POSITION ESTIMATION USING IMAGE DERIVATIVE

Daniele Mortari[*] **Francesco de Dilectis**[†] **and Renato Zanetti**[‡]

This paper describes an image processing algorithm to process Moon and/or Earth images. The theory presented is based on the fact that Moon hard edge points are characterized by the highest values of the image derivative. Outliers are eliminated by two sequential filters. Moon center and radius are then estimated by nonlinear least-squares using circular sigmoid functions. The proposed image processing has been applied and validated using real and synthetic Moon images.

INTRODUCTION

This paper completes a study, previously performed [1], to process cropped Moon and/or Earth images. The theory presented, which has been applied to real and synthetic Moon images, is based on the fact that the edge points are characterized by the highest values of the image derivative.
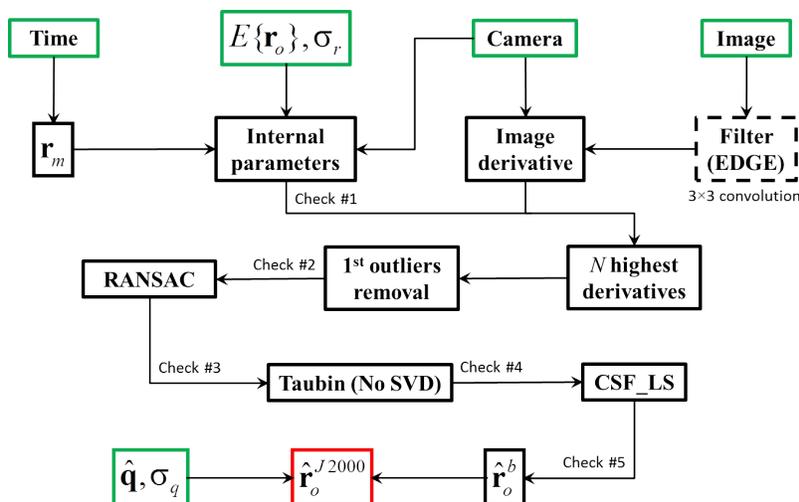

Figure 1. Image Processing General Flowchart

The content of this paper follows the general flowchart given in Fig. 1 which summarizes the main processes. These steps are explained in details in the following sections. The proposed algorithm has the following steps:

[*]Professor, 746C H.R. Bright Bldg., Aerospace Engineering, Texas A&M University, College Station, TX 77843-3141, AAS Fellow, AIAA Associate Fellow. IEEE Senior member. E-mail: MORTARI@TAMU.EDU

[†]PhD Graduate Student, 301B Reed McDonald, Aerospace Engineering, Texas A&M University, College Station, TX 77843-3141. E-mail: F.DE.DILECTIS@NEO.TAMU.EDU

[‡]GN&C Autonomous Flight Systems Engineer, Aeroscience and Flight Mechanics Division, EG6, 2101 NASA Parkway. NASA Johnson Space Center, Houston, Texas 77058. AIAA Senior Member.

- The geometrical illumination conditions and the expected size of the observed Moon are first computed. These data allow to understand if the Moon is new, crescent, gibbous, or full.

- The image differentiation is obtained with a fourth order Richardson extrapolation. An analytical surface example is provided quantifying the accuracy gain obtained using Richardson extrapolation.

- The pixels of the gradient image thus obtained, are sorted in descending mode. This means that the first set of pixels have high probability of being edge points. Outliers are then eliminated by two sequential filters. The first filter validates an edge pixel based on eigenanalysis of a small gradient box around the pixel and by investigation of the same box in the original image. The second filter is an improved version of the RANSAC algorithm.

- The edge point selected are then used to obtain a first estimate of the Moon center and radius. Taubin best fitting [6] (for circles) and the Fitzgibbons algorithm (for ellipses) are used. For the current application, the eccentricity of the Earth is so small that the ellipse angular orientation is a parameter ill defined to estimate (using noisy edge points). In addition, the presence of the atmosphere and high-altitude clouds make the problem even more inaccurate for Earth images. Therefore the orientation of the Earth ellipse is assumed known from the attitude of the camera.

- The final estimation of Moon center and radius is performed by non-linear least-squares using a circular sigmoid function to describe the Moon edge transition.

ILLUMINATION AND IMAGE PROCESSING PARAMETERS

The image processing software require the computation of some internal parameters. With reference to Fig. 2, these parameters depend on the Moon size on the imager, p_r (cannot be smaller than a given value) and on the illumination geometry, p_i (to define the separation between new and crescent Moon and between gibbous and full Moon). In this section, based on the geometry (Moon and Sun positions, J2000) and on a rough estimate of Spacecraft position (J2000), the conditions when the Moon is geometrically considered new, full, gibbous, or crescent are here derived.
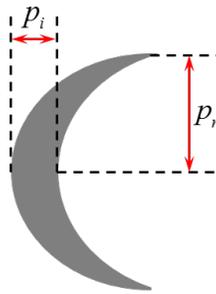

Figure 2. Expected Moon radius (p_r) and illumination parameter (p_i)

The observed Moon radius (in pixel) in the imager, p_r, as well as the illumination size along the axis of symmetry, p_i, dictate the values of important internal driving parameters used in the image processing code.

To estimate the full or new Moon cases, we use the following input data. Let $\hat{r}_s$ be Sun-to-Moon rays direction (unit-vector), r_o be (estimated) Spacecraft-to-Moon vector, and R_m the Moon radius. From these data we can compute

$$p_r = \frac{f \, R_m}{d \, \sqrt{|r_o|^2 - R_m^2}} \qquad \text{(pixels)} \qquad (1)$$

the illumination angle is then computed as

$$\cos \vartheta = \hat{r}_s \cdot \hat{r}_o \qquad (2)$$

from which ϑ is derived. The angle α_m can be derived from the equation

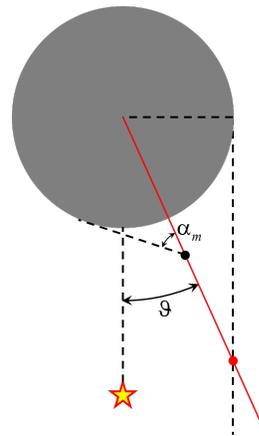$$|r_o| \, \sin \alpha_m = R_m \qquad (3)$$

Figure 3. Full/New Moon geometry

Full or new Moon

while the value of the critical distance, r_{cr}, can be obtained by

$$r_{cr} \, \sin \vartheta = R_m \qquad (4)$$

Therefore,

$$\text{if} \quad |r_o| < r_{cr} \quad \text{and} \quad \begin{cases} \vartheta < \pi/2 & \text{then} \quad p_i = 2p_r \quad \text{(Full Moon)} \\ \vartheta > \pi/2 & \text{then} \quad p_i = 0 \quad \text{(New Moon)} \end{cases} \qquad (5)$$

Gibbous Moon

Gibbous Moon will occur when

$$|r_o| > r_{cr} \qquad \text{and} \qquad \vartheta < \frac{\pi}{2}. \qquad (6)$$

When these conditions are satisfied then (see Fig. 4) the p_i illumination parameter can be computed using the following equations

$$x^2 = |r_o|^2 + R_m^2 - 2 \, |r_o| \, R_m \, \sin \vartheta \qquad \rightarrow \qquad x \quad (7)$$

and

$$\frac{\sin \alpha_e}{R_m} = \frac{\cos \vartheta}{x} \qquad \rightarrow \qquad \alpha_e \qquad (8)$$

Figure 4. Gibbous Moon geometry

and

$$\frac{\tan \alpha_m}{p_r} = \frac{\tan \alpha_e}{r_e^p} \qquad \rightarrow \qquad r_e^p = p_r \, \frac{\tan \alpha_e}{\tan \alpha_m} \qquad (9)$$

and, therefore,

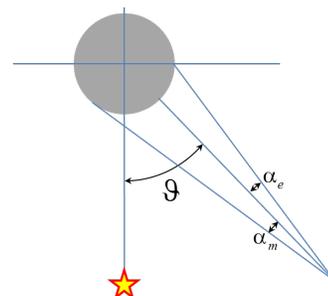$$p_i = p_r + r_e^p = \left(1 + \frac{\tan \alpha_e}{\tan \alpha_m}\right) p_r \qquad (10)$$

