VISION-BASED AUTONOMOUS NAVIGATION USING MOON AND EARTH IMAGES

A Dissertation

by

FRANCESCO DE DILECTIS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Daniele Mortari
Committee Members,	John L. Junkins
	Srinivas R. Vadali
	J. Maurice Rojas
Head of Department,	Rodney D. W. Bowersox

December 2014

Major Subject: Aerospace Engineering

ABSTRACT

This document explores the possibility of a novel approach to trajectory determination based on optical observation of nearby celestial bodies, and describes its development, implementation and testing. The initial idea was spurred by the need to give the Orion capsule, the next-generation manned spacecraft currently in development at NASA, a backup autonomous positioning system. Among other missions, Orion is designed to transport a crew to and from the Moon. The presence of individuals on board tightens the safety requirements, and therefore it is necessary to provide a backup positioning system in case contact with ground is lost. However, rather that develop and install specific hardware on the craft, this work attempts to use data form a pre-existing sensor, an optic camera, for position estimation. This research consists of three main sections: first, analysis of the images captured by camera, to identify the geometrical features of interest of the observed celestial bodies (namely apparent radius, center position and orientation). Second, estimate of these properties with various best fitting algorithms, and derivation of orientation and position of the spacecraft in an inertial frame. And last, best fitting of the gathered position data with a newly developed algorithm based on Bézier functions. Each of these parts are discussed in detail. Results show that the image processing algorithm developed is capable to meet the accuracy requirements imposed by the mission, and that Bézier functions are a suitable tool to efficiently interpolate space trajectories, favorably comparing to more complex techniques like Iterative Batch Least Square and Extended Kalman Filter.

To my parents Lucio and Luciana

because, true to their name,

they always lighten my path.

ACKNOWLEDGEMENTS

The first thing to say is that over four years ago, when I decided to leave my comfortable world to start a Ph.D. on the other side of the Atlantic, I had no idea what I was getting myself into. I arrived in College Station a scorching day of August, my luggage delayed and having never seen the town before, except on Google Maps. Everything and everyone within a several thousand miles radius was completely unknown to me. Thinking about it now, it almost seems I must have had a moment of insanity. But, despite the all-nighters, the bouts of loneliness and the moments when I just wanted to drop it all and run home, I am certain I made the right choice. The things I have learned in my time here go far beyond the realm of academics, and no matter where I will go from now on, I will do so as a better, stronger person.

Of course, no matter my talent and hard work, I would have never successfully completed this journey without the help of many people. First and foremost, I wish to thank my advisor, Daniele Mortari. He gave me the opportunity to start this program and throughout these years, he has always been helpful and understanding, on both a professional and personal level. I would also like to thank the members of my committee, Dr. Junkins, Dr. Vadali and Dr. Rojas. Their advice and their patience in dealing with my efforts to successfully make all the deadlines have been really incredible. And of course, many thanks to Karen Knabe, who was there to save the day whenever the deadly shadow of bureaucracy loomed about.

To my fellow Ph.D. students, both those who have graduated already and those who still have some work ahead of them, thank you all for the peer pressure moti-

vating me to prove myself worthy of the program. :)

Despite the distance, I have always felt the presence of my friends back home, who are like family to me and have cheered me all the way long, and forgiven me for not being close while they went on to make their own amazing dreams into reality. They know I was with them in spirit at every step.

Thank you to my mom, my dad and my sister. I couldn't put into words how great their support was. I will always strive to make them proud of me.

And one last thank to Nick, without whom I quite literally would not be writing these words right now. I look forward to return the favor, tenfold and more.

Francesco de Dilectis,

College Station, Oct. 2014.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF FIGURES

Page

Page

Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

While spacecraft have long been self-sufficient in regard to attitude determination, they currently require external assistance to estimate their position in space. Typically this is provided as range and range rate measurements obtained by either ground stations or signals from GPS receivers. Both methods however have technical (only LEO missions can currently be supported by GPS tracking) or budget (deep space missions tracking is very expensive) limitations. Therefore, autonomous navigation has been the subject of extensive research in the last 20 years.

One crucial aspect of autonomous navigation is the type of sensors used in data acquisition for processing. As one can imagine, this is heavily influenced by the mission; satellites in LEO can take advantage of techniques and hardware not available to deep space probes; and spacecraft flying nearby a known celestial body can make use of such a reference point in ways inaccessible to one whose orbit does not pass sufficiently close to that body. In particular, a spacecraft on a cislunar trajectory will always be in relative vicinity of two major and very well know celestial bodies: the Earth and the Moon. It seems a valid idea then to explore the possibility of extracting position measurements from images of these bodies.

During the travel from Earth to Moon, a spacecraft is mainly subject to gravitational forces from three different celestial bodies (Earth, Sun, Moon), plus other forces of different nature (e.g. Solar Radiation Pressure, Atmospheric Drag), all of whom act with different relative intensity depending on the position along the

trajectory. As a result, models to compute the spacecraft's behavior are extremely complicated and not very reliable. To improve accuracy, estimators are used to account for the inherent imprecision of the models, commonly various implementations of the Kalman filter. A major drawback is then the necessity to manually tune and customize the filter to the specific mission, which can sometimes be prohibitively complicated during the design phase. As an alternative, the approach described in this research completely bypasses the problem by disregarding any considerations about the dynamics involved, while focusing solely on the measurements, and using Bézier functions as interpolating polynomials, thus removing the need for manual tuning and eliminating a complicated phase from the design process.

I.A. Literature Review

Autonomous navigation as a concept has been around for many years. The historical developments up to 1984 can be found in [3], where autonomous navigation is described as having 4 basic characteristics: 1) self-contained, 2) operating in real time, 3) nonradiating and 4) not depending on Earth operations. Ideally an autonomous navigation system should only make use of on-board measurements of natural signals.

Lowrie in [23] distinguishes three types of autonomous navigation concepts: angular position measurements to celestial objects, such as the Sun, the local vertical, Moon-Star and Planet-Star; measurements of Earth target, such as artificial or natural landmarks; and measurements to known artificial beacons, such at the GPS system. It is worth mentioning as new technologies became available, different solu-

tions have been proposed that eschew this classification. Indeed, the method studied in this work does not fall neatly into any of the categories described above.

Historically, initial developments were essentially based on Line-of-Sight (LOS) measurements obtained via a combinations of star, Sun and Earth sensors [21] [42] [25] [24]. Even at these early stages attempts to use CCD sensors in creative ways were made, to try and remove the need for dedicated star trackers and reduce mission costs. Later, the advent of GPS made it a popular choice for autonomous and semiautonomous positioning systems for near-Earth satellites, because of its low cost and pervasiveness [13] [18].

Concurrently, other methods of satellite-to-satellite tracking have been developed [9], where the orbits of two or more spacecraft are determined from measurements of the relative position vector from one spacecraft to the others, a concept especially useful for absolute orbit determination of formation flying spacecraft. Moreover, several studies have been conducted about using Earth magnetic field as the reference to determine a spacecraft position [33] [35] [34]. Regarding deep space missions, the idea of autonomous navigation of a spacecraft using celestial objects has been presented using an extended Kalman filter [12]. In this case, the measurements used in the simulations are the line of sight directions to the celestial objects from high-accuracy attitude sensors and one-way Doppler measurements from ground stations. In [14], Guo has proposed a complete self-contained autonomous navigation system for deep space missions using two types of on-board observation data. The first is a directional data of the spacecraft relative to the Sun, and the second is the optical Doppler shift due to the motion of the spacecraft relative to the Sun.

Similarly, in [19], a method has been developed in which images of planets within the solar system are used to determine the spacecraft position.

Returning to the problem at hand, autonomous navigation for cislunar trajectories has been also studied for a long time, especially during the 1960s and 1970s for the Apollo program. With the new generation of manned spacecraft being designed, a new series of studies has started, trying to take advantage of the latest technological developments to implement techniques that were not feasible at the time. In particular, the authors of [13] discuss the possibility explored in the present research, of using as sources for inertial navigation updates the centroid and radius of Moon and Earth. While their approach uses a Kalman filter to ultimately find the inertial position, they however highlight how the accuracy of the method is very sensitive to the distance from the observed target. This research proposes to overcome such limitation, by using a different approach to the centroid and radius estimation.

I.B. Description of the Project

While the methodology described in these pages can be used in any mission flying reasonably close to a known celestial body, its implementation has focused on a cislunar trajectory, as dictated by Orion mission requirements. Therefore, in the following the generic references to "celestial body" will be replaced primarily by the Moon, and secondarily the Earth.

Because the Orion capsule is slated to have an optic camera among its sensors, pointed towards the bow, the idea came to use images captured by it as a starting point to develop a positioning system without having to add dedicated hardware.

Therefore, a great deal of work has been devoted to image processing, to consistently and accurately extract data useful to estimate the spacecraft position.

Considering the Moon as a perfect sphere (an essentially correct assumption) of known radius ($R_{\leftmoon} = 1737.5$ km), and its position in the inertial space accurately derived from time, the comparison between these known quantities and the values of radius and centroid as seen on the camera CCD can lead to an estimate of the position vector of Orion in a Earth Centered Inertial (ECI) frame. This is the position information to be used in the trajectory estimator.

In a similar manner, the Earth can be considered an ellipsoid of revolution. This means that its projection on the CCD will be an ellipse, whose eccentricity varies with the observer's perspective, albeit in a very small range very close to 1. This sensibly complicates the problem, because it increases the number of unknowns from 3 to 5, one of which is the inclination of the ellipse on the camera plane, and whose observability is extremely low and can sometimes be zero depending on the relative positions of the capsule with respect to Earth.

Once the measurements have been obtained, a newly designed trajectory interpolator is used, capable of working regardless of the dynamics involved in the problem, because it employs a Non-Rational Bézier curve, to fit the data. The choice of Bézier functions over other more advanced polynomial curves, like splines, is due to the fact that Non-Rational Bézier curves can be reformulated in a linear matrix form which makes the implementation of a least square method immediate.

I.C. Significance

The Orion vehicle is being designed to provide nominal crew transport to the lunar transportation stack in low Earth orbit, crew abort prior during transit to the Moon, and crew return to Earth once lunar orbit is achieved [8]. While a primary ground based position determination system is prospected to be installed, the increase in safety requirements due to the presence of a crew prompts the necessity for a backup system capable of working in case of contact loss with ground base. Ideally, the proposed system should give Orion autonomous position estimation capabilities without the need for additional equipment. Its precision should be sufficient to correctly estimate the position within 1 km or less. Moreover, such a system would prove useful when the capsule is behind the Moon, beyond the reach of Earth's signals.

I.D. Methods

This research project consists of two main parts, the first part detailing how the images are processed to extract the position data and the second how these estimates are used to interpolate the trajectory.

I.D.1. Image Processing

Because all the data is extracted from a single sensor, this process must be as efficient as possible to maximize precision and accuracy. As explained above, the desired information to be retrieved are the centroid and radius for the Moon, or centroid, axes and inclination for the Earth. The centroid will allow to determine the attitude of the spacecraft with respect to the observed body (although other sensors

on board will be providing similar measurements, which will have to be integrated to fully describe the attitude), while the geometric dimensions will allow to determine the distance from it, according to:

$$r = \frac{f}{d} \frac{R_{\mathbb{C}}}{\sqrt{D^2 - R_{\mathbb{C}}^2}} \tag{1.1}$$

where f is the camera focal length, d is the dimension of a pixel, and D is the scalar distance between the spacecraft and the target. Determining these geometrical properties is essentially a feature identification problem [28], and as such it can be considered consisting of the following two steps: 1) selection of the pixels within the image and 2) analysis of those pixels to determine the desired parameters. For both tasks, there exists abundant literature, especially regarding the identification of circles and ellipses in an image [31][40][37][11][15]. Therefore, original efforts have been made for the pixel selection task, culminating in the implementation of two completely different techniques, described in chapter II.

The first considers the average brightness of the image to establish a threshold, which is used to create a binary version of it. Then an "inertia tensor" is built allowing for the determination of the axis of symmetry of the bright pixel distribution, which is in turn used to identify pixels belonging to the edge of the body [28](Figure I.1).

The second filters the image as a whole looking for points where the contrast is maximum creating a "derivative" of the original image. Such points will likely belong to the edge of the body. To eliminate eventual extraneous points, called outliers, two algorithm in sequence are used, the first custom designed to find points lying on a bright curve against a dark background, and another based on the Random Sample

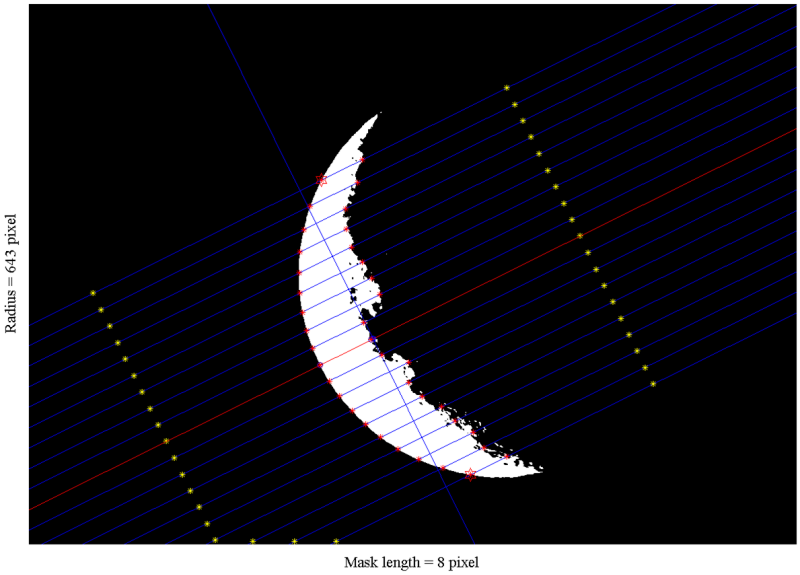


Figure I.1: "Inertia tensor" approach.

Consensus (RANSAC) technique (Figure I.2).

Regarding the actual feature identification, discussed in chapter III, an extensive search in literature has led to the testing of many different algorithms, to find the ones best suited for the problem at hand. It is important to notice that no mention has been found in literature of using such techniques to determine the apparent dimensions of celestial bodies not assumed spherical. Best fitting techniques for this type of problems can be divided in two broad groups:

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 de-

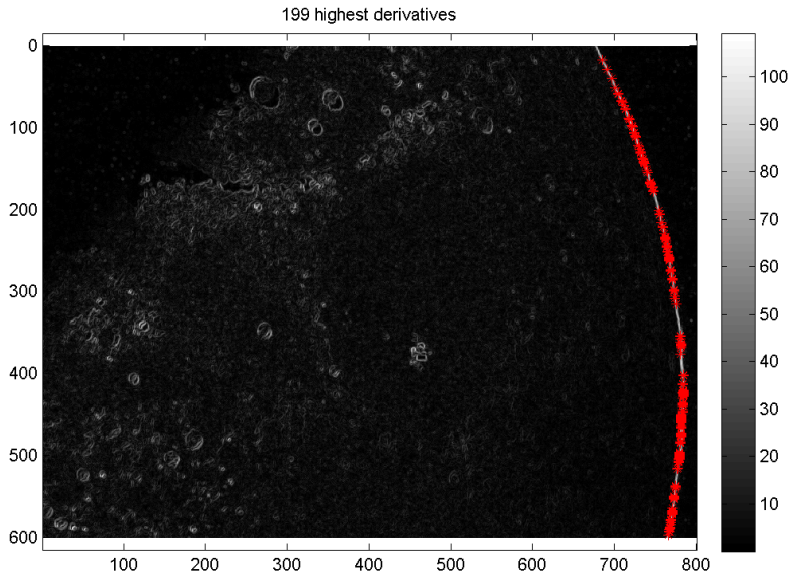


Figure I.2: Gradient approach.

veloped and they differ in accuracy, stability and complexity. They generally give results less accurate than the geometric fit, but hand have closed form solutions and therefore can be implemented in a much simpler way.

For the ellipse, a custom geometric fit has been designed to take advantage of the fact that the sought ellipse has a known axes ratio. In this case, the parameters to be determined are the two coordinates of the centroid, the semi-major axis, and the inclination reducing the number of unknowns to 4. This method is based on non-linear Least Squares applied to the canonical equation of the ellipse in a generic reference frame.

I.D.2. Position Interpolation

A Bézier function is a parametric curve widely used in computer graphics and many other design related applications. While the mathematical foundations were established in 1912 by Sergei Bernstein, its applicability to design was understood much later, between 1959 and 1962, with the work of Pierre Bézier and Paul de Casteljau. The method developed here is based on [30], where the Bézier function is rewritten using linear algebra to estimate its control points via least squares. This basic idea is at the heart of a newly developed iterative process able to optimize the control points, by finding the optimal polynomial degree and parameter distribution. As a validation, this method as been tested on a cislunar trajectory simulated with GMAT, and its results have been compared with popular methods such as Iterative Batch Least Squares and Kalman filtering. A complete discussion is found in chapter V.

I.E. Research Goals

The purpose of this research project are as follows:

- to successfully design a software capable of analyzing in real time images of Moon and Earth taken by an on-board optical camera to extract information relative to the edges of the celestial bodies, including handing of situations in which said bodies are outside the Field-of-View, partially illuminated or non completely within the FOV limits.
- to implement a best fitting procedure able to determine centroid and radius

(for the Moon) or centroid, axes and inclination (for the Earth), with a level of precision sufficient to obtain reliable position measurements for the spacecraft as a whole.

- to interpolate the data so obtained to determine the trajectory of the spacecraft, regardless of the dynamics involved, presence of perturbations or unplanned thrusts.

Because of the very stringent requirements on accuracy, a great effort has been devoted to optimization of the method, especially with regard to the image processing. This has led to a comprehensive analysis of the various causes of error affecting the data extraction process. Also, an error propagation analysis is required to assess the validity and practical usability of the method. Of course, interpolation of existing data does not give any information on the projected behavior of the spacecraft. Therefore, the author hopes that the interpolated data can be used to improve prediction of the spacecraft future trajectory.

CHAPTER II

IMAGE PROCESSING - PIXEL SELECTION

The main source of information for this problem is provided by the images captured with the camera mounted on the spacecraft. Therefore, it is natural that a great effort has been devoted to develop the most efficient and fruitful method to extract the greatest possible amount of information from it. The result is a heavily specialized process, in the sense that is based on several assumptions made on the nature of the image itself. These assumptions are as follows:

- the image contains only one luminous object whose dimension are much greater than any other in the field of view.
- most of the image is "dark", i.e. it has a black background, interspersed with few stars. It is worth here noting that since the camera is calibrated to capture a close and bright object, the level of exposure is not sufficient for most stars to appear, which works conveniently in our favor.
- while the camera works in RGB, no useful information is contained in the different color channels, and only the graytone is used to determine relative brightness of each pixel.
- no attempt to determine any of the internal features of the target is made (craters and *maria* for the Moon, cloud systems and continents for Earth).

Within these criteria, the procedure developed is able to satisfactorily manage all possible cases, in terms of target size and shape, relative positioning of observer

and target, and even situation in which the target is only partially in view.

II.A. Introduction

This chapter describes the methods developed to extract the geometric properties of the observed bodies from the images, and the theory behind it. As the Moon can be considered a sphere, its undistorted projection is a circle, which is uniquely defined by 3 parameters, the two coordinates of the center O and radius r_c) the Earth instead is more accurately described as an ellipsoid of revolution, whose projection is in general an ellipse, uniquely defined by 5 parameters, center O, axes a and b, and inclination θ (later it will be shown that these can be reduced to 4). While a few details have to be changed to account for the differences, the same process can be applied for both cases. This can be considered composed of two subsequent phases:

Pixel Selection , where a subset of all the pixels composing the image are chosen, according to a predetermined criteria. Two techniques have been developed for this task. One, here named *Eigenvalue Method*, analyzes the distribution of bright pixels in the image to determine its principal directions. The search for useful pixel is then conducted in reference to these directions. The other, named *Gradient Method*, applies a Derivative Filter to the whole image, to identify the pixels belonging to the edge of the object.

Feature Recognition , where the requested parameters are extracted via analysis of the pixels previously selected. While insofar it has never been used for space applications, feature recognition is a standard practice in computer graphics, therefore there is ample literature available dealing with this prob-

lem. Specifically, the author has focused on techniques relative to circle and ellipse identification, as these are the shapes of interest.

Because the ultimate purpose is to develop an algorithm, a series of parameters has to be defined to successfully guide the data flow. These will be introduced as the need for them arises.

To begin, it is important to find a reliable method to differentiate between the possible size and shape of the target, which depends on the relative position of observer (the capsule), observed object (Moon or Earth) and the light source (the Sun). In the following section, these different configuration are described relative to the Moon, but they are applicable to Earth as well. It is important to highlight how the following is based on geometric considerations alone and thus does not involve processing of the image data. This is why the parameters are described as "expected" values.

II.A.1. Expected Radius and Illumination Parameter

As the relative positions of observer, target and light source change, the visible part of the target changes accordingly in a manner akin to the classic phases common to all reflective celestial bodies. This can lead to cases in which the target is completely dark from the point of view of the observer, and therefore no analysis can take place. It is important therefore to catch these degenerate cases before they can cause critical errors further down in the process. At this stage the target is considered as perfectly spherical, an acceptable approximation given the fact that, assuming Earth to be an ellipsoid of revolution, its flattening is 0.0033528. Thus

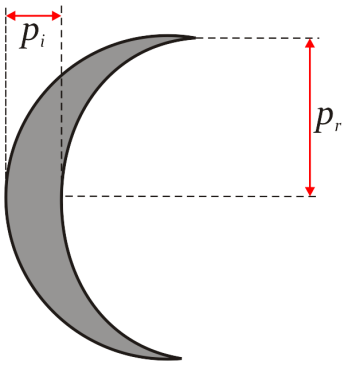


Figure II.1: Expected Moon radius (p_r) and illumination parameter (p_i).

two parameters can be introduced to describe its shape (from now on for simplicity, the term "target" will refer just to the illuminated portion rather than the whole object, as this is what the camera is actually able to observe): one is the radius of the target p_r, and the other is the cross-length measured at its thickest point p_i (see Figure II.1), which will be referred as the *illumination parameter* in the rest of this text. Both these distances are measured in pixels, and they can be found given the camera parameters (focal length f, pixel size d) and the observation geometry. To this purpose, the observer's position is needed. However, even a high uncertainty ($\sigma_D \geq 10,000$ km) has very small impact on p_r and p_i, as long as the spacecraft is further from the target than a certain threshold. Such a rough position estimate can be considered known *a priori*, to start the analysis.

To begin, expected radius and illumination angle are computed as:

$$p_r = \frac{f}{d} \frac{R_m}{\sqrt{|\boldsymbol{r}_o|^2 - R_m^2}} \qquad \text{(pixels)} \tag{2.1}$$

$$\vartheta = \arccos(\hat{\boldsymbol{r}}_s \cdot \hat{\boldsymbol{r}}_o). \tag{2.2}$$

where $\hat{\boldsymbol{r}}_s$ is the Sun-to-Moon rays direction, $\boldsymbol{r}_o$ the Orion-to-Moon vector, and R_m

the Moon radius. The expected radius is of great importance, not only because the illumination parameter is measured in relation to it, but also because it predicts the overall size of the target in the image. The importance of this information will become clear later. The value of p_i is determined differently depending on the values of p_r and ϑ.

II.A.1.a. *Full or New Moon*

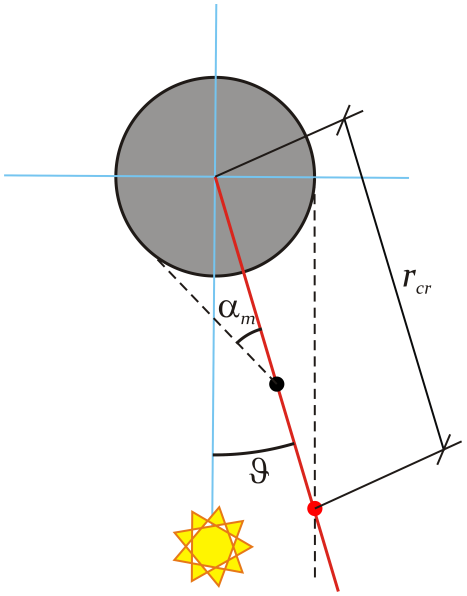


Figure II.2: Full/New Moon geometry.

With reference to Figure II.2, the angle α_m can be derived from simple trigonometry

$$|\boldsymbol{r}_o| \sin \alpha_m = R_m. \tag{2.3}$$

This equation also allows to define a critical distance, r_{cr}, as the distance at which

$\alpha_m = \vartheta$:

$$r_{cr} \sin\vartheta = R_m \tag{2.4}$$

Whether $|\boldsymbol{r}_o|$ is greater or smaller than r_{cr}, together with the value of ϑ, determines the case, according to the following scheme:

$$\text{if} \quad |\boldsymbol{r}_o| < r_{cr} \quad \text{and} \quad \begin{cases} \vartheta < \pi/2 \quad \text{then} \quad p_i = 2p_r \quad \text{(Full Moon)} \\ \vartheta > \pi/2 \quad \text{then} \quad p_i = 0 \quad\quad \text{(New Moon)} \end{cases} \tag{2.5}$$

II.A.1.b. *Gibbous Moon*

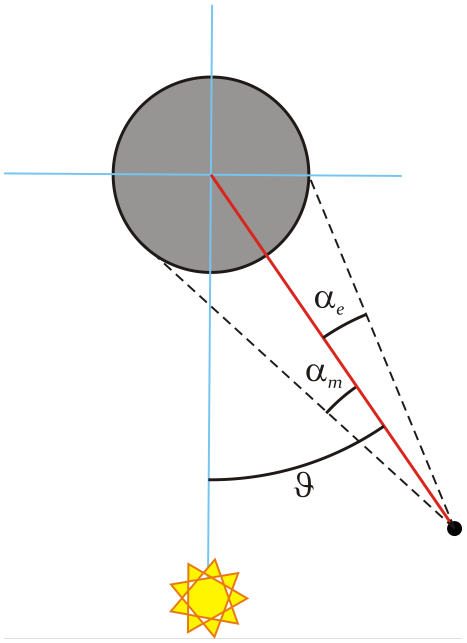


Figure II.3: Gibbous Moon geometry.

Gibbous Moon will occur when

$$|\boldsymbol{r}_o| > r_{cr} \qquad \text{and} \qquad \vartheta < \frac{\pi}{2}. \tag{2.6}$$

For this configuration (see Figure II.3), p_i can be computed by finding the auxiliary variables x and α_e:

$$x^2 = |\boldsymbol{r}_o|^2 + R_m^2 - 2\,|\boldsymbol{r}_o|\,R_m\,\sin\vartheta \quad \rightarrow \quad x \tag{2.7}$$

$$\sin\alpha_e = R_m\,\frac{\cos\vartheta}{x} \quad \rightarrow \quad \alpha_e \tag{2.8}$$

which allows to find the "excess" factor p_e:

$$p_e = p_r\,\frac{\tan\alpha_e}{\tan\alpha_m} \tag{2.9}$$

and, therefore,

$$p_i = p_r + p_e = \left(1 + \frac{\tan\alpha_e}{\tan\alpha_m}\right)\,p_r. \tag{2.10}$$

II.A.1.c. *Crescent Moon*

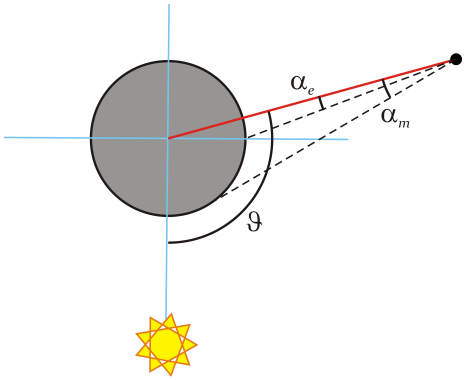


Figure II.4: Crescent Moon geometry.

Crescent Moon will occur when

$$|\boldsymbol{r}_o| > r_{cr} \qquad \text{and} \qquad \vartheta > \frac{\pi}{2}. \tag{2.11}$$

In this case (see Figure II.4) the two variables x and α_e can be found according to:

$$x^2 = |\boldsymbol{r}_o|^2 + R_m^2 - 2\,|\boldsymbol{r}_o|\,R_m\,\sin\vartheta \quad \rightarrow \quad x \tag{2.12}$$

$$\sin\alpha_e = \frac{-\cos\vartheta}{x}\,R_m \quad \rightarrow \quad \alpha_e \tag{2.13}$$

and therefore:

$$p_i = p_r - p_e = \left(1 - \frac{\tan\alpha_e}{\tan\alpha_m}\right)\,p_r \tag{2.14}$$

As explained below, differentiating among these cases is important to properly initialize the algorithm. To this purpose, the analytical relations described above must be supplemented with discrete threshold to ensure the system works properly; so, for instance, if p_i is lower than a certain value, the crescent is deemed too thin to be a reliable source of information, and the target is considered invisible.

II.A.1.d. *Special Case: Cropped Target*

The previous cases do not consider the possibility for the target to partially be in the FOV. However, depending on the relative size of the target on the imager and whether the camera is pointing towards it or not, such a scenario is not at all unlikely and thus it is important to be able to identify it, especially because, as it will be explained later, only one of the two methodologies developed can solve the case of cropped target.

