



HAL
open science

Image processing for precise geometry determination

Ines Belgacem, G. Jonniaux, F. Schmidt

► **To cite this version:**

Ines Belgacem, G. Jonniaux, F. Schmidt. Image processing for precise geometry determination. Planetary and Space Science, 2020, 193, pp.105081. 10.1016/j.pss.2020.105081 . hal-02929664

HAL Id: hal-02929664

<https://hal.science/hal-02929664>

Submitted on 3 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Image processing for precise geometry determination

I. Belgacem^{a,b,*}, G. Jonniaux^c, F. Schmidt^a

^a*Université Paris-Saclay, CNRS, GEOPS, 91405, Orsay, France.*

^b*European Space Astronomy Centre, Urb. Villafranca del Castillo, E-28692 Villanueva de la Cañada, Madrid, Spain*

^c*Airbus Defence & Space, Toulouse, France.*

Abstract

Reliable spatial information can be difficult to obtain in planetary remote sensing applications because of errors present in the metadata of images taken with space probes. We have designed a pipeline to address this problem on disk-resolved images of Jupiter's moon Europa taken with New Horizons' LOnG Range Reconnaissance Imager, Galileo's Solid State Imager and Voyager's Imaging Science Subsystem. We correct for errors in the spacecraft position, pointing and the target's attitude by comparing them to the same reference. We also address ways to correct for distortion prior to any metadata consideration. Finally, we propose a vectorized method to efficiently project images pixels onto an elliptic target and compute the coordinates and geometry of observation at each intercept point.

Keywords: image registration, computer vision, projections, metadata, mapping, SPICE

1. Introduction

For a variety of applications in space exploration remote sensing, it is crucial to have accurate spatial representation of the data. To do so, the user needs precise information about the position and attitude of the spacecraft. The SPICE information system (Acton, 1996) helps planetary scientists and engineers to plan and process remote sensing observations. Groups in the different missions create Camera-matrix Kernels (CK)

*Corresponding author

Email address: ines.belgacem@esa.fr (I. Belgacem)
Preprint submitted to Planetary and Space Science

16 and Spacecraft Kernels (SPK) for SPICE containing the pointing information of the
17 instrument of and the position of the spacecraft. However, the data set to generate
18 these kernels incorporates uncertainties and errors which can make it difficult to accu-
19 rately project pixels into the 3D scene (Jonniaux and Gherardi, 2014; Sidiropoulos and
20 Muller, 2018).

21 Efforts have been made to develop tools that correct and reconstruct the CK kernels
22 using astrometry techniques to generate the so called C-smithed kernels (Cheng, 2014).
23 However, not all C-Kernels have undergone this correction and some of these corrections
24 might be considered by-products and not necessarily be publicly available or easily
25 accessible.

26 Some open-source tools are available such as ISIS3 (Anderson et al., 2004; Edmund-
27 son et al., 2012), but it needs a manual selection of control points for optimal results.
28 Additionally, we have verified after processing Voyager images with this tool that a lot
29 of distortion was still present which is a major issue. The AutoCNet python library
30 (Laura et al., 2018) has been developed with the objective of a more automated tool
31 based on dense feature extraction on high resolution images. The CAVIAR software
32 package (Cooper et al., 2018) is using background star positions to refine spacecraft
33 pointing information and is publicly available to correct CASSINI images metadata.
34 There is also an on-going effort of the PDS Ring-Moon Systems Node to develop tools
35 to easily interact with SPICE geometric metadata (Showalter et al., 2018).

36 We propose here solutions for correcting metadata in disk-resolved images at in-
37 termediate resolution, typically when the full planetary body is observed in a single
38 scene. We hypothesize that the ephemeris of the planetary bodies involved are valid
39 and therefore we correct for the pose of the cameras (position and attitude). We also
40 propose an efficient way to project all pixels onto the target. To illustrate this work,
41 we are using images of Jupiter’s moon Europa taken with three different imagers -
42 the New Horizons’ LOnG Range Reconnaissance Imager (LORRI) (Cheng et al., 2008)

43 Galileo’s Solid State Imager (SSI) (Belton et al., 1992) and Voyager 1 and 2’s Imaging
 44 Science System (ISS) (Smith et al., 1977) i.e. two cameras (narrow and wide angle)
 45 per spacecraft. Images have a 1024x1024 pixel resolution for LORRI, 800x800 for SSI
 46 and 1000x1000 for ISS. This work has been done in the context of the preparation of
 47 ESA’s JUICE mission (Grasset et al., 2013) and NASA’s Europa Clipper (Phillips and
 48 Pappalardo, 2014) but the proposed strategy is general enough to be used for every
 49 past and future space exploration missions. It could have many applications such as
 50 vision-based navigation (Jonniaux et al., 2016), spectroscopic and photometric studies
 51 (Belgacem et al., 2019).

52 Fig. 1 summarizes all the different steps of our approach. We propose several
 53 alternatives and choose the most reliable solution for each brick of the pipeline.

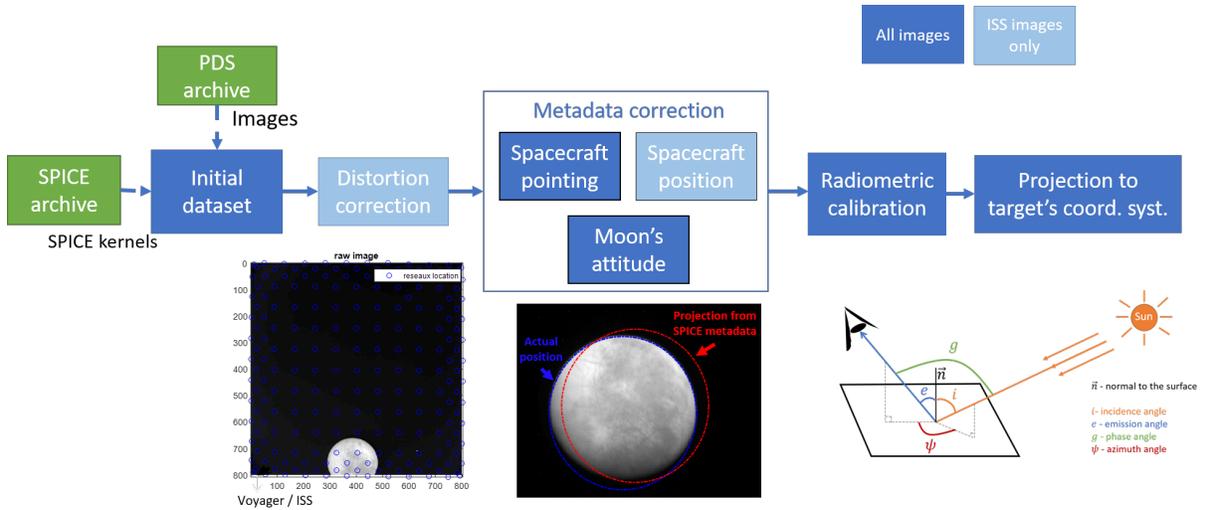


Figure 1: Visualization of entire pipeline to correct metadata and extract geometrical information from the images

54 2. Distortion

55 Before any kind of metadata correction, it is necessary to address the optical dis-
 56 tortion of the camera in the images. The most common type of distortion is radial

57 (Hartley and Zisserman, 2003b). It is symmetric and can follow one of two patterns -
58 "barrel", "pincushion" - or a combination of the two. Some cameras such as the ones
59 embarked on Voyager can have a more complex behavior.

60 2.1. LORRI

61 The geometric distortion has been estimated to be less than 0.5 pix across the entire
62 field of view (Cheng et al., 2008). We considered that there was no need for correction.

63 2.2. Galileo SSI

64 The distortion of the Galileo SSI instrument is well known and expected to be very
65 stable. It is a negative radial distortion ("pincushion") and is maximal at the corners of
66 the image with about 1.2pix and increases from the center of the image as the cube of
67 the distance (Belton et al., 1992). Therefore we can easily correct the images using this
68 power law. For each point in the raw image noted (x, y) or (r, θ) in central cylindrical
69 coordinates, we can compute its undistorted position (x_d, y_d) or (r_d, θ_d) using:

$$r_d = r + \delta_r \quad \text{with} \quad \delta_r = 1.2 \times \left(\frac{d}{d_c}\right)^3 \quad (1)$$

70 Where d is distance from the center and d_c is distance of any corner from center:
71 $d_c = 400\sqrt{2}$

72 Fig. 2 shows the distortion map of Galileo SSI, i.e. the value of δ_r across the field
73 of view of the camera. A simple bilinear interpolation between the positions in the raw
74 image and the corrected coordinates gives us the undistorted image.