3

Crescent Moon will occur when

$$|\boldsymbol{r}_o| > r_{cr} \qquad \text{and} \qquad \vartheta > \frac{\pi}{2}. \qquad (11)$$

When these conditions are satisfied then (see Fig. 5) the p_i illumination parameter can be computed using the following equations

$$x^2 = |\boldsymbol{r}_o|^2 + R_m^2 - 2\,|\boldsymbol{r}_o|\,R_m\,\sin\vartheta \qquad \rightarrow \qquad x \quad (12)$$

and

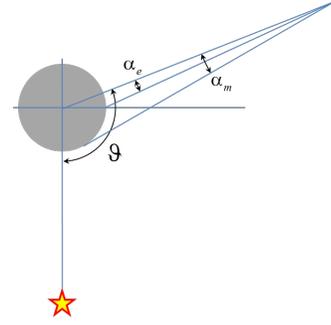$$\frac{\sin\alpha_e}{R_m} = \frac{-\cos\vartheta}{x} \qquad \rightarrow \qquad \alpha_e \qquad (13)$$

Figure 5. Crescent Moon geometry

Crescent Moon

and

$$\frac{\tan\alpha_m}{p_r} = \frac{\tan\alpha_e}{r_e^p} \qquad \rightarrow \qquad r_e^p = p_r\,\frac{\tan\alpha_e}{\tan\alpha_m} \qquad (14)$$

and, therefore,

$$p_i = p_r - r_e^p = \left(1 - \frac{\tan\alpha_e}{\tan\alpha_m}\right)p_r \qquad (15)$$

GRADIENT-BASED IMAGE PROCESSING ALGORITHM

The gradient-based image processing approach is based on the fact that the part of the image with strongest variation of gray-tones is the Moon observable hard edge. Pixels with highest derivatives are the most likely to belong to the hard edge. Some of the selected bright gradient pixels do not belong to the Moon hard edge (outliers). These outliers are eliminated prior to using a best fitting algorithm to determine the sought geometrical properties.

Image differentiation

The gradient of a discrete function (the imager) is computed by numerical differentiation. In our case, the gradient is computed using four point central difference with a single Richardson extrapolation.

The row and column four point central partial differences computed at pixel $[r, c]$ are

$$
\begin{cases}
g_r'(r,c) &= \dfrac{8[I(r+1,c) - I(r-1,c)] - [I(r+2,c) - I(r-2,c)]}{12\,h} \\[4mm]
g_c'(r,c) &= \dfrac{8[I(r,c+1) - I(r,c-1)] - [I(r,c+2) - I(r,c-2)]}{12\,h}
\end{cases}
\qquad (16)
$$

These derivatives are accurate with order h^4, where h is the pixel dimension. The image gradient is then computed as

$$g'(r,c) = \sqrt{g_r'^2(r,c) + g_c'^2(r,c)} \qquad (17)$$

4

A single Richardson extrapolation requires computing Eq. (17) one time with step $2\,h$ (getting g'_{2h}) and one time with step h (getting g'_h). Therefore, the true derivative can be written as

$$\left\{ \begin{array}{ccc} g'|_{true} & = & g'_{2h} + c\,(2h)^4 \\ g'|_{true} & = & g'_h + c\,h^4 \end{array} \right. \tag{18}$$

Equating these two equation we obtain

$$g'_h - g'_{2h} = c\,h^4(2^4 - 1) \quad \rightarrow \quad c\,h^4 = \frac{1}{2^4 - 1}\left(g'_h - g'_{2h}\right) \tag{19}$$

and therefore

$$\boxed{g'|_{true} = g'_h + \frac{1}{2^4 - 1}\left(g'_h - g'_{2h}\right)} \tag{20}$$

An example of application of a single Richardson extrapolation is done using the six-hump Camel back function

$$f(x, y) = \frac{x^2}{4}\left(4 - \frac{21}{40}x^2 + \frac{x^4}{8}\right) + \frac{xy}{4} + y^2\left(\frac{y^2}{4} - 1\right) \tag{21}$$
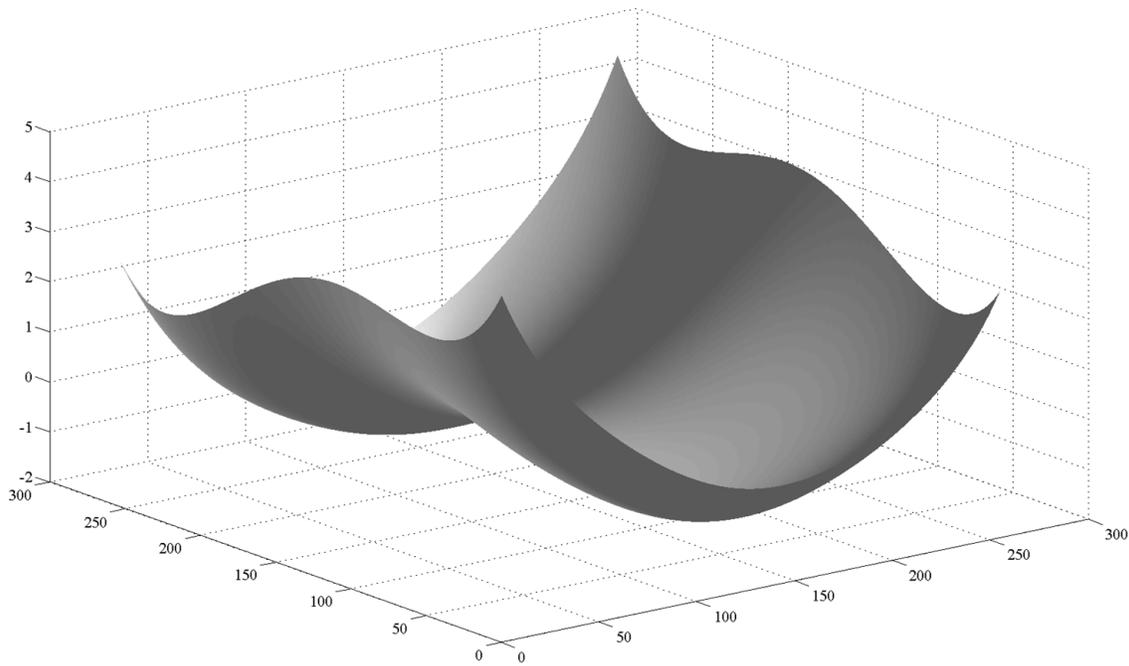

Figure 6. Six-hump Camel back function

Fig. 7 shows the accuracy of a single Richardson extrapolation

In this figure the errors with respect to the true analytic values are show for the four cases:

- Top-left: 2-point central differentiation and no Richardson extrapolation;
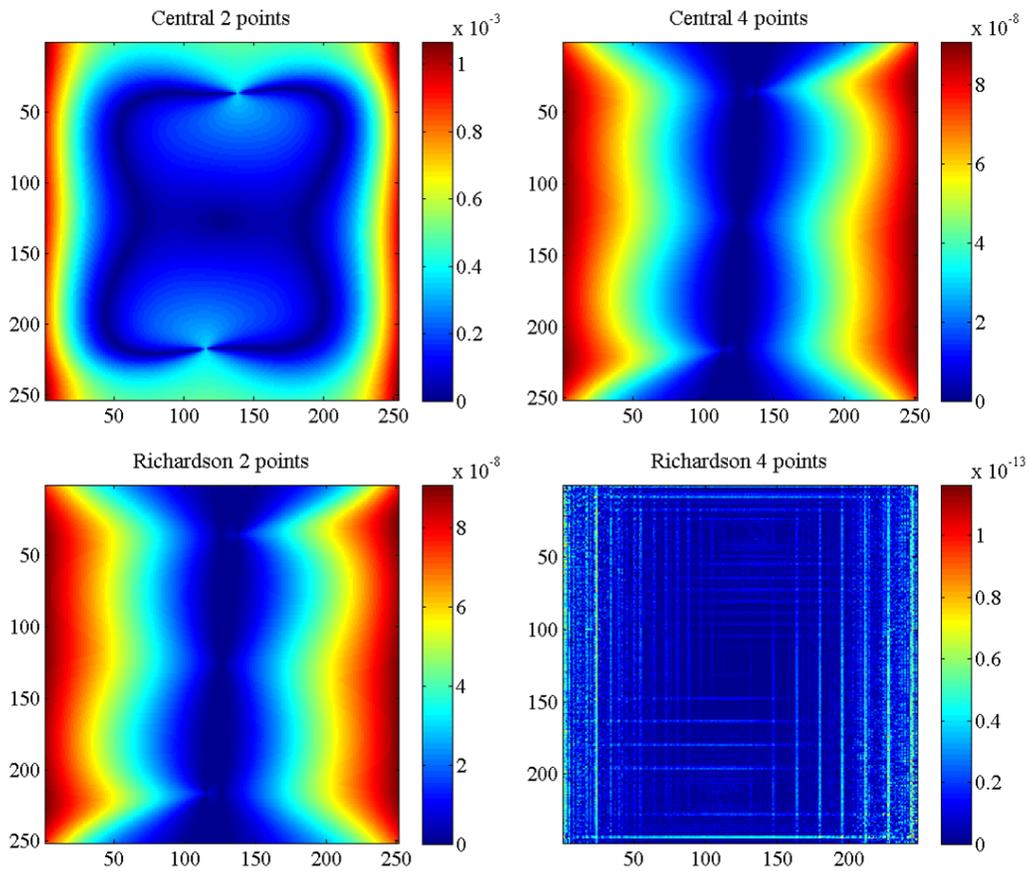
5

Figure 7. Accuracy results for the six-hump Camel back function example

- Top-right: 4-point central differentiation and no Richardson extrapolation;

- Bottom-left: 2-point central differentiation and one Richardson extrapolation;

- Bottom-right: 4-point central differentiation and one Richardson extrapolation;

These results clearly show that: 1) the accuracy provided by 4-point central differentiation and no Richardson extrapolation is equivalent to that provided by 2-point central differentiation and one Richardson extrapolation and 2) the 4-point central differentiation and one Richardson extrapolation provides almost the machine error accuracy. For this reason the 4-point central differentiation and one Richardson extrapolation is selected to evaluate which are the pixels with greatest gradient.

The image gradient pixels are sorted in descending mode. The pixels with highest gradient values (edge pixels) are, therefore, concentrated as the first sorted elements. Unfortunately, some high gradient pixels (outliers) also appear on the Moon body (because of the illumination and local topology) or external (e.g., some bright stars or visible planets). Therefore, the outliers must be eliminated. Two sequential approaches have been implemented in order to eliminate the outliers.

Box-based outliers Identification

The first approach to remove outliers is based on local analysis around the pixels being considered as potential edge pixels (pixels with the highest gradient values). With reference to Fig. 8, this approach is based on using a mask size (the figure shows a 7×7 mask) whose size depend on the expected Moon dimension in pixels, p_r.

In Fig. 8 the central pixel (in blue) is the one being considered as potential pixel on the Moon edge. Using all the pixels of the gradient mask as lumped masses, the eigenvalues and eigenvectors of the inertia matrix are computed. The eigenvalues ratio (min/max) must be lower than a given threshold (e.g., 0.3), the eigenvector associated to the minimum eigenvalue (red line) in inclined by an angle γ with respect to the row axis direction. In the figure, the pixels marked as green should both have high derivative values. The pixels marked as red, on the other hand should have very different gray tones.
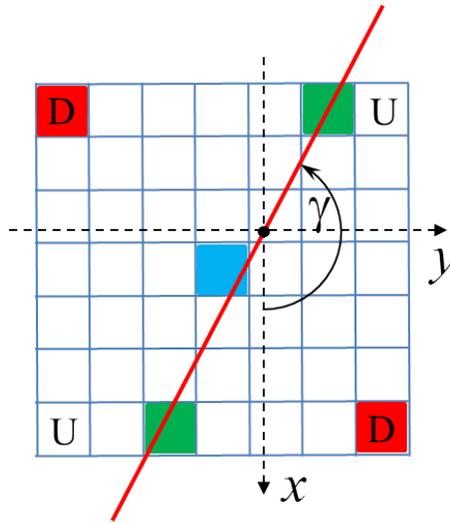

Figure 8. Outliers: first approach

RANSAC for circle and ellipse

A standard algorithm to eliminate outliers is called RANSAC which is an abbreviation for "RANdom SAmple Consensus". The RANSAC algorithm [2] is an algorithm for robust fitting of models in the presence of many data outliers. The algorithm is very simple and can be applied ti lines, circle, ellipse, etc. Given a fitting problem with n data points and unknown m parameters $\boldsymbol{x}$, estimate the parameters. For example, for lines $m = 2$, circles $m = 3$, and for ellipse $m = 5$. Starting with $N_{\min} = n$, the classic RANSAC algorithm perform N_{test} times the following steps:

1. selects m data points at random;

2. estimates the m parameter, meaning to find the curve passing through these m points;

3. counts how many data items are located at a distance greater than a minimum value $d_{\min}$ from the curve. They are N_k.

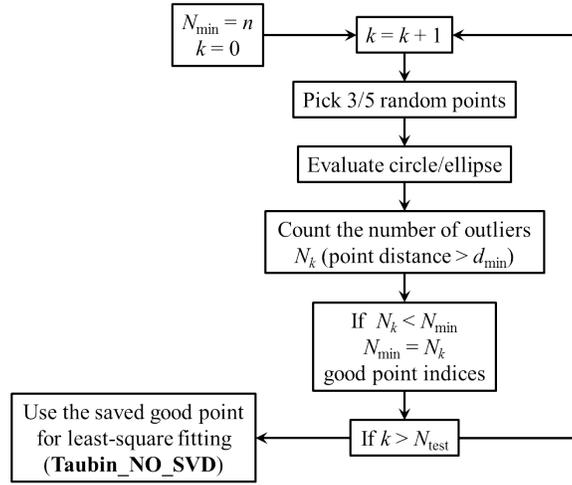4. if $N_k < N_{\min}$ then set $N_{\min} = N_k$.

Figure 9. RANSAC flowchart

One of the limitations of the RANSAC algorithm is the random procedure in which the points are selected. The random selection approach may persistently select outliers. To improve RANSAC, a technique (originally developed for Pyramid Star-Identification [3]) to select always different triads with indices continuously changing, has been adopted.

Best fitting for Circle and Ellipse

Once the pixels in the image have been selected, the algorithm must determine the geometric characteristics of the observed body, that is, radius and center for the Moon, considered a sphere, and axes and center for the Earth, considered an ellipsoid of revolution. Feature recognition for conic curves is based essentially on best fitting, which are typically grouped in two categories:

Geometric fit tries to minimize the geometric distance between the data points and the conic. In general this method is considered the most accurate, however it does not admit closed solution and requires the use of iterative algorithms.

Algebraic fit tries to minimize the "algebraic distance", i.e. the implicit conic equation $F(x, y) = 0$. Many methods based on this approach have been developed and they differ in accuracy, stability and complexity. They generally give results less accurate than the geometric fit, but on the other hand have closed form solutions and therefore can be implemented in a much simpler way.

Taubin for circles

Best fitting algorithms based on geometric fit are iterative methods requiring an initial guess, and their reliability and accuracy is ultimately solely dependent in the iterative scheme implemented. Typically, the Levenberg-Marquardt scheme is considered the most accurate. However, given the need to keep the computational load on-board at a minimum, algorithms based on algebraic fit seem a better choice, even if their accuracy is slightly worse. Among these, the Taubin [6] method is used and described in further detail.

