Analysis of planetary spacecraft images with SPICE

Teresa Peña[1], *Manel Soria*[2], *Paula Betriu*[2], *Enrique García-Melendo*[2]

Abstract
Spacecraft images are an invaluable source of information in Planetary Science. However, they must be processed and the initial stage is to navigate them, i.e., determine the longitude and latitude coordinates of each pixel on the image plane. The main goal of the present work is to develop an open-source tool to do so. It will be independent of proprietary software and implemented in a widely used language (Java, Python). It will be able to analyse planetary images taken by different spacecraft, such as New Horizons, Cassini or Voyager, with minimal user intervention. Here we present the first steps of the process illustrating the techniques to navigate an image of an ellipsoidal body, obtained from mission kernels using NASA Jet Propulsion Laboratory SPICE library, considering that the attitude and position of the spacecraft are available; correct the camera attitude information; determine the image resolution for each pixel; and combine different images of a body to generate mosaics with high resolution.

Keywords
Planetary Science, Planetary Image Processing, SPICE, Open Software

[1] Universitat Politècnica de Catalunya, ESEIAAT, teresa.pena@estudiantat.upc.edu (Student)
[2] Universitat Politècnica de Catalunya, ESEIAAT, Aerospace Engineering / Physics Department, Spain.

Nomenclature

CCD	*Couple-Charged Device*
FOV	*Field of View*
SPICE	*Spacecraft Planet Instrument C-matrix Events [4].*

1. Introduction

Spacecraft images are an invaluable source of information in Planetary Science. To mention just one example, in atmospheric science they can be used to measure wind velocities and track the evolution of storms [1,2]. The first step to process the planetary images is usually to *navigate* them [3] i.e., determine the longitude and latitude coordinates of each pixel on the image plane. In order to do so, accurate information about the position of the spacecraft and the attitude of the optical instrument are crucial. These data are usually presented in the form of *kernel* files generated by the mission and processed with the SPICE library [4]. However, attitude kernels (C-kernels) are sometimes not available at all (Voyager) or not entirely accurate (Cassini [5]).

Many researchers have put their efforts on finding the way to process the images with very precise data sets. In [6] a photogrammetric control network to generate accurate mosaics of Jupiter's moon Europa is developed. In [3], the authors present a software package called PLIA (The Planetary Laboratory for Image Analysis) to navigate and process images from different missions. The present work is aimed to eventually develop an open-source tool for planetary image analysis that does not rely on proprietary software (such as IDL or MATLAB), minimizes the need of human intervention in the navigation process, provides an estimation of the navigation error for each image and can be used to process images of different missions.

Here, as a first step towards the aforementioned goals, we present algorithms to (a) Navigate an image of an ellipsoidal body, assuming that the exact attitude and position of the vehicle are available; (b) Correct the camera attitude information; (c) Obtain the image resolution for each pixel; and (d) Combine different images of the same body in a mosaic to obtain a full projection, choosing the best resolution available for each region.

2. Image projection from known spacecraft position and instrument attitude

Assuming that the attitude of the camera and the position of the spacecraft are perfectly known, the navigation of the images could be performed with the algorithm outlined in this section. An example of an image where little correction is needed is presented in Figure 2.

2.1. Projection of a point in the image plane

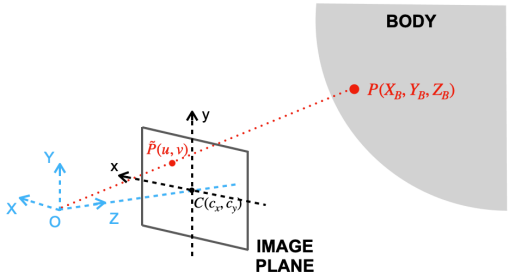

Figure 1. Projection of a surface point in the image plane

Consider a point on the body surface (Figure 1), expressed in its fixed frame, $P(X_B, Y_B, Z_B)$, The rotation matrix from the body frame to the instrument frame, at the instant of the image, can be obtained from the kernel data with the SPICE function *cspice_pxform.* Afterwards, the rotated vector is translated to the location of the instrument by determining the relative position of the frame with SPICE function *cpsice_spkpos*.