75 We should note that another quadratic distortion correction has been proposed by
76 Oberst 2004.

77 2.3. Voyager

78 Voyager is the first generation of space exploration from the 70's and the level of
79 distortion in the images is much higher. It is a lot less stable than in other data sets

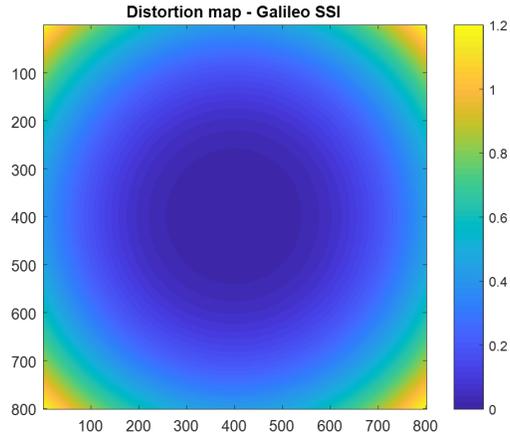


Figure 2: Variation of δ_r across the field of view of Galileo SSI

80 and requires a specific processing. A grid of reseau markings was embedded within the
 81 optics of the cameras to record it. Fig. 3 shows a raw Voyager 1 NAC image where we
 82 can see the markings. The real positions of the reseau were measured during calibration
 83 and are supposed to be fixed. On each image, the apparent position of the reseau can be
 84 measured. We can then correct for distortion using the correspondence. This work has
 85 been done by the NASA Planetary Data System Ring Node service and their corrected
 86 images are available online (Showalter et al., 2013). We have noticed that some residual
 87 distortion was still present so we propose here to improve the correction. We used the
 88 measured positions tabulated in the *geoma* files provided by Showalter et al. 2013 that
 89 were quite precise (see zoomed image in fig. 3). We propose the 3 following methods
 90 that we quantify by their residual RMSD in the prediction of the reseau points of image
 91 C1636902.

92 *2.3.1. Method 1 - radial solution*

93 Let's consider (x, y) the pixel coordinates in the raw image and (x_d, y_d) the pixel
 94 coordinates in the undistorted image, $r = \sqrt{x^2 + y^2}$ the radial coordinate. We are
 95 looking for the radial distortion function f defined by (Hartley and Zisserman, 2003c):

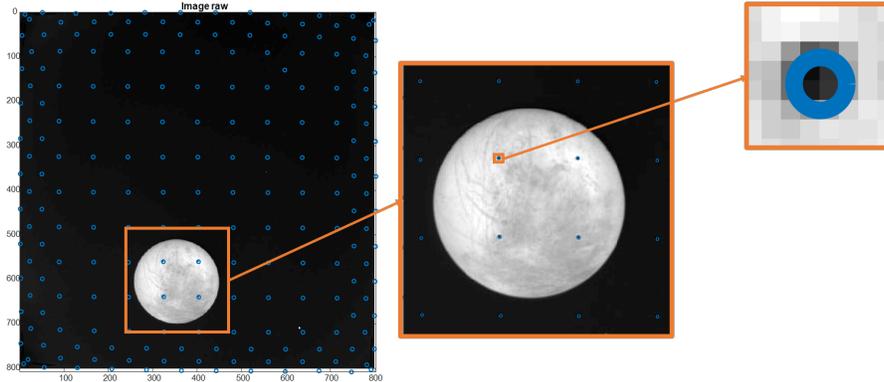


Figure 3: Raw Voyager image with reseau markings

$$r_d = rf(r_1) = r(1 + k_1r + k_2r^2 + k_3r^3 + \dots) \quad (2)$$

106 We use a least-squares approach to estimate the k_i coefficients. Considering that
 107 each of the n reseau point will contribute two equations ($x_d = xf(r)$ and $y_d = yf(r)$)
 108 we have a system of $2n$ equations that we can write $A\mathbf{X} = \mathbf{b}$. The least squares solution
 109 is given by:

$$\mathbf{X} = (A^T A)^{-1} A^T \mathbf{b} \quad (3)$$

100 We could not achieve satisfactory results with this approach. The best result we
 101 achieved using a radial distortion function was an average RMSD (Root Mean Square
 102 Deviation) of 17.3 pix using a 9-degree polynomial in r over 382 Voyager images. The
 103 distortion in the Voyager images cannot be described as radial and needs to be ad-
 104 dressed in a more specific manner. The exhaustive list of images can be found in the
 105 supplementary material.

106 2.3.2. Method 2 - general polynomial

107 More general 2-variables-polynomial functions allow for a non radial symmetric dis-
 108 tortion:

$$\begin{aligned}
x_d &= k_{10} + k_{1a}x + k_{1b}y + k_{1c}xy + k_{1d}x^2 + k_{1e}y^2 \\
y_d &= k_{20} + k_{2a}x + k_{2b}y + k_{2c}xy + k_{2d}x^2 + k_{2e}y^2
\end{aligned}
\tag{4}$$

109 We solve for the k_i coefficients with a least-squares method and make several at-
110 tempts with eq. 4 and higher degree polynomials. Results are improved and the best
111 performance is achieved with a 6-degree polynomial - we obtain an average RMSD of
112 0.47pix over the same 382 Voyager images. The exhaustive list of images can be found
113 in the supplementary material.

114 It is worth noting that these systems are not very stable because the matrix A we
115 define is close to being singular. This problem is more likely to happen the higher the
116 degree of the polynomial.

117 2.3.3. Method 3 - local bilinear transformations

118 Another option is to work on a more local level. We start by dividing the resseau
119 markings grid into triangles using the Delaunay algorithm. For each of them, we com-
120 pute the exact bilinear transformation that would transform it into its undistorted
121 form.

122 For T , a triangle between three points $t_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$, $t_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ and $t_3 = \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$
123 whose undistorted equivalent is T' , a triangle between $t'_1 = \begin{pmatrix} x'_1 \\ y'_1 \end{pmatrix}$, $t'_2 = \begin{pmatrix} x'_2 \\ y'_2 \end{pmatrix}$ and
124 $t'_3 = \begin{pmatrix} x'_3 \\ y'_3 \end{pmatrix}$ we can write:

$$\begin{aligned}
x'_i &= ax_i + by_i + c \\
y'_i &= dx_i + ey_i + f
\end{aligned}
\quad \forall i \in \{1, 2, 3\}
\tag{5}$$

125 If we form the vector $\mathbf{X}_A^T = (a \ b \ c \ d \ e \ f)$, we can rewrite eq. 5 as:

$$\begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} x'_i \\ y'_i \end{pmatrix} \quad (6)$$

$$A_x X_A = B$$

126 Each point of the triangle contributes two lines to the matrix A_x . To find the
 127 coefficients of the transformation matrix stored in the vector \mathbf{X}_A , we need to invert the
 128 system. The least squares solution is given by:

$$\mathbf{X}_A = (A_x^T A_x)^{-1} A_x^T B \quad (7)$$

129 The undistorted image is a 1000*1000 square. Each pixel can be attributed to a tri-
 130 angle and undergo the corresponding transformation to compute its position in the raw
 131 image. With a simple bilinear interpolation between the raw image and the projected
 132 grid of undistorted positions, we have the new undistorted image. The RMSD cannot
 133 be used to evaluate the precision since the the reseau point are perfectly matched by
 134 construction. Thus we cannot have a precise estimation of the accuracy of this correc-
 135 tion method beyond the fact that it is under a pixel if, as we suppose, the distortion is
 136 well sampled by the reseau grid.

137 We choose to apply the local bilinear transformation method since it is the one that
 138 ensures perfect reconstruction for the full reseau markings grid. Fig. 4 illustrates the
 139 matching of the reseau markings computed location in the 1000*1000 grid using the
 140 three methods we present here to their known location.

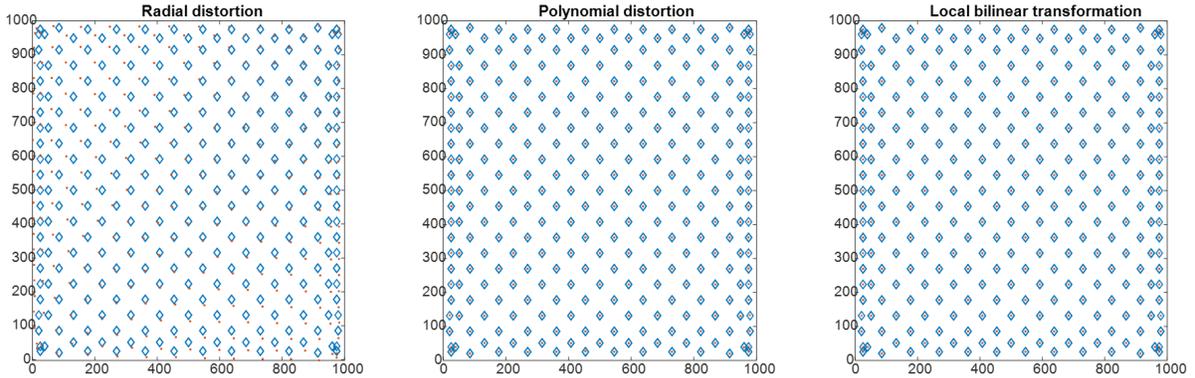


Figure 4: Illustration of the matching of the reseau markings computed location in the 1000*1000 grid using the three methods we present here (magenta dots) to their known location (blue diamonds) for Voyager 1 NAC image C1636858.