Given the circle implicit equation as

$$a\left(x^2 + y^2\right) + b\,x + c\,y + d = a\,z + b\,x + c\,y + d = 0 \tag{22}$$

Method	Geometric (G) Algebraic (A)	Initial guess (Y/N)	Notes
Levenberg-Marquardt	G	Y	Iterative, considered the "best"
Trust Region	G	Y	Improvement on the L-M
Chernov-Lesort	G	Y	Guaranteed convergence
Kåsa	A	N	Fastest. Biased towards small radii
Pratt	A	N	Improvement on Kåsa
Taubin	A	N	Reduces to a 3×3 eigenvalue problem
Kukush-Markovsky-van Huffel	A	N	Works best with big dataset

Table 1. Summary of Best fitting algorithms for circle

the Taubin Best Fit minimizes the function:

$$\mathcal{F}_T(a, b, c, d) = \frac{n \sum_i (a\, z_i + b\, x_i + c\, y_i + d)^2}{\sum_i (4a^2\, z_i + 4ab\, x_i + 4ac\, y_i + b^2 + c^2)} \tag{23}$$

which is equivalent to minimize

$$\mathcal{F}_T(a, b, c, d) = \sum_i (a\, z_i + b\, x_i + c\, y_i + d)^2 \tag{24}$$

subject to the constraint

$$4a^2\, \bar{z} + 4ab\, \bar{x} + 4ac\, \bar{y} + b^2 + c^2 = 1 \tag{25}$$

where the bar sign indicates simple average of the variables.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i, \qquad \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i, \qquad \text{and} \qquad \bar{z} = \frac{1}{n} \sum_{i=1}^{n} (x_i^2 + y_i^2) = \frac{1}{n} \sum_{i=1}^{n} z_i. \tag{26}$$

One important property of this method is that the results are independent of the reference frame. By reformulating this problem in matrix form, it becomes apparent how it is equivalent to a generalized eigenvalue problem. Indeed, define

$$A = \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix}, \quad Z = \begin{bmatrix} z_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ z_n & x_n & y_n & 1 \end{bmatrix}, \quad \text{and} \quad M = \frac{1}{n} Z^{\mathsf{T}} Z = \begin{bmatrix} \bar{z}\bar{z} & \bar{z}\bar{x} & \bar{z}\bar{y} & \bar{z} \\ \bar{z}\bar{x} & \bar{x}\bar{x} & \bar{x}\bar{y} & \bar{x} \\ \bar{z}\bar{y} & \bar{x}\bar{y} & \bar{y}\bar{y} & \bar{y} \\ \bar{z} & \bar{x} & \bar{y} & 1 \end{bmatrix}, \tag{27}$$

then the best fitting problem can be rewritten as

$$A^{\mathsf{T}} M A = 0 \quad \text{subject to} \quad A^{\mathsf{T}} N A = 1, \tag{28}$$

where

$$N = \begin{bmatrix} 4\bar{z} & 2\bar{x} & 2\bar{y} & 0 \\ 2\bar{z} & 1 & 0 & 0 \\ 2\bar{y} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{29}$$

Therefore, the optimization problem is equivalent to the following generalized eigenvalue problem:

$$M A = \eta N A. \tag{30}$$

Since for each solution, $[A, \eta]$

$$A^{\mathsf{T}} M A = \eta A^{\mathsf{T}} N A = \eta \tag{31}$$

then to minimize $A^{\mathsf{T}} M A$ the smallest positive eigenvalue should be chosen. Because the matrix N has rank three, such eigenvalue can be found by explicitly solving the cubic characteristic equation without the necessity to implement SVD or other more sophisticated algorithms.

Fitzgibbons for ellipse

The problem of best fitting for an ellipse is sensibly more complicated and somewhat less studied than the circle. Firstly, compared to the three parameters necessary to describe the circle (x, y, R), a generic ellipse is defined by 5 parameters (x, y, a, b, θ). Moreover, especially for very small values of the eccentricity, the inclination is very hard to discern and the error associated with it is several orders of magnitude greater than for the other parameters.

Because the Earth itself, when considered an ellipsoid of revolution, has very small eccentricity, its planar projections are also almost circular, with ratio between the axes varying between 1 and 0.9966 depending on the latitude of the observer. We thus expect the highest errors in inclination estimation.

While geometric fit methods exist for the ellipse, they tend to not be very accurate and efficient due to the complicated formulation required. An non-linear Least Square approach, using the Jacobian of an ellipse in canonical form has been attempted; however, this method has proven itself not accurate enough for the requirements, and extremely slow to converge, even when advanced algorithms, like Levenberg-Marquardt are used. On the other hand, several studies ([4],[5]) have explored and improved the algebraic fit approach, with promising results. In particular, we have selected the Improved Fitzgibbon method, which reduces to solving a 3×3 eigenvalue problem.

To show this, consider the implicit equation of a conic:

$$f(a, b, c, d, e, f) = ax^2 + bxy + cy^2 + dx + ey + f = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{a} = 0. \tag{32}$$

For this equation to represent an ellipse, it must be $b^2 - 4ac < 0$, but with proper rescaling of the coefficient, which does not change the final result, it can be rewritten as $b^2 - 4ac = 1$. Therefore the problem can be rewritten in matrix form as:

$$\min_{\boldsymbol{a}} \boldsymbol{a}^{\mathsf{T}} D^{\mathsf{T}} D \boldsymbol{a} = \min_{\boldsymbol{a}} \boldsymbol{a}^{\mathsf{T}} S \boldsymbol{a} \qquad \text{subject to} \qquad \boldsymbol{a}^{\mathsf{T}} C \boldsymbol{a} = 1. \tag{33}$$

where

$$D = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_n y_n & y_n^2 & x_n & y_n & 1 \end{bmatrix}, \quad S = D^{\mathsf{T}} D, \quad \text{and} \quad C = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{34}$$

Solving this optimization problem leads to $S\boldsymbol{a} = \lambda C \boldsymbol{a}$ subject to $\boldsymbol{a}^{\mathsf{T}} C \boldsymbol{a} = 1$, and the solution is the eigenvector corresponding to the minimum positive eigenvalue. However, solving the problem

in this form leads to numerical instability. Instead, it is possible to decouple the quadratic part of D from the linear part, to obtain a simplified formulation that, together with being stable, has the added advantage of being of dimension 3 and thus the subsequent eigenanalysis can be performed without using SVD or similar. Therefore, by defining

$$D = [D_1, \; D_2], \quad \text{where} \quad D_1 = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n y_n & y_n^2 \end{bmatrix}, \quad \text{and} \quad D_2 = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}, \tag{35}$$

and where

$$S = \begin{bmatrix} S_1 & S_2 \\ S_2^{\mathrm{T}} & S_3 \end{bmatrix}, \quad \text{where} \quad \begin{cases} S_1 & = D_1^{\mathrm{T}} D_1 \\ S_2 & = D_1^{\mathrm{T}} D_2 \\ S_3 & = D_2^{\mathrm{T}} D_2 \end{cases} \quad \text{and} \quad C_1 = \begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix}. \tag{36}$$

The final system of equation assumes the form:

$$M \boldsymbol{a}_1 = C_1^{-1}(S_1 - S_2 S_3^{\mathrm{T}} S_2) \boldsymbol{a}_1 = \lambda_{\min} \boldsymbol{a}_1 \qquad \text{where} \qquad \boldsymbol{a}_1 = \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} \tag{37}$$

and

$$\boldsymbol{a}_2 = \begin{Bmatrix} d \\ e \\ f \end{Bmatrix} = -S_3^{\mathrm{T}} S_2 \, \boldsymbol{a}_1. \tag{38}$$

This version of the Fitzgibbon algorithm has been implemented with an explicit cubic solver for the characteristic equation.