Once the position of $P(X_B, Y_B, Z_B)$ is expressed in the frame of the instrument as $P(X_I, Y_I, Z_I)$, it is projected on the image plane and converted into pixels, using the intrinsic matrix of the camera [7]:

$$K = \begin{bmatrix} \pm\frac{F}{\rho} & 0 & c_x \\ 0 & \pm\frac{F}{\rho} & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

where F is the focal length of the camera, ρ the size of the pixels and c_x and c_y the coordinates of the optical centre of the FOV (point C in Figure 1). The homogeneous coordinates of $\tilde{P}(u', v', w')$ can be expressed as:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = K \cdot \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \qquad (2)$$

and finally converted to Cartesian with:

$$u = \frac{u'}{w'} \qquad v = \frac{v'}{w'} \qquad (3)$$

in order to obtain the coordinates of $\tilde{P}$, the projection of P.

2.2. *Image navigation*

The navigation of the image is carried out scanning each pixel of the CDD to determine if there is a surface point of the planet projected on it and visible from the spacecraft. To do so, the intersection points between the ellipsoid and a line of sight, which emanates from the centre of the instrument O (Figure 1), goes through the pixel $\tilde{P}$ considered and continues to infinite, has to be obtained.

The equation to determine the intersection between the line of sight and the ellipsoid can be expressed as:

$$(S + \lambda L - C)^T A (S + \lambda L - C) = 1 \qquad (4)$$

where S is the position of the spacecraft, the scalar λ (the unknown) is the distance between the center of the instrument frame O and the body, C is the center of the body, A is a parametrization matrix described below and L is a unitary vector defining the line of sight of each pixel. All the magnitudes are expressed in the reference frame J2000 (equivalent to the *International Celestial Reference Frame* [8]). L is the vector of the line of sight expressed in the frame of the camera. Regarding the matrix A, defined in the principal axes of the ellipsoid and composed of the equatorial r_e and polar r_p radii of the body, it is:

$$A = \begin{bmatrix} \frac{1}{r_e^2} & 0 & 0 \\ 0 & \frac{1}{r_e^2} & 0 \\ 0 & 0 & \frac{1}{r_p^2} \end{bmatrix} \qquad (5)$$

Eq. 4 is expressed as a second-degree equation, whose discriminant Δ is solved. When its value is null or positive, the line of sight intersects the ellipsoid in one or two points, respectively. The smaller value of λ is the one referring to the point in the near-face of the body. Then, the surface points of the ellipsoid are calculated and converted from J2000 to the body-fixed frame by means of a rotation and a translation as described in sub-section 2.1.

Apart from imposing that the intersection point must be on the edge or inside the body, it is also necessary that it is illuminated to be seen from the spacecraft and shown in the image. Function *cspice_illumin* from the SPICE library is used with this purpose.

Once the previous requirements are verified, the longitude and latitude associated to the surface point projected in each pixel are computed. These values can be obtained with two SPICE functions, *cspice_reclat* or *cspice_recpgr*, depending on the system that wants to be used to express the lon/lat values, the planetocentric or the planetographic one.

2.3. *Image projection*

In order to obtain the projected image, the intensity associated with each lon/lat is needed.

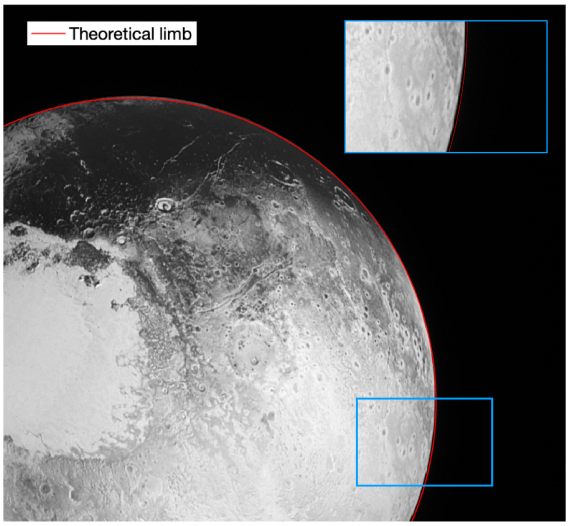

Figure 2. Theoretical limb of Pluto in New Horizon's image 299147481 [9]. The difference between the limbs position predicted with the mission kernels and the image can be seen more clearly in the zoomed area.