141 3. Camera pose

142 In this section, we address the pointing error as an error on the camera pose (space-
 143 craft position and orientation). We should mention that other potential sources of errors
 144 could come from uncertainties in the definition and alignment of the reference frame of
 145 the instrument, or its boresight. Table 2 will show that there is a significant spread in
 146 our pointing corrections which would not be consistent with a systematic equal bias on
 147 all images for a given camera.

148 3.1. Spacecraft Pointing

149 To estimate the pointing error of the instrument, we can use the metadata available
 150 via the C-Kernels to project the shape of the target into the field of view of the camera.
 151 This gives us the predicted position of Europa in the image space in red in fig. 5 and we
 152 can see that it is a few pixels away from its actual position (in blue). We have to keep
 153 in mind that even a slight error in pointing could result in an offset of tens of pixels
 154 in the image. Such errors are not surprising given typical star tracker accuracy (a few
 155 arcseconds (??) which for LORRI is equivalent to a few pixels in each direction). In

156 this case, the center position of the moon is offseted by -16.5 pixels on the x-axis and
 157 -9.4 pixels on the y-axis. More illustrations of the pointing corrections are available for
 158 all data sets in the supplementary material.

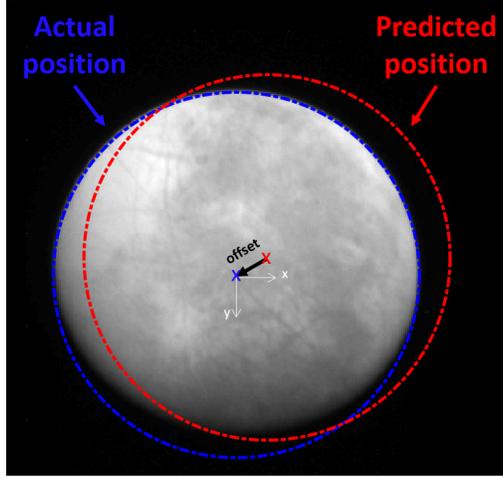


Figure 5: Illustration of remaining pointing error on a LORRI image

159 3.1.1. 2D analysis - measure the offset

160 We are using SurRender, an image renderer developed by Airbus Defence & Space
 161 (Brochard et al., 2018), to simulate images using the metadata from the SPICE kernels
 162 and compare them to the real images (fig. 5). In these conditions, the simplest approach
 163 would be to consider a normalized cross-correlation to compute the translation in the
 164 field of view between the two images using:

$$\rho = \frac{\sum_{x,y} [f_r(x,y) - \overline{f_r}] [f_s(x,y) - \overline{f_s}]}{\sqrt{\sum_{x,y} [f_r(x,y) - \overline{f_r}]^2 \sum_{x,y} [f_s(x,y) - \overline{f_s}]^2}} \quad (8)$$

165 With :

- 166 • f_r : real image, $\overline{f_r}$: mean real image
- 167 • f_s : simulated image, $\overline{f_s}$: mean simulated image

168 However, this solution at best gives pixel-scale errors.

169 To have a better estimate of the pointing error, we finally chose to use an optimization-
170 based method using intensity-based registration. The function we used performs a reg-
171 ular step gradient descent and uses a mean squares metric to compare the two images
172 at each step (imregtform function, (Mathworks, 2018)). We use this function in a loop
173 to ensure that we determine the offset between real image and simulation down to a
174 1/10th of a pixel i.e. we repeat the process and update the simulation with the cor-
175 rected camera orientation until the offset computed between simulation and real image
176 is under 0.1 pixel as described in fig. 7.

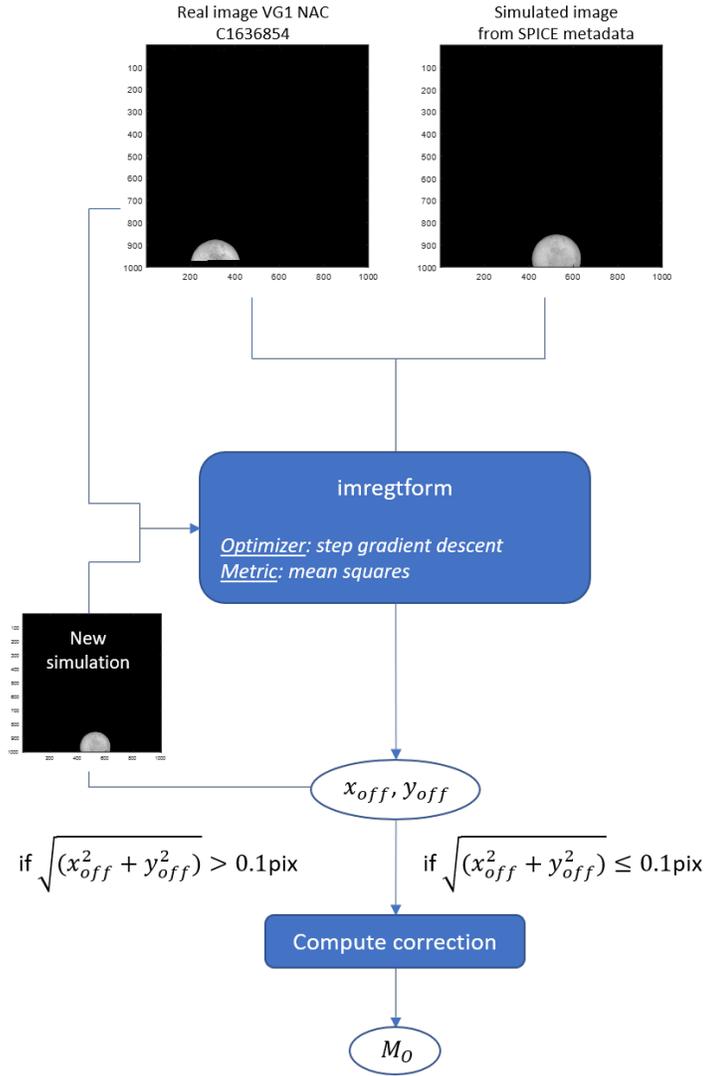


Figure 6: Algorithm used to estimate the translation in the image frame

177 This is the most precise method we have explored and above all the most robust
 178 regardless of the data set. This step can be replaced by any other method of image
 179 registration (Luo and Konofagou, 2010; Ma et al., 2015; Reddy and Chatterji, 1996)
 180 without impacting the rest of the pipeline as long as an offset is computed. We illustrate
 181 that on fig. 7 by comparing the corrected limb with different methods: 2D cross
 182 correlation (in blue), MATLAB function `imregcorr` (in green), MATLAB function

183 imregtform (in yellow) and a limb detection method (in magenta).

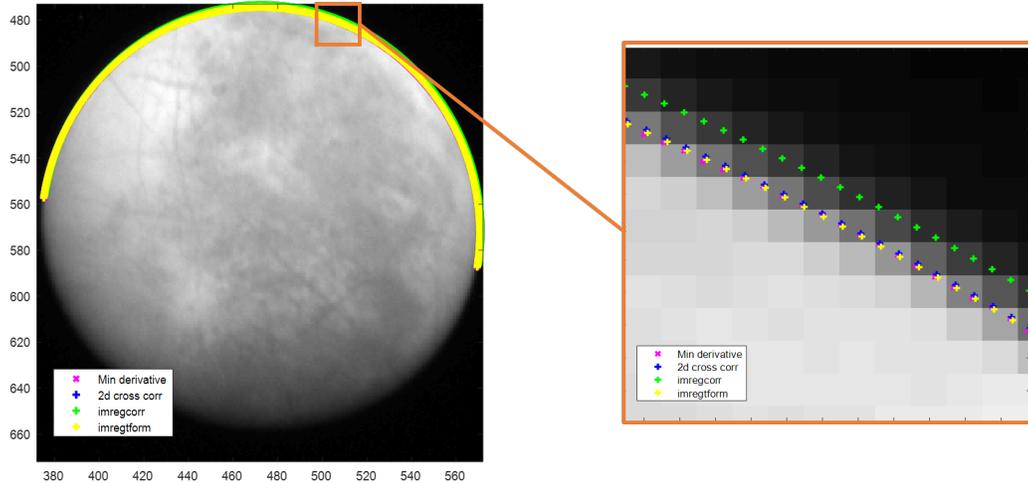


Figure 7: Illustration of different limb corrections on New Horizons' LORRI image LOR_0034849319_0X630_SCI_1

184 We should also note that the methods we considered were all tested on entirely
 185 synthetic situations. As an example, let's take a simulated image of Europa. We have
 186 introduced an offset of 20.32 pixels in the x-direction and an offset of -30.46 pixels in
 187 the y-direction, represented by the initial limb in red. We retrieved these offsets using
 188 three different methods and the results are summarized in table 1.

	Theory	2D cross-correlation	imregcorr	imregtform
xoffset (pix)	20.32	20.0	20.3	22.33
yoffset (pix)	-30.46	-30.4	-30.4	-30.48

Table 1: Results after trying to retrieve a synthetic offset using three different methods

189 Table 2 summarizes the values of the offset we computed for the entirety of the data
 190 sets of Europa images. We have highlighted in bold the mean values in pixel. We can
 191 see that the metadata (extrinsic parameters) errors are much more substantial in the

192 Voyager images. Their distribution is more detailed in fig. 8. We can see that beyond
 193 the fact that the values are significant, the distributions are shifted in the negative
 194 values.