To this purpose, an algorithm has been developed that sweeps the images' boundaries with a rectangular mask proportional to the dimensions of the image itself. At each step the cumulative brightness within the mask is computed and compared with a threshold value, taken equal to 1/10 of the maximum greytone. From the amount of threshold crossings, it is possible to determine if and how many times

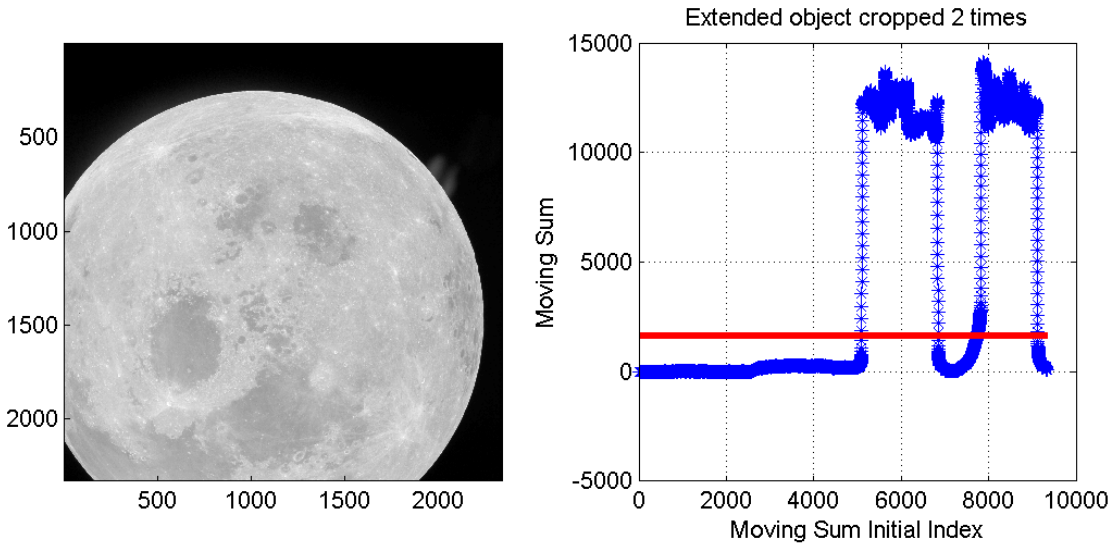


Figure II.5: Processing of cropped image. Left: cropped image of the Moon. Right: results of the cropped detection.

the target has been cropped by the frame. An example application is depicted in Figure II.5.

II.B. Eingenvalue Method

The idea behind this approach is to analyze the distribution of bright pixels in the image to find its symmetries, and thus define a reference system to help determine the pixels to be selected. This works because the method exploits some expected features of the image. Firstly, the image contains only one sizeable bright object, which means any other light source can be considered of negligible size and filtered away quickly. Secondly, this bright object has a symmetric or quasi-symmetric shape. As one can immediately see, whether the target is crescent or gibbous, this condition

is verified (the special case of full Moon, where the symmetry is radial, can be treated separately while maintaining the core of the algorithm untouched). Because of this, the Eigenvalue Method fails when the target is only partially within the FOV, as the symmetry is broken. However, during most of the mission, assuming the camera is pointed somewhat towards the Moon, such scenarios should not present themselves. At any rate, if the image is found to contain a cropped view of the target, this method cannot be applied.

The main cycle of image processing algorithm is summarized in Figure II.6. Further details on the various steps follow.

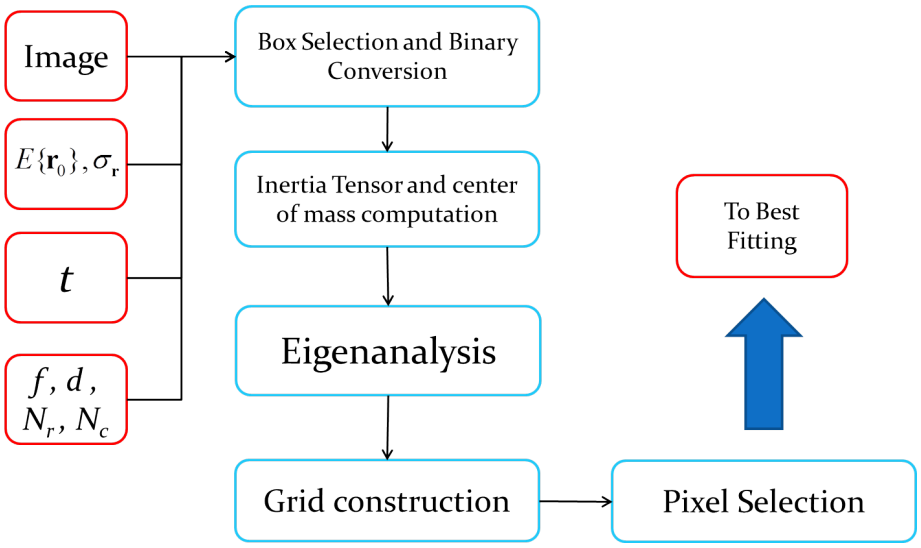


Figure II.6: Eigenvalue method algorithm flowchart.

1. In the first step, a cluster of bright pixels is identified as belonging to the illuminated portion of the observed body. A selection box centered around the cluster is created, equal to twice the length of p_r (see Eq. 2.1). This ensure

that the target is fully included within the boundaries of the box.

2. Greytone values of each pixel within the box image are analyzed to find a threshold which is used to convert the image to binary format.

3. Then, the centroid of the illuminated area is computed. A second check verifies whether the target is fully or partially illuminated, by checking the relative positions of Moon, Sun and spacecraft at the timestamp.

4. For fully illuminated bodies, the points belonging to the edge are identified with a mask swept horizontally across the whole image.

5. For partially illuminated bodies the "inertia tensor" of the binary image is computed. The maximum eigenvalue and its associated eigenvector identify the axis of symmetry of the image itself, which is taken pointing toward the hard edge of the illuminated area. This information about the principal directions of the image is used to identify a series of edge points along both the hard edge of the body and the terminator line.

6. For both fully and partially illuminated body, a SVD-based Least-Squares is used to find an estimate of radius and center. Then, this initial estimate is refined via a Least-Square approach based on circular (for the Moon) or elliptical (for the Earth) sigmoid functions. This is done using a large amount of pixels belonging to an area that partially covers the illuminated area as well as part of the sky next to the hard-edge.

7. Once the center of the observed ellipse has been accurately estimated the ob-

served body center direction is corrected by the offset. Lastly, the computed center and radius are combined with information about the spacecraft attitude to give an estimate of the observer-to-target vector in the J2000 frame.

The last two steps are relative to the feature recognition part of the process and will be described in chapter III. Instead, the next section shows how projective geometry is used to describe the observed ellipse and identify the equations of the hard edge and the terminator. Subsequently, details regarding the eigen-analysis are provided.

II.C. Observed Three-axis Ellipsoid

While tackling the problem of a generic three-axis ellipsoid is slightly more complicated than the case of a sphere or an axial-symmetric ellipse, both of those can be found as special cases of the following development.

The canonical equation, written in the principal reference frame and with center at the origin, is:

$$\boldsymbol{x}^{\mathrm{T}}\, J\, \boldsymbol{x} = 1 \tag{2.15}$$

where $\boldsymbol{x}$ is the vector pointing to the ellipsoid surface, a, b, and c are the three semi-major axes of the ellipsoid, and:

$$J = \begin{bmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/b^2 & 0 \\ 0 & 0 & 1/c^2 \end{bmatrix}.$$

Let $\boldsymbol{p}_0$ be the position vector of the observer. The generic vector $\boldsymbol{x}$ can be rewritten

in terms of $\boldsymbol{p}_0$, the unit vector $\hat{\boldsymbol{v}}$ and a scalar variable t:

$$\boldsymbol{x} = \boldsymbol{p}_0 + t\,\hat{\boldsymbol{v}} \tag{2.16}$$

Substituting this in Eq. (2.15) and expanding yields:

$$\boldsymbol{p}_0^{\mathrm{T}}\, J\, \boldsymbol{p}_0 + 2\,\boldsymbol{p}_0^{\mathrm{T}}\, J\, \hat{\boldsymbol{v}}\, t + \hat{\boldsymbol{v}}^{\mathrm{T}}\, J\, \hat{\boldsymbol{v}}\, t^2 = 1 \tag{2.17}$$

which is a second order equation in t, representing all the possible intersection points between the ellipsoidal surface and the set of all rays intersecting at $\boldsymbol{p}_0$. The observed ellipse is identified by the set of points where such rays are tangent to the surface, where the two roots of Eq. (2.17) coincide. This happens if and only if:

$$(\boldsymbol{p}_0^{\mathrm{T}}\, J\, \hat{\boldsymbol{v}})^2 - (\boldsymbol{p}_0^{\mathrm{T}}\, J\, \boldsymbol{p}_0 - 1)\,\hat{\boldsymbol{v}}^{\mathrm{T}}\, J\, \hat{\boldsymbol{v}} = 0 \tag{2.18}$$

which can be rearranged in the quadratic form:

$$\hat{\boldsymbol{v}}^{\mathrm{T}}(J\boldsymbol{p}_0\boldsymbol{p}_0^{\mathrm{T}}J - (\boldsymbol{p}_0^{\mathrm{T}}J\boldsymbol{p}_0 - 1)J)\hat{\boldsymbol{v}} = \hat{\boldsymbol{v}}^{\mathrm{T}}\, M\, \hat{\boldsymbol{v}} = 0 \tag{2.19}$$

where M is a symmetric matrix. Therefore it can be diagonalized, $M = C\,\Lambda\,C^{\mathrm{T}}$ with C orthogonal. Substituting in Eq. (2.19) we obtain

$$(\hat{\boldsymbol{v}}^{\mathrm{T}}\, C)\,\Lambda\,(C^{\mathrm{T}}\,\hat{\boldsymbol{v}}) = \hat{\boldsymbol{w}}^{\mathrm{T}}\,\Lambda\,\hat{\boldsymbol{w}} = 0 \qquad \text{where} \qquad \hat{\boldsymbol{w}} = C^{\mathrm{T}}\,\hat{\boldsymbol{v}} \tag{2.20}$$

and $\hat{\boldsymbol{w}}$ is a rotated unit vector. Because Λ contains the eigenvalues of M, 2.20 can be explicitly written as:

$$\lambda_1\, w_1^2 + \lambda_2\, w_2^2 + \lambda_3\, w_3^2 = 0 \tag{2.21}$$

where the eigenvalues λ_1, λ_2, and λ_3, are the diagonal terms of the eigenvalue matrix Λ and w_1, w_2, and w_3 are the three components of the unit-vector $\hat{\boldsymbol{w}}$. Now, in order

for Eq. (2.21) to admit solutions, the three eigenvalues cannot have the same sign, that is, one eigenvalue must be negative [1]. Let us consider $\lambda_3 < 0$ and $\lambda_1, \lambda_2 > 0$. Setting, $|\lambda_i| = \xi_i^{-2}$, Eq. (2.21) becomes

$$\frac{w_1^2}{\xi_1^2} + \frac{w_2^2}{\xi_2^2} = \frac{w_3^2}{\xi_3^2} \tag{2.22}$$

which is the equation of an elliptic cone with axis along w_3. Since $\hat{\boldsymbol{w}}$ is a unit-vector then the solution is the intersection of the elliptic cone given in Eq. (2.21) with a unit-radius sphere

$$w_1^2 + w_2^2 + w_3^2 = 1 \tag{2.23}$$

Solving 2.23 for w_3 and substituting in Eq. (2.22) we obtain the equation of the observed ellipse

$$w_1^2 \left(1 + \xi_3^2/\xi_1^2\right) + w_2^2 \left(1 + \xi_3^2/\xi_2^2\right) = 1 \tag{2.24}$$

which has the semi-axes

$$a' = \frac{1}{\sqrt{1 + \xi_3^2/\xi_1^2}} < 1 \qquad \text{and} \qquad b' = \frac{1}{\sqrt{1 + \xi_3^2/\xi_2^2}} < 1 \tag{2.25}$$

Therefore, the ratio of the observed ellipse semi-axes is $\rho = \dfrac{b'}{a'} = \sqrt{\dfrac{\xi_2^2(\xi_1^2 + \xi_3^2)}{\xi_1^2(\xi_2^2 + \xi_3^2)}}$. The observed ellipse, Eq. (2.24), has a not planar behavior. Indeed, on the imager it appears orthogonal to the elliptic cone axis (w_3 direction), as shown in Figure II.7. Since $\hat{\boldsymbol{w}}$ is a unit-vector, when it lies in the w_1-w_3 plane its expression is necessarily $\hat{\boldsymbol{w}}_a = \{a', 0, \sqrt{1 - a'^2}\}^{\mathrm{T}}$ while in the w_2-w_3 plane becomes $\hat{\boldsymbol{w}}_b = \{0, b', \sqrt{1 - b'^2}\}^{\mathrm{T}}$. Conversely, if $a' = b'$, the elliptical cone becomes a regular circular cone and the observed ellipse becomes a planar circle.

[1] If two eigenvalues are negative, then just change sign to all three terms in Eq. (2.21).

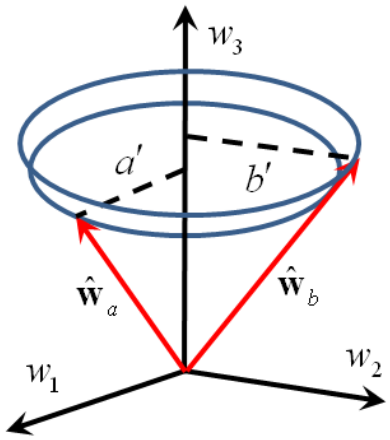


Figure II.7: 3D ellipse projection.

In conclusion, the projection of a three-axis ellipsoid on the imager will generally be an ellipse. Unfortunately, the direction from the camera to the center of the observed ellipse does not coincide with the direction to the center of the ellipsoid. This deviation, and associated correction, is explained and quantified below.

II.C.1. Observed Ellipse Center Offset

The values computed for the semi-axes, a' and b', Eq. (2.25), and the sign of eigenvalues, are necessary to find the body-center offset. Three distinct cases occur:

$$\left\{\begin{array}{lll} \text{if } \lambda_1\,\lambda_2 > 0 & \rightarrow & \left\{\begin{array}{l} \hat{\boldsymbol{w}}_a^{\mathrm{T}} = \{\pm a', \quad 0, \quad \sqrt{1-a'^2}\}^{\mathrm{T}} \\ \hat{\boldsymbol{w}}_b^{\mathrm{T}} = \{0, \quad \pm b', \quad \sqrt{1-b'^2}\}^{\mathrm{T}} \end{array}\right. \\ \text{if } \lambda_1\,\lambda_3 > 0 & \rightarrow & \left\{\begin{array}{l} \hat{\boldsymbol{w}}_a^{\mathrm{T}} = \{0, \quad \sqrt{1-a'^2}, \quad \pm a'\}^{\mathrm{T}} \\ \hat{\boldsymbol{w}}_b^{\mathrm{T}} = \{\pm b', \quad \sqrt{1-b'^2}, \quad 0\}^{\mathrm{T}} \end{array}\right. \\ \text{if } \lambda_2\,\lambda_3 > 0 & \rightarrow & \left\{\begin{array}{l} \hat{\boldsymbol{w}}_a^{\mathrm{T}} = \{\sqrt{1-a'^2}, \quad \pm a', \quad 0\}^{\mathrm{T}} \\ \hat{\boldsymbol{w}}_b^{\mathrm{T}} = \{\sqrt{1-b'^2}, \quad 0, \quad \pm b'\}^{\mathrm{T}} \end{array}\right. \end{array}\right. \tag{2.26}$$

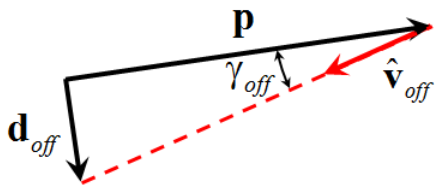


Figure II.8: Offset center vector computation.

Let $\hat{\boldsymbol{v}}_{off}$ be the direction to the center of the observed ellipse. Then,

$$\begin{cases} \text{if } \lambda_1\lambda_2 > 0 \text{ and } \lambda_2\lambda_3 < 0 \quad \rightarrow \quad \hat{\boldsymbol{v}}_{off} = -\text{sign}[\,\boldsymbol{p}^{\mathrm{T}} C(:,3)\,]\, C(:,3) \\ \text{if } \lambda_2\lambda_3 > 0 \text{ and } \lambda_1\lambda_2 < 0 \quad \rightarrow \quad \hat{\boldsymbol{v}}_{off} = -\text{sign}[\,\boldsymbol{p}^{\mathrm{T}} C(:,2)\,]\, C(:,2) \\ \text{if } \lambda_1\lambda_3 > 0 \text{ and } \lambda_2\lambda_1 < 0 \quad \rightarrow \quad \hat{\boldsymbol{v}}_{off} = -\text{sign}[\,\boldsymbol{p}^{\mathrm{T}} C(:,1)\,]\, C(:,1) \end{cases} \tag{2.27}$$

Figure II.8 shows that:

$$\begin{aligned} \cos\gamma_{off} &= -\hat{\boldsymbol{p}}^{\mathrm{T}}\hat{\boldsymbol{v}}_{off} \\ d_{off} &= ||\boldsymbol{p}||\,\tan\gamma_{off} = \ell_{off}\,\sin\gamma_{off} \end{aligned} \tag{2.28}$$

from which the offset vector in the body-fixed reference frame is:

$$\boldsymbol{d}_{off} = \boldsymbol{p} + \ell_{off}\,\hat{\boldsymbol{v}}_{off} = \boldsymbol{p} - \frac{||\boldsymbol{p}||}{\hat{\boldsymbol{p}}^{\mathrm{T}}\hat{\boldsymbol{v}}_{off}}\,\hat{\boldsymbol{v}}_{off} \tag{2.29}$$

Equation (2.29) can be used to evaluate the centroid offset angle, γ_{off}, provided that this angle is greater than the angle associated to the angular uncertainty of the position vector $\boldsymbol{p}$. In other words, let $\boldsymbol{\sigma}_p$ be the uncertainty of the position knowledge. This uncertainty is associated to the angle γ_p, where $|\boldsymbol{p}|\,\sin\gamma_p = |\boldsymbol{\sigma}_p|$. Therefore, the centroid offset can be corrected if and only if

$$\gamma_p < \gamma_{off} \tag{2.30}$$

The approach used above to determine the equation of the observed ellipse can be also used to find the terminator equation.

II.C.2. Terminator Equation

The canonical equation of an ellipsoid (in the observed body reference frame) is provided by Eq. (2.15). Let $\hat{\boldsymbol{v}}_s$ be the Sun rays direction in the body frame. This vector can be computed from time and from the observed body attitude. The vector $\boldsymbol{p} = \boldsymbol{x} + t\,\hat{\boldsymbol{v}}_s$ intersects the ellipsoid if $\boldsymbol{p}$ satisfies Eq. (2.15), $\boldsymbol{p}^{\mathrm{T}}\,J\,\boldsymbol{p} = 1$, obtaining

$$2t\,\hat{\boldsymbol{v}}_s^{\mathrm{T}} J\boldsymbol{x} + t^2\,\hat{\boldsymbol{v}}_s^{\mathrm{T}} J\hat{\boldsymbol{v}}_s = 0 \tag{2.31}$$

This equation has a trivial solution, $t_1 = 0$, and the solution $t_2 = -2\dfrac{\hat{\boldsymbol{v}}_s^{\mathrm{T}} J\boldsymbol{x}}{\hat{\boldsymbol{v}}_s^{\mathrm{T}} J\hat{\boldsymbol{v}}_s}$. By enforcing $t_2 = 0$, the two solutions are made coincident and the locations (on the ellipsoid) where Sun rays are tangent to the ellipsoid are obtained. This implies, $\hat{\boldsymbol{v}}_s^{\mathrm{T}} J\boldsymbol{x} = 0$. Therefore, the terminator equation is described by

$$\hat{\boldsymbol{v}}_s^{\mathrm{T}}\,J\,\boldsymbol{x} = 0 \qquad \text{subject to} \qquad \boldsymbol{x}^{\mathrm{T}}\,J\,\boldsymbol{x} = 1. \tag{2.32}$$

Setting:

$$\boldsymbol{w}_s = \sqrt{J}\,\hat{\boldsymbol{v}}_s = \left\{ \begin{array}{c} \dfrac{v_{s1}}{a} \\ \dfrac{v_{s2}}{b} \\ \dfrac{v_{s3}}{c} \end{array} \right\}^{\mathrm{T}} \qquad \text{and} \qquad \boldsymbol{y} = \sqrt{J}\,\boldsymbol{x} = \left\{ \begin{array}{c} \dfrac{x_1}{a} \\ \dfrac{x_2}{b} \\ \dfrac{x_3}{c} \end{array} \right\}^{\mathrm{T}}, \tag{2.33}$$

Eq. (2.32) becomes

$$\boldsymbol{w}_s^{\mathrm{T}}\,\boldsymbol{y} = 0 \qquad \text{subject to} \qquad \boldsymbol{y}^{\mathrm{T}}\,\boldsymbol{y} = 1. \tag{2.34}$$

This implies that the solution, $\boldsymbol{y}$, is a unit-vector ($\boldsymbol{y} = \hat{\boldsymbol{y}}$) orthogonal to $\boldsymbol{w}_s$. Rewriting $\hat{\boldsymbol{y}}$ in spherical coordinates:

$$\hat{\boldsymbol{y}} = \{\cos\varphi\cos\vartheta,\, \sin\varphi\cos\vartheta,\, \sin\vartheta\}^{\mathrm{T}},$$

and substituting together with Eq. (2.33) in Eqs. (2.32-2.34) gives

$$\frac{v_{s1}}{a}\,\cos\varphi\,\cos\vartheta + \frac{v_{s2}}{b}\,\sin\varphi\,\cos\vartheta + \frac{v_{s3}}{c}\,\sin\vartheta = 0 \tag{2.35}$$

which can be rewritten as:

$$\tan\vartheta = -\frac{c\,v_{s1}}{a\,v_{s3}}\,\cos\varphi - \frac{c\,v_{s2}}{b\,v_{s3}}\,\sin\varphi. \tag{2.36}$$

By varying φ between 0 and 2π, Eq. (2.36) allows to compute to corresponding value for ϑ. Therefore, the $\boldsymbol{x}$ vector in the body-fixed frame is given by

$$\boldsymbol{x} = \left(\sqrt{J}\right)^{-1}\hat{\boldsymbol{y}} = \left\{\begin{array}{c} a\,\cos\varphi\,\cos\vartheta \\ b\,\sin\varphi\,\cos\vartheta \\ c\,\sin\vartheta \end{array}\right\}. \tag{2.37}$$

Finally, the terminator in the camera frame, $\boldsymbol{t}_{\mathrm{c}}$, is provided by

$$\boldsymbol{t}_{\mathrm{c}} = C_{\mathrm{CO}}\,C_{\mathrm{OI}}\,C_{\mathrm{IB}}\,(\boldsymbol{x} - \boldsymbol{P}_0) \tag{2.38}$$

where $\boldsymbol{P}_0$ is the camera position vector in the body-fixed frame and the product $C_{\mathrm{CO}}\,C_{\mathrm{OI}}\,C_{\mathrm{IB}}$ is the transformation matrix moving from body-fixed to camera frame.

For a three-axis generic ellipsoid a simulation example is shown in Figure II.9

The above analysis allows for a complete characterization of the type of features which needs to be identified in the image, and also quantifies the center offset error. However, prerequisite to feature extraction is the pixel selection phase. The following sections describe this process in detail for the Eigenvalue method.

II.C.3. Principal Axes of Illuminated Area

The image is converted in binary by choosing a threshold greytone value, which is selected given the maximum and minimum values contained in the image. Then,

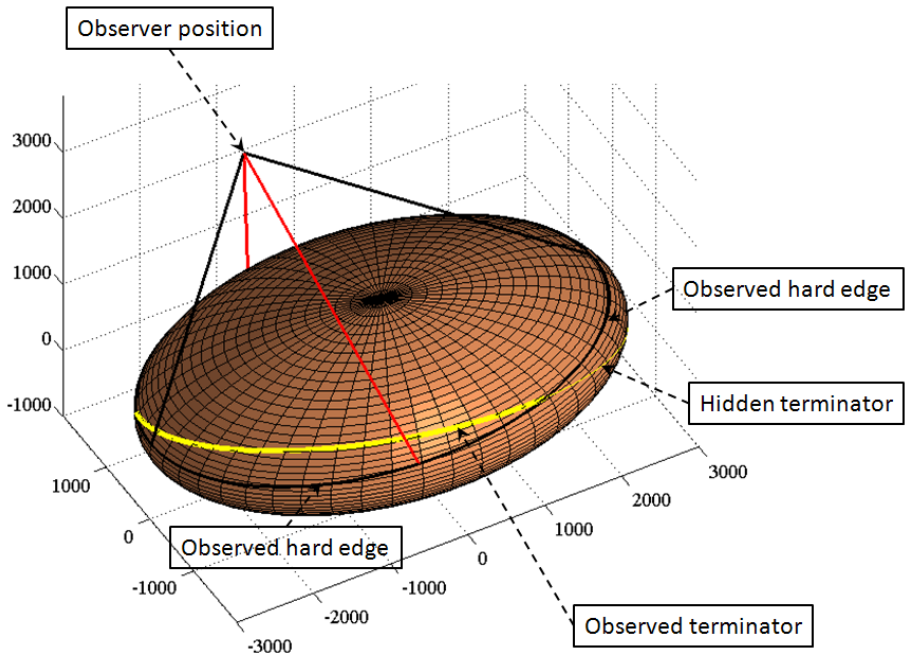


Figure II.9: Simulation example of observed ellipsoid.

the coordinates of the centroid can be found from:

$$r_b = \frac{\Sigma_{i,j}(\mathfrak{I}_{i,j}\, r_{i,j})}{\Sigma_{i,j}\mathfrak{I}_{i,j}} \quad \text{and} \quad c_b = \frac{\Sigma_{i,j}(\mathfrak{I}_{i,j}\, c_{i,j})}{\Sigma_{i,j}\mathfrak{I}_{i,j}} \tag{2.39}$$

where $[r_{i,j}, c_{i,j}]$ are the coordinates of a generic pixel, $\mathfrak{I}_{i,j}$ is the associated binary value, and the indexes i and j span the entirety of the selection box. Similarly, its inertia tensor is built according to:

$$T = \begin{bmatrix} \Sigma_{i,j}\, \mathfrak{I}_{i,j}\, (r_{i,j} - r_b)^2 & -\Sigma_{i,j}\, \mathfrak{I}_{i,j}\, (r_{i,j} - r_b)(c_{i,j} - c_b) \\ -\Sigma_{i,j}\, \mathfrak{I}_{i,j}\, (r_{i,j} - r_b)(c_{i,j} - c_b) & \Sigma_{i,j}\, \mathfrak{I}_{i,j}\, (c_{i,j} - c_b)^2 \end{bmatrix}. \tag{2.40}$$

From the eigen-analysis two values are obtained (λ_1 and λ_2), from which the inclination of the principal axes ($\vartheta_{\min}, \vartheta_{\max}$) can be determined according to the

following scheme:

$$\begin{cases} \text{if} \quad \lambda_1 > \lambda_2 \quad \rightarrow \quad \begin{cases} \vartheta_{\max} = \text{atan2}(w_{11}, w_{21}) \\ \vartheta_{\min} = \text{atan2}(w_{12}, w_{22}) \end{cases} \\ \text{if} \quad \lambda_2 > \lambda_1 \quad \rightarrow \quad \begin{cases} \vartheta_{\min} = \text{atan2}(w_{11}, w_{21}) \\ \vartheta_{\max} = \text{atan2}(w_{12}, w_{22}) \end{cases} \end{cases} \tag{2.41}$$

Finally, the axis of symmetry can be determined as the line passing through $[r_b, c_b]$ and having slope m, where $m = \tan \vartheta_{\max}$ if $\lambda_1 > \lambda_2$ and $m = \tan \vartheta_{\min}$ if $\lambda_1 < \lambda_2$. Moreover the ratio between the eigenvalues is a measure, albeit approximated, of how much of the target's surface is illuminated; values close to 0 indicating an almost invisible target and values close to 1 indicating an almost full Moon.

The determination of the axes allows to establish a reference frame within the image. Next, several pixels belonging to the edges of the target have to be selected, as explained below.

