POLITECNICO
MILANO 1863

School of Industrial and Information Engineering
Department of Aerospace Science and Technology
Space Engineering
Politecnico di Milano, Milan, Italy

DEVELOPMENT AND TEST OF AN ALGORITHM FOR DETECTION AND POST-PROCESSING FOR LUMIO MISSION

Master of Science Thesis of:
Andrea Degli Innocenti
ID: 899437

Supervisor:
Dr. Francesco Topputo

Academic Year 2018/19

Andrea Degli Innocenti: *Development and test of an algorithm for detection and post-processing for LUMIO Mission*

Supervisor:
Dr. Francesco Topputo

Location:
Milano, Italy

Time Frame:
2019

A Mamma, Papà,
Giulia ed Elena.

Abstract

The LUMIO mission, presented as response to the fourth edition of the SysNova competition organized by ESA, proposes to insert a CubeSat in a halo orbit near the L_2 Lagrangian Point of the Earth-Moon system. The purpose of this mission is to observe, quantify and characterizes the impacts of meteoroids that collide with the Moon Far-Side. The process is performed through an optical instrument, thanks to which it is possible the detect flashes generated by the impacts. For this reason it is necessary to develop a methodology able to detect and quantify the impacts that are captured by the optical instrument and process the images in order to obtain scientific information about the phenomenon.

The methodology developed for this work is divided into an algorithm able to find the flashes within the acquired image and to produce as output data about the intensity and position, and by an algorithm that acquires these information previously produced, returning physical data about the impact; those are developed according to the most recent model concerning the LUMIO optical camera, called LUMIO-Cam.

To guarantee the consistency of the detection with the environment captured by the optical camera e to improve the reliability of the information produced, some check have been inserted inside the algorithm used to detect the flash; the number of check is however limited due to the constraints about the execution time (less than 66 ms).

A further restriction imposed by LUMIO team concerns the amount of information to save; for this reason data referred to impact detected consecutively are merged, if they are classified as consistent, improving the post-processing analysis. This, putting together data coming from the detection together with the ones from the telemetry, evaluates the temperature, the area of the impact and the magnitude of the flash.

The development, validation and simulation of the performances of the methodology are conducted through MATLAB®, using synthetic images generated through POV-Ray that faithfully reproduce the images that will be acquired by LUMIO.

The simulation environment uses models of asteroids and impacts of those with other celestial bodies, so that it is coherent with the one LUMIO will be subjected to during its operational life.

The success rate for the only part of the detection found during the validation phase is just below the 98%, which can be considered as a positive result, since many non-detection are due to non-feasible flash intensity. Furthermore the execution time in within the constraints imposed bu LUMIO-Cam.

Regarding the result produced by the simulation, the correct detection is close to 100% an the information produces are coherent with the data used as input, meeting the expectations.

Sommario

La missione LUMIO, presentata come risposta alla quarta edizione della competizione SysNova organizzata dall'ESA, propone di inserire un CubeSat in un'orbita halo in prossimità del Punto Lagrangiano L_2 del sistema Terra-Luna. Il fine di questa missione è di osservare, quantificare e caratterizzare gli impatti di meteoroidi che colpiscono il lato nascosto della Luna. Il processo è eseguito attraverso uno strumento ottico, grazie al quale è possibile il rilevamento di bagliori luminosi generati dagli impatti. Per questo motivo si rende necessario lo sviluppo di una metodologia capace di rilevare e quantificare gli impatti che vengono catturati dallo strumento ottico e processare le immagini al fine di ottenere informazioni scientifiche sul fenomeno

La metodologia sviluppata per questo lavoro è suddivisa in un algoritmo capace di trovare il bagliore luminoso all'interno dell'immagine acquisita e di produrre dati riguardo l'intensità e la posizione e in un algoritmo che acquisisce le informazioni prodotte precedentemente, restituendo i dati fisici relativi all'impatto; questi sono sviluppati in accordo con il più recente modello riguardante la camera ottica di LUMIO, chiamata LUMIO-Cam.

Per garantire la consistenza della rilevazione con l'ambiente catturato dalla camera ottica e per aumentare l'affidabilità delle informazioni prodotte, sono stati inseriti dei controlli all'interno dell'algoritmo per rilevare il bagliore luminoso; il numero di controlli è però limitato in quanto vi sono vincoli riguardo il tempo di esecuzione (inferiore a 66 ms).