		Voyager 1		Voyager 2		New Horizons	Galileo
		NAC	WAC	NAC	WAC	LORRI	SSI
offset	mean	186.1	131.1	63.5	81.2	18.1	22.4
	min	0.0	7.2	1.2	52.5	11.0	$1e^{-5}$
	max	796.4	675.9	191.1	340.5	23.4	120.9
xoffset	mean	160.9	104.8	44.9	57.4	15.0	15.9
	min	0.0	2.0	0.9	37.1	3.8	$1e^{-5}$
	max	537.2	478.2	135.1	240.8	22.6	85.5
yoffset	mean	64.6	63.6	44.9	57.4	9.4	15.77
	min	0.0	0.0	0.9	37.1	6.0	$1e^{-5}$
	max	777.0	447.7	135.1	240.8	13.6	85.5

Table 2: Summary of all the offsets computed on images of Europa taken with the different cameras

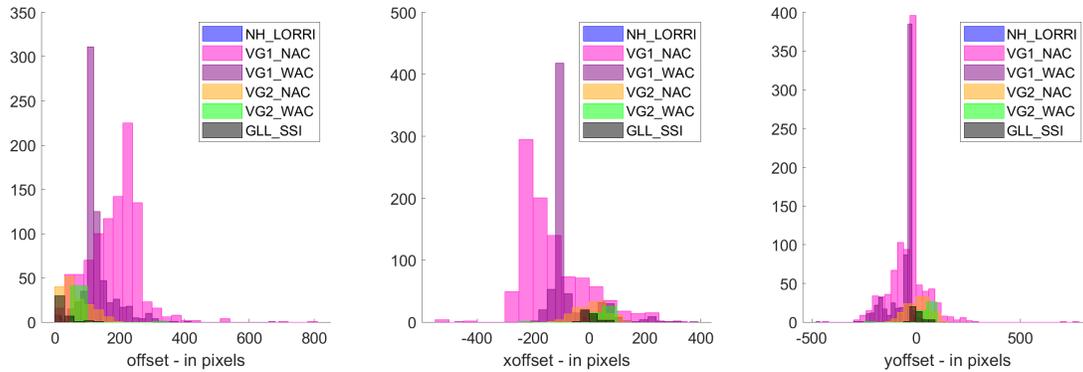


Figure 8: Distribution of pointing errors in pixels for the different cameras

195 3.1.2. Computing the 3D transform

196 Once the offset computed in the image space, we need to find the associated 3D
 197 transform (rotation) to actually correct for spacecraft pointing. For that, we need to
 198 consider the expected boresight of the instrument corrected by the computed offset in
 199 the image:

$$\bar{\mathbf{b}} = K^{-1} \left[\begin{pmatrix} x_{off} \\ y_{off} \end{pmatrix} + \begin{matrix} c_x \\ c_y \end{matrix} \right] \quad (9)$$

200 Where:

- 201 • K : the intrinsic matrix of the camera (Hartley and Zisserman, 2003a) describing
 202 the field of view and focal of the camera
- 203 • x_{off}, y_{off} : offsets along the x and y-axis
- 204 • c_x, c_y : pixel coordinates of the camera center

205 $\bar{\mathbf{b}}$ is the normalized vector (in the camera frame) representing the actual boresight
 206 of the camera. To derive the correcting Euler angles, we can use fig. 9 showing the yz
 207 plane. Around the x-axis: $\alpha = \arctan\left(\frac{-b_y}{b_z}\right)$. A similar approach leads to deriving the
 208 angle around the y-axis $\beta = \arctan\left(\frac{b_x}{b_z}\right)$.

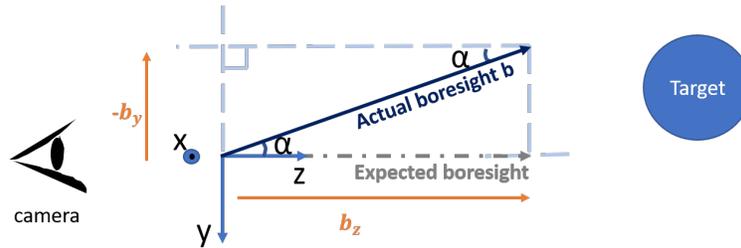


Figure 9: Visualization of boresight in the yz plane

209 The rotation matrix M_o associated to the Euler angles $[\alpha, \beta, 0]$ is the correction
 210 factor we apply to the camera orientation to correct the pointing. Please note that we
 211 do not estimate the rotation around the boresight which will be addressed in section 5.

212 3.2. Distance

213 Fig. 10 shows an example of a Voyager observation, after correction of spacecraft
 214 pointing, where we can see that the moon is actually expanding beyond the limb in red.

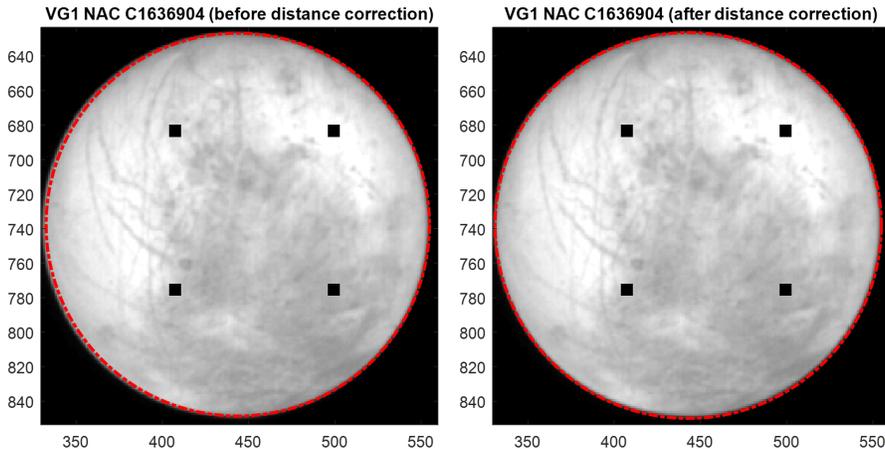


Figure 10: VG1 NAC image with Europa out of limb bounds, especially on the left of the moon (left) and the result after distance correction (right)

215 This can be the result of either an underestimated field of view, or distance. We
 216 found that this effect is dependent on the image, thus it is more likely that a correction
 217 of distance is needed. We continue using comparisons to simulated images, the changing
 218 parameter being, this time, the distance between camera and target.

219 To evaluate the match between real and simulated images we chose the Structural
 220 Similarity index (SSIM) which robustness to noise was noted by (Łoza et al., 2007) who
 221 used it as a tracking technique in videos. The SSIM of two images a and b is defined
 222 by the combination of three terms - luminance ($l(a, b)$), contrast ($c(a, b)$) and structure
 223 ($s(a, b)$):

$$ssim(a, b) = l(a, b)c(a, b)s(a, b) = \left[\frac{2\mu_a\mu_b}{\mu_a^2 + \mu_b^2} \right] \left[\frac{2\sigma_a\sigma_b}{\sigma_a^2 + \sigma_b^2} \right] \left[\frac{\sigma_{ab}}{\sigma_a\sigma_b} \right] \quad (10)$$

224 Where μ is the sample mean, σ is the standard deviation and σ_{ab} is the sample
 225 covariance. We use the SSIM index to define our cost function. To minimize the cost

226 function (or maximize the similarity index), we try doing a simple gradient descent but
 227 the algorithm was thrown off by local minima. We decided to resort to a less optimized
 228 but safer method: computing the cost function for a set of distance values between 99%
 229 and 101% of the predicted distance and picked the distance for which we had the best
 230 index a posteriori.

231 Fig. 11 shows the distribution of the computed correction factors over the totality
 232 of the Voyager data set for Europa. We can note that the distance has always been
 233 overestimated and that Europa is actually always closer than expected from the SPICE
 234 kernels.

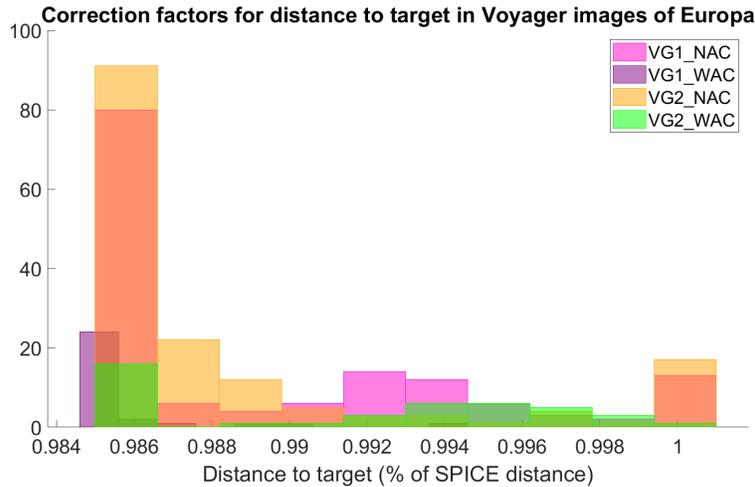


Figure 11: Distribution of correction factors for the distance to target in Voyager images of Europa

235 4. Validation

236 4.1. Distortion

237 We compared our undistorted images using a local bilinear transformation (see sec-
 238 tion 2.3.3) to the GEOMED images made available in the Ring Node archive (Showalter
 239 et al., 2013). Fig. 12 and 13 illustrate the differences we have noted.