II.D. Pixel Selection

To create a mesh to analyze the image, a set of lines parallel to the axis of symmetry is considered, evenly spaced along the orthogonal axis. The spacing of the lines is chosen relatively to the p_r and p_i parameters. In particular, it is desirable that the lines extend across the majority of the target, but not so close to its limits, where its thickness is too little and may lead to errors. Experimentally, it has been found that this problem is avoided by choosing no more than 9 lines per side, which cover up to 80% of p_r length. Once the grid has been established, a mask is swept along it in the direction of the axis of symmetry, to find the target edges. The

mask is composed of a sequence of zeros and ones, for a total length of 8 pixels: $[0, 0, 0, 0, 1, 1, 1, 1]$. The positions were the correlation between the mask and the image is maximum are selected, as these will correspond to the pixels where the image switches from 0 to 1 value and vice versa. As can be seen in the example

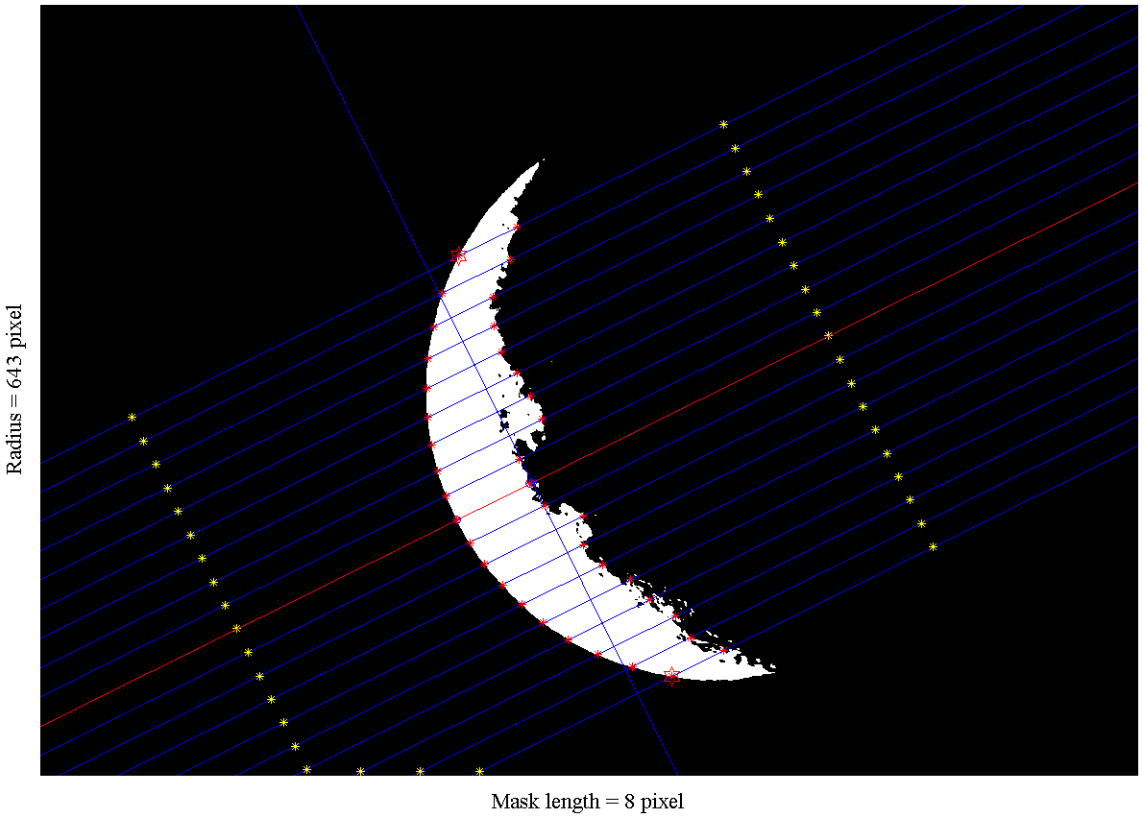


Figure II.10: Selection of hard edge and terminator points.

(Figure II.10), each line, swept in a pre-determinate direction, identifies two points, thus defining two sets of points, each arranged in a more-or-less arc configuration. Only one of those will correspond to the hard edge, while the other will be the terminator. It is however easy to distinguish between the two during the feature identification phase, because the terminator will be characterized by a much higher

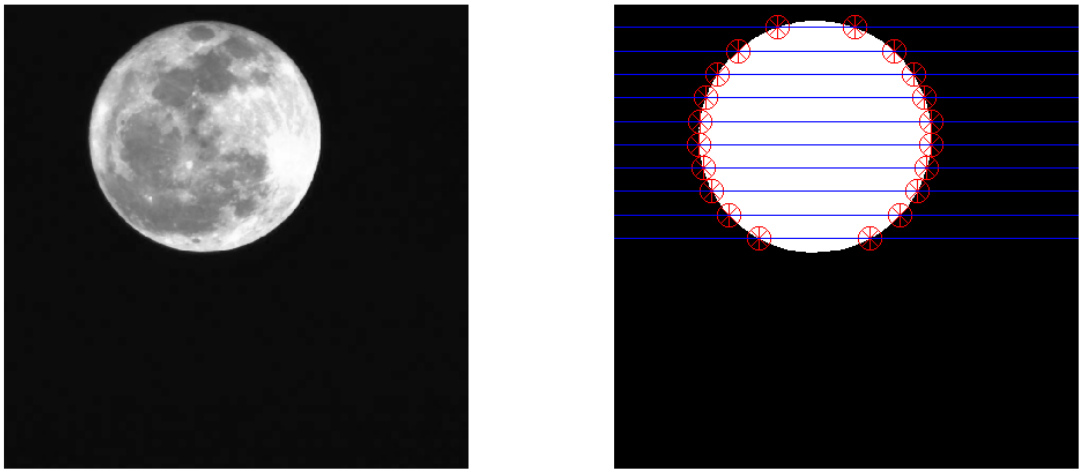


Figure II.11: Pixel selection for fully illuminated target. Left: Original Image. Right: Pixel selection.

value of standard deviation of the residuals than the hard edge. This is due to the shadows of the internal features of the target, which result in a relatively irregular boundary compared to the actual edge.

II.D.1. Fully Illuminated Moon

In the case of fully illuminated target, λ_1 and λ_2 are almost identical and it is impossible to define a principal axis. This obstacle is easily overcome, however, by noticing that the absence of a preferential direction in the image means any direction will work fine. Therefore, for simplicity the mask is swept along horizontal lines. Another difference with the general case is that now all points identified supposedly belong to the hard edge, and thus only one set of pixels is defined. An example of this case is depicted in Figure II.11.

II.E. Example

This section provides the numerical results for a real Moon image taken from ground (Houston area) on March 6, 2013 at 06:08:10 CDT. The original image is shown in the top-left of Figure II.12. The Moon distance is known with very low

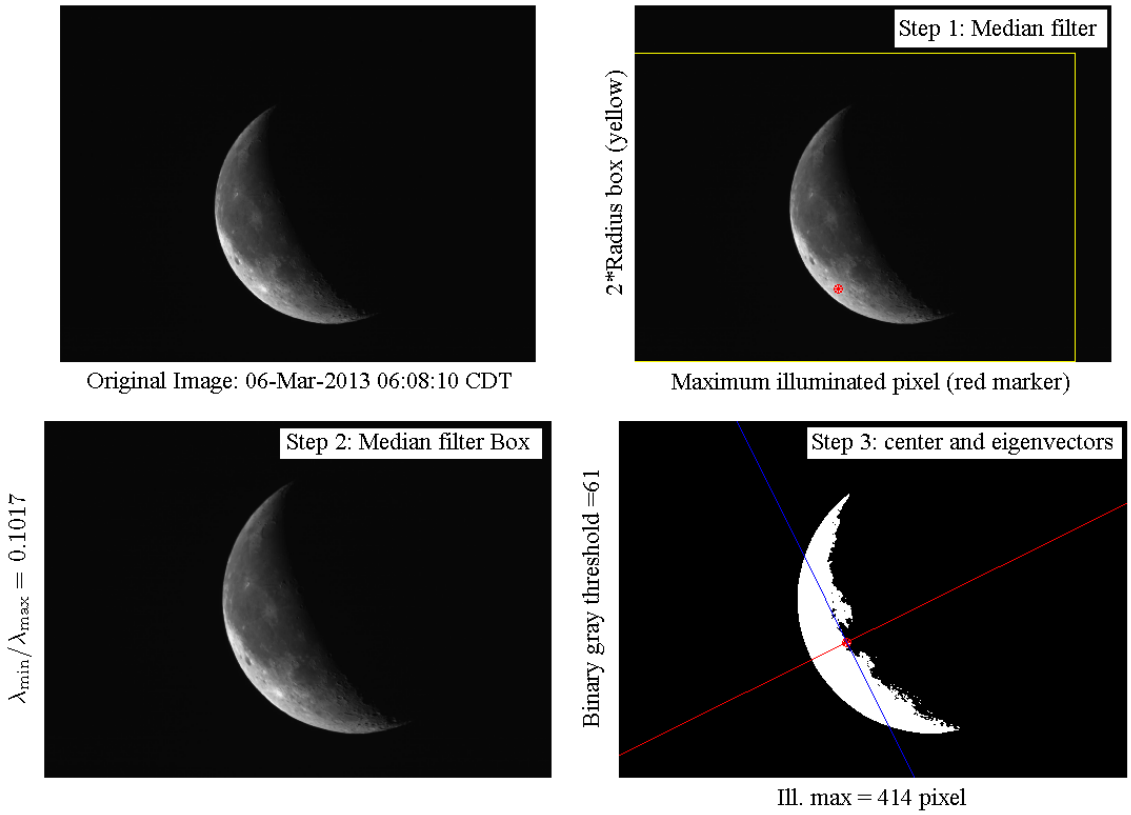


Figure II.12: Eigenvalue method example.

precision (some Earth radii). A simple 3×3 Gaussian filter is applied to smooth the image, mainly to identify a single pixel belonging to the Moon. The filter is first applied to remove eventual bright pixels of stars. The brightest pixel is shown in the top-right of Figure II.12. Around that pixel a box whose dimension is the observed

diameter of the Moon (computed using the approximated Moon distance and the camera parameters) is also shown. This box allows to restrict the image processing to a rectangular subset of the original image, sub-image shown in the bottom-left of Figure II.12. Using the value of the brightest pixel in the filtered image (414), a threshold value (61) is obtained to obtain the binary image shown in the bottom-right of Figure II.12. Eigenanalysis of the binary images, as provided by Eq. (2.39) through Eq. (2.41), gives a ratio of eigenvalues, $\frac{\lambda_{\min}}{\lambda_{\max}} = 0.107$, an indicator of the image portion being illuminated.

The axis of symmetry (eigenvector associated to the maximum eigenvalue) of the illuminated part is shown in red in the bottom-right of Figure II.12, while in blue is indicated the orthogonal direction (eigenvector associated to the minimum eigenvalue). The axis of symmetry, of course, just approximates the real symmetric axis of the illuminated area. Nevertheless, this approximate direction can be used to draw a set of 2×9 lines along the orthogonal axis with constant step size (1/11 of the estimated Moon radius). Along these lines a mask of 8 pixels ([0, 0, 0, 0, 1, 1, 1, 1]) is used to identify where is maximum the correlation with the two transitions (one associated with the terminator and one with the hard edge) in the binary images. These two sets of points are then sent to the best fitting subroutine. It is immediate to recognize the terminator because of its much higher deviation.

II.F. Drawbacks of the Eigenvalue Method

As described above, the Eigenvalue method provides a reliable and relatively straightforward way to select pixels from the image to initiate the feature identifica-

tion process. It works equally well for fully and partially illuminated targets, with only minor modifications to the algorithm. However it is unable to treat images where the target is cropped, because the image symmetry, which is at the basis of the method, is broken. To deal with this situation, a different approach was sought. This new technique, named Gradient method, later proved to be much more general than the Eigenvalue method, and therefore has supplanted it altogether.

II.G. Gradient Method

The Gradient method, not unlike the Eigenvalue method, takes advantage of the properties of the input images. In this case, the driving principle is that the greater variation of graytone within the image is expected to be localized at the target's edge. Therefore, pixels belonging to the edge will appear extremely bright in comparison, if the image is treated with a gradient filter, i.e. a filter that computes the derivative of the graytone at each point. Then, selecting the brightest pixels in the filtered image should provide with the desired data. However, because of the possible presence of disturbances, such as bright stars, target surface topology or atmospheric phenomena, and artificial objects orbiting Earth, some of these pixels will not belong to the edge. A procedure to remove these points, dubbed "outliers", from the data set is thus required. The remaining pixels can be used to find the best fitting circle and determine the sought geometric properties.

As a whole, the Gradient method is composed of two steps:

- First the image is processed with a differentiation filter. As the image is a discretized, bidimensional matrix of integer between 0 and 255, a numerical

differentiation scheme has been implemented, followed with a single application of Richardson extrapolation, to improve accuracy. The pixels with highest value of the gradient are then selected.

- Then the dataset is scanned to identify and remove outliers. Two different algorithm in sequence are applied, to minimize the possibility of any outlier polluting the data.

Figure II.13 illustrates the information flow of the algorithm, while the theory behind these two steps is discussed in detail further below.

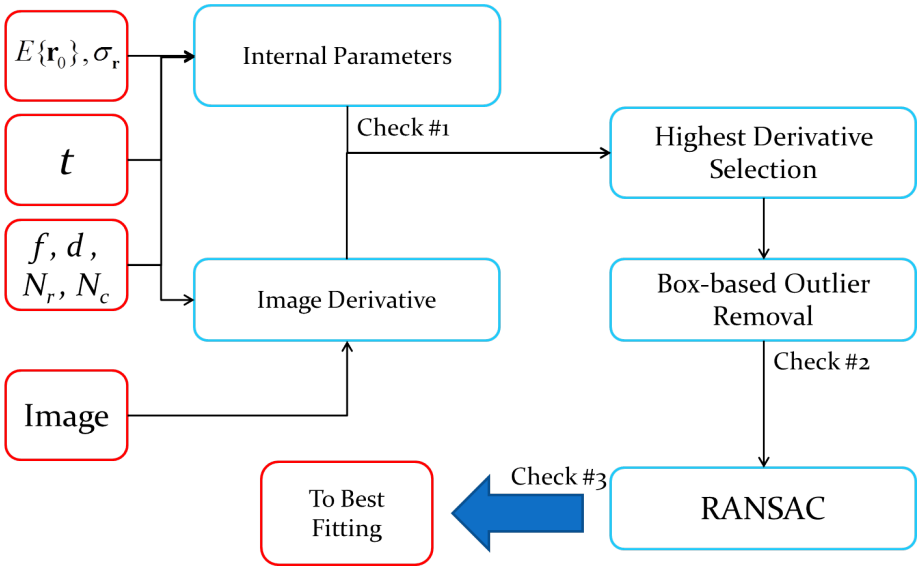


Figure II.13: Gradient method flowchart.

II.H. Image Differentiation

The image is a bidimensional array of integer values, ranging from 0 to 255, representing the graytone of each pixel. The algorithm to compute the gradient has

been chosen as a four-point central difference followed by Richardson extrapolation to increase the accuracy.

Given a generic pixel of row and column index $[r_i, c_i]$, its row and column four-point central differences are computed according the the following scheme:

$$\begin{cases} g'_r(r_i, c_j) &= \dfrac{8\,(\mathfrak{I}_{i+1,j} - \mathfrak{I}_{i-1,j}) - (\mathfrak{I}_{i+2,j} - \mathfrak{I}_{i-2,j})}{12\,h} \\ \\ g'_c(r_i, c_j) &= \dfrac{8\,(\mathfrak{I}_{i,j+1} - \mathfrak{I}_{i,j-1}) - (\mathfrak{I}_{i,j+2} - \mathfrak{I}_{i,j-2})}{12\,h} \end{cases} \tag{2.42}$$

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

$$g'(r_i, c_i) = \sqrt{g'^2_r(r_i, c_i) + g'^2_c(r_i, c_i)} \tag{2.43}$$

II.H.1. Richardson Extrapolation

Richardson extrapolation is a sequence acceleration method. Among its many applications, it can be used to improve the accuracy of a difference scheme, with a much smaller computational burden that it would cost increasing the order of the scheme itself. The basic idea is to extrapolate from two different estimates of a derivative, obtained with different stepsizes [5]. Thus consider two approximations of Eq. (2.43), one with step $2\,h$ and one time with step h:

$$\begin{cases} g'|_{true} &= g'_{2h} + a_0\,(2h)^4 \\ g'|_{true} &= g'_h + a_0\,h^4 \end{cases} \tag{2.44}$$

By equating:

$$g'_h - g'_{2h} = a_0\,h^4(2^4 - 1) \quad \rightarrow \quad a_0\,h^4 = \frac{1}{2^4 - 1}\,(g'_h - g'_{2h}) \tag{2.45}$$

and therefore

$$\boxed{g'|_{true} = g'_h + \frac{1}{2^4 - 1}\left(g'_h - g'_{2h}\right)} \tag{2.46}$$

To show the effectiveness of this approach, the results of a test done with the six-hump Camel back function (Eq. (2.47)), depicted in Figure II.14, are here reported.

$$f(x,y) = \frac{x^2}{4}\left(4 - \frac{21}{40}x^2 + \frac{x^4}{8}\right) + \frac{xy}{4} + y^2\left(\frac{y^2}{4} - 1\right) \tag{2.47}$$

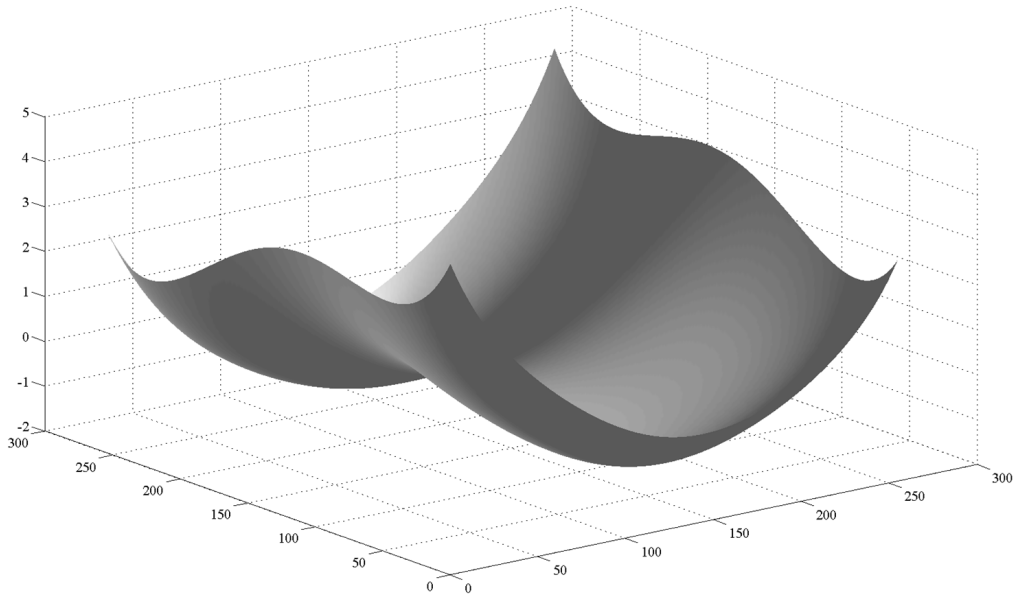


Figure II.14: Six-hump camel-back function.

This function belongs to a family of polynomials widely used as benchmarks for numerical methods because of their peculiar shape. For this test, four different approximations are obtained. The errors with respect to the analytic values are shown in Figure II.15, in regard to the following four cases:

1. Top-left: 2-point central differentiation without Richardson extrapolation;

2. Top-right: 4-point central differentiation without Richardson extrapolation;

3. Bottom-left: 2-point central differentiation with Richardson extrapolation;

4. Bottom-right: 4-point central differentiation with Richardson extrapolation;

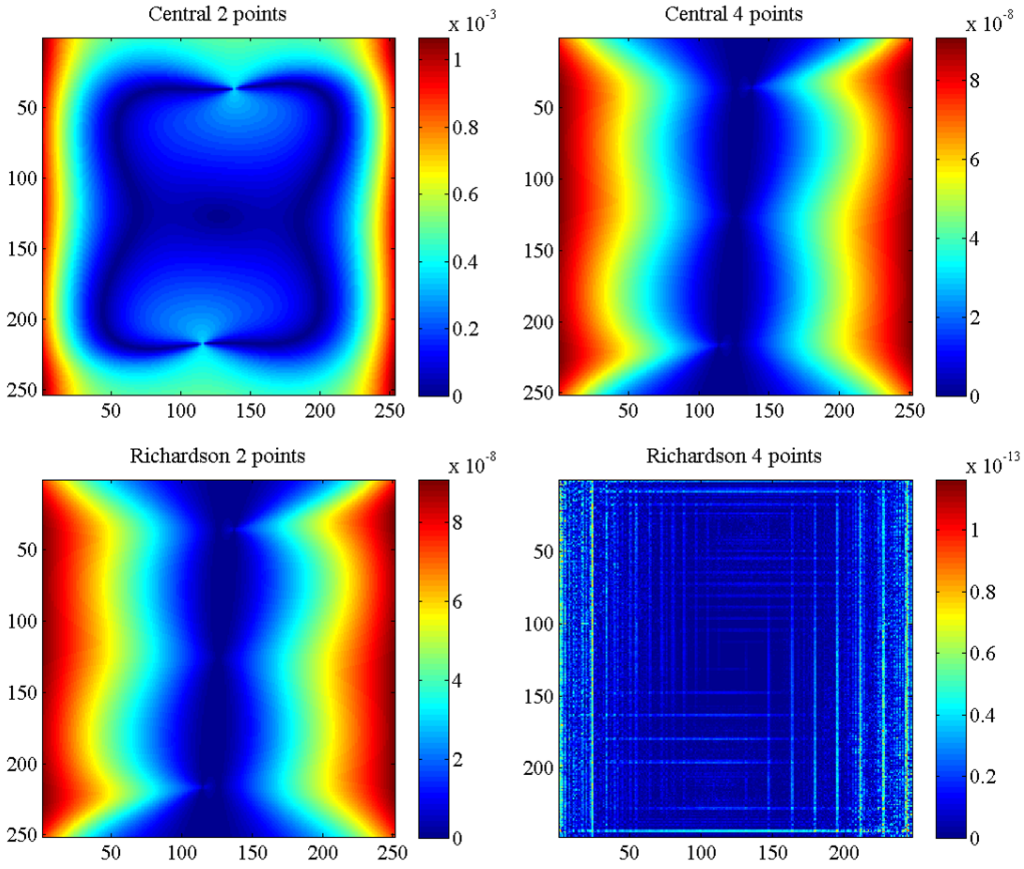


Figure II.15: Accuracy results for the six-hump camel back function example.

These results show that: 1) the accuracy provided by 4-point central differentiation without Richardson extrapolation is the same as provided by 2-point central

differentiation with Richardson extrapolation and 2) the 4-point central differentiation with Richardson extrapolation reaches an error of the same order of magnitude as the machine precision. This last combination has been used in the rest of the work to compute the image gradient.

Once the gradient image has been created, the pixels are sorted in descending order. As mentioned before, not all of these pixels belong to the target's hard edge, and thus these "outliers" need to be filtered out. To this purpose two methods, applied in succession for maximum effect, have been coded.

II.I. Box-based Outliers Identification

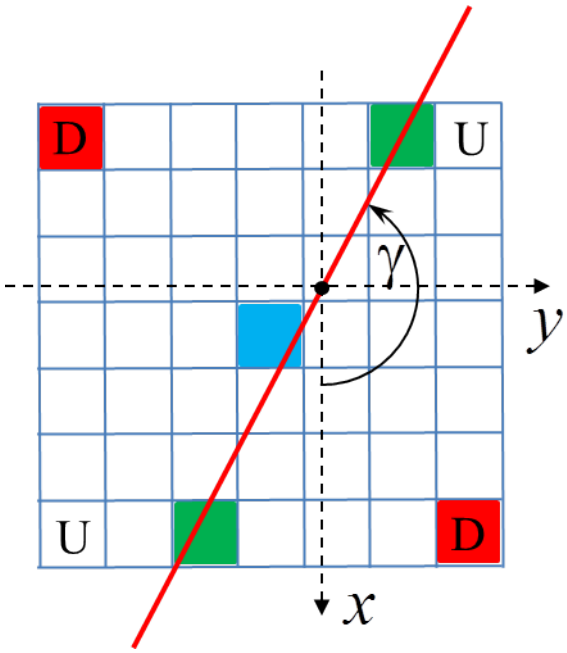


Figure II.16: Mask for the box-based outlier identification technique (mask size: 7×7).

The first approach is based on a local analysis around the pixel being considered as potentially belonging to the target's edge. A mask like the one in Figure II.16, whose size depends on the parameter p_r, is centered on a candidate pixel. If the pixel does indeed belong to the target's edge, then it is likely to be part of a sequence of pixels of comparable brightness distributed along a preferred direction. Moreover, the pixel should sit on the border between two areas where the graytone "flattens" (the inside and outside of the target) and which therefore should appear as dark in the gradient image. If the pixel does not satisfy all of these conditions, it is rejected. To perform this analysis, first the tensor of inertia of the gradient image within the mask is computed, using the same equations as in II.C.3, and then the eigenvectors and eigenvalues are found. The eigenvector associated to the minimum eigenvalue (red line direction in Figure II.16) is inclined by an angle γ with respect to the row axis direction and represents the direction of the supposed edge. If $\gamma \mod \pi < \pi/2$ then the farthest pixels from the edge line are those identified in green while if $\gamma \mod \pi > \pi/2$ the farthest pixels are the red ones. Several conditions have to be satisfied for the pixel to be accepted:

- The ratio between the eigenvalues must be lower than a certain threshold, chosen experimentally at 0.3. This indicates the distribution of bright pixels has a preferential direction.

- The values of the gradient in the pixels furthest from the edge line should be small, as both should be far from areas of variable brightness. Moreover, their graytones should differ noticeably, as one belongs to the target and the other to the night sky.

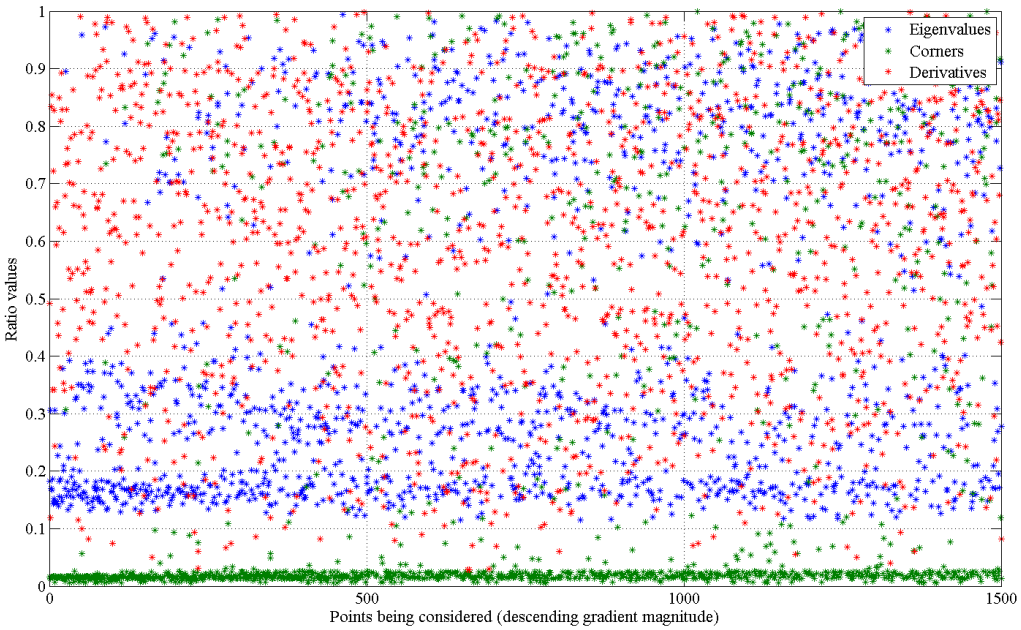


Figure II.17: Outliers: test results example.

- For the pixels highlighted in green in Figure II.16, the ratio of their gradient values should be close to 1 ($g_{\min}/g_{\max} > 0.6$) as they both supposedly belong to the edge.

- For the pixels highlighted in red in the same figure, the ratio of their graytone values should be close to 0 ($g_{\min}/g_{\max} < 0.3$) as one belongs to the nught sky and the other to the target.

An example for these value in a processed image is given in Figure II.17. In this figure the green markers indicate the ratios between the gray tones of the farthest pixels from the edge line, the blue markers the ratios between the eigenvalues of the gradient box, and the red markers the derivatives ratios of pixels at the extremes of the edge line.

II.J. RANSAC for Circle and Ellipse

The second technique implemented in the code is a standard algorithm to eliminate outliers called RANSAC which is an abbreviation for "RANdom SAmple Consensus" [10]. It was developed for robust fitting of models in the presence of many data outliers. The algorithm is very simple and can be applied in the fitting of many geometric entities, such as lines, circles, ellipses, etc. It is a non-deterministic iterative algorithm, in the sense that it produces a reasonable result only with a given probability, which increases as more iterations are allowed. It works by selecting random subsets of size equal to the number of unknown parameters, and then checks whether the remaining datapoints lie further than a certain threshold from the curve determined by the initial subset. Repeating this process for many randomly chosen subsets is likely to find and remove the outliers. More formally, given a fitting problem with n data points and unknown m parameters, starting with $N_{\min} = n$, the classic RANSAC algorithm perform N_{test} times the following steps:

1. selects m data points at random.

2. finds the curve passing through these m points, thus estimating the m parameters associated.

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 $N_{\min} = N_k$.

The basic cycle is depicted in Figure II.18.

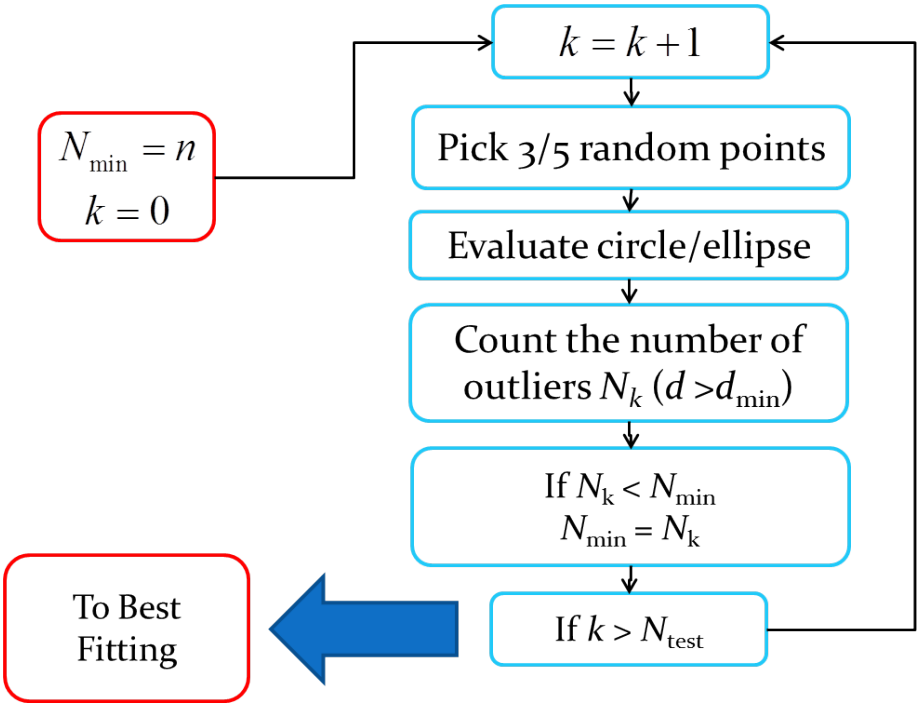


Figure II.18: RANSAC flowchart.