NUMERICAL TESTS

To support the capability of the developed image processing code, three numerical examples of partially observed Moon are here presented. All these tests have been performed with no information about time, camera data, and observer, Moon, and Sun positions. The image derivative is obtained using the 4-point central approach with no Richardson extrapolation. These information clearly help to define the best values of the internal parameters. Since this help is missing, the internal parameters have been adopted with same values in all three tests. These values are:

1. Maximum number of highest derivatives selected: $N = 1500$;

2. Box-based outliers elimination: box size = 21×21;

3. Box-based outliers elimination: derivative ratio > 0.6;

4. Box-based outliers elimination: corner ratio < 0.3;

5. Box-based outliers elimination: eigenvalue ratio < 0.3;

6. RANSAC: minimum distance $d_{\min} = 4$ pixels;

7. RANSAC: number of tests = 100.

Example #1: Real image, four-times cropped

The original image of example #1, shown in Fig. 10, was intentionally cropped from a real image (taken from Earth) to validate the capability of the software to face a multiple cropped Moon images. Four times cropped is the also the maximum number of cropped segments possible. The four figures of Fig. 10 show the four main phases of the image processing. Once the image differentiation has been computed, the first 1,490 points with highest derivatives have been selected. This number differs from $N = 1500$ because the image is small and the number of points selected is also a function of the total number of pixels ($N = 1500$ is also the maximum number).
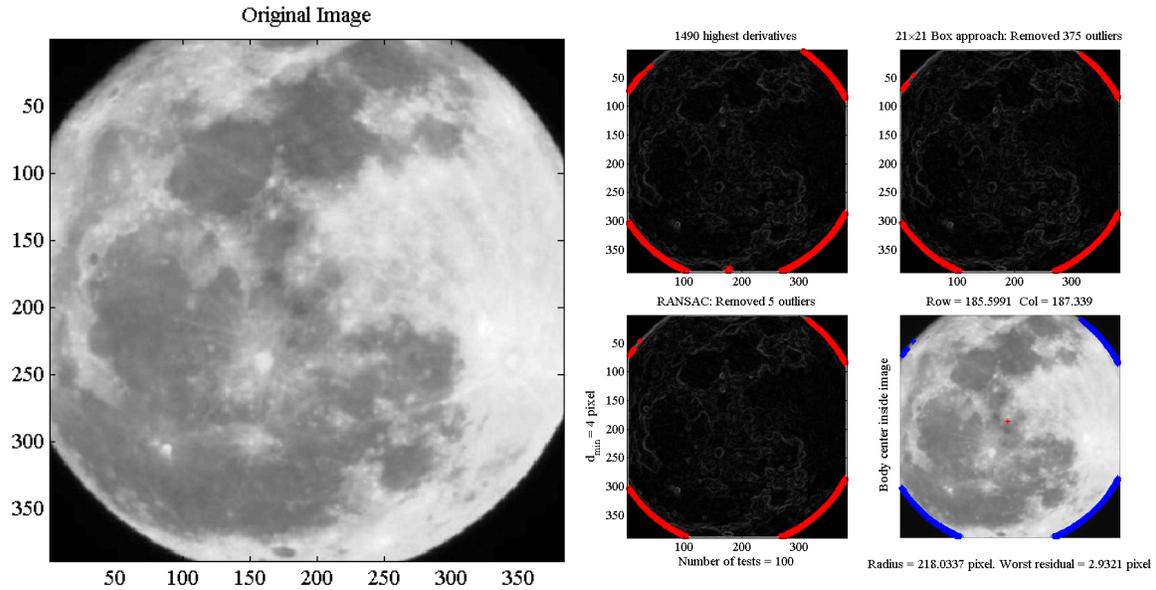
Figure 10. Example #1: Original image

Figure 11. Example #1: Image processing phases

The box-based outliers elimination approach discarded 375 points based on the results shown in Fig. 12 while RANSAC eliminated additional 5 points. The resulting points are shown in the bottom right figure of Fig. 11 using blue markers.

The bottom right figure of Fig. 11 shows the results obtained by Taubin fitting approach to estimate the circle parameters (Moon center and radius). The maximum distance from the estimated circle (worst residual) of the selected points was 2.9 pixels. The Taubin estimation gives a Moon center located at row 185.56 and column 187.34 and radius 218.03. Using this estimate, the set of points shown by red and blue markers in Fig. 13 over of Moon edge have been selected. These points are then used by CSF-LS to perform the nonlinear least-square estimation of Moon center and radius using circular sigmoid function. The new estimate implies a variation of -0.7, -0.5, and 0.4 pixels for row and column center coordinates and for radius, respectively.
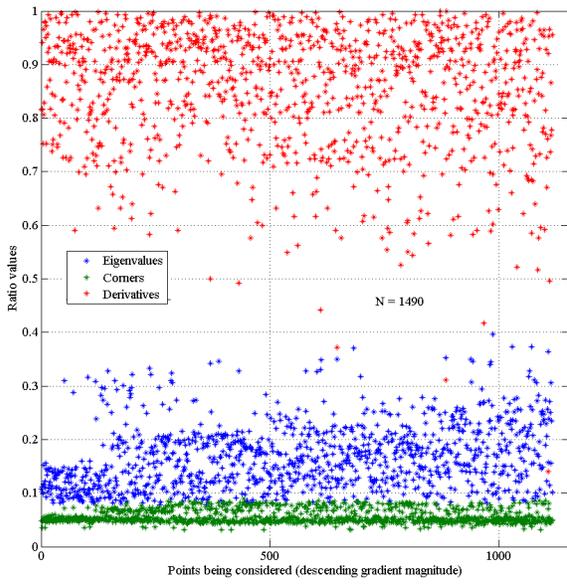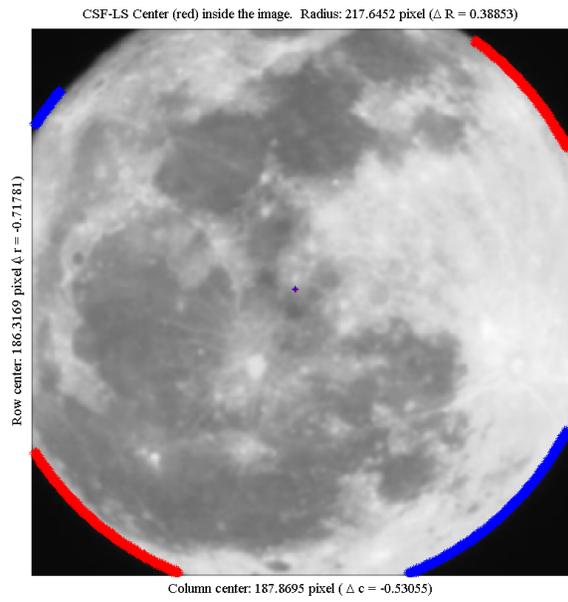
Figure 12. Box-based outliers statistics
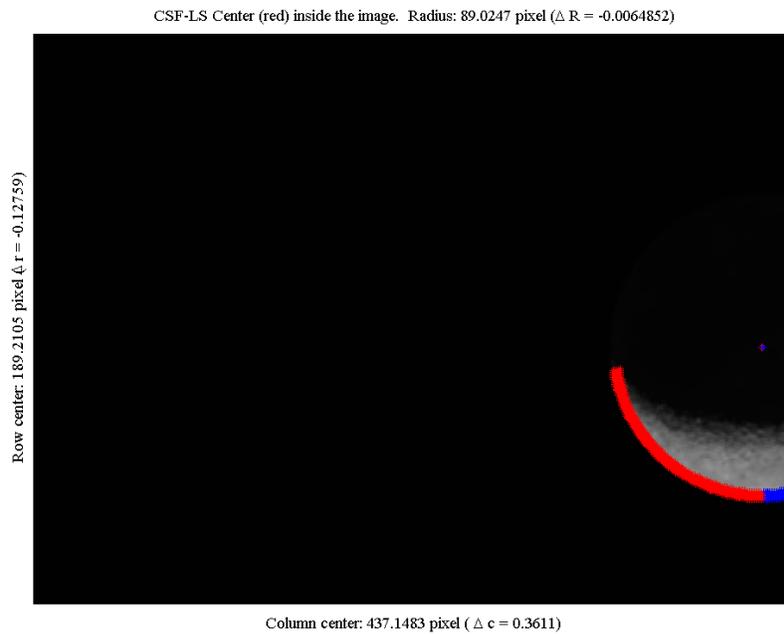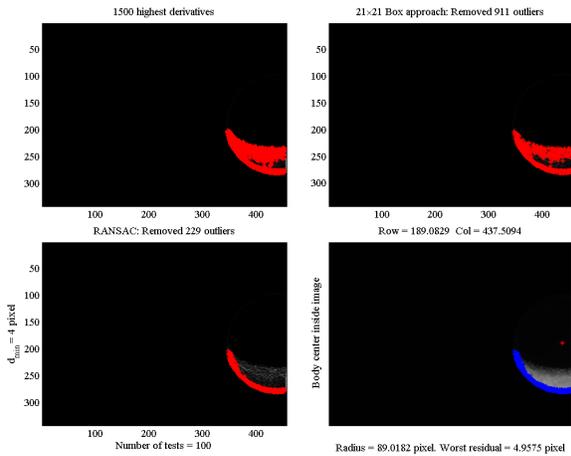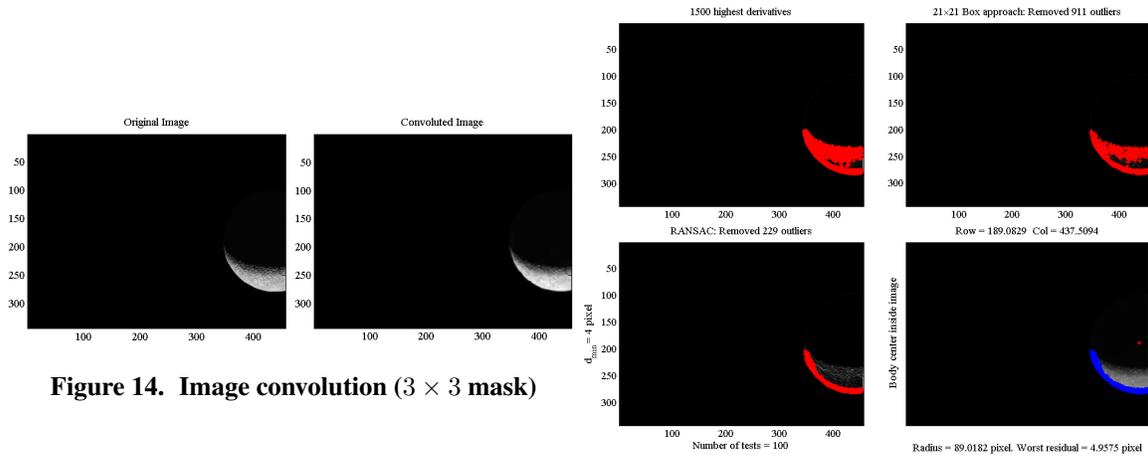