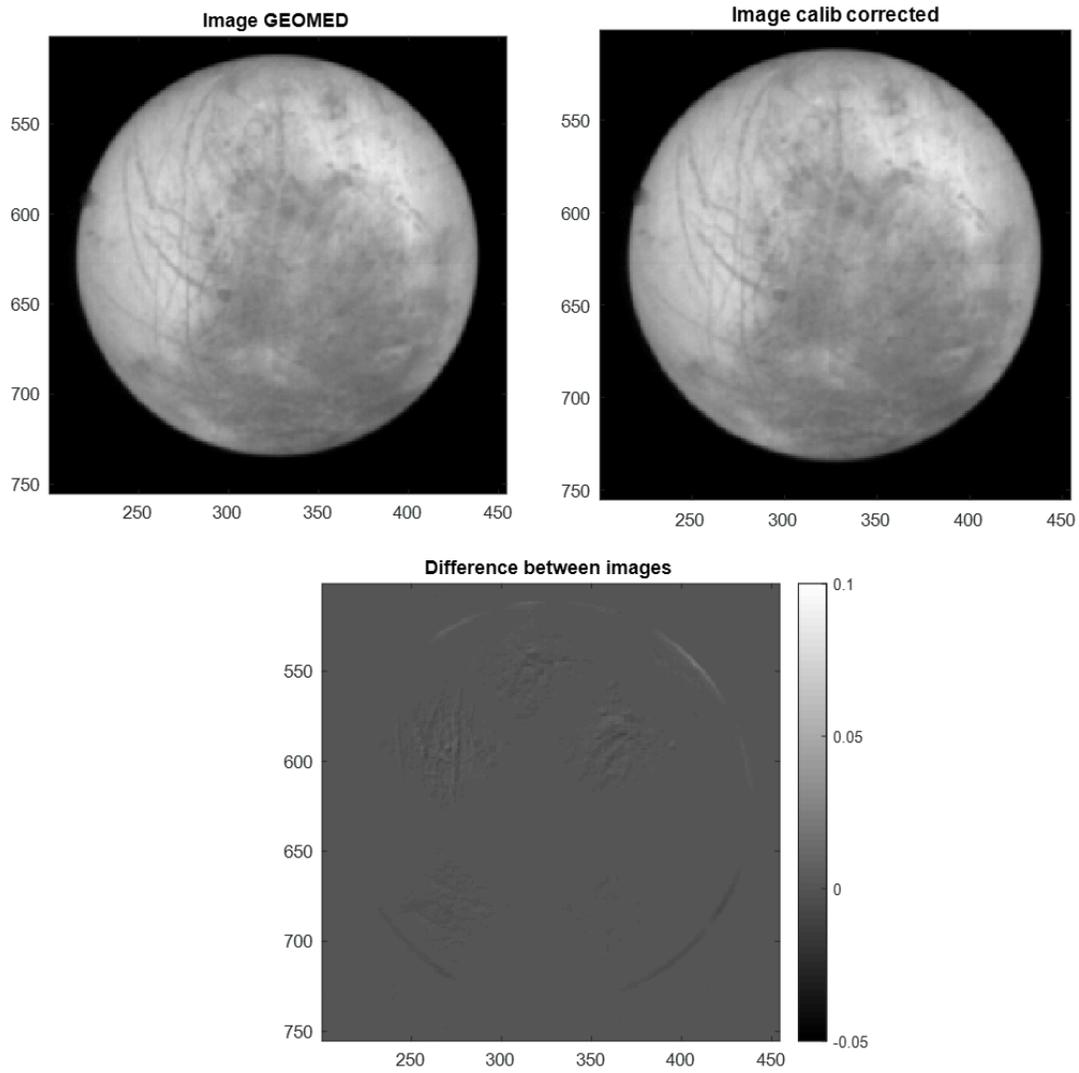


Figure 12: Illustration of the differences between our undistorted images and the GEOMED images made available by the PDS Ring Node (Showalter et al., 2013) with Voyager 1 NAC image C1636902.

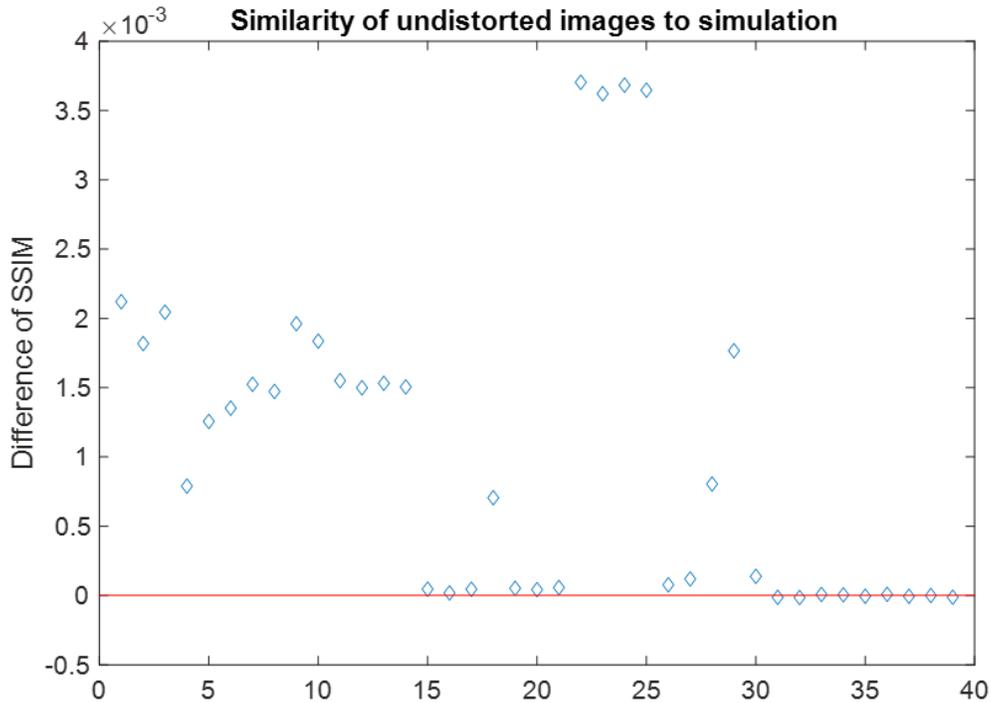


Figure 13: Values of the difference between the structural similarity index (ssim) comparing our undistorted images and a simulation and the ssim value comparing the GEOMED images and the same simulation. The absolute values of the index are around 0.99. The images represented here are the 39 Voyager images used in Belgacem et al. 2019 and listed in the supplementary.

240 We should note that our correction is only slightly better than the GEOMED images
 241 (images corrected for distortion by VICAR software). One way of looking at it is
 242 comparing the GEOMED image and our undistorted image to a simulation. When
 243 doing that, we find that our correction gives more similar images (quantified with the
 244 structural similarity index) to the simulation than the GEOMED. However, we are
 245 looking at differences in the index of the order of $1e-3$ and ssim values around 0.99.

246 *4.2. Camera pose*

247 For a complete validation of the camera pose, we look closely at the new predicted
 248 limb, after the different corrections. Although we are strictly looking at the limb here,

249 we do not only validate the correction of the spacecraft orientation: if the distortion
250 is not corrected or if the camera position is still wrong, it will also show as a poorly
251 corrected limb.

252 For each point on the limb of the target, we can trace two segments in the image
253 frame - one vertical and one horizontal. We only keep the one less tangent to the limb.
254 The relevant information is in the fully illuminated part of the limb (highlighted in
255 orange in fig. 14).

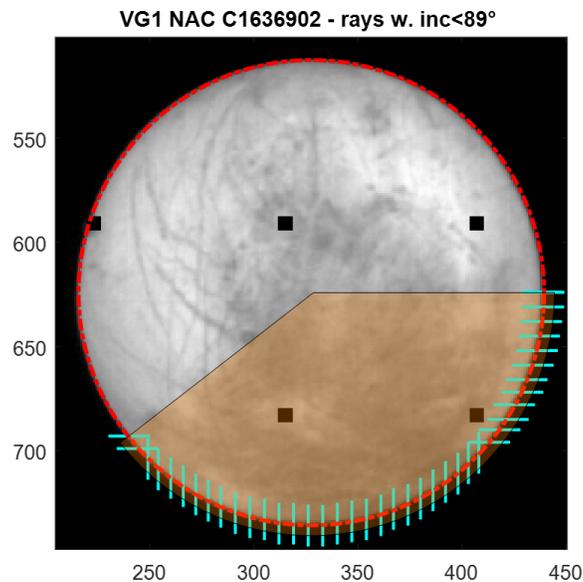


Figure 14: Example of a Voyager 1 NAC image with segments drawn on the illuminated part of the limb

256 Each of these segments gives a piece of information about the position of the limb.
257 If we visualize each of them individually we can see quite precisely how well the new
258 predicted limb fits the target on the image. The limb is supposed to be at the extremum
259 of the derivative of the segment: it is the strongest change from the illuminated target to
260 the blackness of the background sky. Fig. 15 shows the illuminated segments displayed
261 on fig. 14 compared to their simulated equivalent. Fig. 16 shows the derivatives.