Crucial to a successful implementation of RANSAC is the method used for the random selection of subsets. A purely random selection approach may persist choosing outliers, with great waste of computational time. On the other hand, a procedure based on deterministic inner loops can also be inefficient, if by chance one outlier is selected by the outmost loop. Therefore, as an improvement on the basic method, a technique originally developed for Pyramid Star-Identification [29], has been adopted, which always selects different subsets by continuously rotating the dataset indexes.

II.J.1. Gradient Method Parameters

To properly implement the outlier removal routines, a series of parameters must be defined, which help define thresholds and safety checks to ensure the algorithm is working correctly. These parameters are defined in terms of p_r and p_i, as defined in section II.A.1. They are:

- N: the number of pixels in the gradient image to be selected for analysis. While it speeds up the process to choose a smaller set, it is important to ensure enough pixels are selected, as some outliers might have higher brightness values than some of the "good" pixels. This is especially true for the cases of crescent or gibbous Moon, where pixels on the terminator have higher gradient than pixels on the extremities of the illuminated arc. Ultimately N is calculated according to the following expression

$$N = k\pi p_r \tag{2.48}$$

  where k is a factor that varies linearly with p_i, from a minimum of 2 when $p_i \leq 10$ to a maximum of 4, when $p_i \approx 2p_r$:

$$k = 2\left(1 + \frac{p_i - 10}{2\,p_r - 10}\right) \tag{2.49}$$

- s: size of the box for box based outliers removal. The box has to be chosen sufficiently small, so that the segment of Moon edge within it appears almost straight. Extensive testing has shown that the best choice is to take $s = \lceil 0.8 p_r \rceil$, with a lower limit of $s = 3$ and an upper limit of $s = 21$.

- $d_{\max}$: maximum distance for RANSAC. Chosen proportional to p_r, according to $d_{\max} = 5\,p_r/100$.

- N_T: Number of tests for RANSAC. This is chosen as a fraction of all the possible combinations of p elements taken from the set of N_1 pixels remaining after the box-based filtering.

$$N_T = \frac{1}{5}\binom{N_1}{p} \tag{2.50}$$

  Additionally, an upper limit for this parameter is set at 500.

- $\Delta X_{\min}$: convergence tolerance for CSF Least Squares. This parameter is obviously necessary for the latter best fitting rather than the pixel identification, but it is here described for completeness. Assuming the requirement accuracy is 0.3 pixel (in center and radius estimation), the tolerance is chosen to be 1/50 of that value, $\Delta X_{\min} = 0.3/50$.

CHAPTER III

IMAGE PROCESSING - FEATURE RECOGNITION

Once the pixel in the image have been selected, it is finally possible to best fit the data to determine the geometric characteristics of the observed body. As mentioned before, these are center O and apparent radius r_c for a circle, and center O, axes a and b and inclination θ for an ellipse. Now, the problem of recognizing geometrical entities and reduce them to a minimal set of parameters, sometimes called *feature extraction*, is very important in image processing for a number of applications, foremost among them computer graphics and computer vision. Consequently a great amount of research has gone into developing methods to efficiently and univocally recognize geometrical shapes within an image; in particular, much more literature is available regarding circle identification than ellipse. This is partially due to the fact that the first case is more treatable. Nevertheless, there are some reliable methods available for ellipse identification, which will be described below. Of course, trying to identify ideal geometrical figures within a discrete representation of the physical world reduces essentially to a best fitting problem, which is way all the methods discussed are by all intents and purposes best fitting techniques. Currently, feature recognition techniques for conic curves fall within one of the following categories:

In **Geometric Fit**, given a set of n data points, one tries to iteratively converge to the minimum of

$$\mathbb{L}_g = \sum_{i=1}^{n} d_i^2(\boldsymbol{x}) \tag{3.1}$$

where d_i are geometric distances from the data points to the conic, and $\boldsymbol{x}$ is the vector of the parameters to be found.

In **Algebraic Fit**, the d_i are replaced with some other function of the parameters f_i, and thus a new cost function is defined:

$$\mathbb{L}_a = \sum_{i=1}^{n} f_i^2(\boldsymbol{x}) \tag{3.2}$$

These functions typically are algebraic expression of the conic in question, which are usually simpler than the expression for geometric distances.

Historically, geometric fit has been regarded as being the more accurate of the two. However, this accuracy comes at the price of added computational cost, and possibility of divergence. Algebraic fit, instead, is typically reliable, simple and fast. A common practice is to use an algebraic fit to generate an initial guess for a subsequent iterative geometric fit, but in recent years, new algebraic fitting routines have shown such a degree of accuracy that geometric fits cannot improve it noticeably [2].

To find the algorithm that best suits the mission needs, an extensive analysis of various methods has been done (both geometric and algebraic) for the circle and the ellipse. From the abundance in literature, there are many advanced circle algebraic fits with accuracy comparable to iterative methods making the first surely the better choice. For the ellipse the possibility of a geometric fit has been explored to compare with the few algebraic fits found in literature. The methods for circle best fitting are described first, and then those for ellipse best fitting.

III.A. Circle Best Fitting

A great deal of best fitting algorithm for the circle is available in literature. Many of those are geometric methods, for which Eq. (3.1) assumes the form:

$$\mathbb{L}_g = \sum_{i=1}^{n} [\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r_c]^2 \tag{3.3}$$

and their reliability and accuracy is ultimately solely dependant on the iterative scheme implemented. Typically, the Levenberg-Marquardt method is considered the most accurate [26]. However, a great number of algebraic fits have been developed as well, and several of these are accurate enough to compete with more complex geometric fits. This, coupled with the desire to keep the computational load on the spacecraft system low, makes algebraic fits more suitable for the mission. Table III.1 summarizes several algebraic fits which have been thoroughly analyzed by Chernov [2, chap. 5]. Among these, the Taubin fit has been chosen to be implemented in the code, because of its accuracy, ease of implementation and reliability. Therefore, this method is described in further detail.

III.A.1. Taubin Best Fit

It has been shown in [2, pp. 127-130] that Kåsa, Pratt and Taubin fit can be described with very similar formalism using matrix representation. For convenience, that treatment is briefly reported here.

Given the implicit equation for a circle

$$a\,(x^2 + y^2) + b\,x + c\,y + d = a\,z + b\,x + c\,y + d = 0, \tag{3.4}$$

Table III.1: Summary of algebraic best fitting algorithms for circle.

Method	**Pros**	**Cons**
Kåsa [20]	Simplest, fastest method, consistent for vanishing noise	Underestimate radius with data distributed over small arcs
Pratt [32]	Same as Kåsa, plus can treat collinear data	Numerically unstable depending on the implementation
Taubin [41]	Similar to Pratt, but can be extended to other curves	n/a
Kukush-Markovsky-van Huffel [36]	Consistent estimator	Works best with big dataset

an algebraic best fit minimizes the function:

$$\mathcal{F}(\boldsymbol{A}) = \frac{1}{n}\sum_{i=1}^{n}(a\,z_i + b\,x_i + c\,y_i + d)^2 = n^{-1}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z})\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{A} \tag{3.5}$$

subject to a constraint

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{N}\boldsymbol{A} = 1 \tag{3.6}$$

where

$$\boldsymbol{A} = \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix}, \quad \boldsymbol{Z} = \begin{bmatrix} z_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ z_n & x_n & y_n & 1 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{M} = \frac{1}{n}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z} = \begin{bmatrix} \overline{zz} & \overline{zx} & \overline{zy} & \overline{z} \\ \overline{zx} & \overline{xx} & \overline{xy} & \overline{x} \\ \overline{zy} & \overline{xy} & \overline{yy} & \overline{y} \\ \overline{z} & \overline{x} & \overline{y} & 1 \end{bmatrix}, \tag{3.7}$$

and the shorthand $z_i = x_i^2 + y_i^2$ and $\overline{z} = \frac{1}{n}\sum z_i, \ldots$ have been used. The constraint matrix $\boldsymbol{N}$ determines the specific fit. In particular,

$$\boldsymbol{N} = \boldsymbol{K} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \rightarrow \quad a = 1, \tag{3.8}$$

for Kåsa fit,

$$\boldsymbol{N} = \boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & 0 & 0 & 0 \end{bmatrix} \quad \rightarrow \quad b^2 + c^2 - 4ad = 1, \tag{3.9}$$

for Pratt, and

$$\boldsymbol{N} = \boldsymbol{T} = \begin{bmatrix} 4\overline{z} & 2\overline{x} & 2\overline{y} & 0 \\ 2\overline{z} & 1 & 0 & 0 \\ 2\overline{y} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \rightarrow \quad 4a^2\overline{z} + 4ab\overline{x} + 4ac\overline{y} + b^2 + c^2 = 1, \tag{3.10}$$

for Taubin. It is also worth mentioning that the constraint (3.10) can be rewritten as two constraints, namely $d = -a\overline{z} - b\overline{x} - c\overline{y}$ and $b^2 + c^2 - 4ad = 1$, which shows how Taubin is identical to Pratt but with an additional constraint to the minimization problem. The constrained minimization problem (3.5) - (3.6) can be solved by using Lagrange multipliers to define the new unconstrained function:

$$\mathcal{G}(\boldsymbol{A}, \eta) = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{A} - \eta(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{N}\boldsymbol{A} - 1). \tag{3.11}$$

Differentiation leads to

$$\boldsymbol{M}\boldsymbol{A} = \eta\boldsymbol{N}\boldsymbol{A}. \tag{3.12}$$

which indicates $\boldsymbol{A}$ must be a generalized eigenvector of the pair $(\boldsymbol{M}, \boldsymbol{N})$, subject to (3.6). Because for each solution it is always true that

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{A} = \eta\boldsymbol{A}^{\mathrm{T}}\boldsymbol{N}\boldsymbol{A} = \eta \tag{3.13}$$

then the solution with with the smallest η will minimize the $\mathcal{G}(\boldsymbol{A}, \eta)$. Note that η is never negative since $\boldsymbol{M}$ is a positive semi-definite matrix. It can be seen that the constraint matrix (3.10) has rank 3, which indicates the problem can be reduced. This is important because an eigenvalue problem of rank 3 can be solved explicitly with a cubic equation without using SVD decomposition or other advanced methods.

Assuming the data set is centered, i.e $\overline{x} = \overline{y} = 0$, the first constrain for Taubin becomes $d = -a\overline{z}$. This allows to eliminate the fourth parameter,and the minimization problem is reduced to

$$\mathcal{F}_T(a, b, c) = \sum_{i=1}^{n} [a(\overline{z} - z_i) + bx_i + cy_i]^2 \quad \text{subject to} \quad 4\overline{z}a^2 + b^2 + c^2 = 1. \tag{3.14}$$

Further simplification is obtained by defining $a_0 = 2\overline{z}^{1/2}a$, which leads to the final system

$$\mathcal{F}_T(\boldsymbol{A}_0) = \sum_{i=1}^{n} \left[\frac{z_i - \overline{z}}{2\overline{z}^{1/2}} a_0 + bx_i + cy_i \right]^2 = \boldsymbol{A}_0^{\mathrm{T}} (\boldsymbol{Z}_0^{\mathrm{T}} \boldsymbol{Z}_0) \boldsymbol{A}_0 \quad \text{subject to} \quad ||\boldsymbol{A}_0|| = 1 \tag{3.15}$$

where $\boldsymbol{A}_0 = [a_0, b, c]^{\mathrm{T}}$ and

$$\boldsymbol{Z}_0 = \begin{bmatrix} (z_1 - \overline{z})/(2\overline{z}^{1/2}) & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ (z_n - \overline{z})/(2\overline{z}^{1/2}) & x_n & y_n \end{bmatrix}$$

which is the required formulation of rank three.

III.B. Ellipse Best Fitting

The best fitting of an ellipse is a sensibly more complicated problem than for the circle. First of all, a generic ellipse is uniquely defined by 5 parameters rather than 3; this leads immediately to a more complicated expression for geometric fit, to the point that most authors focus exclusively on algebraic fit for this conic section. Moreover, for ellipses with very low eccentricity the inclination becomes unobservable. This is an important problem when observing Earth, since, having an estimated equatorial radius of $6,378.1$ km and a polar radius of $6,356.8$ km, its flattening is only 0.0033528,

which means the Earth projects an ellipse with $e = 0.0067$ when observed from the equator. This number rapidly approaches 0 as the latitude of the observer reaches the poles. Such small values of e will be easily masked by the presence of noise and indeed it is necessary to assume that above a certain latitude, considering the typical levels of noise encountered in this research, it is impossible to discern the elliptical shape. Therefore, the target must be considered a circle. Even when the target can be considered elliptical, the greatest errors will be experienced in the estimate of the inclination.

III.B.1. Geometric Fit for the Ellipse

To attempt a geometric fit of the ellipse, it is important to have a good parametric representation, as this will greatly simplify the successive work. Also, by considering the fact that the target flattening is known, the number of parameters can be reduced to four given that $f = 1 - \frac{b}{a} = 1 - \rho$. A compact way to write the equation of a generic ellipse as explicit function of its center $[x_c, y_c]$, semi-major axis a and inclination θ is

$$F(\boldsymbol{B}) = F(x_c, y_c, a, \theta) = (\boldsymbol{X} - \boldsymbol{C})^{\mathrm{T}} \, R(\theta)^{\mathrm{T}} \, J \, R(\theta) \, (\boldsymbol{X} - \boldsymbol{C}) - 1 = 0 \tag{3.16}$$

where $\boldsymbol{X} = [x, y]^{\mathrm{T}}$, $\boldsymbol{C} = [x_c, y_c]^{\mathrm{T}}$ and

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}; \quad J = \begin{bmatrix} 1/a^2 & 0 \\ 0 & 1/(\rho a)^2 \end{bmatrix} \tag{3.17}$$

Given a set of points, it is possible to use Eq. (3.16) to set up a Least Squares based best fit. To this purpose, the Jacobian of F is required:

$$\boldsymbol{J} = \frac{\partial F}{\partial \boldsymbol{B}} = \begin{bmatrix} -2R(\theta)^{\mathrm{T}}\, J\, R(\theta)(\boldsymbol{X} - \boldsymbol{C}) \\ -\frac{2}{a}(F+1) \\ (\boldsymbol{X} - \boldsymbol{C})^{\mathrm{T}}\,(\dot{R}(\theta)^{\mathrm{T}}\, J\, R(\theta) + R(\theta)^{\mathrm{T}}\, J\, \dot{R}(\theta))(\boldsymbol{X} - \boldsymbol{C}) \end{bmatrix}. \tag{3.18}$$

Then, the solution for the parameter vector $\boldsymbol{B}$ has the form [4, pp. 24-29]

$$\boldsymbol{B} = -(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\,\boldsymbol{J}\,F. \tag{3.19}$$

which is to be solved at every iteration, as this approach is based on a linearization of a non-linear problem. To this purpose, to improve the stability and convergence speed, the Levenberg-Marquard [26] correction has been implemented, replacing $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$ with $\boldsymbol{J}_\lambda = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} + \lambda\,\boldsymbol{I}$ where λ is an additional positive control parameter to be changed at every iteration.

III.B.2. Geometric Fit Results

A statistical analysis has been conducted to test the effectiveness of the method described above. To this purpose, simulated data have been generated by superimposing Gaussian noise to a known ellipse. Moreover, for each test a new data set has been used, with inclination of the target ellipse randomly varying. A typical result, obtained over $10,000$ runs is depicted in Figure III.1. As one can see, the error for center and semi-major axis is the same, less than 3 pixels almost always, while the inclination error is below 3 degrees. However, these results are for an ellipse with $\rho = 0.9$. As this number approaches 1, the accuracy on inclination declines dramati-

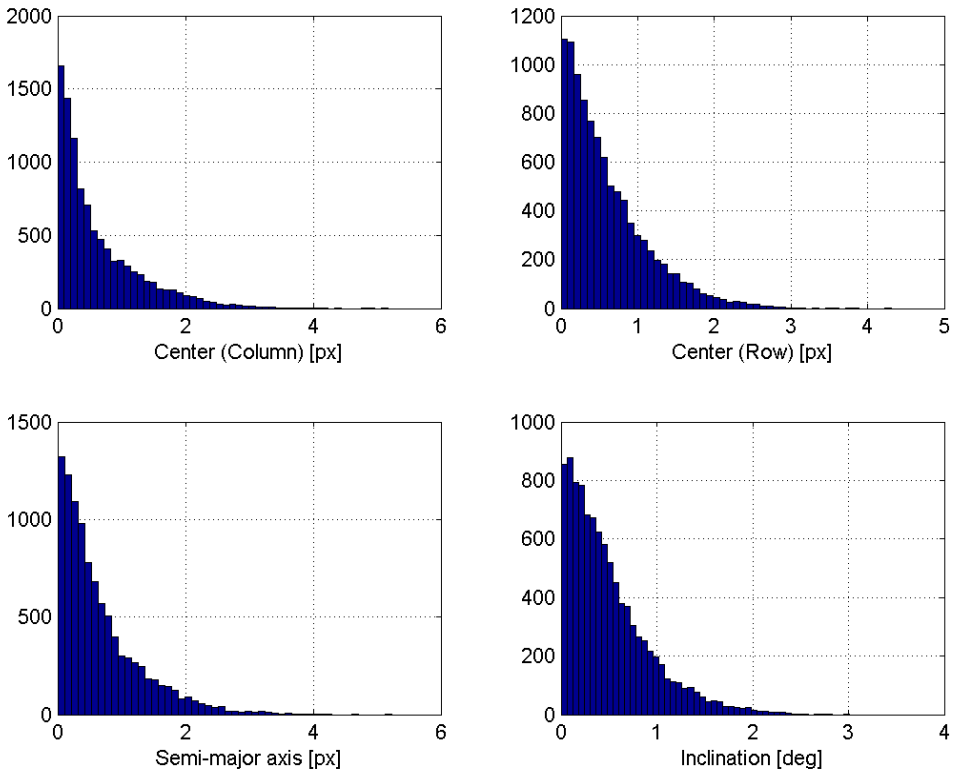
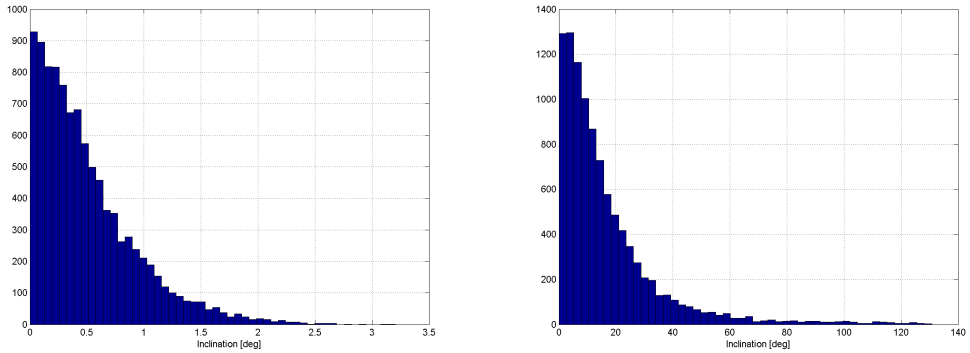


Figure III.1: Geometric Best Fit of ellipse over 10,000 runs ($\rho = 0.9$).



(a) $\rho = 0.99$ **(b)** $\rho = 0.9966$

Figure III.2: Inclination error for different values of ρ.

cally (see Figure III.2). The algorithm thus becomes completely unreliable for values of flattening within the range of interest. Therefore this geometric fit, while capable of good performance in general, it is not suitable for this application. It is necessary to turn to algebraic fits.

III.B.3. Algebraic Fit for the Ellipse

Among the few algebraic fits for ellipses, the method developed by Fitzgibbon[11] has been found to give good results but its original formulation has several drawbacks, specifically numerical instabilities and inability to treat ideal cases (i.e. data perfectly fitting the model). However, these issues have been solved in [15] which has the added benefit of reducing the problem to the eigenanalysis of a 3×3 matrix, which as mentioned before can be solved with elementary techniques. The original Fitzgibbon method and its modification are detailed next.

III.B.4. Improved Fitzgibbon Ellipse Fit

The implicit equation of a conic is

$$f(a,b,c,d,e,f) = ax^2 + bxy + cy^2 + dx + ey + f = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{A} = 0. \tag{3.20}$$

For this equation to represent an ellipse, the relation $b^2 - 4ac < 0$ must be satisfied. Since inequality constraints are more complicated to enforce, it is convenient to note that the parameter vector $\boldsymbol{A}$ can be freely scaled, as this does not change the final solution. Then, the constraint can be changed into an equality constraint, $4ac - b^2 = 1$. Therefore the minimization problem can be written in matrix form as

$$\min_{\boldsymbol{A}} \boldsymbol{A}^{\mathrm{T}} D^{\mathrm{T}} D \boldsymbol{A} = \min_{\boldsymbol{A}} \boldsymbol{A}^{\mathrm{T}} S \boldsymbol{A} \qquad \text{subject to} \qquad \boldsymbol{A}^{\mathrm{T}} C \boldsymbol{A} = 1. \tag{3.21}$$

where

$$D = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_n y_n & y_n^2 & x_n & y_n & 1 \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.22}$$

The solution of this optimization problem is the generalized eigenvalue problem

$$S\boldsymbol{A} = \lambda C \boldsymbol{A} \quad \text{subject to} \quad \boldsymbol{A}^{\mathrm{T}} C \boldsymbol{A} = 1, \tag{3.23}$$

which has already been encountered for circle best fit, and whose solution is the eigenvector associated with the smallest positive eigenvalue. As mentioned before,

solving Eq. (3.23) in this form leads to numerical instability. Instead, it is possible to decouple the quadratic and the linear part of the problem by splitting D:

$$D_1 = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n y_n & y_n^2 \end{bmatrix}, \quad \text{and} \quad D_2 = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}, \tag{3.24}$$

Consequently, the matrix S can also be considered composed of three submatrices

$$S = \begin{bmatrix} S_1 & S_2 \\ S_2^{\mathrm{T}} & S_3 \end{bmatrix}, \quad \text{where} \quad \begin{cases} S_1 &= D_1^{\mathrm{T}} D_1 \\ S_2 &= D_1^{\mathrm{T}} D_2 \\ S_3 &= D_2^{\mathrm{T}} D_2 \end{cases} \tag{3.25}$$

Finally, the non-zero part of C can be isolated

$$C_1 = \begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \tag{3.26}$$

and the parameter vector can be split in the two parts $\boldsymbol{A}_1 = [a, b, c]^{\mathrm{T}}$ and $\boldsymbol{A}_2 = [d, e, f]^{\mathrm{T}}$. With these modifications, Eq. (3.23) is equivalent to the system of equations

$$S_1 \boldsymbol{A}_1 + S_2 \boldsymbol{A}_2 = \lambda C_1 \boldsymbol{A}_1 \tag{3.27}$$

$$S_2^{\mathrm{T}} \boldsymbol{A}_1 + S_3 \boldsymbol{A}_2 = 0 \tag{3.28}$$

Since S_3 is always invertible unless all the points lie on a line [16], in which case fitting an ellipse is impossible, it is possible to find $\boldsymbol{A}_2 = -S_3^{-1} S_2^{\mathrm{T}} \boldsymbol{A}_1$ and substitute in Eq. (3.27) to obtain the reduced optimization problem

$$C_1^{-1}(S_1 - S_2 S_3^{\mathrm{T}} S_2)\boldsymbol{A}_1 = M \boldsymbol{A}_1 = \lambda \boldsymbol{A}_1 \quad \text{subject to} \quad \boldsymbol{A}_1^{\mathrm{T}} C_1 \boldsymbol{A}_1 = 1. \tag{3.29}$$

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

III.B.5. Algebraic Fit Results

To test the accuracy of this method, a series of runs have been performed with artificially generated data sets superimposed with Gaussian noise. The number of points is taken equal to 100, the noise equal to 1 pixel (1σ), and the target ellipse is considered having center aligned with the optical axis and inclination of 20 degrees with respect to the horizontal axis. A total of 1000 tests are performed. The results are depicted in Figure III.3. Aside from the fact that the accuracy of this method is comparable with those of the geometric method described above (see Figure III.1), what is important to remark is that the inclination again stands out for the much higher lever of error. Truly, the small values of eccentricities expected in this application make the inclination of the ellipse almost unobservable. It is ultimately necessary to understand that in many cases it is not possible to discern the difference in axes for Earth, and it must be therefore treated as a circle.

III.C. Sigmoid Function Least Square

To further improve the results, it is possible to apply another least square optimization. In this case, a class of functions, called *Sigmoid Functions* (SFs) is used to estimate with greater precision the location of the target edge within the image. Sigmoid functions are distinguished by the geometric shape on which they are modelled. Thus there exist Linear, Circular and Elliptical SFs. Similarly, in three dimensions

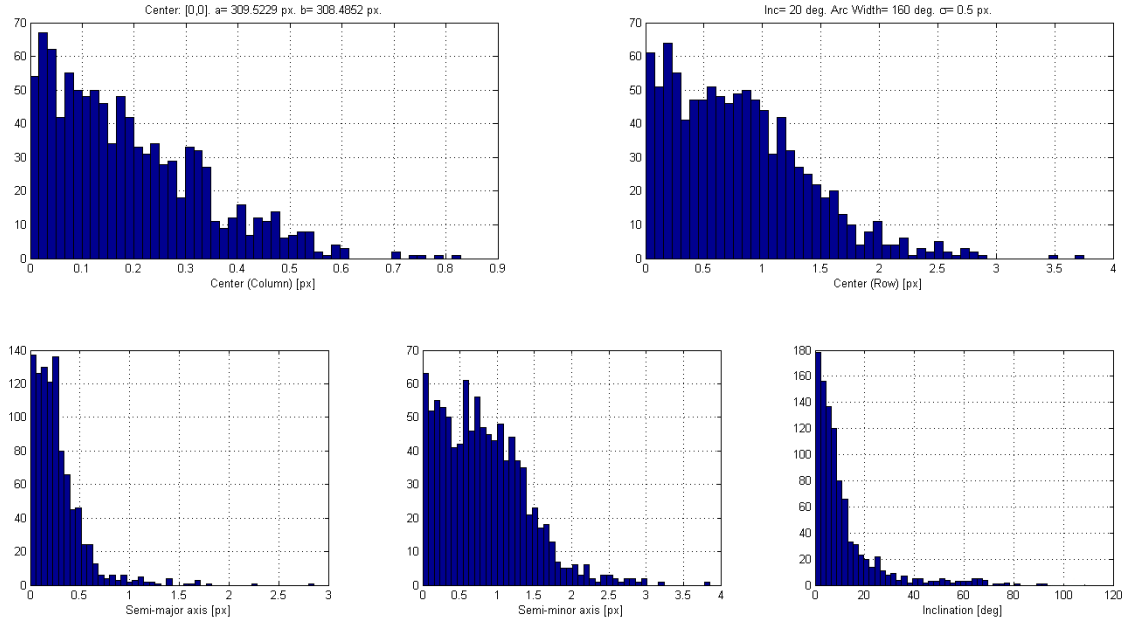


Figure III.3: Statistical analysis of the Fitzgibbon fit.

one has Spherical and Ellipsoidal SFs. These functions are introduced in this section along with the partial needed to use them for least-squares estimation.

III.C.1. Linear Sigmoid Function (LSF)

A Linear Sigmoid Function is described by

$$f = y_{\max} + \frac{y_{\min} - y_{\max}}{1 + e^{k(x_t - x)}} \tag{3.30}$$

where $y_{\max}$ and $y_{\min}$ are the initial and final levels of the sigmoid functions, x_t the the distance from the origin where the step transition occurs, and k is the *sigmoid function constant*, indicating how rapid the step transition is. The higher the value of k the shorted the step transition is. Using $x_t = 50$ and for $y_{\max} = 150$ and $y_{\min} = 20$, four examples for various values of $k = [0.1, 0.3, 0.9, 2.7]$ are shown in Figure III.4

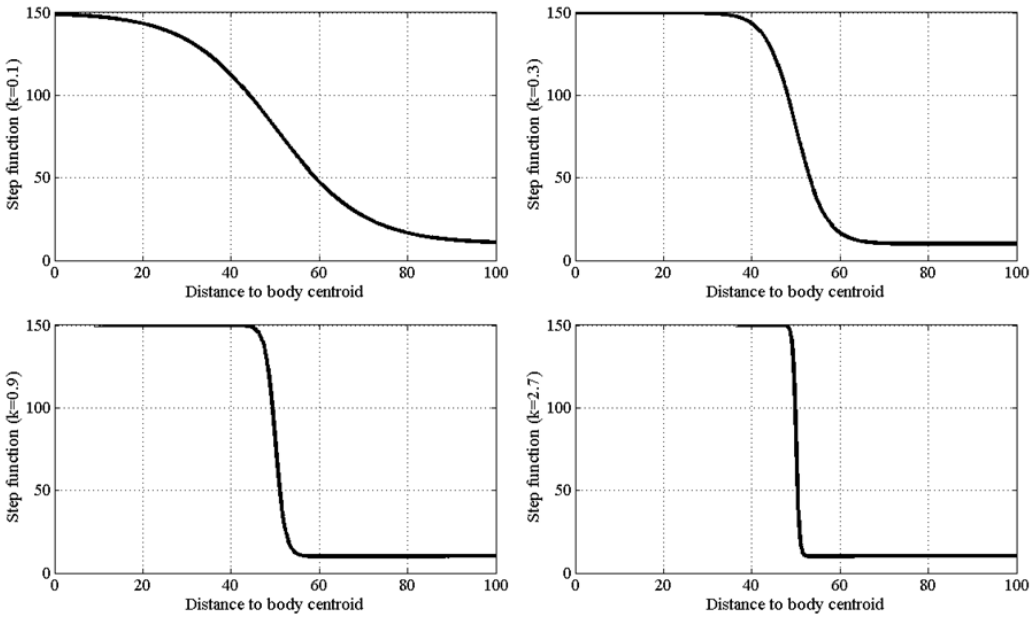


Figure III.4: Examples of Linear Sigmoid Functions.