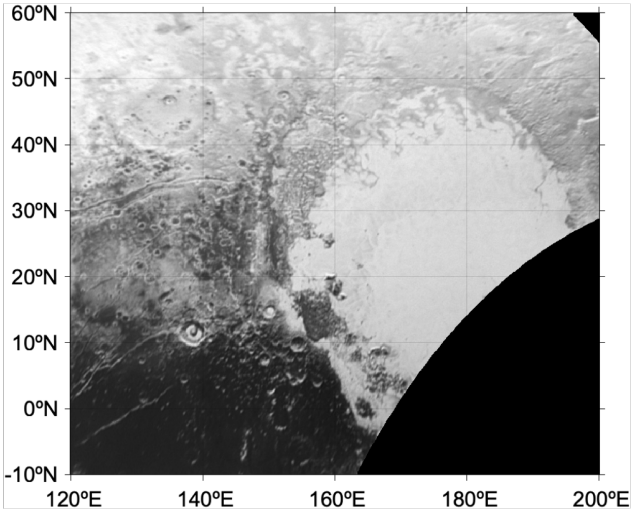

Figure 3. Image projection of Pluto from New Horizon's image 299147481 [9] generated from Figure 2.

To compute it, we use an interpolation algorithm based on the triangulation of the intensity data obtained from the original image.

In order to consider that the intensity of a pixel obtained from the interpolation is valid, the surface point associated to it has to be (a) inside the FOV of the camera and (b) in the near face of the planet or moon. The first condition is verified by computing the location of the pixel in the image plane and, for the second, the angle between the position vector of the surface point and the one that has its origin in this point and its end in the spacecraft is computed. If this angle is smaller than 90º, the point is located in the near-side of the body and the interpolated intensity of the pixel is maintained.

If one of the previous two conditions is not fulfilled, the interpolated intensity is changed into a specific value, such as 0, so in the image the pixel under consideration is displayed in black.

To increase the quality of the projection, image processing techniques are used and, more specifically, sharpening and contrast adjustment. The final result can be seen in Figure 3.

3. Attitude correction from known planetary limb position

If the attitude of the camera is not perfectly known, the image navigation provides wrong results that lead to unreal image projections or mosaics. In fact, even the projection of Figure 3, where the kernel is quite accurate, does not coincide perfectly with that provided in [10], since the longitude is slightly different.

The inaccuracies in the data are manifested as a displacement of the position of the body with respect to the one that can be seen in the image. In order to correct this mismatch, the procedure that is proposed here consists of determining the rotation around the axes of the instrument (with the attitude provided by SPICE) necessary to make coincide the positions of the body.

3.1. Limb points on the image

The aim of this work is to provide procedures that can be used with the majority of images in which all or a part of the limb of the body is displayed. However, when trying to generate the limb of the planet or moon directly from the original image, there are factors that make it very complicated to do it with completely automatic methods that do not require the intervention of the user, such as the presence of rings in the images of Saturn or the dark zones of Pluto in the images taken by New Horizons.

Because of this, here we propose a method characterized by the generation of a limb defined by as many points as desired, which follow the limb that can be seen in the image and are computed in the same way. Thus, is the user who chooses the zones in which the limb points should be located and the algorithm computes the exact pixel. To do so, it first determines the average intensity of the pixels of the background and those of the planet. Then, the mean of these two values is calculated to know the intensity that the pixel where the limb point will be located should have.

3.2. Theoretical limb position

As the points of the limb from the image are expressed in pixels, the limb generated with the SPICE data (i.e., the theoretical limb) also has to be expressed in this units in order to be compared.

The theoretical limb is defined in the body-fixed frame by the main parameters of a conic (i.e., center and major and minor semi-axes), obtained from SPICE function *cspice_edlimb*. From them and the parametric expression of an ellipse in 3-D, an arbitrary number of points that are part of the conic can be generated and afterwards converted into pixels to be shown in the image. To do so, a rotation and translation of their position vector from the body-fixed frame to that of the instrument is performed as described in previous sections.

3.3. Correction procedure

To correct the mismatch between the limbs of the body, the frame of the instrument is rotated causing a variation in the position of the theoretical limb. To do this, the rotation matrix that makes coincide almost perfectly both limbs has to be found.