Un'ulteriore restrizione imposta dal team di LUMIO riguarda la quantità di informazioni da salvare; per questa ragione i dati riferiti a impatti rilevati consecutivamente sono uniti, se classificati come consistenti, migliorando così l'analisi durante la post produzione. Quest'ultima unendo i dati dalla rilevazione con quelli della telemetria valuta la temperatura, l'area dell'impatto e la magnitudine del bagliore luminoso.

Lo sviluppo, la validazione e la simulazione delle prestazioni della metodologia sono eseguiti attraverso MATLAB®, utilizzando immagini sintetiche generate attraverso POV-Ray che riproducono fedelmente le immagini che saranno acquisite da LUMIO.

L'ambiente di simulazione comprende modelli riguardanti gli asteroidi e gli impatti di essi con altri corpi celesti, in modo da essere coerente con quello a cui sarà sottoposto LUMIO durante la sua vita operativa.

Il tasso di successo riscontrato durante la fase di validazione per la sola parte di rilevamento si attesta poco al di sotto del 98%, il quale si può considerare un risultato positivo, in quanto molte non rilevazioni sono riconducibili a intensità di bagliori non fisiche. Inoltre il tempo di esecuzione è all'interno dei limiti imposti da LUMIO-Cam.

Per quanto riguarda il risultato prodotto della simulazione, la corretta rilevazione si avvicina al 100% e le informazioni prodotte sono coerenti con i dati inseriti, rispecchiando le aspettative.

Acknowledgements

Voglio innanzitutto ringraziare il mio relatore Dr. Francesco Topputo per avermi dato l'opportunità di lavorare ad un argomento così stimolante ed interessante, per la sua disponibilità e per i suoi preziosi consigli.

Un grazie particolare ai miei genitori che mi hanno dato la possibilità di intraprendere questo percorso, sostenendomi quotidianamente con la vicinanza che solo loro sanno dare. Grazie alle mie sorelle Giulia e Elena, per avermi sopportato e supportato in questi anni.

Grazie ai miei amici di sempre, per i momenti seri e per quelli di distrazione dallo studio e dagli esami con una buona birra, un giretto, con una partita a calcetto o semplicemente quattro chiacchiere.

Grazie a Matteo, Nicolò, Andrea, Fudu, al so Affi, al bomber Enrico, a Zighi, Riki e Martina e gli altri miei compagni conosciuti qui al Politecnico in questi 5 anni, con i quali ho condiviso esami, progetti e intere, lunghissime giornate nella bellissima Bovisa; grazie a voi le fatiche e la stanchezza sono state allietate. Grazie in particolare alla mia amica Alessandra per la sua pazienza, soprattutto nella lettura...

Grazie infine al Politecnico per avermi insegnato a non accontentarmi ma anche ad apprezzare i risultati di un lavoro fatto con fatica e impegno.

Grazie,
Andrea

Contents

List of Figures

List of Tables

List of Acronyms

A/D Analogue to Digital. 19–21, 24, 28, 48, 86

CCD Charged-Coupled Device. 6, 14, 17, 19, 21, 24, 28, 30, 65, 74
CDF Concurrent Design Facility. 1, 9, 11, 14, 16, 17, 19, 22, 85
CPU Central Processing Unit. 22

EMCCD Electron Multiplying Charged-Coupled Device. 19, 20
EOL End of Life. 13, 16
ESA European Space Agency. 1, 6, 9, 11–13, 27, 85

FL Focal Length. 19
FOV Field of View. 16, 19, 29, 65
FPGA Floating Point Gate Array. 22
FPS Frames Per Second. 14, 22

GCR Galactic Cosmic Rays. 6, 17
GPU Graphics Processing Unit. 22

JAXA Japan Aerospace Exploration Agency. 6
JPL Jet Propulsion Laboratory. 19

LUCE Lunar Cubesat for Exploration. 11
LUMIO Lunar Meteoroid Impacts Observer. xi, xvii, 1, 2, 4, 6–9, 11–13, 16–18, 21–23, 28, 37, 40, 42, 46, 48, 50, 54, 59, 66, 71, 75, 79, 85, 86

MIDAS Moon Impacts Detection and Analysis System. 6, 7

NASA National Aeronautics and Space Administration. 5, 7
NEA Near-Earth Asteroid. 2, 3
NEC Near-Earth Comet. 2
NEO Near-Earth Object. 1, 2, 10, 62
NIR Near-Infrared Spectrum channel. 14, 17–19, 37, 38, 59, 60, 69, 71, 74, 75, 79, 82