Least-squares with a LSF requires building a Jacobian. In the general case the Jacobian will be an $n \times 4$ matrix associated to the four unknowns, x_t, $y_{\max}$, $y_{\min}$, and k. The Jacobian can be of smaller dimension when some of these variable can be assumed to be known (e.g., k, or the transition levels $y_{\max}$ and $y_{\min}$). Setting

$$\alpha_i = e^{k(x_t - x_i)} \tag{3.31}$$

the Jacobian has the form

$$J = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_t} & \dfrac{\partial f_1}{\partial y_{\max}} & \dfrac{\partial f_1}{\partial y_{\min}} & \dfrac{\partial f_1}{\partial k} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial f_n}{\partial x_t} & \dfrac{\partial f_n}{\partial y_{\max}} & \dfrac{\partial f_n}{\partial y_{\min}} & \dfrac{\partial f_n}{\partial k} \end{bmatrix} \tag{3.32}$$

where

$$\begin{aligned}
\frac{\partial f_i}{\partial y_{\min}} &= \frac{1}{1+\alpha_i}, \\
\frac{\partial f_i}{\partial y_{\max}} &= \frac{\alpha_i}{1+\alpha_i} = \alpha_i \frac{\partial f_i}{\partial y_{\min}}, \\
\frac{\partial f_i}{\partial x_t} &= -\frac{(y_{\min}-y_{\max})\alpha_i\, k}{(1+\alpha_i)^2} = -\frac{\partial f_i}{\partial y_{\min}} \frac{\partial f_i}{\partial y_{\max}} (y_{\min}-y_{\max})\, k, \quad \text{and} \\
\frac{\partial f_i}{\partial k} &= -\frac{(y_{\min}-y_{\max})\alpha_i}{(1+\alpha_i)^2} (x_t - x_i) = \frac{(x_t-x_i)}{k} \frac{\partial f_i}{\partial x_t}.
\end{aligned}$$

The iterative least squares approach is the described by

$$\begin{Bmatrix} \Delta x_t \\ \Delta y_{\max} \\ \Delta y_{\min} \\ \Delta k \end{Bmatrix} = (J^{\mathrm{T}} J)^{-1} J^{\mathrm{T}} \begin{Bmatrix} f_1 - f(x_1) \\ \vdots \\ f_n - f(x_n) \end{Bmatrix} \tag{3.33}$$

Sometime, a good estimation of $y_{\max}$ and $y_{\min}$ is available by statistics taken from the head and the tail of the data. In this case, the second and third rows of the Jacobian are omitted. In some other cases k can also be assigned. For these cases, which are associated to some kind of knowledge about the step transition, the Jacobian becomes an n vector.

In order to successfully perform least-squares with sigmoid functions, the following considerations must be well understood. Particular attention must be given to the range of the x_i vector. In fact, when the difference, $x_t - x_i$, is given in pixels then the values of α_i for pixels far from x_t may become too big. When this happens, $\alpha_i \to \infty$ and, consequently, the terms $\dfrac{\partial f_i}{\partial y_{\max}}$ become undefined. To avoid this problem a particular attention must be given to the range size around the transition. This must be the smallest possible. Scaling the x_i is not a solution as it will modify

the product $k(x_t - x_i)$, which is equivalent of using a smaller value of k.

III.C.2. Circular Sigmoid Function (CSF)

Equation (3.30) can be extended to two-dimensional space with radial distribution

$$f = y_{\max} + \frac{y_{\min} - y_{\max}}{1 + e^{k(r_e - d)}} \tag{3.34}$$

where $d = \sqrt{(c_0 - c)^2 + (r_0 - r)^2}$ is the square distance (here provided in pixel) with coordinates $[r, c]$ from the origin $[r_0, c_0]$ and r_e is the radial step transition. Setting

$$\beta_i = \sqrt{(c_0 - c_i)^2 + (r_0 - r_i)^2} \quad \text{and} \quad \alpha_i = e^{k(r_e - \beta_i)}, \tag{3.35}$$

the Jacobian requires the computation of the following derivatives

$$\begin{cases} \dfrac{\partial f_i}{\partial r_0} = \dfrac{(y_{\min} - y_{\max})\,\alpha_i}{(1+\alpha_i)^2}\, k\, \dfrac{r_0 - r_i}{\beta_i} \\ \dfrac{\partial f_i}{\partial c_0} = \dfrac{(y_{\min} - y_{\max})\,\alpha_i}{(1+\alpha_i)^2}\, k\, \dfrac{c_0 - c_i}{\beta_i} \\ \dfrac{\partial f_i}{\partial r_e} = -\dfrac{(y_{\min} - y_{\max})\,\alpha_i}{(1+\alpha_i)^2}\, k \\ \dfrac{\partial f_i}{\partial k} = -\dfrac{(y_{\min} - y_{\max})\,\alpha_i}{(1+\alpha_i)^2}\, (r_e - \beta_i) \\ \dfrac{\partial f_i}{\partial y_{\max}} = \dfrac{\alpha_i}{1+\alpha_i} \\ \dfrac{\partial f_i}{\partial y_{\min}} = \dfrac{1}{1+\alpha_i} \end{cases} \tag{3.36}$$

CSFs are used for high accurate estimation of Moon center and radius. Spice provides the semi-axes of the Moon ellipsoid as all equals, $a = b = c = 1737.0$ km[1]. Dealing with a spherical body greatly simplifies the data processing complexity as no body center offset and observed ellipse orientation must be considered.

[1]Reference [44] provides the following values for the Moon semi-axes: $a = 1737.8414$ km (Equatorial semi-major axis), $b = 1737.59$ km (Equatorial semi-minor axis), and $c = 1737.17$ km (Polar axis)

III.C.3. Elliptical Sigmoid Function (ESF)

An Elliptical Sigmoid Function is associated with an ellipse with observed semi-axes a and b along the α and β axes, respectively. These observed semi-axes are actually a' and b' and they are computed using Eq. (2.25). The equation of an ellipse with respect to its own axes is

$$b^2\,\alpha^2 + a^2\,\beta^2 = a^2\,b^2 \tag{3.37}$$

Setting $\rho = b/a$, Eq. (3.37) becomes a function of a only

$$\rho^2\,\alpha^2 + \beta^2 = \rho^2\,a^2 \tag{3.38}$$

An ESF is described by

$$f = y_{\max} + \frac{y_{\min} - y_{\max}}{1 + e^{k(\rho\,a - d)}} \tag{3.39}$$

where $d = \sqrt{\rho^2\,\alpha^2 + \beta^2}$, and the coordinate transformations between $[\alpha, \beta]$ (in pixel) and $[r, c]$ (row and column, pixel) are provided in Eq. (A.5) and (A.6), respectively. The ellipse has center $[r_0, c_0]$ and the ellipse oriented angle ϑ (see Figure A.2) can be computed using by the relationship $C\,\hat{\boldsymbol{w}} = \hat{\boldsymbol{v}}$, given in Eq. (2.22), specified for the greatest semi-major axis of the observed ellipse, $\hat{\boldsymbol{w}}_a = \{a',\, 0,\, \sqrt{1 - a'^2}\}^{\mathrm{T}}$.

Setting

$$\beta_i = \sqrt{\rho^2 x_i^2 + y_i^2} \quad \text{and} \quad \alpha_i = e^{k(\rho a - \beta_i)} \tag{3.40}$$

where the $[x_i, y_i]$ from/to $[r_i, c_i]$ conversions are given in Eq. (A.3) and Eq. (A.4).

The Jacobian requires the computation of the following derivatives

$$\begin{cases} \dfrac{\partial f_i}{\partial r_0} = \dfrac{(y_{\min} - y_{\max})\,\alpha_i}{(1+\alpha_i)^2}\, k\, \dfrac{\rho^2 x_i \sin\vartheta + y_i \cos\vartheta}{\beta_i} \\ \dfrac{\partial f_i}{\partial c_0} = \dfrac{(y_{\min} - y_{\max})\,\alpha_i}{(1+\alpha_i)^2}\, k\, \dfrac{y_i \sin\vartheta - \rho^2 x_i \cos\vartheta}{\beta_i} \\ \dfrac{\partial f_i}{\partial a} = -\dfrac{(y_{\min} - y_{\max})\,\alpha_i}{(1+\alpha_i)^2}\, k\,\rho \\ \dfrac{\partial f_i}{\partial \vartheta} = \dfrac{(y_{\min} - y_{\max})\,\alpha_i}{(1+\alpha_i)^2}\, k\, \dfrac{x_i y_i (\rho^2 - 1)}{\beta_i} \\ \dfrac{\partial f_i}{\partial y_{\max}} = \dfrac{\alpha_i}{1+\alpha_i} \\ \dfrac{\partial f_i}{\partial y_{\min}} = \dfrac{1}{1+\alpha_i} \\ \dfrac{\partial f_i}{\partial k} = -\dfrac{(y_{\min} - y_{\max})\alpha_i}{(1+\alpha_i)^2}\,(\rho a - \beta_i) \end{cases} \tag{3.41}$$

CHAPTER IV

IMAGE PROCESSING EXAMPLES

To better illustrate the methods discussed so far, four examples of image processing will be discussed. These try to cover a wide range of possible conditions for the target.

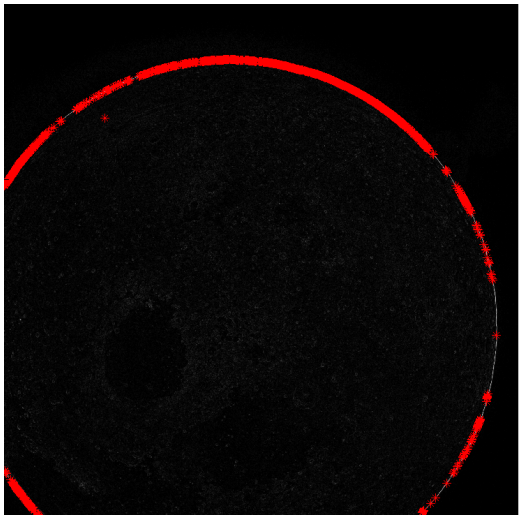
All these tests have been performed with no information about time, camera data, and observer, Moon, and Sun positions. The image derivative is obtained using the 4-point central approach with no Richardson extrapolation. These information clearly help to define the best values of the internal parameters.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

1. Box-based outliers elimination: box size = 21×21;
2. Box-based outliers elimination: derivative ratio > 0.6;
3. Box-based outliers elimination: corner ratio < 0.3;
4. Box-based outliers elimination: eigenvalue ratio < 0.3;
5. RANSAC: minimum distance $d_{\min} = 3$ pixels;
6. Max number of iteration for CSF-LS: 50;
7. Tolerance for CSF-LS: 0.3/50 pixels.

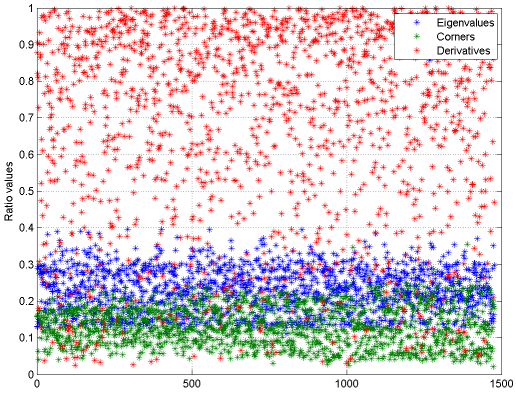
IV.A. Example #1: Real Moon Image, Two-times Cropped

The first example shows the capability of the code to deal with situations in which the target is cropped by the frame. In this case, the target is cropped twice, out of a maximum possible of 4 times. The original image is depicted in Figure IV.1. The pixel selection process is illustrated in Figure IV.2. For this image, an overall 323 points have been found to be outliers and removed. Regarding the best fitting process (Figure IV.3), applying the Sigmoid least square has changed the best fitting by less than 2 pixels. The convergence is reported in Table IV.1.

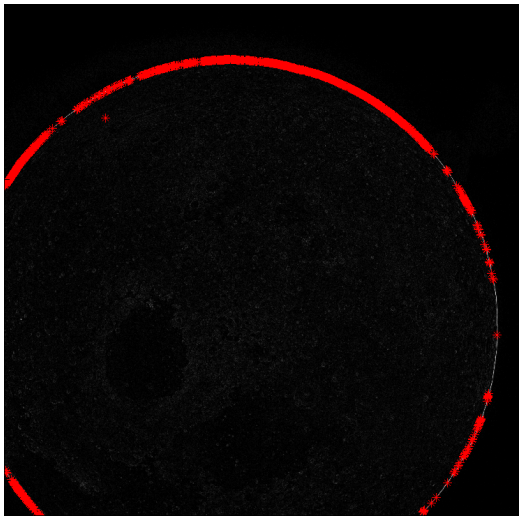
Figure IV.1: Example #1: Original image.



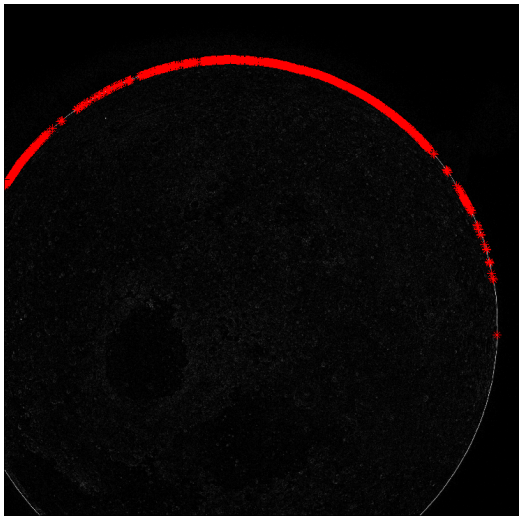
(a) 1497 **derivatives selected.**



(b) **Box-based Test.**



(c) **Box approach:** 179 **removals.**

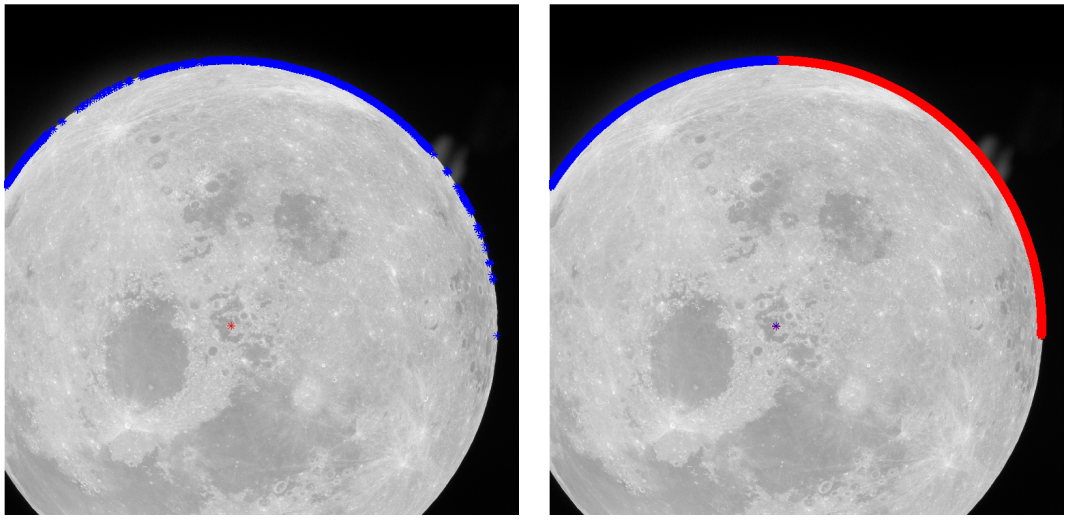


(d) **RANSAC:** 144 **removals.**

Figure IV.2: Pixel selection process for example#1.

Table IV.1: CSF-LS convergence process for example #1.

Iter #	r	y_c	x_c
0	1208.0027	1463.1014	1031.6035
1	1208.5013	1463.7112	1031.7029
2	1208.8818	1464.1634	1031.7398
3	1209.1436	1464.4729	1031.7559
4	1209.2979	1464.6546	1031.7643
5	1209.3768	1464.7473	1031.769
6	1209.4148	1464.7916	1031.7715
7	1209.4325	1464.8124	1031.7728
8	1209.4408	1464.8219	1031.7734
9	1209.4445	1464.8264	1031.7737



(a) Taubin Best Fit. $x_c = 1031.6$, $y_c = 1463.1$, $r = 1208.0$ **pixels.**

(b) CSF-LS. $\Delta x_c = -0.170$, $\Delta y_c = -1.725$, $\Delta r = -1.442$ **pixels.**

Figure IV.3: Best fitting process for example#1.

IV.B. Example #2: Barely Visible Moon

The second example considers a situation in which the target (specifically the Moon) is barely illuminated, and therefore the algorithm has to deal with a relatively thin target (see Figure IV.4). For this case, while a total of 306 points are removed before the best fitting process, none is found via RANSAC test (see Figure IV.5d). This is because most of the brightness of the image is concentrated in a very thin arc. There are therefore no points not able to satisfy the requirements of RANSAC. For the best fitting procedure, it is interesting to notice that the correction along the y-axis is over one order of magnitude greater than along the x-axis (see Figure IV.6) . This is due to the fact that the point distribution is completely unbalanced in the horizontal direction. Thus, Sigmoid least square has to compensate for a bigger error. Also, the convergence process requires 11 iterations to satisfy the tolerance

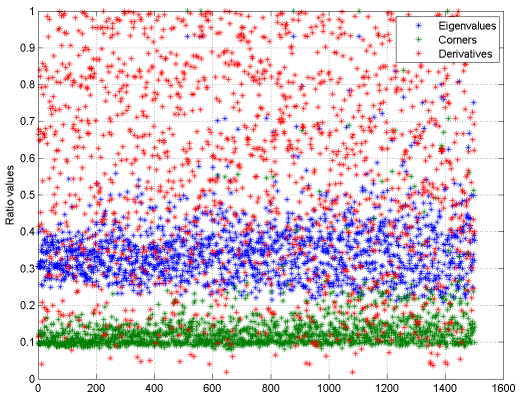
(Table IV.2).

Figure IV.4: Example #2: Original image.

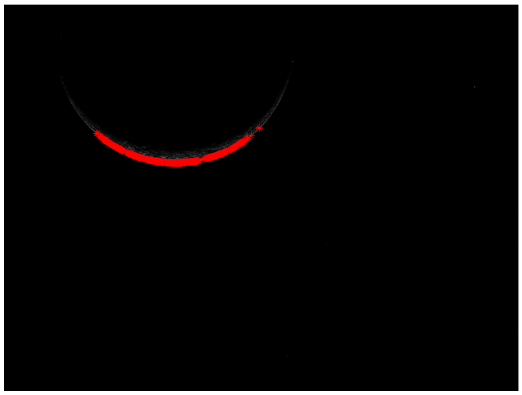
IV.C. Example #3: Cropped Earth

In Figure IV.7, a cropped view of the Earth is depicted. While the Earth is supposed to project an ellipse on the imager, as described in the previous chapters this information is lost due to the fact that expected eccentricity is so small that is inevitably masked by the noise. Therefore, here the target is assumed to be once more circular. Compared to images of the Moon, there is a net increase in outliers (see Figure IV.8). Indeed, between box-based and RANSAC removal methods, a total of 1222 points is removed. On the other end, the remaining pixels are sufficient to estimate the parameters with great precision, which is barely improved by the Sigmoid least squares (Figure IV.9). This is best illustrated by the fact that only 4 iterations are required to reach convergence, as reported in Table IV.3.

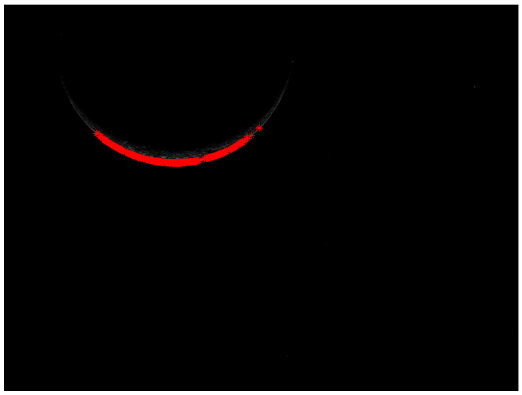
(a) 1501 **derivatives selected.**



(b) **Box-based Test.**

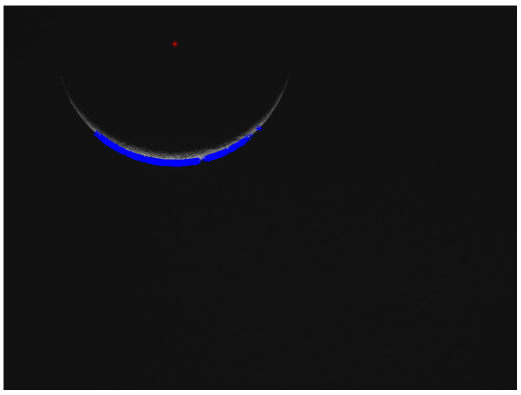


(c) **Box approach:** 306 **removals.**

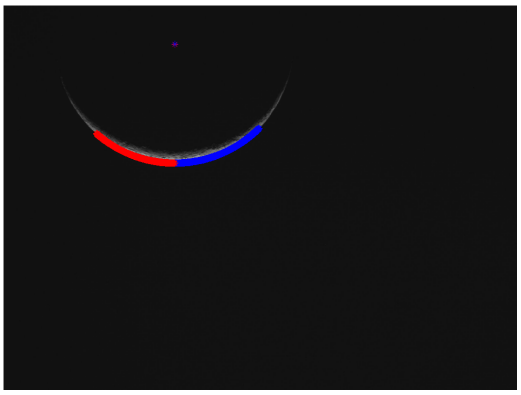


(d) **RANSAC:** 0 **removals.**

Figure IV.5: Pixel selection process for example #2.



(a) **Taubin Best Fit.** $x_c = 864.158$, $y_c = 192.760$, $r = 600.726$ **pixels.**



(b) **CSF-LS.** $\Delta x_c = -0.126$, $\Delta y_c = -4.704$, $\Delta r = 4.370$ **pixels.**

Figure IV.6: Best fitting process for example #2.

Table IV.2: CSF-LS convergence process for example #2.

Iter #	r	y_c	x_c
0	600.7265	192.7597	864.1576
1	598.3991	195.2876	864.2523
2	597.3463	196.4184	864.284
3	596.8473	196.9489	864.291
4	596.6038	197.2055	864.2907
5	596.4822	197.3328	864.2888
6	596.4204	197.3972	864.287
7	596.3885	197.4303	864.2857
8	596.372	197.4474	864.2848
9	596.3633	197.4564	864.2843
10	596.3587	197.4611	864.2839
11	596.3563	197.4636	864.2837

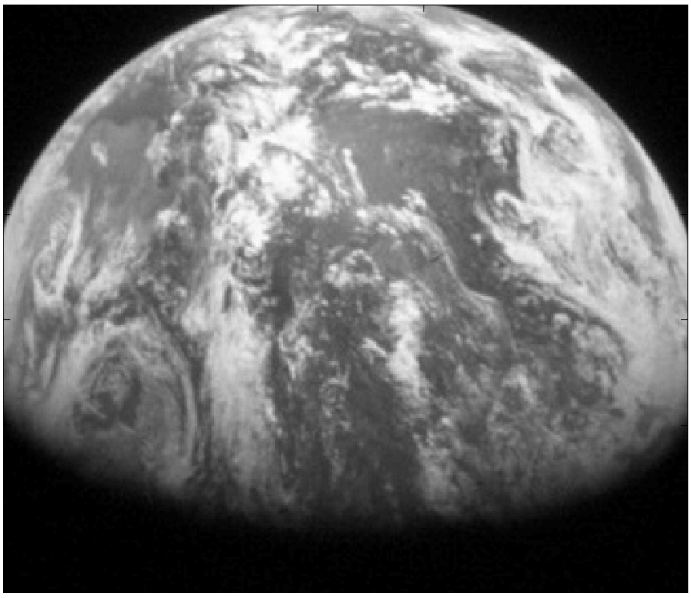
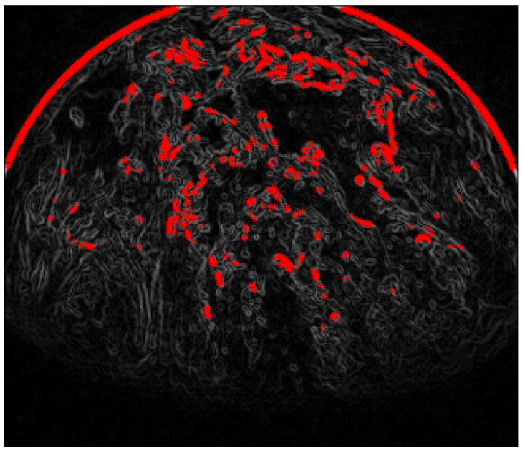


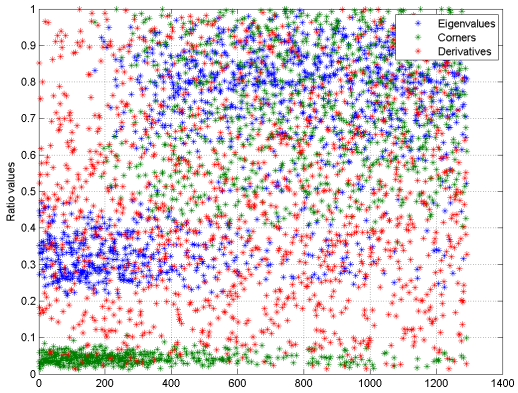
Figure IV.7: Example #3: Original image.

Table IV.3: CSF-LS convergence process for example #3.

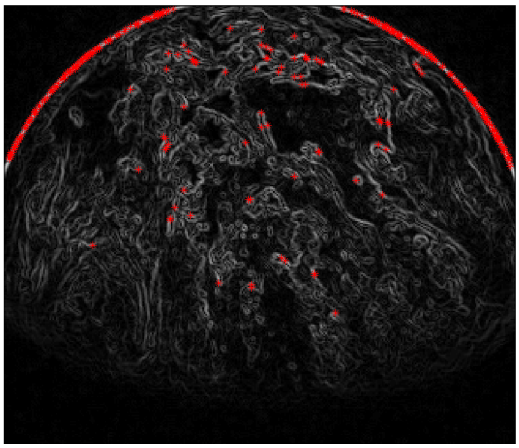
Iter #	r	y_c	x_c
0	172.6639	166.9863	162.2505
1	172.6059	166.9053	162.3508
2	172.5849	166.8783	162.3717
3	172.5788	166.8707	162.3762
4	172.5772	166.8688	162.3772



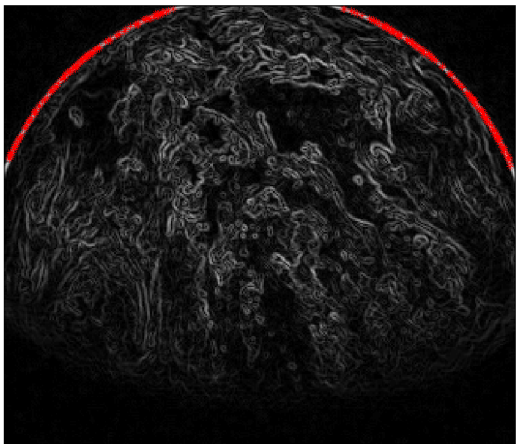
(a) 1464 **derivatives selected.**



(b) **Box-based Test.**

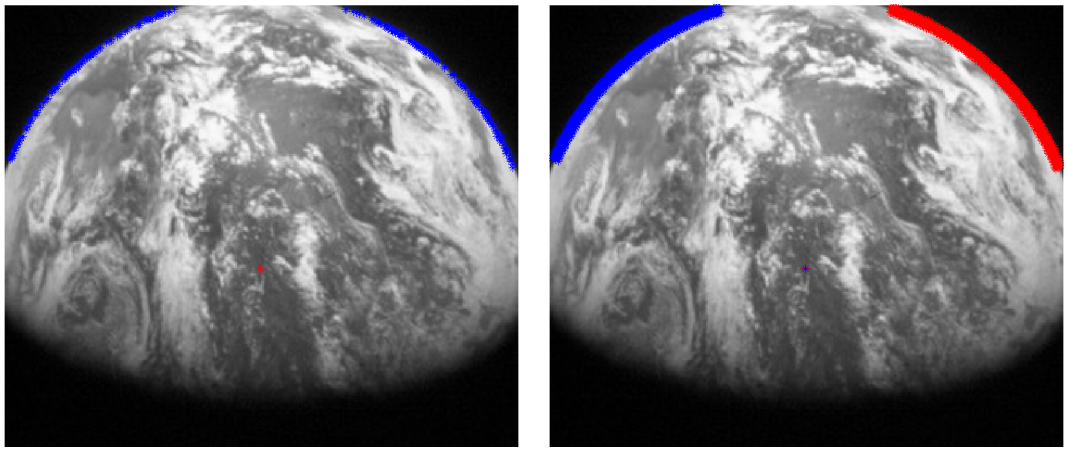


(c) **Box approach:** 1146 **removals.**



(d) **RANSAC:** 76 **removals.**

Figure IV.8: Pixel selection process for example #3.



(a) Taubin Best Fit. $x_c = 162.250$, $y_c = 166.986$, $r = 172.664$ **pixels.**

(b) CSF-LS. $\Delta x_c = -0.126$, $\Delta y_c = -0.117$, $\Delta r = 0.087$ **pixels.**

Figure IV.9: Best fitting process for example #3.