This matrix is defined by the angles of Euler (α, β, γ) and is applied to the coordinates of each point of the conic expressed in the frame of the instrument:

$$\begin{bmatrix} X_{I'} \\ Y_{I'} \\ Z_{I'} \end{bmatrix} = R_z(\gamma) R_y(\beta) R_x(\alpha) \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \qquad (6)$$

To find the combination of angles that generates the best fitting, an iterative minimization procedure is used. The parameter

to minimize is D, the sum of the distances d_i between each limb point p_i and its projection on the theoretical limb conic t_i.

$$D = min\left[\sum\left(||p_i - t_i||\right)\right] \quad (7)$$

The minimization starts with arbitrary angles, such as a tenth of the angular field of view of the camera, and after a few iterations the error is notably reduced, provided that the number of limb points is sufficient. Once the right Euler angles have been obtained, they are used to generate the corresponding rotation matrix.

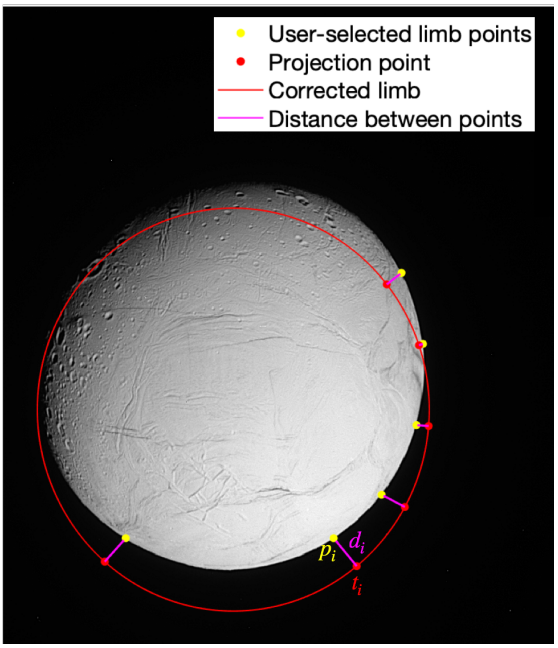

Figure 4. Iteration of the minimization in Cassini's image N1516169656 of Enceladus [9].

3.4. Application of the correction to the image navigation

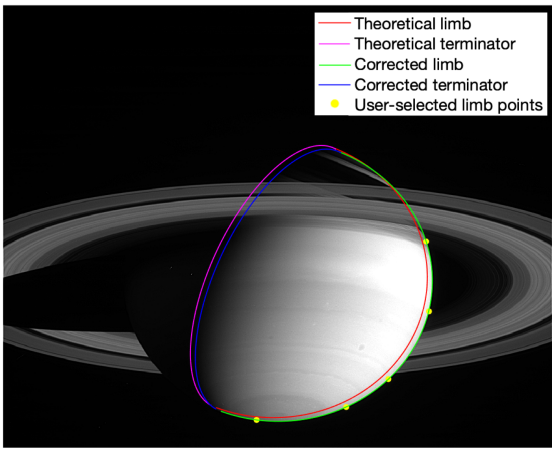

Figure 5. Comparison of the theoretical and corrected limbs and terminators of Saturn in Cassini's image 1461406214 [9].

When the rotation matrix that corrects the attitude of the spacecraft has been computed, it has to be applied to the image navigation procedure and, more specifically, to the pointing vector (L in Eq. 4) used to defined the position of each pixel of the CCD.

In the case presented in Figure 5, the Euler angles are (0.0024°, -0.0032°, 0.0352°) and the total distance or error between limbs is 0.814 pixels.

4. Generation of image mosaics

An image mosaic can be defined as the union of multiple image projections, in such a way that the range of longitude and latitude displayed is wider than the one that could be shown in each individual image projection. For each pixel, the image with best resolution is selected.

The resolution in pixels/meter is computed as the square root of the surface that covers each pixel, which is defined by the mean vertical and horizontal distances between the pixel whose resolution wants to be determined and the adjacent pixels in each direction. As the pixels are associated to a longitude and latitude, Vincenty's formula [11] is used to compute the geodesic distance between two pixels on the surface of an ellipsoidal body.

As with the intensity of the pixels in the image projection, the resolution is also interpolated to obtain the correct value associated to the lon/lat of the pixels. The interpolation algorithm is based, as well, on the triangulation of the resolution data obtained from the original image.

Once the intensity of the pixels and its resolution for each image have been filtered, the data of the image mosaic can be generated by selecting, from the multiple images, the pixels with a higher resolution.