Figure 13. Example #1: CSF-LS final estimation

13

Example #2: Synthetic image cropped

In the second example the test image is a synthetically generated image. All the relevant data and results are shown in Figs. 14, 15, and 16.

Figure 14. Image convolution (3×3 mask)

Figure 15. Example #2: Image processing phases

Figure 16. Example #2: CSF-LS final estimation

Example #3: Real image with center far outside the imager
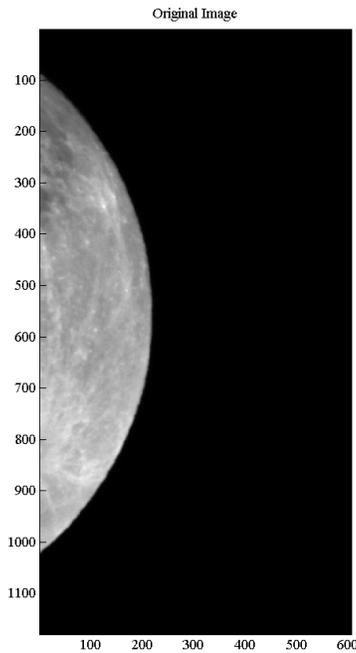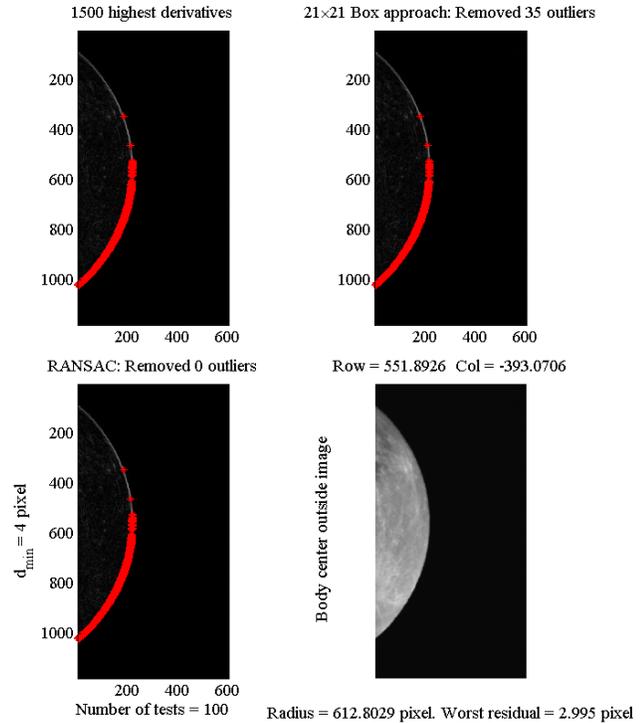

Figure 17. Example #3: Original image

Figure 18. Example #3: Image processing phases

ERROR ANALYSIS

This section is dedicated to the error analysis associated to the following sources:

1. **Estimated position error**. This analysis quantifies how the error in the measured Moon radius (in pixel) affects the Spacecraft-to-Moon distance computation (in km).

2. **Estimated attitude error**. This analysis quantifies how the error in the camera attitude knowledge (angle between quaternion) affects the Spacecraft-to-Moon direction (unit-vector).

3. **Image processing errors**. This Montecarlo analysis shows the distribution of the Moon radius and center estimated by the image processing software when the image is perturbed by Gaussian noise. This is done in the three cases of crescent, gibbous, and full Moon illumination.

Position and attitude uncertainty propagation

This section investigates the error in the Moon estimated radius because of the position estimation error. That is, how the Spacecraft position estimate affects the measurement of the Moon radius in the imager. It also investigates the error in the Moon center direction because of the attitude estimation error. That is, since the Spacecraft position estimate requires the evaluation of the Spacecraft-to-Moon vector in J2000 while it is measured in the camera reference frame, how the camera to J2000 change of coordinate affect the Spacecraft-to-Moon direction?
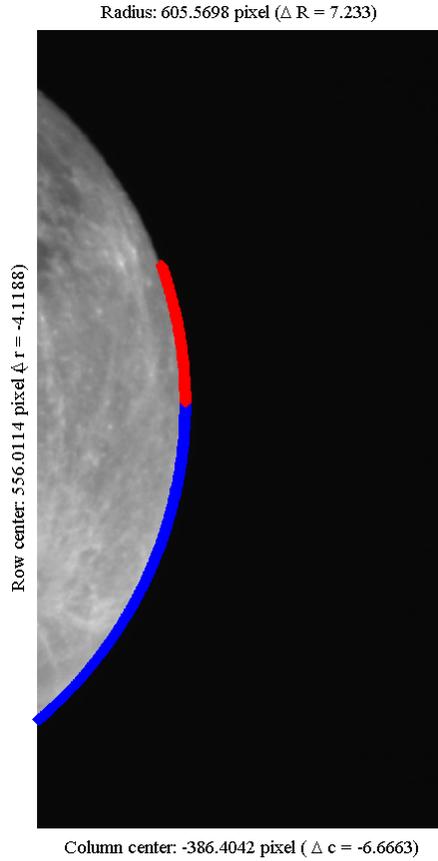
Figure 19. Example #3: CSF-LS final estimation