IV.D. Example #4: Synthetic Crescent Moon

Because the selection of images available for testing was limited, this has been augmented with a series of artificially generated images, obtained with the NASA visualization tool EDGE. The issue with this system is that the final images have an unnatural "crispness", due to being completely artificially created. This creates problems to the pixels selection algorithm, and also is not representative of real images. To resolve this issue, a simple convolution filter is applied in preprocessing, of the form $F = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.

Except for this detail, the process is analogous to previous cases. The original image is depicted in Figure IV.10. The derivative selection process (see Figure IV.11) removes a total of 736 pixels, most of which via box-based removal. Figure IV.12 shows the results of Taubin Best Fit for this example, and the convergence of the CSF algorithm is described in Table IV.4.

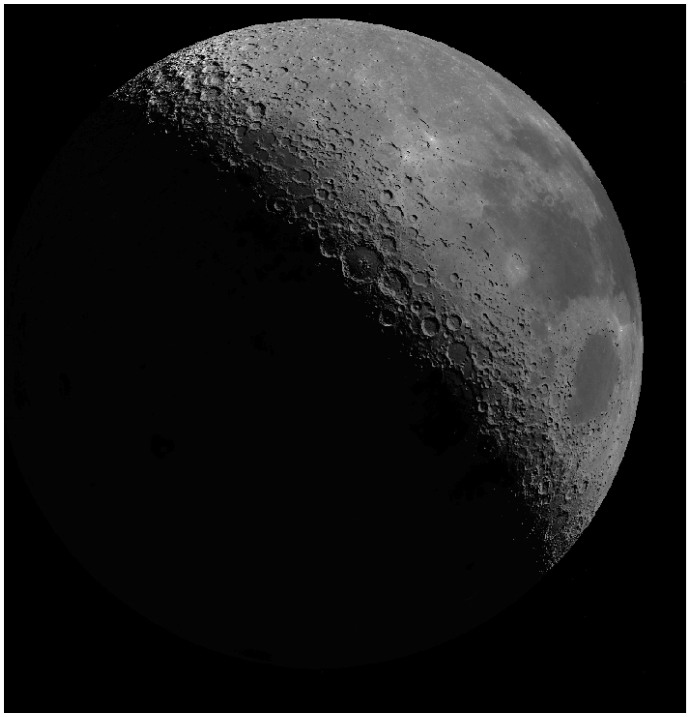
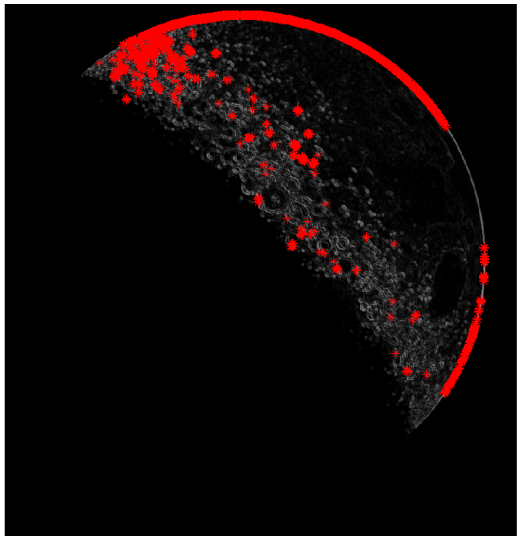


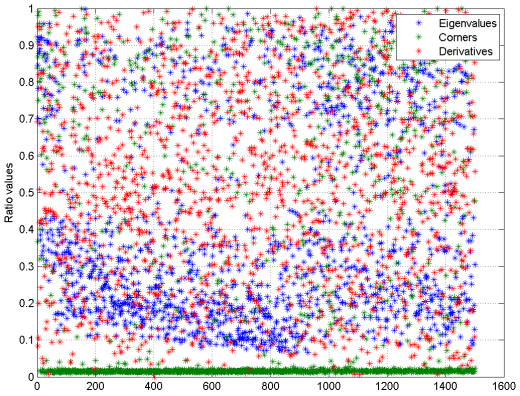
Figure IV.10: Example #4: Original image.

Table IV.4: CSF-LS convergence process for example #4.

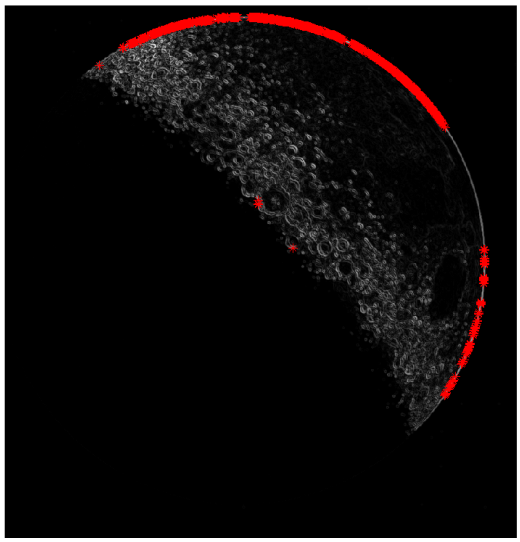
Iter #	r	y_c	x_c
0	390.6188	409.1497	380.7349
1	390.744	409.155	380.4779
2	390.7965	409.1737	380.3782
3	390.8168	409.1842	380.341
4	390.8245	409.1889	380.3272
5	390.8274	409.1909	380.3221
6	390.8285	409.1916	380.3202



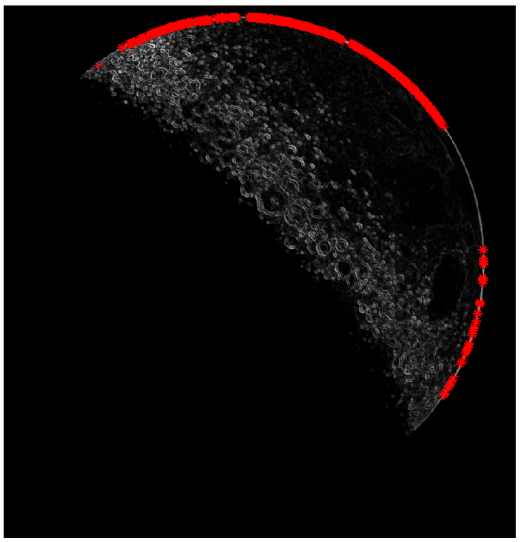
(a) 1501 **derivatives selected.**



(b) **Box-based Test.**

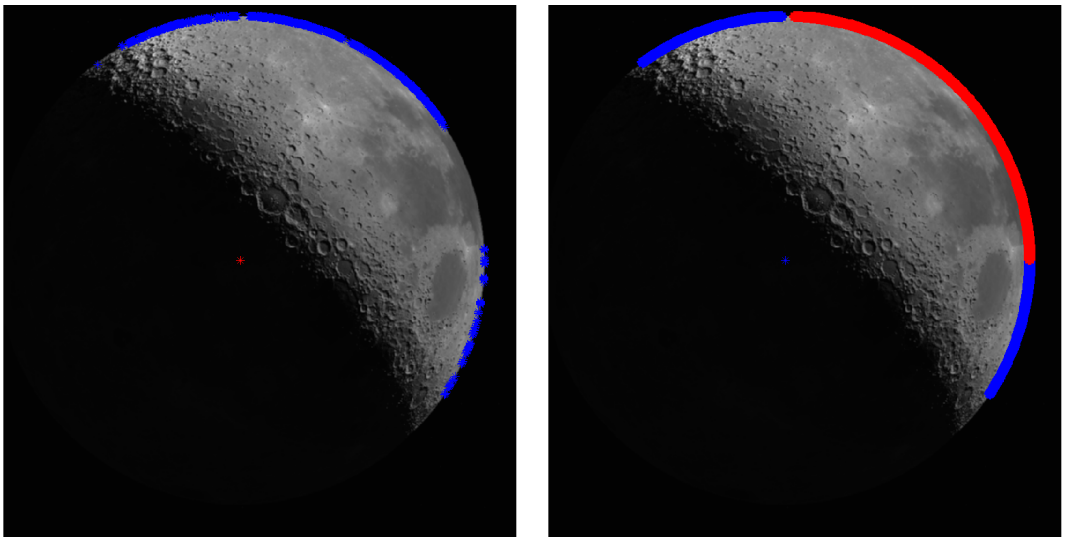


(c) **Box approach:** 732 **removals.**



(d) **RANSAC:** 4 **removals.**

Figure IV.11: Pixel selection process for example #4.



(a) Taubin Best Fit. $x_c = 380.735$, $y_c = 409.150$, $r = 390.619$ **pixels.**

(b) CSF-LS. $\Delta x_c = 0.415$, $\Delta y_c = -0.042$, $\Delta r = -0.210$ **pixels.**

Figure IV.12: Best fitting process for example #4.

CHAPTER V

BÉZIER CURVES

Exact orbit determination is completely solvable only for the Two Body Problem and a few restricted cases involving three bodies. The search for solutions in more general cases has lead to many important advancements in mathematics in the last 400 years [27]. Many approaches have been used, by formulating the problem as an Initial Value Problem, or a Two Points Boundary Value Problem [1] [17]. In other cases, especially since orbit determination has become necessary for satellites and not just for planets, statistical or filter-based techniques have been developed [38] [6]. In general however, the solution is specifically tailored for the application [43] [22] [45].

Given the work previously done to extract position information from the images, it is sensible to use an estimation technique that dispenses completely of any type of physical models, and instead relies completely on the measurements. In particular Bézier polynomials are used, because they are extremely versatile, while at the same time completely defined by a limited number of parameters. They also allow for a compact matrix representation [30], that allows for a very efficient least square best fitting technique to be implemented. A Bézier function is a parametric curve widely used in computer graphics and many other design related applications. While the mathematical foundations were established in 1912 by Sergei Bernstein, its applicability to design was understood much later, between 1959 and 1962, with the work of Pierre Bézier and Paul de Casteljau, who used them to design automobile hulls at

Renault and Citroën, respectively. Mathematically, a Bézier curve in 3D is defined as a polynomial function of a parameter s and a set of control points, organized in a $3 \times n$ matrix $\mathbb{C}$, where n is the curve degree. While $\mathbb{C}$ is the ultimate target of the estimation, as any Bézier curve is completely described by its control points, the values of the parameter s corresponding to the chosen measurements have to be found as an intermediate step. This leads to an iterative process involving two different but connected least square problems, one to find the control points and the other to find the parameter values. The solution is considered achieved once a certain measure of the error, defined as the difference between estimates and measurements, is below a chosen tolerance. Then, a second Bézier curve is used to approximate the behaviour of time along the trajectory. Computation of this second curve is not subject to an iterative process, because it uses the set of parameter values previously found. Further details are given in the following sections.

V.A. Bézier Least Square

The results in this section come from work first described in [7], which in turn was originally based on [30]. A Non-Rational Bézier function can be written as linear combination of Bernstein polynomials,

$$\boldsymbol{r} = \sum_{k=0}^{n} \boldsymbol{c}_k B_k^n(s) \quad \text{where} \quad B_k^n(s) = \binom{n}{k} s^k (1-s)^{n-k}, \;\; s \in [0,1]. \tag{5.1}$$

and by using the binomial theorem, a generic Bernstein polynomial can be rewritten as

$$B_k^n(s) = \binom{n}{k} s^k \left[\sum_{j=0}^{n-k} \binom{n-k}{j} (-s)^{n-(k+j)} \right] = \binom{n}{k} \left[\sum_{j=0}^{n-k} \binom{n-k}{j} (-1)^{n-(k+j)} s^{n-j} \right]. \tag{5.2}$$

Therefore, any Non-Rational Bézier function has an expression of the form

$$\boldsymbol{r} = \sum_{k=0}^{n} \left[\boldsymbol{c}_k \binom{n}{k} \left[\sum_{j=0}^{n-k} \binom{n-k}{j} (-1)^{n-(k+j)} s^{n-j} \right] \right]. \tag{5.3}$$

While Eq. (5.3) has a rather cumbersome form, when m points along a curve and the corresponding values of the parameter s are chosen, it can be rewritten in a more compact form using linear algebra

$$\mathbb{R} = \mathbb{C}\,\mathbb{M}\,\mathbb{S} \tag{5.4}$$

where

$$\begin{cases} \mathbb{R} = [\, \boldsymbol{r}_1,\ \boldsymbol{r}_2,\ \ldots,\ \boldsymbol{r}_m \,] \\ \\ \mathbb{C} = [\, \boldsymbol{c}_0,\ \boldsymbol{c}_1,\ \ldots,\ \boldsymbol{c}_n \,] \end{cases} \quad \text{and} \quad \mathbb{S} = \begin{bmatrix} s_1^n & s_2^n & \cdots & s_{m-1}^n & s_m^n \\ s_1^{n-1} & s_2^{n-1} & \cdots & s_{m-1}^{n-1} & s_m^{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_1 & s_2 & \cdots & s_{m-1} & s_m \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \tag{5.5}$$

and the elements of matrix $\mathbb{M}$ are of the form $\mathbb{M}(k,j) = \binom{n}{k}\binom{n-k}{j}(-1)^{n-(k+j)}$, $j = 1, \ldots, n-1$. It follows from this formula that $\mathbb{M}(k,j) = \mathbb{M}(j,k)$, i.e. $\mathbb{M}$ is a symmetric matrix, and also $\mathbb{M}(k,j) = 0$ whenever $k + j > n + 1$. To reiterate, R is a matrix of measurements, C is the control points matrix, M is a coefficient matrix completely determined by the degree n of the polynomial chosen, and S contains all the $n+1$ powers of the parameter values s_k corresponding to the m measurements. If the degree of the curve and the values of the parameter are known, Eq. (5.3)allows to estimate the control points via least squares. Indeed,

$$\mathbb{C} = \mathbb{R}\,\mathbb{S}^{\mathrm{T}}\,(\mathbb{S}\mathbb{S}^{\mathrm{T}})^{-1}\,\mathbb{M}^{-1} \tag{5.6}$$

The crucial part is therefore to determine n and the distribution of parameter $\{s_k\}$.

V.A.1. Degree of the Bézier Curve

As is to be expected with polynomials, Bézier functions of different degree provide different levels of approximation. Since all but the simplest trajectories have variable curvature, and small values of curvature are often masked by noise, it is not possible to define *a priori* the optimal degree to use in a specific application. As a rule of thumb, polynomials of higher degree will provide better approximations, up to a point, because of the greater number of degree of freedoms (the number of control points of a polynomial of degree n is $n+1$). On the other hand, the maximum degree usable for a given interpolation is limited by the number of data points available, namely:

$$n = \left\lceil \frac{m}{d} - 1 \right\rceil . \tag{5.7}$$

where the number at denominator represents the number of dimensions (3 in the present case) As n increases, matrix S becomes progressively ill-conditioned, becoming unstable for $n \simeq 10$. This happens because the first columns are n-th powers of numbers less than 1, and therefore quickly become infinitesimal, while the last column is constant and composed of 1's. Therefore, the practical solution implemented in the algorithm to find the "optimal" degree consists by starting the analysis with a polynomial of degree 2 and then increasing this value until a local minimum in the average position error has been found, or the degree reaches either the "hard" limit enforced by Eq. (5.7) or the "soft" limit due to the numerical problems discussed above (whichever is smaller). It has been found experimentally that in most cases the optimal degree will be close or equal to the maximum defined by Eq. (5.7).

V.A.2. Distribution of the Parameter

To solve Eq. (5.4), a distribution of the parameter s corresponding to the measurements has to be assumed. If the data is provided at constant time step, a uniform distribution can be used:

$$s_k = \frac{k-1}{n-1} \qquad \text{where} \qquad k \in [1, m]. \tag{5.8}$$

Another valid approximation relates the distribution with the relative distance of subsequent data points, as per the following:

$$\begin{cases} s_1 = 0 \\ s_m = 1 \end{cases} \qquad \text{and} \qquad s_k = \frac{\sum_{j=2}^{k} |\tilde{\boldsymbol{r}}_j - \tilde{\boldsymbol{r}}_{j-1}|}{\sum_{j=2}^{m} |\tilde{\boldsymbol{r}}_j - \tilde{\boldsymbol{r}}_{j-1}|} \quad \text{where} \quad k \in [2, m-1]. \tag{5.9}$$

where the $\tilde{\boldsymbol{r}}_j$ are the position measurements. In general, solving Eq. (5.4) with either distribution leads to estimates displaced from the actual measurements, by a value of:

$$d_k = |CM\boldsymbol{s}_k - \tilde{\boldsymbol{r}}_k| \quad k \in [1, m]. \tag{5.10}$$

where $\boldsymbol{s}_k$ is a vector of all the powers of s_k. Obviously, changing a certain s_k will only affect the value of the corresponding d_k. Thus, it is possible to find a new set of parameters by minimizing a set of aptly chosen cost functions $\{L_k\}$:

$$L_k = (CM\boldsymbol{s}_k - \tilde{\boldsymbol{r}}_k)^{\mathrm{T}}(CM\boldsymbol{s}_k - \tilde{\boldsymbol{r}}_k) \quad k \in [2, m-1]. \tag{5.11}$$

By expanding Eq. (5.11) becomes

$$L_k = \boldsymbol{s}_k^{\mathrm{T}} M C^{\mathrm{T}} C M \boldsymbol{s}_k - 2\boldsymbol{r}_k^{\mathrm{T}} C\, M \boldsymbol{s}_k + \boldsymbol{r}_k^{\mathrm{T}} \boldsymbol{r}_k. \tag{5.12}$$

Applying the necessary condition for a stationary point leads to a polynomial in s_k:

$$\frac{dL_k}{ds_k} = \frac{dL_k}{d\boldsymbol{s}_k} \cdot \frac{d\boldsymbol{s}_k}{ds_k} = 2\,\boldsymbol{s}_k^{\mathrm{T}} M C^{\mathrm{T}} C M \frac{d\boldsymbol{s}_k}{ds_k} - 2\boldsymbol{r}_k^{\mathrm{T}} C\, M \frac{d\boldsymbol{s}_k}{ds_k} = 0 \tag{5.13}$$

or equivalently

$$F(s_k) = \boldsymbol{s}_k^{\mathrm{T}} M C^{\mathrm{T}} C M \frac{d\boldsymbol{s}_k}{ds_k} - \boldsymbol{r}_k^{\mathrm{T}} C\, M \frac{d\boldsymbol{s}_k}{ds_k} = 0 \tag{5.14}$$

Eq. (5.14) is a polynomial in s_k which can be solved for instance via Newton-Raphson method, given an initial guess $\bar{s}_k$:

$$s_k^* = \bar{s}_k - \frac{F(\bar{s}_k)}{F'(\bar{s}_k)} \tag{5.15}$$

where

$$F'(s_k) = \frac{dF(s_k)}{ds_k} = \frac{d\boldsymbol{s}_k^{\mathrm{T}}}{ds_k} M C^{\mathrm{T}} C M \frac{d\boldsymbol{s}_k}{ds_k} + \boldsymbol{s}_k^{\mathrm{T}} M C^{\mathrm{T}} C M \frac{d^2\boldsymbol{s}_k}{ds_k^2} - \boldsymbol{r}_k^{\mathrm{T}} C\, M \frac{d^2\boldsymbol{s}_k}{ds_k^2} \tag{5.16}$$

In Eqs. (5.10 - 5.16), vector $\boldsymbol{s}_k$ and its derivatives have the following expressions:

$$\boldsymbol{s}_k = \begin{Bmatrix} s_k^n \\ s_k^{n-1} \\ \vdots \\ s_k \\ 1 \end{Bmatrix} \qquad \frac{d\boldsymbol{s}_k}{ds_k} = \begin{Bmatrix} n s_k^{n-1} \\ (n-1) s_k^{n-2} \\ \vdots \\ 1 \\ 0 \end{Bmatrix} \qquad \frac{d^2\boldsymbol{s}_k}{ds_k^2} = \begin{Bmatrix} n(n-1) s_k^{n-2} \\ (n-1)(n-2) s_k^{n-3} \\ \vdots \\ 0 \\ 0 \end{Bmatrix} \tag{5.17}$$

Minimizing Eq. (5.11) (note how the first and last point are left unchanged, to ensure the parameter always spans the range $[0, 1]$), leads to a new parameter distribution, that can in turn be used in Eq. (5.4) to find a new set of control points. In conclusion, Eqs. (5.8) - (5.9) can be used as initial guess for the iterative process.

V.A.3. Non-linear Time Best Fitting

Analogous to what has been done for position, time along the trajectory can also be expressed as a function of the parameter s. When measurements are taken at constant time step, the implicit assumption is that t varies linearly with the parameter, as in $t = t_i\,(1-s)+t_f\,s$, which can be seen as a Bézier function of degree 1. However, non-linear time models better accommodate velocity variations along the trajectory. Therefore, time can described with polynomials of higher degree, and not necessarily the same used for the trajectory. In principle, the two functions exist in different dimensional spaces, with the "time curve" being effectively one dimensional. However, they still are related via the parameter s, because each measurement is associated with a time instant. Therefore, upon reaching convergence in the estimation of the $\boldsymbol{s}_k$, the control points τ for the "time curve" are easily found by applying just once the same equation used for $\mathbb{C}$:

$$\tilde{\boldsymbol{t}} = \tau \mathbb{M}_p \mathbb{S}_p \quad \rightarrow \quad \tau = \tilde{\boldsymbol{t}}\,\mathbb{S}_p^{\mathrm{T}}\,(\mathbb{S}_p \mathbb{S}_p^{\mathrm{T}})^{-1}\,\mathbb{M}_p^{-1} \tag{5.18}$$

where the subscript "p" is used to highlight how the degree used in this calculation is not the same used for the estimation of the position.

V.B. Bézier Least Square Algorithm

In conclusion, the algorithm is composed of three cycles in sequence: the first finds the optimal degree for the polynomial interpolating the position, the second optimizes the parameter distribution, and the last finds the optimal degree for the polynomial interpolating time, based on the parameter distribution found earlier.

An alternative formulation, where the parameter distribution is optimized before the degree, has also been tried. However, its results were of equal or lower accuracy, while causing a considerable increase in computational cost. The overall algorithm of the Bézier Least Squares (BLS) is described in Figure reffig:flowchart.

V.B.1. Estimate of the Velocity

As an added result of the method, once the "space" and "time" curve have been found, it is straightforward to obtain an estimate for the velocity at the measured positions; indeed:

$$\boldsymbol{v}(t) = \frac{d\boldsymbol{r}}{dt} = \left(\frac{d\boldsymbol{r}}{d\boldsymbol{s}}\right)\Big|_{s_k} \left(\frac{ds}{dt}\right)\Big|_{s_k} = \left(\frac{d\boldsymbol{r}}{d\boldsymbol{s}}\right)\Big|_{s_k} \left(\frac{dt}{d\boldsymbol{s}}\right)\Big|_{s_k}^{-1} \tag{5.19}$$

Because both $\boldsymbol{r}(s)$ and $t(s)$ are polynomials, these derivatives are extremely simple to find. It is worth noting that Eq. (5.19) does not return the Bézier curve for the velocity (i.e. its control points), but only the estimated velocity at the s_k.

V.C. Weights of the Measurements

The next logical step is to extend the method by including relative weights for the measurements. This can be done easily, from a mathematical standpoint, by comparing Eq. (5.2) with the classic equations for Weighted Least Squares. Assuming, as is often the case, all measurements to be uncorrelated,the weights matrix $\mathbb{W}$ will be diagonal. Then:

$$\min_{\mathbb{C}} (\mathbb{C}\mathbb{M}\mathbb{S} - \mathbb{R})^{\mathrm{T}}\mathbb{W}(\mathbb{C}\mathbb{M}\mathbb{S} - \mathbb{R}) = \min_{\mathbb{C}} \mathbb{S}^{\mathrm{T}}\mathbb{M}\mathbb{C}^{\mathrm{T}}\mathbb{W}\mathbb{C}\mathbb{M}\mathbb{S} - 2\mathbb{R}^{\mathrm{T}}\mathbb{W}\mathbb{C}\mathbb{M}\mathbb{S} + \mathbb{R}^{\mathrm{T}}\mathbb{W}\mathbb{R} \tag{5.20}$$

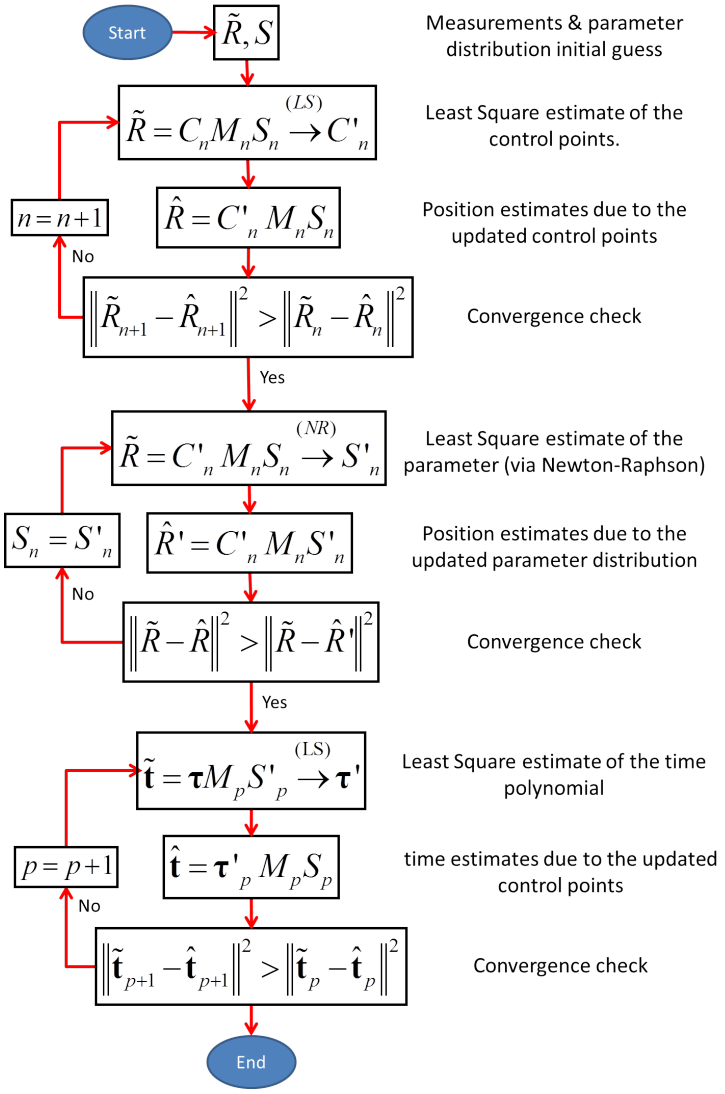


Figure V.1: Algorithm scheme.

And the necessary condition for a local minimum is:

$$\frac{d\mathbb{K}}{d\mathbb{C}} = \mathbb{C}(\mathbb{M}\mathbb{S}\mathbb{W}\mathbb{S}^{\mathrm{T}}\mathbb{M}) - \mathbb{R}\mathbb{W}\mathbb{S}^{\mathrm{T}}\mathbb{M} = 0 \tag{5.21}$$

which can be further simplified:

$$\mathbb{C}(\mathbb{M}\mathbb{S}\mathbb{W}\mathbb{S}^{\mathrm{T}}\mathbb{M}) = \mathbb{R}\mathbb{W}\mathbb{S}^{\mathrm{T}}\mathbb{M} \quad \rightarrow \quad \mathbb{C} = \mathbb{R}\mathbb{W}\mathbb{S}^{\mathrm{T}}\mathbb{M}(\mathbb{M}\mathbb{S}\mathbb{W}\mathbb{S}^{\mathrm{T}}\mathbb{M})^{-1} \tag{5.22}$$

and results into the following expression:

$$\mathbb{C} = \mathbb{R}\mathbb{W}\mathbb{S}^{\mathrm{T}}(\mathbb{S}\mathbb{W}\mathbb{S}^{\mathrm{T}})^{-1}\mathbb{M}^{-1} \tag{5.23}$$

On the other hand, choosing the actual values to populate $\mathbb{W}$ is not at all easy. Indeed, they are dependent both on the sensors used to acquire the data and the methods used to process these data to obtain the actual measurements. Therefore, each set of weights is specific to the mission and generalizations cannot be made. Implementing the use of weights for the Bézier Least Square methods require a very detailed description of the data acquisition and elaboration system, whose specifications might not be established enough in the initial stages of mission design. Depending on the application, it is sometime possible to run custom-made Monte Carlo simulations to obtain approximate values for the weights. Other times, a possible estimate can be derived by analysis of the data processing model.

When generating measurements from analysis of images of celestial bodies the precision of such measurements is dependent primarily on the relative distance between observer and target (if the sensor properties can be assumed constant throughout the mission), according to the following relationship, derived from the geometry of Figure V.2:

$$\frac{dr}{dD} = -\frac{f\,R\,D}{(D^2 - R^2)^{3/2}} \qquad \rightarrow \qquad \sigma_D^2 = \frac{(D^2 - R^2)^3}{f^2\,R^2\,D^2}\,\sigma_r^2 \tag{5.24}$$

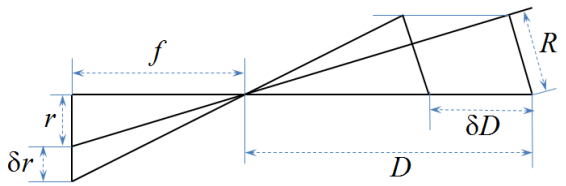


Figure V.2: Distance estimation sensitivity geometry.

In accordance with the Gauss-Markov theorem, the weights can be chosen as $w_k = 1/\sigma_D^2$, so that the Least Square estimator is also a Minimum Variance Estimator.

V.D. Bézier Least Squares Sensitivity Analysis

Because a Bézier function in multiple dimensions is simply an ordered vector of one-dimensional functions, all the testing is done in 1D and the results are immediately generalizable. For these tests, the target trajectory has been chosen as a known Bézier curve, with artificially superimposed Gaussian noise. This simplifies the analysis and allows the investigation of the sensitivity of the algorithm with respect to changes in the parameter distribution. The target trajectory is a 5^{th} degree curve, depicted in Figure V.3.

