figure 5: A comparison of “real” and shape model images.

For this purpose, ray-tracing functionality was developed and put to various uses as detailed in the following. This ray-tracer is designed to specifically highlight those aspects that are important to the validation process. In particular these include the incorporation of the camera distortion model. Shadow casting is considered, but not reflection (specular or diffuse).

The most obvious use of the ray-tracer is to render an image of the shape model for each of the real images and compare these pairs of images.

Note that POV-Ray [11] could have been reused for this purpose. However, we did not want to rely on third-party software for this operationally critical aspect of the validation process. Also, the ray-tracer functionality can be used for additional checks as already mentioned in section 4.2.2. and further described in section 4.3.3. below.

In general, we cannot expect real and ray-traced image to exhibit the same color space characteristics. Therefore, before comparing the images, histogram matching is performed, see, e.g., [18]. This method was adapted to the particular case at hand. Since in the ray-traced image, the background is a true black, we first push the real image's background noise to black. Since the comet will stand out fairly brightly against the background, this can be accomplished fairly easily by cutting off all color values below a threshold. For the histogram matching, we subsequently ignore black in the matching, since different comet sizes in the images would otherwise push dark areas of the larger comet to be black as well. The remainder of the real image's histogram is matched to the ray-traced image, usually aligning the colors of the comet fairly well.

We considered aligning the images using phase correlation, see [19], for both positional and rotational matching. The idea was that this would counteract thermal warping effects which can introduce a variation in the relative alignment of star trackers and camera of a few millidegrees. However, we found that other differences in the models greatly outweighed these effects, and have opted not to perform this alignment process.

After ray-tracing the shape model in this fashion, a way has to be found to compare it to the real image.

4.3.1. Manual Image Comparison

The most obvious comparison method is manual comparison of the images. Rather than looking at the images side-by-side, the two images are superimposed using false colors to convey meaning, see Fig. 5d. Violet indicates a larger or better-lit shape model, and red indicates a larger real comet.

We initially found the false-color superimposition to be hard to interpret and considered different color maps as well as interactive transparency variation (using third-party software). However, after some practice, the false-color scheme used in Fig. 5d becomes easier to read.

In section 4.2.3. we claimed that the estimated shape will always be larger than the real comet. Yet, in Fig. 5d there are red areas indicating a larger real comet. This can be partly attributed to statistical errors in the estimation process, as well as to the shape model's smoothing process, which may err on the order of $\frac{1}{2}150\sqrt{3}\text{m}$. See also section 4.3.3. below.

The violet areas in Fig. 5d are those where the real comet exhibits small craters and pockmarks, which are not easily carved out by the shape model estimation method, in particular with a granularity of 150m-a-side cubes.

4.3.2. Automated Metric-Based Image Comparison

Manual comparison of images is useful to get a feeling for the differences we may expect. However, comparing several hundred images manually may become tedious.

Therefore, we use a method called *Mean Structural Similarity*, or MSSIM, to compute a metric which is an indication of the similarity of two images, see [20].

Using the method as-is does not scale well for images which have large black portions (as we expect at greater distances to the comet). Imagine one image with a few non-black pixels in one area, the other with a few non-black pixels in another area. The images will be considered “similar” since they are both mostly black.

Therefore, we use a simple form of the weighted average modification suggested in [20], by weighting windows which are completely black with 0, all others with 1:

$$\text{WMSSIM}(\mathbf{X}, \mathbf{Y}) := \frac{1}{\mu} \sum_{j=1}^M \delta_{\mathbf{x}_j \mathbf{y}_j} \text{SSIM}(\mathbf{x}_j, \mathbf{y}_j), \quad \text{with} \quad (11)$$

$$\delta_{\mathbf{x}_j \mathbf{y}_j} := \begin{cases} 0 & \text{if } \mathbf{x}_j \text{ and } \mathbf{y}_j \text{ are fully black,} \\ 1 & \text{otherwise,} \end{cases} \quad \text{and} \quad (12)$$

$$\mu := \sum_{j=1}^M \delta_{\mathbf{x}_j \mathbf{y}_j}. \quad (13)$$

Here, the structural similarity index $\text{SSIM}(\mathbf{x}_j, \mathbf{y}_j)$ is given as in [20]. Note that M is the number of windows, \mathbf{X}, \mathbf{Y} are the image maps, and $\mathbf{x}_j \subset \mathbf{X}, \mathbf{y}_j \subset \mathbf{Y}$ are moving 8×8 pixel windows of these image maps.