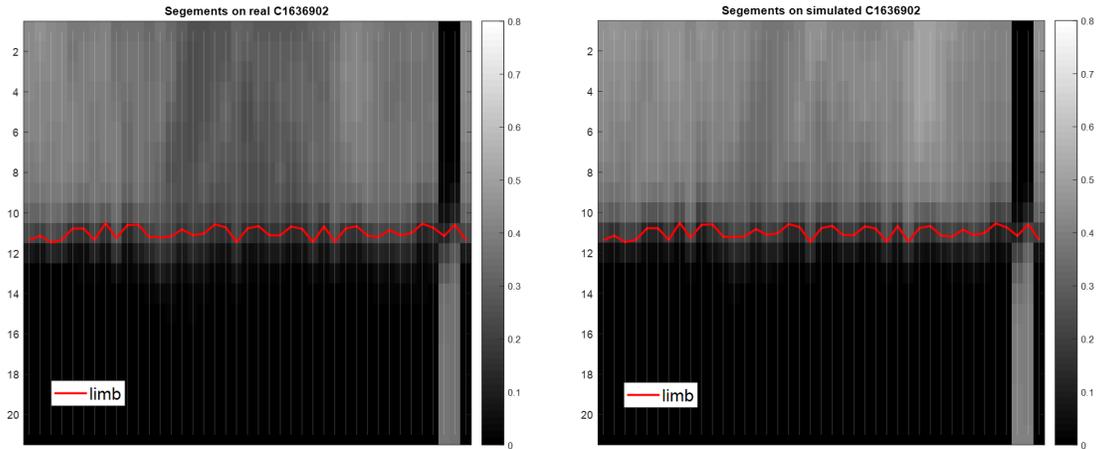


Figure 15: Comparison of limb segment between real image and simulation. In red is the newly corrected limb after metadata correction.

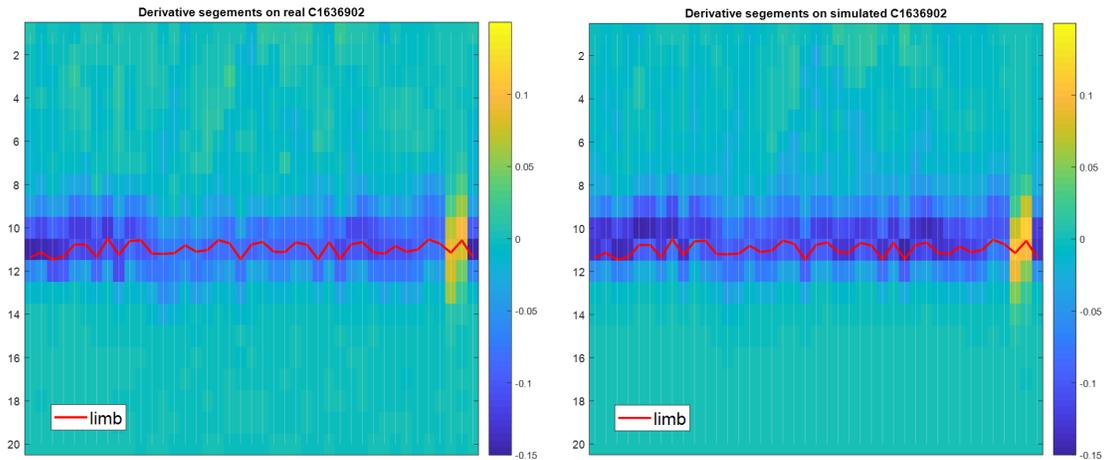


Figure 16: Comparison of derivatives of limb segment between real image and simulation. In red is the newly corrected limb after metadata correction.

262 The form of the derivative is not trivially described. A first order approximation
 263 would be a Gaussian but a few tests showed quickly that it was not enough. That is
 264 why we chose to compare each segment to its equivalent in the simulation. If the new

265 predicted limb fits perfectly, both derivatives should have the same extremum (fig. 17).
 266 However, if the camera pose is still off, this will show as a shift between simulation and
 267 reality both in the segment itself and its derivative (fig. 18).

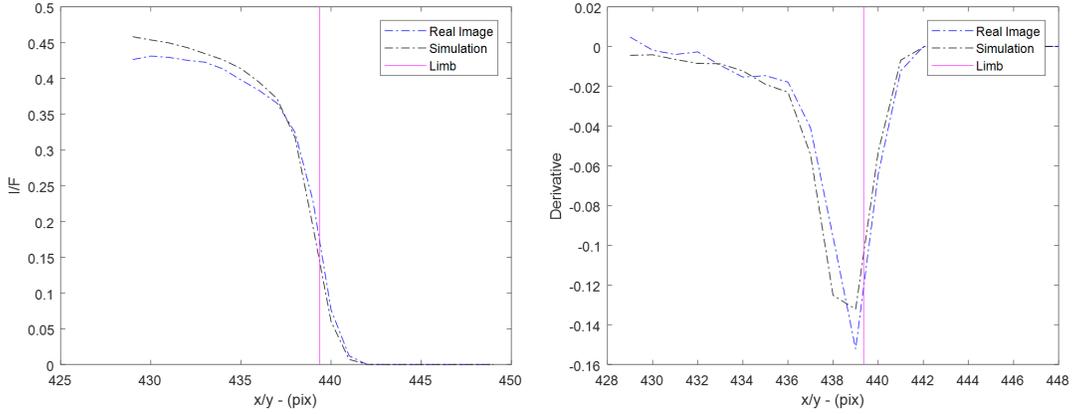


Figure 17: Example of a segment and its derivative after complete correction of camera pose

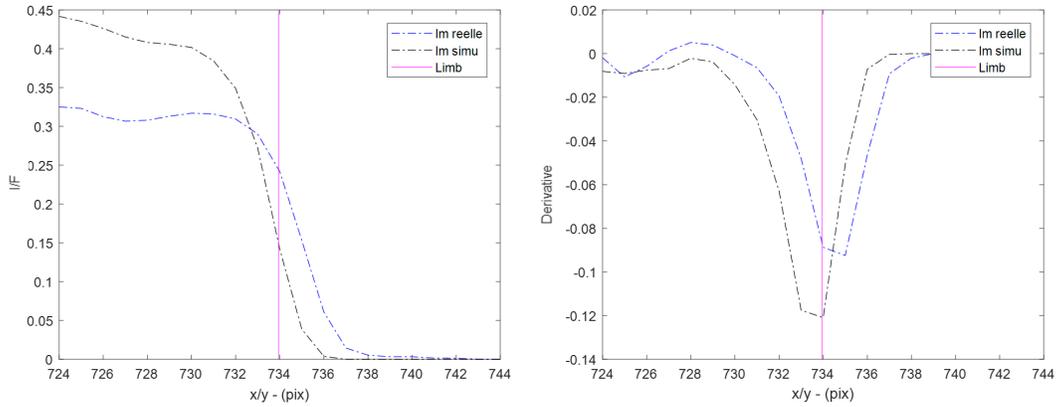


Figure 18: Example of a segment and its derivative with a remaining error in the camera pose. In red is the newly corrected limb after metadata correction.

268 5. Target's attitude

269 After correcting the camera pose (attitude + position with respect to Europa), some
 270 differences remain between the images and their simulations. We perform an optical

271 flow measurement to interpret these differences. At least some of them can be explained
 272 by an imprecision of the moons' attitude with respect to the camera. There is indeed
 273 a global movement that can be corrected by a slight rotation of the target - Europa
 274 here. The rotation around the boresight of the camera - not corrected so far - can also
 275 contribute to that effect.

276 5.1. 2D analysis - optical flow

277 An optical flow describes the apparent motion of an object in an image. We choose
 278 a local approach: for each pixel we define a region of interest consisting of a 13-pixel
 279 square box in the simulated image and search for the best normalized cross-correlation
 280 in a 21-pixel square box in the real image (fig. 19a) using equation 11.

$$\rho_{u,v} = \frac{\sum_{x,y} [f(x,y) - \bar{f}_{u,v}] [t(x-u, y-v) - \bar{t}]}{\sqrt{\sum_{x,y} [f(x,y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x-u, y-v) - \bar{t}]^2}} \quad (11)$$

281 Where:

- 282 • t is the ROI in the simulated image, \bar{t} is the mean of the ROI in the simulated
 283 image
- 284 • f is the search box in the real image, $\bar{f}_{u,v}$ is the mean of the search box in the
 285 equivalent ROI in the real image

286 Thus, for each pixel, we have a displacement vector pointing to the best local corre-
 287 lation between the simulated and real images (fig. 19b). At the image scale, we obtain a
 288 global pattern of displacement indicating in which direction the moon has to be rotated
 289 in the simulated image to better match the real one.