First, we study the effect of noisy measurements. Gaussian noise with increasing values of standard deviation (σ_r), up to 10% of the average value of the signal, is superimposed to the truth, and a total of 100 tests are run for each value of σ_r (Figure V.4).

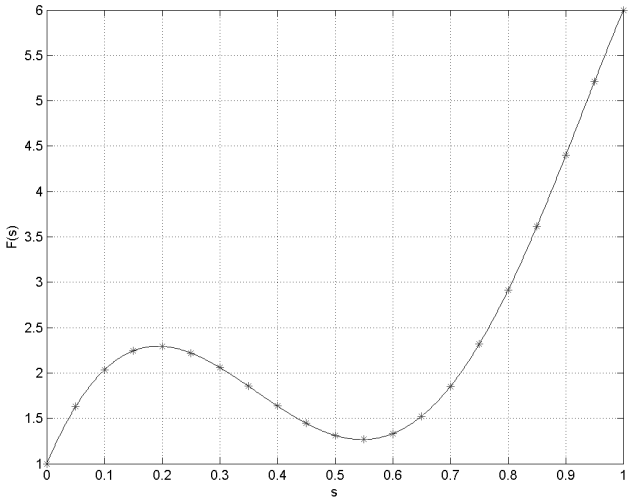


Figure V.3: Target function.

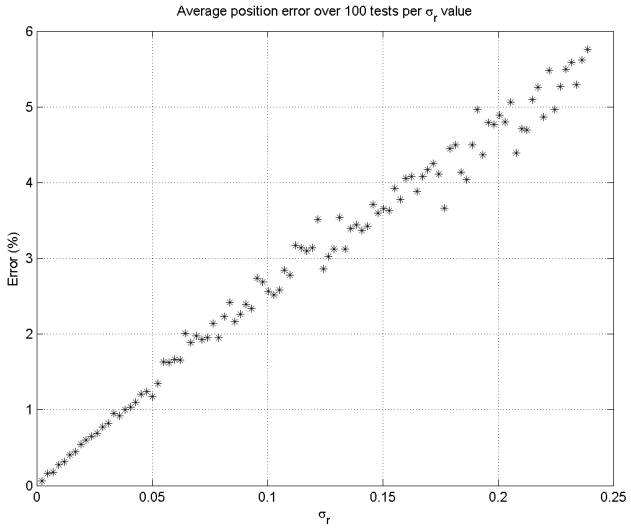


Figure V.4: Average error for increasing noise in the measurements.

One of the steps in the algorithm consists in shifting the values of s along the curve to minimize the error for a given set of control points. Because of the need

to assume a starting distribution to initialize the algorithm, it is worth analyzing convergence when starting with poorly modeled distributions. The target trajectory is generated using a uniform distribution, while the algorithm is initialized with a different distribution. This is obtained as follows: each s_k of the true uniform distribution is perturbed from its original position by a Gaussian noise whose standard deviation (σ_s) is proportional to the width of the interval between two subsequent s_k, so as to keep the sequence ordered. Similarly to what has been done for the case of noisy measurements, the standard deviation is increased linearly within the boundary described above and 100 tests are performed for each value of σ_s (Figure V.5).

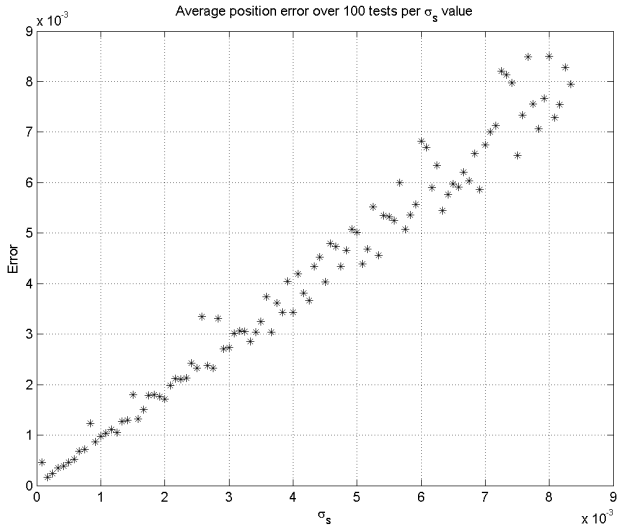


Figure V.5: Average error for increasing disturbances in the s distribution.

V.E. Estimation of a Cislunar Trajectory

As a second test, Bézier Least Squares is used to estimate segments of a cislunar trajectory for a Earth-to-Moon mission. The measurements are obtained with a camera of focal length 300 mm and then random Gaussian noise, whose standard deviation is based on the optical properties of the sensor, is artificially added. Using Eq. (5.24), it is possible to introduce weights in the formulation, as the the reciprocal of σ_D^2. Along the trajectory, three segments are considered, close to the Earth, midway, and close to the Moon (Figure V.6). The second segment has a very small average curvature compared to the first and the third. The first segment is the most curved, and along it the spacecraft is mainly subject to Earth gravitational effects, while along the third segment the Moon is the principal gravitational body Each segment consists of 25 measurements, equally spaced over 2 hours. The trajectory is simulated with the GMAT software, taking into account the gravitational effects of Sun, Earth and Moon, in addition to higher order effects due to the Earth gravitational field and the effects of solar radiation pressure.

The results show the performance of the Bézier least squares method point by point along the segments, compared with a Weighted Iterative Least Square (ILS) and an Extended Kalman Filter (EKF), both of which require full state propagation to compute the estimate. For these two methods, the simplified physical model implemented only includes the point mass effects of the Earth and the Moon.

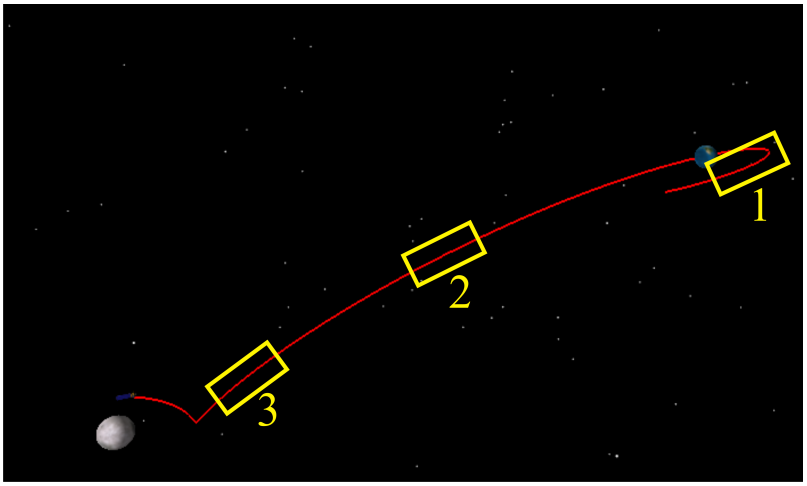


Figure V.6: Simulated Earth-to-Moon trajectory (GMAT).

V.E.1. Weighted Batch Iterative Least Square

The standard Batch Least-Squares orbit determination technique [39, pp. 199-210] is used as a mean to assess the performance of the proposed method. Rather then processing the Moon direction $\tilde{\boldsymbol{b}}_k$ and Radius $\tilde{R}_k$ separately, a single derived measurement is obtained. From the apparent Moon radius and the focal length, the "measured" distance from the Moon $\tilde{\rho}_k$ is deduced. Then a measurement $\tilde{\boldsymbol{y}}_k$ of the vehicle position from the center of the Moon is calculated

$$\tilde{\boldsymbol{y}}_k = \tilde{\rho}_k C_k^{\mathrm{T}} \, \tilde{\boldsymbol{b}}_k = f \, \frac{R_{\leftmoon}}{\tilde{R}_k} \, C_k^{\mathrm{T}} \, \tilde{\boldsymbol{b}}_k \tag{5.25}$$

where $R_{\leftmoon}$ is the Moon radius, f is the camera focal length, and C_k is the J2000-to-camera attitude.

To obtain the batch least-squares solution we start from an initial estimate $\hat{\boldsymbol{x}}_0^{\mathrm{T}} = \{ \, \hat{\boldsymbol{r}}_0^{\mathrm{T}}, \, \hat{\boldsymbol{v}}_0^{\mathrm{T}} \, \}$ and propagate it forward to each measurement time using Earth and Moon's central gravities. Together with the state we integrate the state tran-

sition matrix $\Phi(t_k, t_0)$ that takes linearized state deviations from the time of the initial estimate to the time of the k-th measurement. At each measurement time we accumulate the current observation:

$$\tilde{H}_k = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{5.26}$$

$$H_k = \Phi(t_k, t_0) \tag{5.27}$$

$$\Lambda = \Lambda + H_k^{\mathrm{T}} W_k H_k \tag{5.28}$$

$$N = N + H_k^{\mathrm{T}} W_k \left[\tilde{\boldsymbol{y}}_k - (\boldsymbol{r}_{mk} - \hat{\boldsymbol{r}}_k)\right] \tag{5.29}$$

where Λ and N are initialized at zero, $\tilde{H}_k$ is the partial of the measurement with respect to the current state, H_k is the partial of measurement with respect to the initial state, $\hat{\boldsymbol{r}}_k$ is the estimated position of the vehicle with respect to the center of Earth and $\boldsymbol{r}_{mk}$ is the position of the Moon also with respect to Earth. The measurement weight W_k is given by the inverse of the measurement error covariance Σ_k.

$$W_k = \Sigma_k^{-1} \tag{5.30}$$

$$\begin{aligned} \Sigma_k &= \sigma_R^2 \left(\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \tilde{R}_k}\right)\left(\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \tilde{R}_k}\right)^{\mathrm{T}} + \sigma_b^2 \left(\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \tilde{\boldsymbol{b}}_k}\right)\left(\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \tilde{\boldsymbol{b}}_k}\right)^{\mathrm{T}} \\ &= \sigma_R^2 f^2 \frac{R_{\leftmoon}^2}{\tilde{R}_k^4} C_k^{\mathrm{T}} \tilde{\boldsymbol{b}}_k \tilde{\boldsymbol{b}}_k^{\mathrm{T}} C_k + \sigma_b^2 \tilde{\rho}_k^2 \left(\boldsymbol{I} - C_k^{\mathrm{T}} \tilde{\boldsymbol{b}}_k \tilde{\boldsymbol{b}}_k^{\mathrm{T}} C_k\right) \end{aligned} \tag{5.31}$$

where $sigma_R^2$ is the variance of the error in determining the Moon radius and σ_b^2 is the variance of the error in determining the Moon direction.

After all measurements are accumulated the initial estimate is updated as:

$$\hat{\boldsymbol{x}}_0 = \hat{\boldsymbol{x}}_0 + \Lambda^{-1} N \tag{5.32}$$

This procedure can be iterated multiple times, until convergence is reached.

V.E.2. Extended Kalman Filter

The extended Kalman filter incorporates one measurement at the time rather than using all of them at once. Therefore using the EKF algorithm only the estimate at the very last measurement time contains information from all the measurements. This is in contrast with the Iterative Least-Squares or the Bézier Least Squares approach in which the estimate at any point of the trajectory is obtained by best fitting all measurements available. The EKF estimate at the time of the last measurement is almost identical to the Weighted Iterative Least-squares estimate at that same time.

The EKF is initialized with an initial estimate $\hat{\boldsymbol{x}}_0^{\mathrm{T}} = \{\, \hat{\boldsymbol{r}}_0^{\mathrm{T}},\, \hat{\boldsymbol{v}}_0^{\mathrm{T}} \,\}$ with associated initial estimation error covariance given by

$$P_0 = \begin{bmatrix} (5 \text{ km})^2 \boldsymbol{I}_{3\times 3} & \boldsymbol{O}_{3\times 3} \\ \boldsymbol{O}_{3\times 3} & (0.01 \text{ km}/s)^2 \boldsymbol{I}_{3\times 3} \end{bmatrix}$$

The state is propagated forward to the next measurement time using Earth and Moon central gravities. Together with the state we integrate the state transition matrix $\Phi(t_k, t_{k-1})$ that takes linearized state deviations from the prior measurement time to the current measurement time. This is different from batch least-squares where the state transition matrix was always referring to the initial epoch. In-between measurements the covariance is propagated as

$$P_k = \Phi(t_k, t_{k-1}) P_{k-1}^{+} \Phi(t_k, t_{k-1})^{\mathrm{T}}$$

notice the absence of process noise. The measurement update is given by

$$K_k = P_k \tilde{H}_k^{\mathrm{T}} (\tilde{H}_k P_k \tilde{H}_k^{\mathrm{T}} + \Sigma_k)^{-1} \tag{5.33}$$

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k + K_k[\tilde{\boldsymbol{y}}_k - (\boldsymbol{r}_{mk} - \hat{\boldsymbol{r}}_k)] \tag{5.34}$$

$$P_k^+ = P_k - K_k(\tilde{H}_k P_k \tilde{H}_k^{\mathrm{T}} + \Sigma_k) K_k^{\mathrm{T}} \tag{5.35}$$

where $\tilde{H}_k$ and Σ_k are defined in Eqs. (5.26) and (5.31), respectively.

V.E.3. Simulation Results

For all segments, the ILS reaches convergence after 8 iterations, and the optimal degree found by BLS is 7. As mentioned above, Gaussian noise is superimposed to the simulated measurements. Results for segment 1 are reported in Figure V.7 and V.8, results for segment 2 are in Figure V.9 and V.10, and results for segment 3 in Figure V.11 and V.12. As it can be seen, BLS performs comparably with ILS and EKF. Each figure shows the root sum squared of the errors at each measurement time for the three algorithms. The overall performance of the three methods is summarized by taking the average error along each segment in both position and velocity. The results are reported in Table V.1.

It can be seen that the EKF provides progressively worse results as the spacecraft moves further from Earth, while BLS improves. This is mainly due to the fact that EKF (like the ILS) uses a model which does not take into account any perturbation and has zero process noise, therefore its results are less accurate where such perturbations are more prominent. In contrast, BLS only uses measurements and it is "closer" to the truth along segment 3. It is worth noting that even when BLS has a greater margin of error than the other methods, it remains a sensibly sim-

	Close to Earth	Midway	Close to Moon
ILS Position error (km)	1.420	0.695	0.328
BLS Position error (km)	2.997	1.022	0.124
EKF Position error (km)	2.613	1.187	0.298
ILS Velocity error (km/s)	0.0019	0.0004	3.20×10^{-4}
BLS Velocity error (km/s)	0.0063	0.0026	6.88×10^{-5}
EKF Velocity error (km/s)	0.0067	0.0185	0.0157

Table V.1: Average error comparison.

pler method to implement, because it does not require any of the tuning procedures typically associated with the Kalman filters. Moreover, the velocity estimates are obtained very simply as a secondary result according to Eq. (5.19), while EKF and ILS perform a full state propagation to obtain the same result.

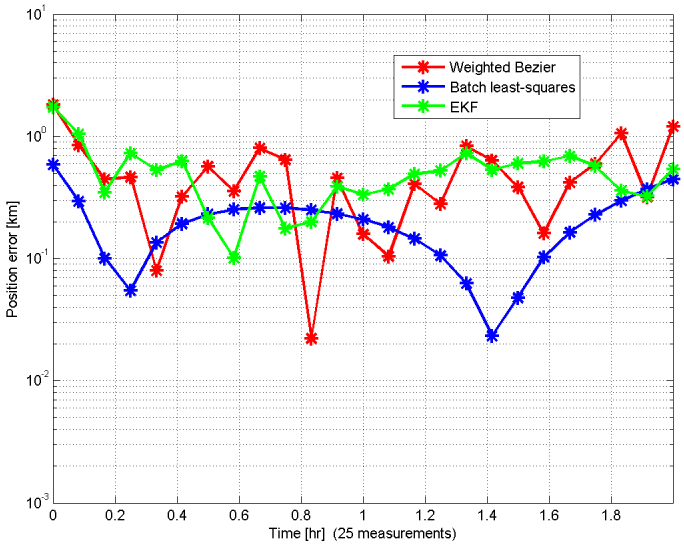


Figure V.7: Position estimation error along segment 1.

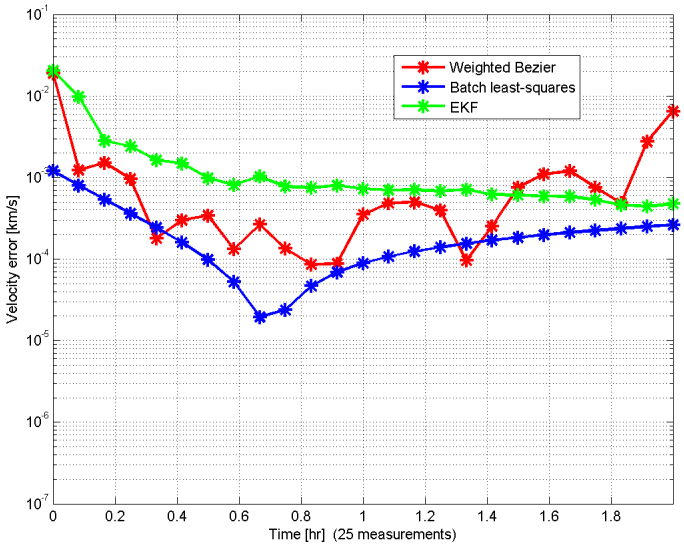


Figure V.8: Velocity estimation error along segment 1.

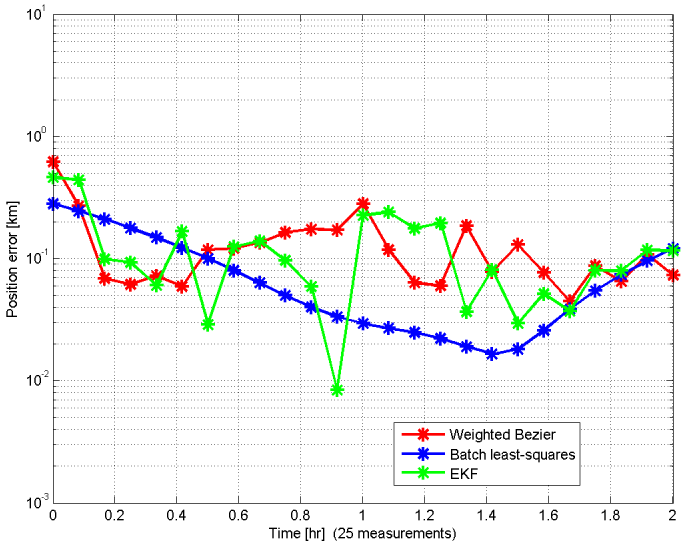


Figure V.9: Position estimation error along segment 2.

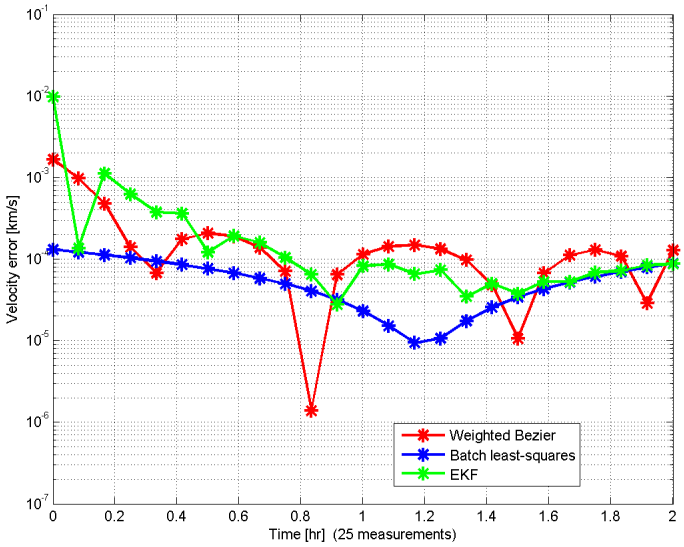


Figure V.10: Velocity estimation error along segment 2.

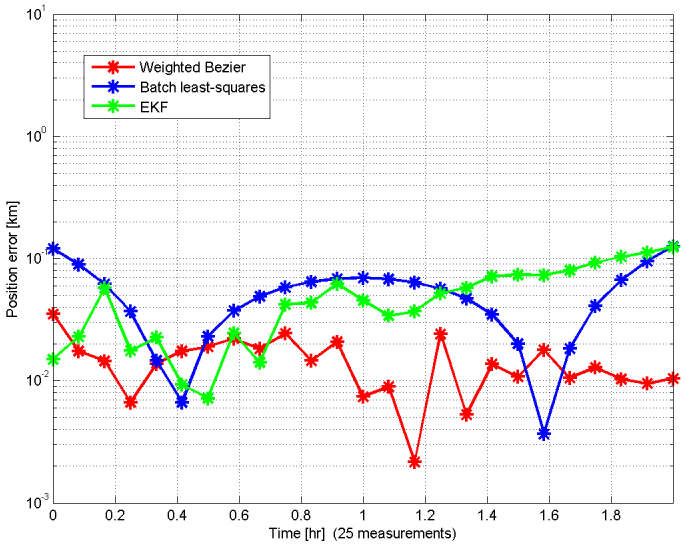


Figure V.11: Position estimation error along segment 3.

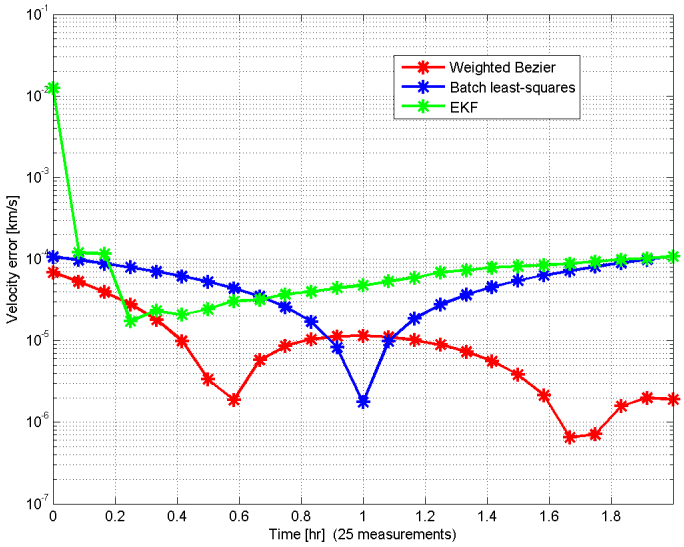


Figure V.12: Velocity estimation error along segment 3.

CHAPTER VI

ERROR ANALYSIS

The previous three chapters of this dissertation have discussed the techniques to extract position data from images of Moon and Earth and interpolate this data with a trajectory. It is now necessary to investigate how errors are generated and propagated during the computation. Such errors can be divided in four categories, depending on what they affect or at which step in the algorithm they occur.

1. **Estimated position error**. This is the error in the Orion-to-Moon distance computation (in km) due to errors in measured Moon radius (in pixel).

2. **Estimated attitude error**. This is the error made in the estimate of Orion-to-Moon direction due to errors in the camera attitude knowledge (angle between quaternion).

3. **Image processing errors**. The presence of noise in the image affects the estimation. To quantify this effect, the same image is artificially perturbed by random Gaussian noise multiple times, and then results of the estimation process are compared, in a Montecarlo analysis. This is done for both fully and partially illuminated target.

4. **Pin-hole projection error**. Lastly, it is necessary to consider the radial distortion created by the pin-hole model (gnomonic projection) on the target shape. Even for a spherical target, if its center is not aligned with the camera

optical axis, the pin-hole model projection results in a conic section. The source of error is not discussed in this work however.

This analysis will focus primarily on the Moon as a target.

VI.A. Position and Attitude Uncertainty Propagation

First and foremost, it is necessary to investigate how errors in the estimate of the target's center and dimensions propagate to position and attitude.

VI.A.1. Moon Radius

There is a direct relation between the target radius and the Observer-to-Target distance. The Observer-to-Target vector can be defined as $\boldsymbol{r}_{om} = \boldsymbol{r}_m - \boldsymbol{r}$ where $\boldsymbol{r}_m$ and $\boldsymbol{r}$ are the target and observer position vectors in J2000, respectively. The vector $\boldsymbol{r}_m$ is known while the observer position vector, $\boldsymbol{r}$, 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 uniformly distributed in the unit-radius sphere. Obviously, the uncertainty in the observer vector position coincides with the uncertainty in the Observer-to-Target distance, $\sigma_D = \sigma_{\boldsymbol{r}}$.

Now, focusing on the Moon, its radius in pixels in the imager is equal to

$$r = \frac{f}{d}\,\frac{R_m}{\sqrt{D^2 - R_m^2}} \tag{6.1}$$

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 Observer-to-Moon distance. This equation

allows us to derive the uncertainty in the Moon radius in the imager

$$\sigma_r = \left| \frac{\partial r}{\partial D} \right|_{\bar{\boldsymbol{r}}} \sigma_D = \frac{f\, D\, R_m}{d\,(D^2 - R_m^2)^{3/2}}\, \sigma_{\boldsymbol{r}} \tag{6.2}$$

The inverse of Eq. (6.2) is

$$\sigma_D = \frac{d\,(D^2 - R_m^2)^{3/2}}{f\, D\, R_m}\, \sigma_r \tag{6.3}$$

and provides an important information about the accuracy required by the radius estimation. This equation describes how the radius estimate accuracy is affecting the Observer-to-Moon vector length, and, consequently, how it affects the estimation of the position.

Equation (6.3) has been plotted in Figure VI.1 as a function of the Moon radius error and for four values of distance, $D = 50{,}000$ km, $D = 150{,}000$ km, $D = 250{,}000$ km, and $D = 350{,}00$ km. Figure VI.2 provides σ_D as function of the Moon distance for the specific value of $\sigma_r = 0.3$ pixels.

VI.A.2. Moon Center Direction

The Moon direction uncertainty 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 $\bar{\boldsymbol{q}}$ and standard deviation $2\sigma_{\boldsymbol{q}}$. 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 camera optical axis. Therefore, the radial uncertainty in the Moon center direction is provided by

$$\sigma_c = \frac{f}{d} \left| \frac{1}{\cos^2\vartheta} \right|_{\bar{\boldsymbol{q}}} \sigma_{\boldsymbol{q}} \tag{6.4}$$

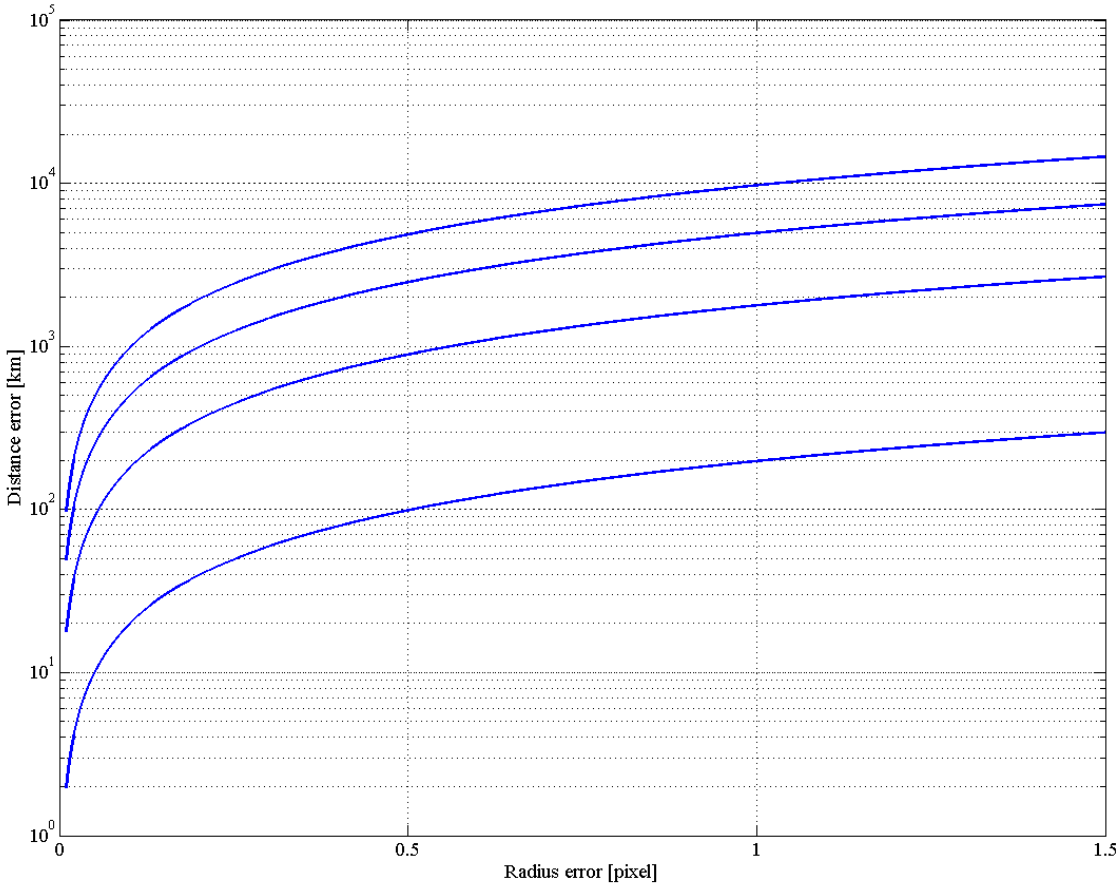


Figure VI.1: Estimation error on Moon distance for various values of distances.