We chose the weighting power factors of the SSIM algorithm to be $\alpha = 1$ and $\beta = \gamma = 0$, i.e. the comparison metric is reduced to a luminance comparison. For details please refer to [20].

For Fig. 5, the WMSSIM value is 0.849971, the MSSIM 0.913076.

We have run an analysis of the MSSIM over 168 images and have found that the WMSSIM succeeds fairly well in identifying potential problems. See Fig. 6 for some examples. These examples also demonstrate that the unweighted MSSIM is not very useful in our case. Since the image in Fig. 6a is mostly black, the MSSIM is very high, while the WMSSIM gives a more realistic value for the agreement between the images.

In theory, values of the WMSSIM “close” to 1 indicate that the images are similar, values close to 0 indicate that they are different. We have found that values > 0.8 indicate very similar images, and that values < 0.6 potentially indicate some sort of problem. Values in-between may have to be looked at manually.

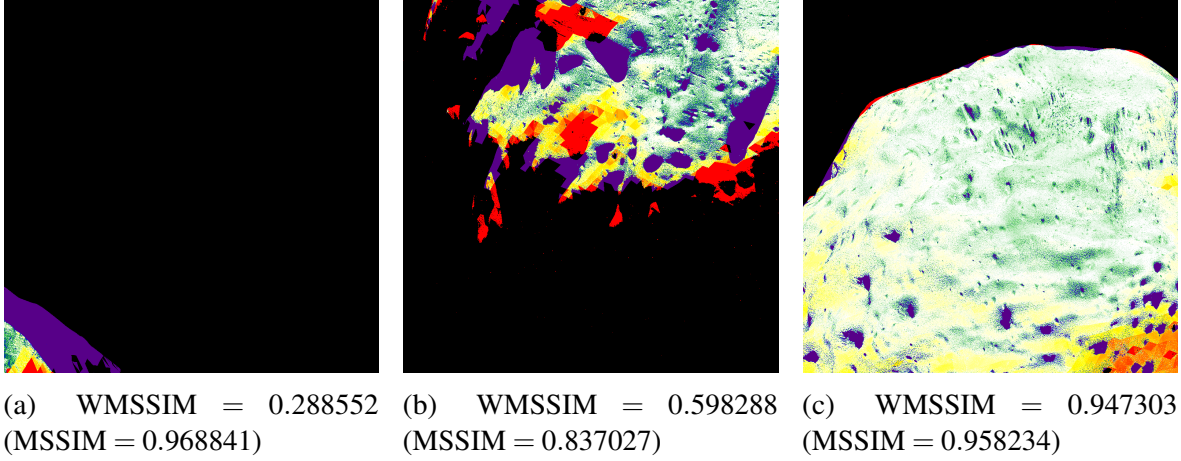


Figure 6: Some false-color comparison images for different values of the WMSSIM.

This process cuts down the number of images that actually have to be manually compared to a fraction of the total number of images. Of the analyzed images, 38% had an WMSSIM above 0.8, and only 20% had an WMSSIM below 0.6.

4.3.3. Shape vs Image Distance Checks

As mentioned in the introduction to this section, the ray-tracer can also be put to other uses. It is used to quantify the areas where the real comet is larger than the shape model (the red areas in Fig. 5d).

For any pixel (x, y) in the real image whose value is above a certain threshold (and hence probably part of the comet), a ray $r : \mathbb{R} \mapsto \mathbb{R}^3$ is cast and the closest distance $d_{x,y}$ of this ray to the shape model is computed. This is done in a fashion similar to Eq. 4, with the additional free parameter of the point on the ray:

$$d_{x,y} = \min_{f \in \mathcal{F}, \tau \in \mathbb{R}} \left\{ \left\| r(\tau) - \prod_f r(\tau) \right\| \right\}, \quad (14)$$

where \mathcal{F} again denotes the set of faces in our shape model.