OBC On-Board Computer. 22
OBPDP On-Board Payload Data Processing System. 22

P/L Payload. 6, 12

QE Quantum Efficiency. 19

S/C Spacecraft. 1, 12, 16, 29, 40, 65, 66, 71, 75, 79, 82, 85
SADA Solar Array Drive Assembly. 16
SNR Signal to Noise Ratio. 15, 28, 64
SOI Sphere of Influence. 3, 4

TEC-SF Directorate of Technology, Engineering and Quality. 11
TRL Technology Readiness Level. 19

VIS Visible Spectrum channel. 14, 17–19, 37, 38, 59, 60, 69, 71, 74, 75, 79, 82

List of Symbols

A_p Plume area. 37, 38, 59, 60, 63, 64, 67, 69, 70, 72, 76, 80

B_t Background function at time t. 27, 28, 40, 42

B_{t-1} Background function at time $t-1$. 27, 40, 42

C_x Centre of the image reference freme (X-coordinate) [for LUMIO=512]. 70

C_y Centre of the image reference freme (Y-coordinate) [for LUMIO=512]. 70

C_{max} Maximum charge handling capacity. 20, 38, 59, 65, 70

D Aperture diameter. 19, 37, 38, 59, 64, 69

D Crater diameter. 63, 67, 72, 76, 80

D_t Delta image at time t. 28, 42

E_r Radiated energy. 5

I_t Acquired image at time t. 27, 28, 42

I_{t-1} Acquired image at time $t-1$. 27, 40

$L(\lambda, T)$ Spectral emissive power. 37, 38

N Pixel value. 67, 72, 75, 76, 80

N_{NIR} Pixel value retrieved by the image in NIR channel. 59, 70, 71, 75, 79, 82

N_{VIS} Pixel value retrieved by the image in VIS channel. 59, 69, 70, 74, 79, 82

$N_{e^-}(\lambda, T)$ Flux of electron per unit time per wavelength. 37, 38, 59

N_{max} Maximum quantizied value ($2^{N_{bit}}$). 38, 59, 65, 70

P Radiated power. 38

P_{imp} Estimated power released. 39, 61, 70

Q Power per unit area. 5

$QE(\lambda)$ Quantum efficiency. 38, 60, 70

R Body radius. 4, 64

$R_{\leftmoon}$ Moon's radius. 19, 62, 70, 89

R_{eq} Equivalent radius. 36

S_{imp} Impact signal. 37, 38, 59, 64, 65, 69, 70, 75

S_{imp}^{VIS} Signal in VIS bandwidth. 59, 70, 71, 75, 79, 82

S_{imp}^{NIR} Signal in NIR bandwidth. 59, 70, 71, 75, 79, 82

T_{imp} Impact temperature. 59, 64, 67, 72, 76, 80

V Geocentric velocity (velocity at SOI). 3, 4

V_{imp} Impact velocity on the Moon surface. 62, 63, 67, 72, 76, 80

X_f X-coordinate of the flash in the image. 55, 67, 72, 76, 80

Y_f Y-coordinate of the flash in the image. 55, 67, 72, 76, 80

$\Delta\lambda$ Wavelength beam-width. 38, 39

α Weighting coefficient for the background function. 27, 40, 41

η Luminous efficiency. 5, 63, 64, 67, 72, 76, 80

γ Impact angle with respect to the horizon. 63, 64, 67, 72, 76, 80

$\hat{A}_p$ Estimated plume area. 60, 61, 70, 71, 75, 79, 82

$\hat{C}$ Estimated flash column in pixels matrix. 69, 74, 79, 82

$\hat{N}$ Pixel value retrieved by the image. 38, 59

$\hat{R}$ Estimated flash row in pixels matrix. 69, 74, 79, 82

$\hat{R}_{\leftmoon}$ Estimated Moon radius from the image. 31, 36, 69, 70, 74, 79, 82

$\hat{S}_{imp}$ Estimated impact signal. 38, 59

$\hat{T}_{imp}$ Estimated impact temperature. 38, 59–61, 70, 71, 75, 79, 82

$\hat{X}_{c\leftmoon}$ Estimated Moon centre X-coordinate from the image. 31, 35, 55, 69, 74, 79, 82

$\hat{X}_f$ Estimated X-coordinate of the flash in the image. 35, 70

$\hat{Y}_{c\leftmoon}$ Estimated Moon centre Y-coordinate from the image. 31, 35, 55, 69, 74, 79, 82