290 5.2. Computing the 3D solid rotation - Kabsch algorithm

291 We need to compute the correcting rotation associated with the movement pattern.
 292 From the optical flow, we have two sets of matching 2D points. We first project these

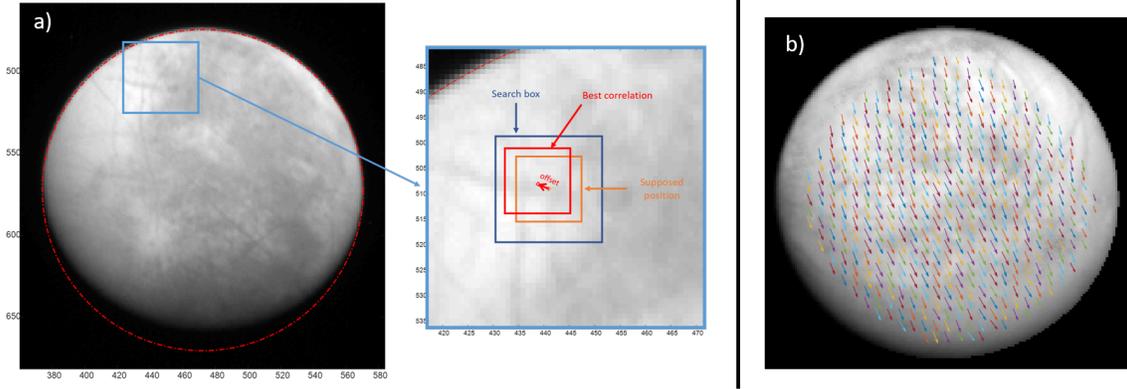


Figure 19: a) Definition of ROI and search box for a pixel b) Example of optical flow resulting from a simulated rotation of Europa

293 points to obtain two sets of 3D points. We choose to implement Kabsch algorithm
 294 (Kabsch, 1978) to compute the rotation that minimizes the RMSD between the two
 295 sets of points. Let's represent both sets of n points by matrices P and Q

$$P = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{pmatrix}_{set\#1} \quad Q = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{pmatrix}_{set\#2} \quad (12)$$

296 Then, we compute the covariance matrix $C = P^T Q$. The matrix C is not necessarily
 297 invertible, we thus need to use the single value decomposition. Let U , Σ and V be the
 298 matrices of this decomposition such as $C = U \Sigma V^T$. Finally, the rotation matrix that
 299 best matches the two sets of points P and Q is given by:

$$R = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(VU^T) \end{pmatrix} U^T \quad (13)$$

300 To ensure that R is expressed in a direct right-handed coordinate system, we need
 301 $\det(VU^T) > 0$. If it is not the case, we have to invert the sign of the last column of

302 matrix V before calculating R . This rotation matrix is the correction factor to apply
 303 to the target’s attitude.

304 We have to note that the choice of the texture is decisive in this approach. For
 305 instance, in the case of Europa, we used the color map (Jónsson, 2015) to produce the
 306 simulated images. If the map is erroneous, every measurement made in comparison
 307 to the simulations will be erroneous as well. We have identified very few patches that
 308 seem to be badly registered with respect to the rest of the map but do not affect the
 309 measurements overall.

310 6. Projections: camera to scene

311 After correction of all the metadata, we can safely project each pixel of the images
 312 onto the target to compute the corresponding coordinates (latitude and longitude) and
 313 observation geometry (incidence, emission and phase angles).

We are modeling Europa as an ellipsoid. Each point $X = \begin{pmatrix} x & y & z \end{pmatrix}^T$ on the surface
 verifies the equation:

$$(X - V)^T A (X - V) = 1 \tag{14}$$

314 Where:

- 315 • S : spacecraft position
- 316 • X : point on the ellipsoid
- 317 • V : center of the ellipsoid
- 318 • A : positive definite matrix parametrising the quadric

319 A is a parametrisation matrix in any arbitrary orientation of the quadric. In the
 320 principal axes of the ellipsoid, it can be simplified to $A = \begin{pmatrix} \frac{1}{r_e} & 0 & 0 \\ 0 & \frac{1}{r_e} & 0 \\ 0 & 0 & \frac{1}{r_p} \end{pmatrix}$ where r_e is the
 321 equatorial radius of the ellipsoid and r_p , the polar radius.

322 We chose to express every coordinate in the J2000 frame which means that we will
 323 use:

$$A = R_T^{-1} \begin{pmatrix} \frac{1}{r_e} & 0 & 0 \\ 0 & \frac{1}{r_e} & 0 \\ 0 & 0 & \frac{1}{r_p} \end{pmatrix} R_T \quad (15)$$

324 Where R_T is a matrix that transforms any coordinates in the target's fixed frame
 325 to J2000 - i.e. the target's attitude depending on time.

326 Each pixel of the detector has a line of sight - a 3-D vector. In order to project all
 327 the pixels onto the moon, we target to compute the intersection of these lines of sight
 328 with the ellipsoid modeling the planetary body. This is equivalent to solving equation
 329 14 after replacing X by:

$$X = S + kL \quad (16)$$

330 Where $k \in \mathbb{R}$ is the distance from the pixel to the target and L is a $3 \times N$ matrix of
 331 unitary vectors, each being the line of sight of a pixel on the detector.

We obtain:

$$(S + kL - V)^T A (S + kL - V) = 1 \quad (17)$$

332 We have a second degree equation to solve for k that - once developed - can be
 333 written:

$$(L^T AL)k^2 + (2S^T AL - 2L^T AV)k + (V^T AV - 2S^T AV + S^T AS - 1) = 0 \quad (18)$$

334 We can compute the determinant by:

$$\Delta = (2S^T AL - 2L^T AV)^2 - 4(L^T AL)(V^T AV - 2S^T AV + S^T AS - 1) \quad (19)$$

335 Three cases can arise:

- 336 • $\Delta < 0$: no solution, the line of sight doesn't intersect the ellipsoid, the pixel
337 doesn't see the target
- 338 • $\Delta = 0$: one solution, the pixel intersects the target on its exact edge
- 339 • $\Delta > 0$: two solutions, the line of sight intersects the ellipsoid twice, on the
340 spacecraft-facing side and on the other side of the target along the same axis.
341 In this case, we keep the closest point (spacecraft-facing face) which means the
342 lowest k .

343 Solving the equation gives us the exhaustive collection of pixels in a position to
344 "see" the moon. We still need to eliminate the pixels seeing the night side of the moon.
345 To do so, we compute the geometry of observation at each intersection and eliminate
346 all pixels seeing an area where the incidence angle is greater than 90° .

347 This approach shows a clear advantage compared to existing functions in SPICE
348 - SINCPT, ILLUMIN, RECLAT - that are not vectorized. A vectorized projection of
349 the entire 1024×1024 pixel grid of a New Horizons' LORRI image took 0.45 seconds
350 compared to a limiting 1 minute and 33 seconds using the SINCPT function in a loop.
351 We should mention that the PDS Ring-Moon systems node has also developed a Python
352 toolbox - OOPS - that simplifies the use of these SPICE functions (Showalter et al.,
353 2018).

354 7. Conclusion

355 We have developed a complete pipeline to process images and convert them into
356 usable and precise science products for a variety of applications. As an example of ap-
357 plication, we have used these tools in a regional photometric study of Europa (Belgacem
358 et al., 2019) for which an accurate projection of the individual pixels in the images was
359 crucial to obtain the right coordinates and geometry of observation. We successfully
360 ran the pipeline in its entirety on the full pertinent collection of 57 images taken with

361 New Horizons' LORRI and Voyager's ISS. An exhaustive list of the images used in
362 Belgacem et al. 2019 is available in the supplementary material. As a future work, we
363 will compute and make our corrected metadata available for the Europa images at this
364 link: <https://github.com/InesBlgcm/ImageProcessing>. We will also reach out to
365 SPICE experts to generate C-smithed kernels for the relevant data set. Our vectorized
366 solution for projecting pixels onto an ellipsoid target will also be very useful to estimate
367 the geometry efficiently.

368 We have to note that our approach is dependent on a reliable image renderer and
369 most of all a reliable texture for the target, especially for correcting the target's attitude.
370 Without these resources a less precise pointing correction would still be possible using
371 a projected ellipsoid in the field of view in place of a more thorough simulated image.

372 Another major hypothesis is to consider the ephemeris of the planetary bodies in-
373 volved to be perfectly known. An improved approach would also correct for planetary
374 ephemeris. This could be achieved with a more general use of a software such as
375 CAVIAR (Cooper et al., 2018) that is for now dedicated to correcting CASSINI's ISS
376 images. After a first correction based on background stars, our image processing ap-
377 proach would enable an improved knowledge of the ephemeris of the planetary bodies
378 in the field of view.