Note that the resulting distance is a lower bound for the distance between estimated shape and the real comet, since we are only considering the projection of both estimated shape and real comet onto the camera focal plane.

As mentioned in section 3.2.1., we expect hot pixels and single event upsets in the images. These will in general cause false triggering of the pixel/shape distance check. However, such outliers generally are isolated pixels. Therefore only clusters of two or more adjacent pixels with a distance of more than $\frac{1}{2}150\sqrt{3}\text{m}$ are considered to violate the distance check.

In the case of Fig. 5, the worst ray/shape distance according to Eq. 14 is just shy of 40m, well below the $\frac{1}{2}150\sqrt{3} \approx 130\text{m}$ we posited earlier. This worst offending pixel is pointed out in Fig. 5d by a green arrow.

In Fig. 6b, which has an WMSSIM of just under 0.6, the maximum ray/shape model distance according to Eq. 14 is 19m. The many (partly large) red areas can therefore be attributed to lighting of the shape model. In this particular case, the Sun incidence angle is fairly shallow (local dusk or dawn), making the illumination very sensitive to the surface properties.

We computed the maximum ray/shape distance for the same set of 168 images used in section 4.3.2. for the WMSSIM analysis, and found this distance to usually be between 20m and 40m. The worst observed distance was 62m.

It is of interest to note that these errors are of the same order as the systematic bias of the landmark altitude below the shape surface, see Fig. 4. This indicates that the landmark position estimation is of better quality than can be initially assumed from the results presented in section 4.2.2.

The shape vs image distance checks described above have proved the most robust verification method. Though it can only augment the other methods described in this paper, it is often the best way to quantify the error that we see.

5. Conclusion

We have presented a method to validate the comet shape reconstruction. This was performed by the Flight Dynamics Team at ESOC in preparation of the anticipated arrival at the comet 67P/Churyumov-Gerasimenko.

Care was taken to simulate the expected situation around the comet as realistically as necessary. Naturally, no knowledge exists of the physical shape or surface properties of the real comet. The model and surface texturing chosen to simulate the images can therefore only be representative at best. Furthermore, the surface of the simulated comet model was imbued with sufficient detail in order to facilitate the whole estimation process presented in Procedure 1. The ray-traced images were not anti-aliased due to our implementation of the camera distortion models. Neither coma nor stars were added to the final image, since we do not expect to see them on the real images either.

Only once we arrive at 67P/Churyumov-Gerasimenko, will we know how representative the simulated images are.

The whole campaign from first images up to lander delivery was exercised with mean expected orbital performance. External forces (from the comet coma and solar radiation pressure) were simulated to have what we expect to be mean levels and variation. In particular the coma model is subject to similar uncertainties as the shape and surface mentioned above. Dispersion errors of the trajectory correction maneuvers performed with Rosetta were assumed to have the same magnitude as we have come to expect from past maneuvers.

Having assumed a certain trajectory around the comet, no further errors were introduced in the attitude of the navigation cameras. Such errors may stem from inaccuracies in the spacecraft attitude control, transient effects due to, e.g., solar array flexible modes, or thermal warping modifying the relative alignment between star trackers and navigation cameras.

Similarly, though imbuing the simulated comet with nutation, no dynamic effects (due to, e.g., jets applying torque to the comet) were simulated.

Given the described simulated scenario, it was found that the shape model accuracy is well within the design accuracy of the shape reconstruction process. Both landmark/shape consistency checks (section 4.2.2.) and shape/image distance checks (section 4.3.3.) have shown the error to be on the order of 50m. The shape reconstruction process uses cubes with 150m-sides, so the accuracy cannot be expected to be more than half the spatial diagonal.

Thus given what believe to be a sufficiently accurate scenario, the shape reconstruction method has been successfully validated. The Flight Dynamics Team at ESOC in Darmstadt is therefore in a position to successfully support Rosetta upon its arrival at the comet 67P/Churyumov-Gerasimenko in August 2014.

Since the validation methods presented here are largely independent of any knowledge of the real comet, they will be applied during real near-comet operations. Any limitations of the simulated scenario and images listed above will be mitigated by this fact. This is of considerable importance since we will only have three months between arrival at the comet and lander delivery. By this time the availability of an accurate shape model is essential for successfully delivering the Philae lander to the surface of the comet 67P/Churyumov-Gerasimenko.