5. Results & Discussion

To illustrate the procedure described in this article, the image mosaic of Figure 6 is presented. Here each image has been represented with a different color for clarity.

6. Conclusions

The results provided confirm that the first steps of the work that will end in the development of an open-source tool for planetary image analysis have been completed.

We are now able to navigate images obtaining accurate longitudes and latitudes, by means of a correction procedure based on an estimation of the camera kernel error.

However, there are still a few aspects that should be addressed in the near future: (a) the Newtonian light time correction and stellar aberration correction; (b) the analysis of images in which the *kernel* files are not available, such as those of Voyager 1 and 2; (c) the treatment of images without limb; (d) the estimation of the error in the image projection due to inaccuracies in the attitude of the instrument or the selection of limb points; and (e) the estimation and correction of the spacecraft position when multiple stars and a body are visible in the image.

The final algorithm will be implemented in a widely available and portable language such as Java or Python.

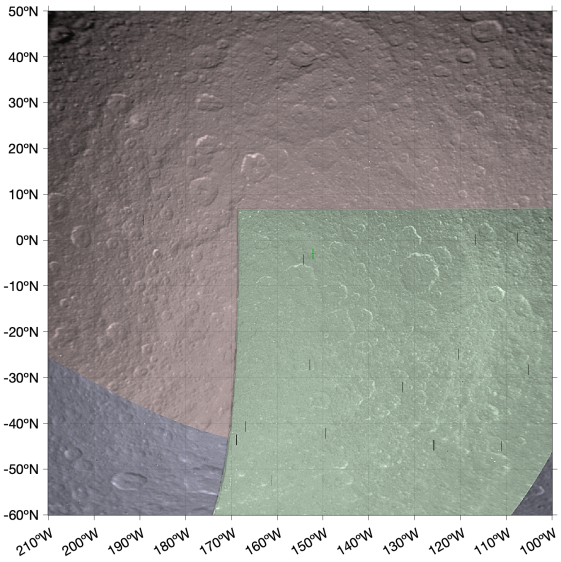

Figure 6. Colored image mosaic of Rhea from Cassini's images N1511700504, N1511717371 and N1499997169 [9].

References

[1] A. Sánchez, E. Garcia-Melendo et al, A complex storm system in Saturn's north polar atmosphere in 2018, *Nature Astronomy*, 1-17, 2019.

[2] E. García-Melendo, S. Pérez-Hoyos et al, Saturn's zonal wind profile in 2004–2009 from Cassini ISS images and its long-term variability, *Icarus*, 215, 62-74, 2011.

[3] R. Hueso, J. Legarreta et al, The Planetary Laboratory for Image Analysis (PLIA), *Advances in Space Research*, 46, 1120-1138, 2010.

[4] The SPICE Concept. NAIF Website: https://naif.jpl.nasa.gov/naif/spiceconcept.html, last visited: 18th March 2022.

[5] Th. Roatsch, M. Wählsich et al, Mapping of the icy Saturnian satellites: First results from Cassini-ISS, *Planetary and Space Science,* 54, 1137-1145, 2007.

[6] M. T. Bland, L. A. Weller et al, Improving the usability of Galileo and Voyager images of Jupiter's moon Europa, *Earth and Space Science*, 12, e2021EA001935, 2021.

[7] Geometry of Image Formation. LearnOpenCV Website: https://learnopencv.com/geometry-of-image-formation/, last visited: 10th March 2022.

[8] Reference Frames. NAIF Website: https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/req/frames.html, last visited: 19th March 2022.

[9] OPUS3 Website: https://opus.pds-rings.seti.org/#/cols=opusid,instrument,planet,target,time1,observationduration&widgets=instrument,observationtype,target&order=time1,opusid&view=search&browse=gallery&cart_browse=gallery&startobs=1&cart_startobs=1&detail=, last visited: 18th March 2022.

[10] S. A. Stern, F. Bagenal et al, The Pluto system: Initial results from its exploration by New Horizons, *Science*, 6258, aad1815-2, 2015.

[11] C. Thomas and W. Featherstone, Validation of Vincenty's Formulas for the Geodesic Using a New Fourth-Order Extension of Kivioja's Formula, *Journal of Surveying Engineering-asce - J SURV ENG-ASCE*, 131, 2005.