379 Although we have carried out this work with images of Europa, this approach should
380 be easily adaptable on any other target. We validate here the pipeline on images from
381 six different cameras, demonstrating its versatility. We also could imagine carrying out
382 a similar approach for small bodies as long as a precise shape model is available to
383 simulate our images.

384 **Acknowledgment**

385 This work is supported by Airbus Defence & Space, Toulouse (France) as well
386 as the "IDI 2016" project funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02

387 and the “Institut National des Sciences de l’Univers” (INSU), the Centre National
388 de la Recherche Scientifique (CNRS) and Centre National d’Etudes Spatiales (CNES)
389 through the Programme National de Planétologie. We would also like to thank Mark
390 Showalter for his insight on the Voyager data set as well as two anonymous reviewers
391 for their valuable comments and feedback.

392 **References**

- 393 Acton, C.H., 1996. Ancillary data services of NASA’s navigation and ancillary informa-
394 tion facility. *Planetary and Space Science* 44, 65–70. doi:10.1016/0032-0633(95)
395 00107-7.
- 396 Anderson, J., Sides, S., Soltesz, D., Sucharski, T., Becker, K., 2004. Modernization
397 of the integrated software for imagers and spectrometers, in: *Lunar and Planetary*
398 *Science Conference*.
- 399 Belgacem, I., Schmidt, F., Jonniaux, G., 2019. Regional study of europa’s photometry.
400 *Icarus* doi:10.1016/j.icarus.2019.113525.
- 401 Belton, M., Klaasen, K., Clary, M., Anderson, J., Anger, C., Carr, M., Chapman,
402 C., Davies, M., Greeley, R., Anderson, D., Bolef, L., Townsend, T., Greenberg, R.,
403 Head, J., Neukum, G., Pilcher, C., Veverka, J., Gierasch, P., Fanale, F., Ingersoll,
404 A., Masursky, H., Morrison, D., Pollack, J., 1992. The galileo solid-state imaging
405 experiment. *Space Science Reviews* 60. doi:10.1007/bf00216864.
- 406 Brochard, R., Lebreton, J., Robin, C., Kanani, K., Jonniaux, G., Masson, A., Despré,
407 N., Berjaoui, A., 2018. Scientific image rendering for space scenes with SurRender
408 software. 69th International Astronautical Congress (IAC), Bremen, Germany .
- 409 Cheng, A., 2014. NEW HORIZONS Calibrated LORRI JUPITER ENCOUNTER V2.0,
410 NH-J-LORRI-3-JUPITER-V2.0, ckinfotxt file, in: NASA Planetary Data System.

- 411 Cheng, A.F., Weaver, H.A., Conard, S.J., Morgan, M.F., Barnouin-Jha, O., Boldt,
412 J.D., Cooper, K.A., Darlington, E.H., Grey, M.P., Hayes, J.R., Kosakowski, K.E.,
413 Magee, T., Rossano, E., Sampath, D., Schlemm, C., Taylor, H.W., 2008. Long-
414 Range Reconnaissance Imager on New Horizons. *Space Science Review* 140, 189–215.
415 doi:10.1007/s11214-007-9271-6, arXiv:0709.4278.
- 416 Cooper, N.J., Lainey, V., Meunier, L.E., Murray, C.D., Zhang, Q.F., Baillie, K., Evans,
417 M.W., Thuillot, W., Vienne, A., 2018. The caviar software package for the astrometric
418 reduction of cassini ISS images: description and examples. *Astronomy & Astrophysics*
419 610, A2. doi:10.1051/0004-6361/201731713.
- 420 Edmundson, K.L., Cook, D.A., Thomas, O.H., Archinal, B.A., Kirk, R.L., 2012. JIG-
421 SAW: THE ISIS3 BUNDLE ADJUSTMENT FOR EXTRATERRESTRIAL PHO-
422 TOGRAMMETRY. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial*
423 *Information Sciences I-4*, 203–208. doi:10.5194/isprsannals-i-4-203-2012.
- 424 Grasset, O., Dougherty, M., Coustenis, A., Bunce, E., Erd, C., Titov, D., Blanc,
425 M., Coates, A., Drossart, P., Fletcher, L., Hussmann, H., Jaumann, R., Krupp,
426 N., Lebreton, J.P., Prieto-Ballesteros, O., Tortora, P., Tosi, F., Hoolst, T.V.,
427 2013. JUpiter ICy moons explorer (juice): An ESA mission to orbit ganymede
428 and to characterise the jupiter system. *Planetary and Space Science* 78, 1 – 21.
429 doi:10.1016/j.pss.2012.12.002.
- 430 Hartley, R., Zisserman, A., 2003a. Camera models, in: *Multiple View Geometry in*
431 *Computer Vision*. 2nd edition. Cambridge University Press. chapter 6, pp. 153–195.
- 432 Hartley, R., Zisserman, A., 2003b. Computation of the camera matrix p, in: *Multi-*
433 *ple View Geometry in Computer Vision*. 2nd edition. Cambridge University Press.
434 chapter 7, pp. 196–212.

- 435 Hartley, R., Zisserman, A., 2003c. Multiple View Geometry in Computer Vision. 2nd
436 edition. 2 ed., Cambridge University Press, New York, NY, USA.
- 437 Jonniaux, G., Gherardi, D., 2014. Robust extraction of navigation data from images for
438 planetary approach and landing, in: 9th International ESA Conference on Guidance,
439 Navigation & Control Systems.
- 440 Jonniaux, G., Kanani, K., Regnier, P., Gherardi, D., 2016. Autonomous vision-based
441 navigation for JUICE. 67th International Astronautical Congress (IAC) .
- 442 Jónsson, B., 2015. Mapping europa, in: [http://www.planetary.org/blogs/guest-](http://www.planetary.org/blogs/guest-blogs/2015/0218-mapping-europa.html)
443 [blogs/2015/0218-mapping-europa.html](http://www.planetary.org/blogs/guest-blogs/2015/0218-mapping-europa.html).
- 444 Kabsch, W., 1978. A discussion of the solution for the best rotation to relate two
445 sets of vectors. *Acta Crystallographica Section A* 34, 827–828. doi:10.1107/
446 s0567739478001680.
- 447 Laura, J., Rodriguez, K., Paquette, A.C., Dunn, E., 2018. AutoCNet: A python library
448 for sparse multi-image correspondence identification for planetary data. *SoftwareX*
449 7, 37–40. doi:10.1016/j.softx.2018.02.001.
- 450 Loza, A., Mihaylova, L., Bull, D., Canagarajah, N., 2007. Structural similarity-based
451 object tracking in multimodality surveillance videos. *Machine Vision and Applica-*
452 *tions* 20, 71–83. doi:10.1007/s00138-007-0107-x.
- 453 Luo, J., Konofagou, E.E., 2010. A fast normalized cross-correlation calculation method
454 for motion estimation. *IEEE Transactions on Ultrasonics, Ferroelectrics and Fre-*
455 *quency Control* 57, 1347–1357. doi:10.1109/tuffc.2010.1554.
- 456 Ma, J., Zhou, H., Zhao, J., Gao, Y., Jiang, J., Tian, J., 2015. Robust feature matching
457 for remote sensing image registration via locally linear transforming. *IEEE Transac-*

458 tions on Geoscience and Remote Sensing 53, 6469–6481. doi:10.1109/tgrs.2015.
459 2441954.

460 Mathworks, 2018. User guide, in: Matlab Image Processing Toolbox.

461 Oberst, J., 2004. Vertical control point network and global shape of io. Journal of
462 Geophysical Research 109. doi:10.1029/2003je002159.

463 Phillips, C.B., Pappalardo, R.T., 2014. Europa clipper mission concept: Exploring
464 jupiter’s ocean moon. Eos, Transactions American Geophysical Union 95, 165–167.
465 doi:10.1002/2014E0200002.

466 Reddy, B., Chatterji, B., 1996. An fft-based technique for translation, rotation, and
467 scale-invariant image registration. IEEE Transactions on Image Processing 5, 1266–
468 1271. doi:10.1109/83.506761.

469 Showalter, M., Gordon, M., Olson, D., 2013. VG1/VG2 JUPITER ISS PROCESSED
470 IMAGES V1.0 VGISS_5101-5214, in: NASA Planetary Data System.

471 Showalter, M.R., Ballard, L., French, R.S., Gordon, M.K., Tiscareno, M.S., 2018. De-
472 velopments in geometric metadata and tools at the pds ring-moon systems node, in:
473 Informatics and Data Analytics.

474 Sidiropoulos, P., Muller, J.P., 2018. A systematic solution to multi-instrument coreg-
475 istration of high-resolution planetary images to an orthorectified baseline. IEEE
476 Transactions on Geoscience and Remote Sensing 56, 78–92. doi:10.1109/tgrs.2017.
477 2734693.

478 Smith, B., Briggs, G., Danielson, G., Cook, A., Davies, M., Hunt, G., Masursky, H.,
479 Soderblom, L., Owen, T., Sagan, C., Suomi, V., 1977. Voyager imaging experiment.
480 Space Science Reviews 21. doi:10.1007/bf00200847.