Let us consider the Moon radius (in pixel) uncertainty first. Moon radius is directly related to the Spacecraft-to-Moon distance. The Spacecraft-to-Moon vector is $\boldsymbol{r}_{om} = \boldsymbol{r}_m - \boldsymbol{r}$ where $\boldsymbol{r}_m$ and $\boldsymbol{r}$ are the Moon and Spacecraft position vectors in J2000, respectively. Vector $\boldsymbol{r}_m$ is simply derived from time while the Spacecraft position vector, $\boldsymbol{r}$, it can be considered a random vector variable with mean $\bar{\boldsymbol{r}}$ and standard deviation $\sigma_{\boldsymbol{r}}$, that is, $\boldsymbol{r} \sim \mathcal{N}(\bar{\boldsymbol{r}}, \sigma_{\boldsymbol{r}}\hat{\boldsymbol{u}})$, where $\hat{\boldsymbol{u}}$ is a random unit-vector variable uniformly distributed in the unit-radius sphere. Neglecting the ephemeris errors, the uncertainty in the Spacecraft vector position coincides with the uncertainty in the Spacecraft-to-Moon distance, $\sigma_D = \sigma_{\boldsymbol{r}}$.

The Moon radius in pixels in the imager is provided by

$$r = \frac{f}{d} \frac{R_m}{\sqrt{D^2 - R_m^2}} \tag{39}$$

where d is the pixel dimension (in the same unit of the focal length, f), $R_m = 1,737.5$ km is the Moon radius, and D is the Spacecraft-to-Moon distance. This equation allows us to derive the uncertainty in the Moon radius in the imager

$$\sigma_r = \left|\frac{\partial r}{\partial D}\right|_{\bar{\boldsymbol{r}}} \sigma_D = \frac{f\,D\,R_m}{d\,(D^2 - R_m^2)^{3/2}}\,\sigma_{\boldsymbol{r}} \tag{40}$$

16

The inverse of Eq. (41)

$$\sigma_D = \frac{d\,(D^2 - R_m^2)^{3/2}}{f\,D\,R_m}\,\sigma_r \qquad (41)$$

provides an important information about the accuracy required by the radius estimation. This equation tells you how the radius estimate accuracy is affecting the Spacecraft-to-Moon vector length, and, consequently, how this estimate is affecting the estimation of the Spacecraft position.
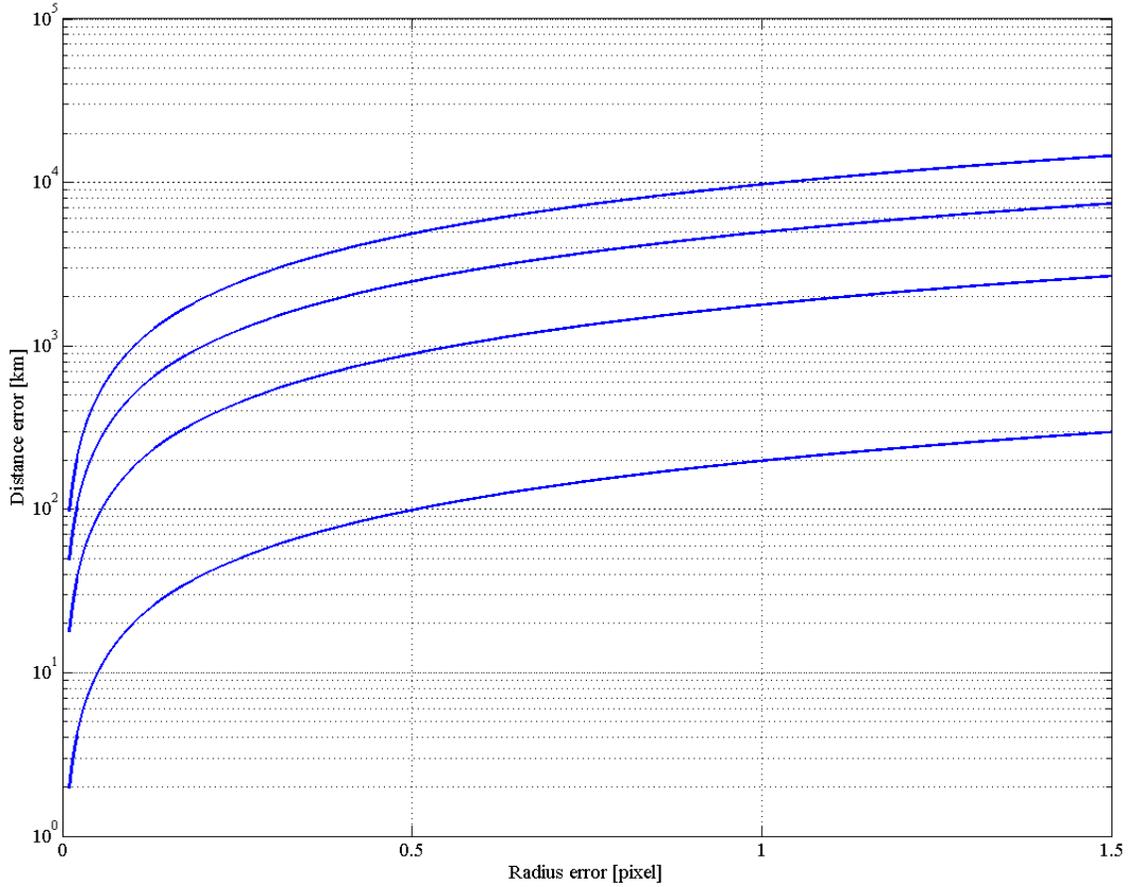

Figure 20. Calculated Moon distance error as a function of radius determination error

Equation (41) has been plotted in Fig. 20 as a function of the Moon radius error and for four values of the Moon distance, $D = 50,000$ km, $D = 150,000$ km, $D = 250,000$ km, and $D = 350,00$ km.

The Moon direction uncertainty (in pixel) is caused by the uncertainty in the attitude knowledge. Consider the attitude $\mathbf{q} \sim \mathcal{N}(\bar{\mathbf{q}}, 2\sigma_{\mathbf{q}}\hat{\mathbf{u}})$ with mean $\bar{q}$ and standard deviation $2\sigma_{\mathbf{q}}$. Now the displacement with respect to the camera optical axis is $c = \dfrac{f}{d}\,\tan\vartheta$, where ϑ is the angle between estimated Moon center and camer optical axis. Therefore, the radial uncertainty in the Moon center direction is provided by

$$\sigma_c = \frac{f}{d}\,\left|\frac{1}{\cos^2\vartheta}\right|_{\bar{q}}\,\sigma_{\mathbf{q}} \qquad (42)$$

17

Image processing error

Image processing error quantification requires performing statistics using a substantial set of images with known truth data. Since no such ideal database is currently available, the analysis of the resulting estimated parameters (Moon radius and center) dispersion is performed by adding Gaussian noise to the same image. The Moon illumination scenario plays here a key role since when the Moon is fully illuminated the number of data (pixels) used to perform the estimation will be twice as much as in the partial illumination case (under the same other conditions: distance to the Moon and selected camera). For this reason two scenarios have been considered:

1. Partially illuminated Moon (results provided in Fig. 21)

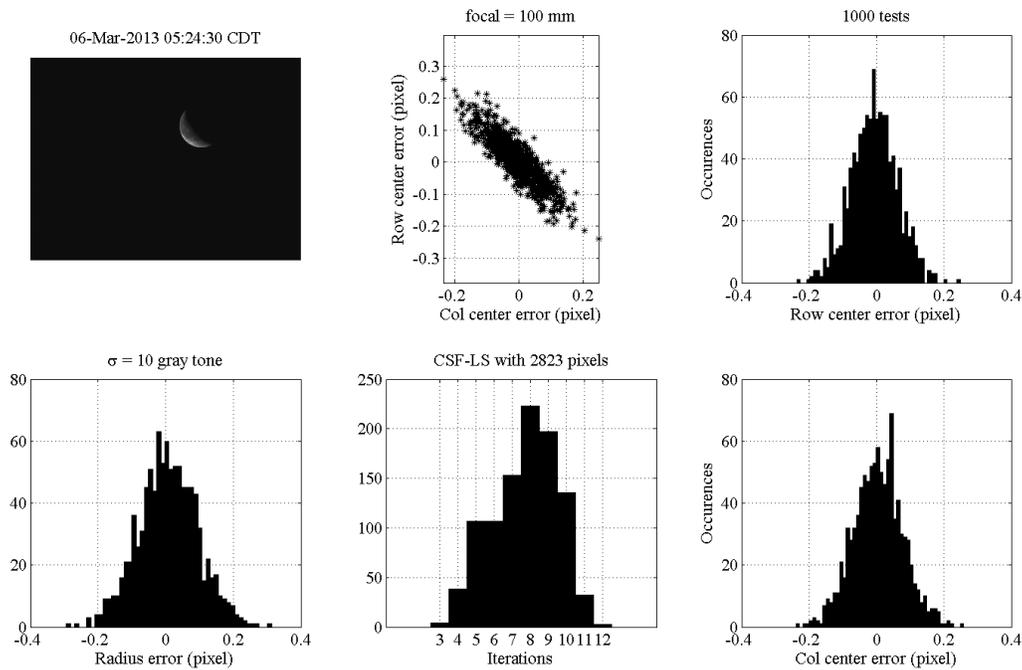2. Full illuminated Moon (results provided in Fig. 22)

Figure 21. Image processing error results: partial illumination case

The results of the tests done using partial illumination Moon are shown in Fig. 21. The image selected is real and it was taken on March 6, 2013 at 05:24:30 using a focal length $f = 100$ mm. The statistics of the results obtained in 1,000 tests by adding Gaussian noise with zero-mean value and standard deviation $\sigma = 10$ gray tone are given. The code estimates the Moon radius and center (row and column) by least-squares using circular sigmoid function (CSF-LS). The results show the three parameters estimated as unbiased and with a maximum error of the order of 0.2 pixels (with respect to the original picture). The histogram of the number of iterations required by the least-squares to converge shows an average of 8 iterations. The central-upper plot in Fig. 21 shows the ellipsoid distribution of the Moon center error due to the fact that the estimation uses just data on one side of the Moon, on the illuminated part.
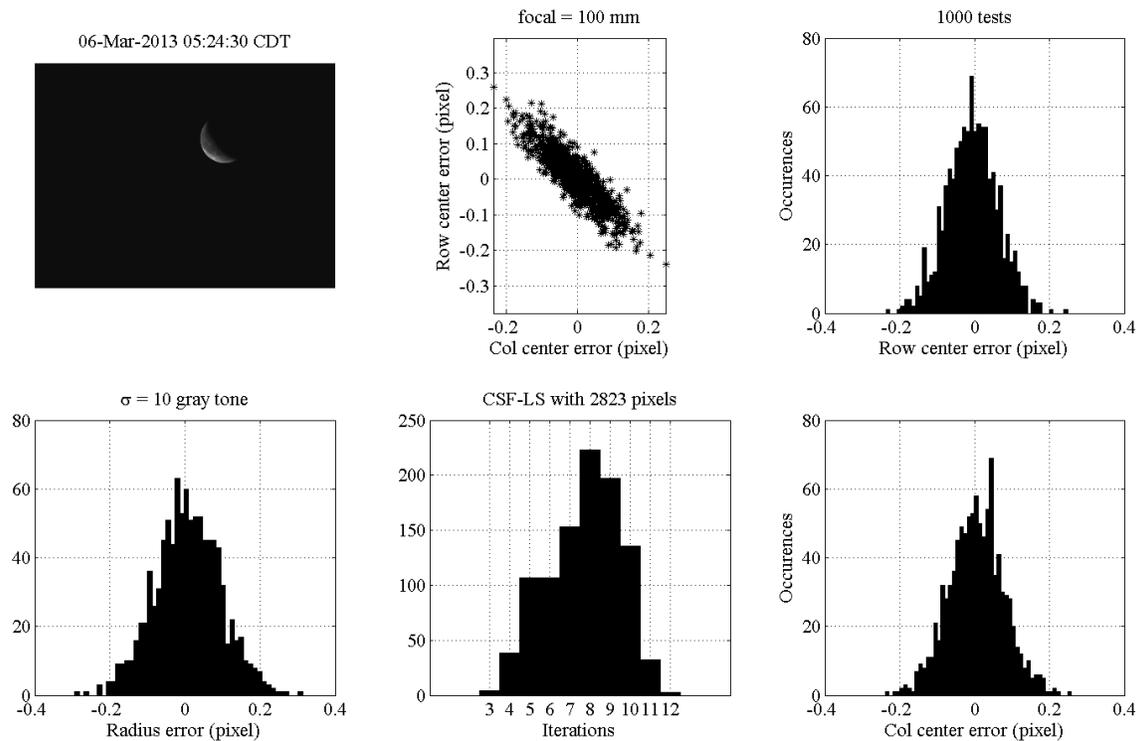
18

Figure 22. Image processing error results: full illumination case

The results of the tests done using full illuminated Moon are shown in Fig. 22. Also in this case the image is real and was taken in Houston on February 25, 2013 at 22:33:00 CDT using a focal length $f = 105$ mm. The statistics of the results obtained in 1,000 tests by adding Gaussian noise with zero-mean value and standard deviation $\sigma = 10$ gray tone are given. The code estimates the Moon radius and center (row and column) by least-squares using circular sigmoid function (CSF-LS). The results show the three parameters estimated as unbiased and with a maximum error of the order of 0.01 pixels, one order magnitude more accurate than what obtained in the partial illuminated case. The histogram of the number of iterations required by the least-squares to converge shows an average of 3 iterations. The central-upper plot in Fig. 21 shows the unbiased Gaussian distribution of the Moon center error. These more accurate results are due to the fact that the estimation uses data all around the Moon edge.

CONCLUSIONS

This paper presents an image processing algorithm capable of determining the apparent location of the Earth or moon and its apparent diameter. This information can then be used for autonomous onboard trajectory determination in cislunar space. Tow method of removing potential outliers are introduced in order to increase the algorithm robustness. Preliminary analysis of the algorithm's performance is studied via processing both real and synthetic lunar images.

REFERENCES

[1] Mortari, D., de Dilectis, F., and D'Souza, C. "Image Processing of Illuminated Ellipsoid," AAS 13-853, 2013 AAS/AIAA Astrodynamics Specialist Conference, Hilton Head, SC, August 11–15, 2013.

[2] Fischler, M.A. and Bolles. R. C. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Comm. of the ACM, Vol 24, pp 381-395, 1981.

[3] Mortari, D., Samaan, M.A., Bruccoleri, C., and Junkins, J.L. "The Pyramid Star Pattern Recognition Algorithm," ION *Navigation*, Vol. 51, No. 3, Fall 2004, pp. 171-183.

[4] Fitzgibbon, a., Pilu, M., and Fisher, R.B "Direct Least Square Fitting of Ellipses," IEEE Trans. Pattern Anal. Mach. Intell. 21, 5 (May 1999), 476-480.

[5] Halir, R. and Flusser, J. Numerically Stable Direct Least Squares Fitting of Ellipses. 1999

[6] Taubin, G. "Estimation Of Planar Curves, Surfaces And Nonplanar Space Curves Defined By Implicit Equations, With Applications To Edge And Range Image Segmentation," IEEE Trans. PAMI, Vol. 13, pages 1115-1138, 1991