6. References

- [1] “JPL Ephemerides.” <http://ssd.jpl.nasa.gov/?ephemerides>. Accessed: 2014-04-08.
- [2] Lauer, M., Kielbassa, S., and Pardo de Santayana, R. “Optical Measurements for Attitude Control and Shape Reconstruction at the Rosetta Flyby of Asteroid Lutetia.” Proceedings of the 23rd International Symposium on Space Flight Dynamics — 23rd ISSFD. Pasadena, CA, USA, 2013.
- [3] Pardo de Santayana, R., Lauer, M., Muñoz, P., and Castellini, F. “Surface Characterization and Optical Navigation at the Rosetta Flyby of Asteroid Lutetia.” Proceedings of the 24th International Symposium on Space Flight Dynamics — 24th ISSFD. Laurel, MD, USA, 2014.
- [4] Wokes, D. and Essert, J. “Development of Rosetta’s Initial Stage Comet Rendezvous Guidance Systems.” AIAA GNC/AFM/MST/ASC. Minneapolis, MN, USA, 2012.
- [5] Casas, C., Companys, V., Herfort, U., and Pardo de Santayana, R. “Navigation Analysis for Comet Operations up to SSP Delivery.” RO-ESC-TN-5562, Issue 2.2 Draft, 2014. Technical Note.
- [6] Muñoz, P., Budnik, F., Godard, B., Morley, T., Companys, V., Herfort, U., and Casas, C. “Preparations and Strategy for Navigation During Rosetta Comet Phase.” Proceedings of the 23rd International Symposium on Space Flight Dynamics — 23rd ISSFD. Pasadena, CA, USA, 2013.

- [7] Coppola, P. and Zega, E. “Rosetta Navigation Camera (CAM) Software Software Requirements Document.” RO-GAL-RS-2007, Issue 6, 2001.
- [8] Lauer, M. “Rosetta OSIRIS/FD Optical Navigation Interface.” RO-ESC-IF-5503, Issue 3, 2013. Technical Note.
- [9] “Rosetta.” http://www.esa.int/Our_Activities/Space_Science/Rosetta. Accessed: 2014-04-06.
- [10] Lowry, S., Duddy, S. R., Rozitis, B., Green, S. F., Fitzsimmons, A., Snodgrass, C., Hsieh, H. H., and Hainaut, O. “The Nucleus of Comet 67P/Churyumov-Gerasimenko: A New Shape Model and Thermophysical Analysis.” *Astronomy and Astrophysics*, Vol. 548, 2012. Article A12.
- [11] “Persistence of Vision Raytracer.” <http://www.povray.org/>. Accessed: 2014-03-30.
- [12] “The Persistence of Ignorance — Mesh Camera.” <http://www.ignorancia.org/en/index.php?page=mesh-camera>. Accessed: 2014-04-07.
- [13] Slusser, S. M. “ShareCG Meteor.” <http://www.sharecg.com/v/54542/browse/5/3D-Model/Meteor>. Accessed: 2014-04-01.
- [14] “ImageMagick.” <http://www.imagemagick.org>. Accessed: 2014-04-12.
- [15] “Flexible Image Transport System.” <http://fits.gsfc.nasa.gov/>. Accessed: 2014-04-17.
- [16] “Rosetta/MarsExpress/VenusExpress Mission Control System Data Delivery Interface Document, Appendix H: FD Products.” RO-ESC-IF-5003, Issue 4.4, November 2013.
- [17] Hatcher, A. *Algebraic Topology*. Cambridge University Press, 2002.
- [18] Shapira, D., Avidan, S., and Hel-Or, Y. “Multiple Histogram Matching.” *Proceedings of The IEEE International Conference on Image Processing*. Melbourne, Australia, 2013.
- [19] Sarvaiya, J. N., Patnaik, S., and Kothari, K. “Image Registration Using Log Polar Transform and Phase Correlation to Recover Higher Scale.” *Journal of Pattern Recognition Research*, Vol. 7, pp. 90–105, 2012.
- [20] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. “Image Quality Assessment: From Error Visibility to Structural Similarity.” *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600–612, 2004.