VI.B. Image Processing Error

Image processing error quantification would ideally require performing statistics using a substantial set of images with known true data. Since no such ideal database is available, the following analysis is performed by adding random 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, all other variables being

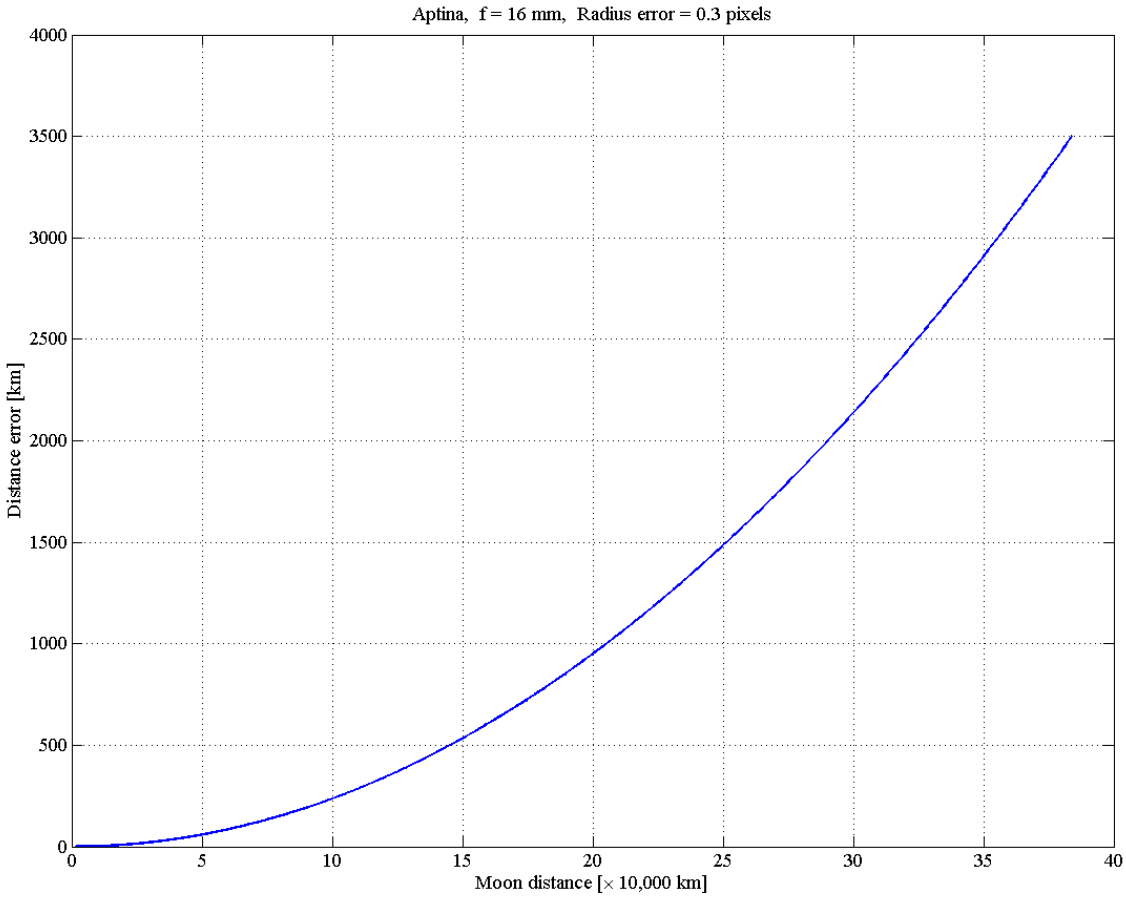


Figure VI.2: Estimation error on Moon distance for Moon radius error of 0.3 pixels.

unchanged. For this reason two scenarios have been considered:

1. Partially illuminated Moon

2. Full illuminated Moon

The results of the tests for partially illuminated target are shown in Figure VI.3. The original image is a real picture taken on March 6, 2013 at 05:24:30 from the Earth surface near Houston using a focal length $f = 100$ mm. The statistics of the results

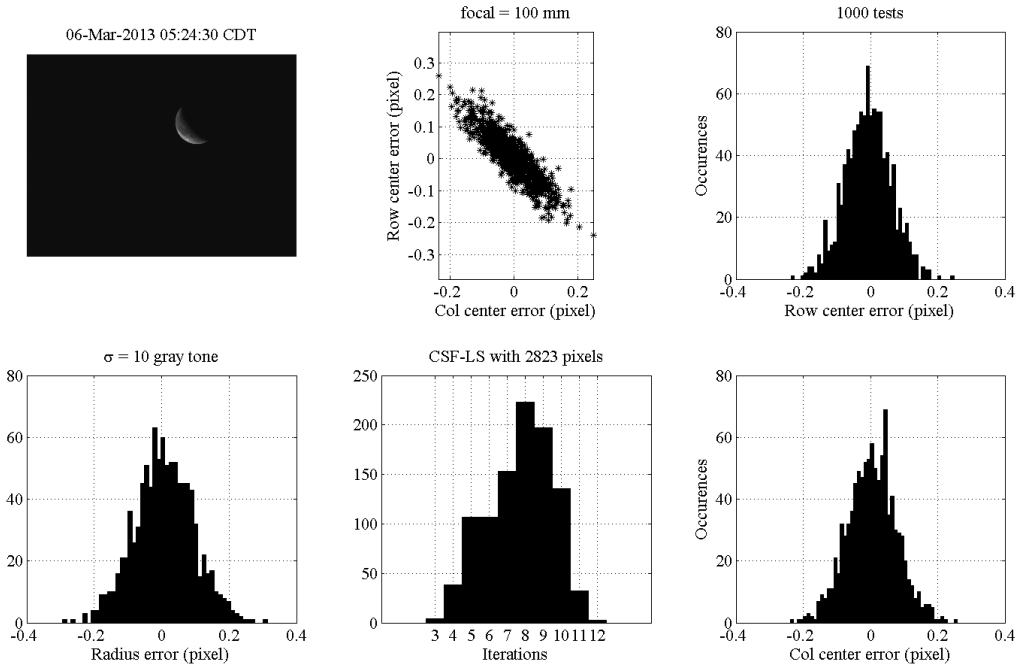


Figure VI.3: Image processing error results: partial illumination case.

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 second image in the first row in Figure VI.3 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.

The results of the tests done using full illuminated Moon are shown in Figure VI.4. Also in this case the image is real and was taken in Houston on February

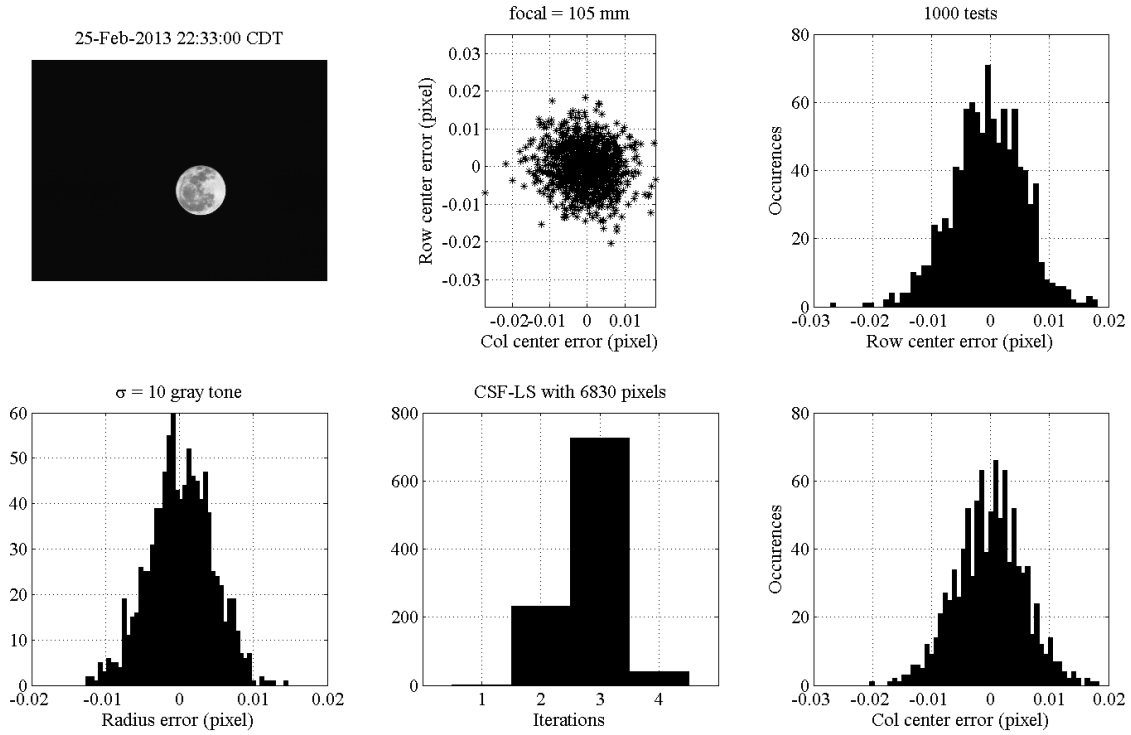


Figure VI.4: Image processing error results: full illumination case.

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 FigureVI.3 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.

CHAPTER VII

CONCLUSIONS

Ultimately, this research succeeded in developing a reliable method to extract position measurements from optical images of celestial bodies and accurately interpolate a trajectory based on those measurements. As the work is composed of two main sections, one dealing with the data extraction, and the other with the trajectory interpolation, so the results will be described accordingly.

VII.A. Image Processing

For the data extraction, it is important to achieve very high accuracy when estimating the target dimensions, as even very small error will generate errors several orders of magnitude greater in the position, a problem that only worsen with distance from the target. To compensate, every effort has been made to minimize such errors at the source. Firstly, the best possible set of pixels has to be selected from the image. This means exclusively pixels belonging to the edge of the target, spread across the widest arc possible. Of the two techniques developed, the Gradient Method has proven to be more reliable and more versatile, being able to correctly work in any condition, whether the target is fully or partially illuminated, cropped or fully within the field of view, or even barely visible. Therefore, the final code only uses this method for pixel selection. To improve the reliability of the code, several safety checks have also been included, as depicted in Figure VII.1. These also ensure the final result will be consistent.

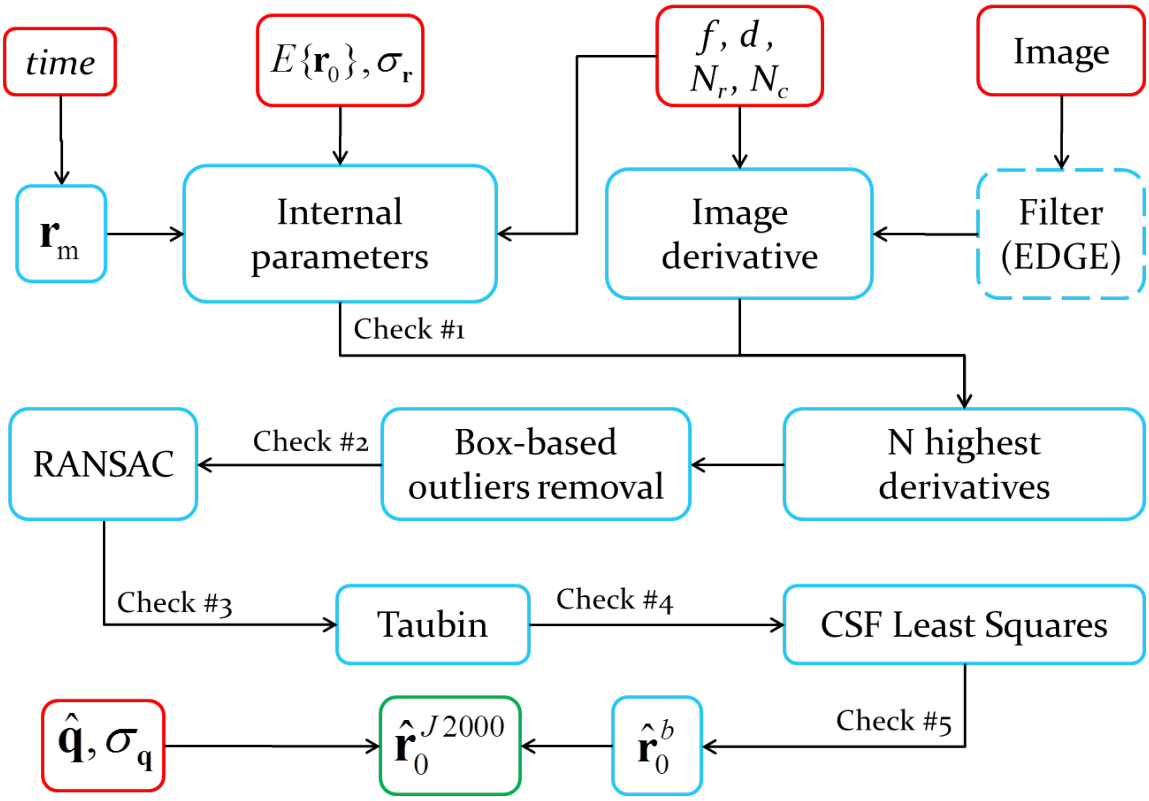


Figure VII.1: Flowchart of the implemented algorithm.

- **Check #1** (after Internal Parameters computation).

  1. If the estimated radius p_r (see Eq. 2.1) is less than a minimum threshold, the target is considered too small to be observed.

  2. If the illumination parameter p_i is less than a minimum fraction of p_r, the target is assumed invisible. Also can be triggered if the target is completely out of frame, or covers more than 95% of the imager.

- **Check #2** (after box based outliers removal).

  1. If too many outliers have been removed during this phase, the data set is

considered fundamentally flawed and therefore rejected altogether.

- **Check #3** (after RANSAC based outliers removal).

    1. If too many outliers have been removed by RANSAC, then the data set does not satisfy the basic assumption behind the method, and therefore it is rejected.
    2. If the distances found by RANSAC are not consistent with p_r, the data set is also considered flawed.

- **Check #4** (after Taubin).

    1. If the radius estimated by Taubin (or the semi major axis for Fitzgibbon) is not consistent with p_r, the best fitting procedure is considered failed.
    2. If the computed center position is too far away from the optical axis, the image is likely to be too distorted for good estimate.

- **Check #5** (inside and after CSF-LS).

    1. As the CSF-LS is an iterative process, a standard check on convergence is performed.
    2. If the correction applied to the initial guess is too big, the procedure is assumed to have diverged.

If any of these checks are failed, the process is aborted and the system waits for the next image.

To maximize the best fitting procedure, a great number of methods have been tested and compared, to find those who best suit the application. These results have

also been improved with original work. Particularly, the use of Sigmoid least square, based on initial guess obtained with algebraic fit, allows to reach precisions well below pixel size (less than 0.3 pixels). Despite these efforts, it has become apparent that best fitting for ellipses constitute a much harsher problem than for circles. This is not due to the conic section itself, as the latest algebraic fits can overcome the mathematical complications easily. The real difficulty lies in the target itself being the projection of an ellipsoid with extremely small flattening, which in turn projects ellipses whose eccentricity is almost zero. This has two consequences:

- Of all the parameters, the inclination θ is the hardest one to observe, and consequently the associated average error is one order of magnitude greater than for the others.

- No matter how sophisticated the technique, eventually the eccentricity will be completely obscured by noise, and therefore the ellipse can only be best fitted as a circle.

VII.B. Trajectory Estimation

Regarding the trajectory estimation, this work shows that Least Squares applied to Non-Rational Bézier functions provide a reliable approach to describe space trajectories based on a set of measured position vectors. The proposed approach has been tested in three segments of a simulated cislunar trajectory and compared with iterative Least Squares and extended Kalman filter estimators. Indeed, the average error of all three methods was comparable, despite the Bézier least squares being much simpler in implementation, and not requiring any dynamical model to work, instead

relying completely upon measurements. For this reason it is expected to be suitable in those situations where accurate dynamics models are too complicated to describe accurately, whether because they are constantly changing or unknown altogether. Typical examples are the solar radiation pressure model (it depends on attitude and on solar activity) or venting and accidental pipe leaking events. Moreover, since dynamical models are not used, knowledge of process noise and/or dynamical model uncertainty are not needed. Compared to other best fitting methods, this approach also provides an interpolation of the measurement times, which can be used to obtain velocity estimates.

Ultimately, Bézier least squares is not meant to replace more advanced methods, when based on accurate dynamical models. Rather, it can be used when these estimators need to work far from their nominal (and optimal) range of applicability or when they provide doubtful results. In these cases the proposed method still provides good estimates of both position and velocity.

REFERENCES

[1] R. Bellman, H. Kagiwada, and R. Kalaba. "Orbit Determination as a Multi-Point Boundary-Value Problem and Quasilinearization". In: *Proceedings of the National Academy of Sciences of the United States of America*. Vol. 48. Aug. 1962, pp. 1327–1329. DOI: 10.1073/pnas.48.8.1327.

[2] Nikolai Chernov. *Circular and Linear Regression. Fitting Circles and Lines by Least Squares*. Vol. 117. Monographs on Statistics and Applied Probablity. CRC Press, 2011.

[3] M. A. Chory et al. "Autonomous Navigation - Where We Are in 1984". In: *Guidance and Control*. Keystone, CO, 1984, pp. 27–37.

[4] J. L. Crassidis and J. L. Junkins. *Optimal Estimation of Dynamical Systems*. 2nd. CRC Press, 2011.

[5] G.F. Curtis and P.O. Wheatley. "Applied Numerical Analysis". In: 7th. Pearson, 2004. Chap. 5, pp. 269–270.

[6] K.J. DeMars and M.K. Jah. "Probabilistic Initial Orbit Determination Using Gaussian Mixture Models". In: *Journal of Guidance, Control and Dynamics* 36.5 (Sept. 2013), pp. 1324–1335. DOI: 10.2514/1.59844.

[7] F. de Dilectis, D. Mortari, and R. Zanetti. "Trajectory Determination with Unknown Perturbations". In: *AAS/AIAA Astrodynamics Specialist Conference*. Hilton Head, SC, 2013.

[8] C. D'Souza et al. "Orion Cislunar Guidance and Navigation". In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2007.

[9] B. T. Fang. "Satellite-to-Satellite Tracking Orbit Determination". In: *Journal of Guidance, Control, and Dynamics* 2.1 (1979), pp. 57–64.

[10] M.A. Fischler and Bolles. R. C. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Communications of the ACM* 24 (July 1981), pp. 381–395.

[11] A. Fitzgibbon, M. Pilu, and R. B. Fisher. "Direct Least Square Fitting of Ellipses". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21.5 (1999), pp. 476–480.

[12] D. Folta et al. "Autonomous Navigation Using Celestial Objects". In: *AAS/AIAA Astrodynamics Specialist Conference*. Girdwood, AK, 1999.

[13] J. R. Guinn and R. J. Boain. "Spacecraft Autonomous Navigation for Formation Flying Earth Orbiters Using GPS". In: *AAS/AIAA Astrodynamics Specialists Conference*. San Diego, CA, 1996, pp. 29–31.

[14] Yanping Guo. "Self-Contained Autonomous Navigation System for Deep Space Missions". In: *AAS/AIAA Space Flight Mechanics Meeting*. Breckenridge, CO, 1999, pp. 1099–1113.

[15] R. Halir and J. Flusser. "Numerically Stable Direct Least Squares Fitting of Ellipses". In: *6th Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*. Pilsen, Czech Republic, 1998.

[16] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*. Vol. 1. Addison-Wesley, 1993.

[17] K.C. Howell and H.J. Pernicka. "Numerical determination of Lissajous trajectories in the restricted three-body problem". In: *Celestial Mechanics* 41.1-4 (1987), pp. 107–124. DOI: 10.1007/BF01238756.

[18] E. F. Jochim et al. "GPS Based Onboard and Onground Orbit Operations for Small Satellites". In: *Acta Astronautica* 39.912 (1996), pp. 917–922.

[19] Reza Raymond Karimi. "Designing an Interplanetary Autonomous Spacecraft Navigation System Using Visible Planets". PhD thesis. Texas A&M University, May 2012.

[20] I. Kåsa. "A Curve Fitting Procedure and its Error Analysis". In: *IEEE Transactions on Instrumentation and Measurement* 25 (1976), pp. 8–14.

[21] N. P. Laverty et al. "Multi-Mission Attitude Determination and Autonomous Navigation/MADAN". In: *Guidance and Control*. Vol. 1. Keystone, CO, 1980, pp. 551–609.

[22] W. Li et al. "Jason-2 precise orbit determination using DORIS RINEX phase data". In: *Geomatics and Information Science of Wuhan University* 38.10 (Oct. 2013), pp. 1207–1211.

[23] J. W. Lowrie. "Autonomous Navigation Systems Technology". In: *17th Aerospace Sciences Meeting*. New Orleans, LA, 1979.

[24] F. Markley. "Autonomous Navigation Using Landmark and Intersatellite Data". In: *AAS/AIAA Astrodynamics Specialist Conference*. Seattle, WA, 1984, pp. 1–11.

[25] F. L. Markley. "Autonomous Satellite Navigation Using Landmarks". In: *AIAA, Astrodynamics Specialist Conference*. Vol. 1. Lake Tahoe, NV, 1981.

[26] D. Marquart. "An Algorithm for Least Squares Estimation of Nonlinear Parameters". In: *Journal of the Society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441. DOI: 10.1137/0111030.

[27] B.G. Marsden. "Initial orbit determination - The pragmatist's point of view". In: *The Astronomical Journal* 90.1541-1547 (Aug. 1985). DOI: 10.1086/113867.

[28] D. Mortari, F. de Dilectis, and C. D'Souza. "Image Processing of Illuminated Ellipsoid". In: *AAS/AIAA Astrodynamics Specialist Conference*. Hilton Head, SC, 2013.

[29] D. Mortari et al. "The Pyramid Star Pattern Recognition Algorithm". In: *ION Navigation* 51.3 (Fall 2004), pp. 171–183.

[30] T. A. Pastva. "Bézier Curve Fitting". PhD thesis. Monterey, California. Naval Postgraduate School, 1998.

[31] V. Pratt. "Direct Least-Squares Fitting of Algebraic Surfaces". In: *SIGGRAPH Comput. Graph.* 21.4 (1987), pp. 145–152.

[32] V. Pratt. "Direct Least Squares Fitting of Algebraic Surfaces". In: *Computer Graphics* 21 (1987), pp. 145–152.

[33] M. Psiaki and F. Martel. "Autonomous Magnetic Navigation For Earth Orbiting Spacecraft". In: *3rd Annual AIAA/USU Conference on Small Satellites*. Logan, UT, 1989.

[34] M. L. Psiaki. "Autonomous Orbit and Magnetic Field Determination Using Magnetometer and Star Sensor Data". In: *Journal of Guidance, Control, and Dynamics* 18.3 (1995), pp. 584–592.

[35] M. L. Psiaki, L. Huang, and S. M. Fox. "Ground Tests of Magnetometer-Based Autonomous Navigation (MAGNAV) for Low-Earth-Orbiting Spacecraft". In: *Journal of Guidance, Control, and Dynamics* 16.1 (1993), pp. 206–214.

[36] S. Shklyar et al. "On the Conic Section Fitting Problem". In: *Journal of Multivariate Analysis* 98 (2007), pp. 588–624.

[37] A. Strandlie et al. "Particle Tracks Fitted on the Riemann Sphere". In: *Computer physics communications* 131.1 (2000), pp. 95–108.

[38] B.D. Tapley. "Recent Advances in Dynamical Astronomy". In: *Astrophysics and Space Science Library* 39 (1973), pp. 396–423. DOI: 10.1002/asna.19752960409.

[39] B.D. Tapley, B.E. Shutz, and G.H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, 2004.

[40] G. Taubin. "Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range

Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.11 (1991), pp. 1115–1138.

[41] G. Taubin. "Estimation of Planar Curves, Surfaces and Nonplanar Space Curves Defined by Implicit Equations, with Applications to Edge and Range Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (1991), pp. 1115–1138.

[42] R. L. White et al. "Attitude and Oribit Estimation Using Stars and Landmarks". In: *Aerospace and Electronic Systems, IEEE Transactions on* AES-11.2 (1975), pp. 195–203.

[43] Y. Yang et al. "Satellite orbit determination combining C-band ranging and differenced ranges by transfer". In: *Chinese Science Bulletin* 58.19 (July 2013), pp. 2323–2328. DOI: 10.1007/s11434-013-5871-6.

[44] C. Z. Zhang. "A Note on the Physical Parameters of the Moon". In: *Earth, Moon and Planets* 64.1 (1993), pp. 31–38.

[45] X. Zhang, P. Li, and X. Zuo. "Kinematic precise orbit determination based on ambiguity-fixed PPP". In: *Geomatics and Information Science of Wuhan University* 39.9 (Sept. 2013), pp. 1009–1013.

APPENDIX A

REFERENCE FRAMES AND COORDINATES TRANSFORMATIONS

As for time definition and reference frames definitions and orientations, the JPL SPICE software has been adopted.

Reference Frames

The proposed image processing requires the use of several Reference Frames (RFs). These are:

- $[\hat{\boldsymbol{i}}_1, \hat{\boldsymbol{i}}_2, \hat{\boldsymbol{i}}_3]$ This RF identifies the J2000 RF. Earth Centered and inertial.
- $[\hat{\boldsymbol{o}}_1, \hat{\boldsymbol{o}}_2, \hat{\boldsymbol{o}}_3]$ This is the spacecraft principal axes RF.
- $[\hat{\boldsymbol{b}}_1, \hat{\boldsymbol{b}}_2, \hat{\boldsymbol{b}}_3]$ This is the observed body (Earth or Moon) RF. Body-centered and body-fixed.
- $[\hat{\boldsymbol{c}}_1, \hat{\boldsymbol{c}}_2, \hat{\boldsymbol{c}}_3]$ This is the camera RF. $\hat{\boldsymbol{c}}_1$ and $\hat{\boldsymbol{c}}_2$ axes lie on the camera imager while $\hat{\boldsymbol{c}}_3$-axis is pointing from imager center to the lens.
- $[\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2, \hat{\boldsymbol{w}}_3]$ This is the elliptical cone RF, as provided by Eq. (2.22).

Coordinates Transformations

To move from/to five different RFs four coordinates transformation DCMs (Direction Cosine Matrix) are needed. These are provided in Table A.1, where C in-

dicates the DCM and $\hat{\boldsymbol{q}}_{ba}$ the quaternion moving the RFs indicated by the arrow:

Inertial	$\rightarrow$	Spacecraft	$\hat{\boldsymbol{o}} = C(\hat{\boldsymbol{q}}_{oi})\,\hat{\boldsymbol{i}}$
Inertial	$\rightarrow$	Observed Body	$\hat{\boldsymbol{b}} = C(\hat{\boldsymbol{q}}_{bi})\,\hat{\boldsymbol{i}}$
Spacecraft	$\rightarrow$	Camera	$\hat{\boldsymbol{c}} = C(\hat{\boldsymbol{q}}_{co})\,\hat{\boldsymbol{o}}$
Observed Body	$\rightarrow$	Elliptical cone	$\hat{\boldsymbol{w}} = C(\hat{\boldsymbol{q}}_{wb})\,\hat{\boldsymbol{b}}$

Table A.1: Coordinate Transformations.

In addition to the previous RFs, the coordinate transformations between imager (row, column and x, y) from/to camera RF must be given (see Fig. A.1). These are provided by the standard pin-hole camera model.

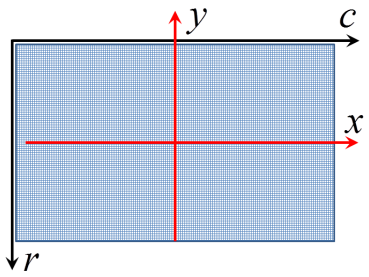


Figure A.1: $[r, c]$ and $[x, y]$ reference frames for the imager.

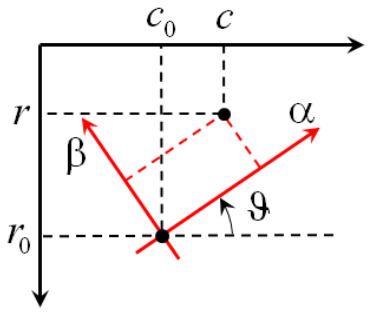


Figure A.2: $[r, c]$ and $[\alpha, \beta]$ reference frames for the imager.

Direct transformations from camera RF to $[x, y]$ RF are

$$x = -f\,\frac{\hat{\boldsymbol{c}}(1)}{\hat{\boldsymbol{c}}(3)} \qquad \text{and} \qquad y = -f\,\frac{\hat{\boldsymbol{c}}(2)}{\hat{\boldsymbol{c}}(3)} \tag{A.1}$$

where coordinates, x and y, and focal length, f, are provided in mm. The inverse transformation is

$$\hat{\boldsymbol{c}} = \frac{1}{\sqrt{x^2 + y^2 + f^2}} \begin{Bmatrix} -x \\ -y \\ f \end{Bmatrix} \tag{A.2}$$

The coordinates $[x, y]$ can be transformed in pixel coordinates $[r, c]$ (row, column) by

$$r = \frac{n_r + 1}{2} - \frac{y}{d_r} \qquad \text{and} \qquad c = \frac{n_c + 1}{2} + \frac{x}{d_c} \tag{A.3}$$

where d_r and d_c are row and column dimensions of a pixel [mm]. The inverse transformations are

$$x = d_c \left(c - \frac{n_c + 1}{2} \right) \qquad \text{and} \qquad y = d_r \left(\frac{n_r + 1}{2} - r \right) \tag{A.4}$$

In addition to the previous RFs a new RF is introduced that is useful to estimate the orientation of observed ellipses. The axes of this RF are as shown in Figure A.2 and the transformations are

$$\begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} = \begin{bmatrix} \cos\vartheta & -\sin\vartheta \\ -\sin\vartheta & -\cos\vartheta \end{bmatrix} \begin{Bmatrix} c - c_0 \\ r - r_0 \end{Bmatrix} \tag{A.5}$$

and

$$\begin{Bmatrix} c \\ r \end{Bmatrix} = \begin{Bmatrix} c_0 \\ r_0 \end{Bmatrix} + \begin{bmatrix} \cos\vartheta & -\sin\vartheta \\ -\sin\vartheta & -\cos\vartheta \end{bmatrix} \begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} \tag{A.6}$$