$\hat{Y}_f$ Estimated Y-coordinate of the flash in the image. 35, 70

λ Wavelength. 37, 38, 60, 61, 70

λ_f Longitude of the flash. 35, 36, 55, 58, 69, 74, 79, 82

μ Gravitational mass parameter. 4

$\mu_{\oplus}$ Moon's gravitational mass parameter. 62, 89

$\mu_{\leftmoon}$ Estimated flash column in pixels matrix. 62, 89

μ_{img} Mean value of the image. 29, 46, 48

ϕ_f Latitude of the flash. 35, 36, 55, 69, 74, 79, 82

$\rho_{\leftmoon}$ Moon bulk density. 63

ρ_i Impactor bulk density. 63, 64, 67, 72, 76, 80

σ Stefan-Boltzmann's constant. 5

σ_{img} Maximum value of standard deviation of the image. 29, 46, 48

τ Estimated impact duration from the image. 38, 59, 60, 69, 74, 79, 82

$\tilde{R}$ Fictitious radius. 4

ϑ Angle between LUMIO-Cam boresight direction and impact vector. 37, 38, 59, 64, 66, 69–71, 74, 75, 79, 82

ξ Optics lenses reduction factor. 20, 38, 60, 70

c Light speed in vacuum. 37, 38, 60, 61, 70, 89

d Distance from the Moon. 37–39, 59, 61, 64, 66, 69–71, 75, 79

$d_{\oplus\leftmoon}$ Reference distance between Earth and Moon. 62, 63, 89

$d_{\mathbf{CCD}}$ CCD side length. 19

d_{min} Minimum distance from the Moon surface. 19

f Anisotropy degree of light emission. 37–39, 59, 61, 64, 69, 70

f_{ld} Degradation coefficient due to light diffraction. 37, 38, 59, 64, 69

g Standard acceleration of gravity. 63, 64, 89

h Plank's constant. 37, 38, 60, 61, 70, 89

k_B Boltzmann's constant. 37, 61, 89

m Apparent magnitude. 38, 64

m Meteoroid mass. 63, 67, 72, 76, 80

m_1 Angular coefficient of the first secant. 31

m_2 Angular coefficient of the second chord. 31

$m_\oplus$ Apparent magnitude at Earth distance. 67, 72, 76, 80

$m_{\perp 1}$ Angular coefficient of the perpendicular to the first chord. 31

$m_{\perp 2}$ Angular coefficient of the perpendicular to the second chord. 31

m_{imp} Estimated apparent magnitude at LUMIO distance. 39, 61, 70, 71, 75, 79, 82

KE Kinetic energy. 5, 63, 64, 67, 72, 76, 80

t Impact duration. 37, 38, 59, 60, 64, 67, 69, 72, 75, 76, 80

f f number. 19

1

Introduction

1.1 Context

Mapping the objects near Earth, called Near-Earth Objects (NEOs), could improve either the knowledge of the Solar System, since formation of those objects goes back to the formation of the Solar System itself, either the awareness of meteoroid showers for future human missions in space. Indeed impact of these objects with Spacecrafts (S/Cs) or astronauts would lead to problems for the mission or to the failure in the worst of the cases. First studies about this topic were done by using Earth atmosphere as detector, by looking at those meteor that entering in it bright up due to drag, giving rise to the well known tail; by the way these kind of measurements are limited by many factors, as conditions of the atmosphere's illumination, the weather or the available portion of sky to be observed. An evolution of this investigation has been developed recently, at the beginning of the new millennium, and consists on detecting the presence of a meteoroid in an indirect way, by looking at some flashes on the Moon surface generated by the impact; thanks to this solution it is possible to look at a wider area, the whole lunar Near-Side, than the single portion of sky upper the observatory, collecting more information. First confirmed data about this research date back to 1999 (Suggs et al., 2014), and are carried out by looking at the Moon's Near-Side using optical telescopes; in this way it is possible to characterize relatively small meteoroids (up to 1 kg), whose flux density is low and few data are available. However, also these observations are subjected to limitation due to weather, Moon-Earth-Sun geometry and illuminations conditions. A further evolution for this research field is to use orbiting satellite around Moon to detect impact flashes, reducing a lot the distance from the Moon and getting rid of many limitation, like weather and inappropriate geometries. In this context LUMIO has been developed, as response at the fourth edition of the SYSNova challenge. Phase-*0* design has been concluded (Topputo et al., 2017), as well as the Concurrent Design Facility (CDF) (Walker et al., 2018) performed by the European Space Agency (ESA).

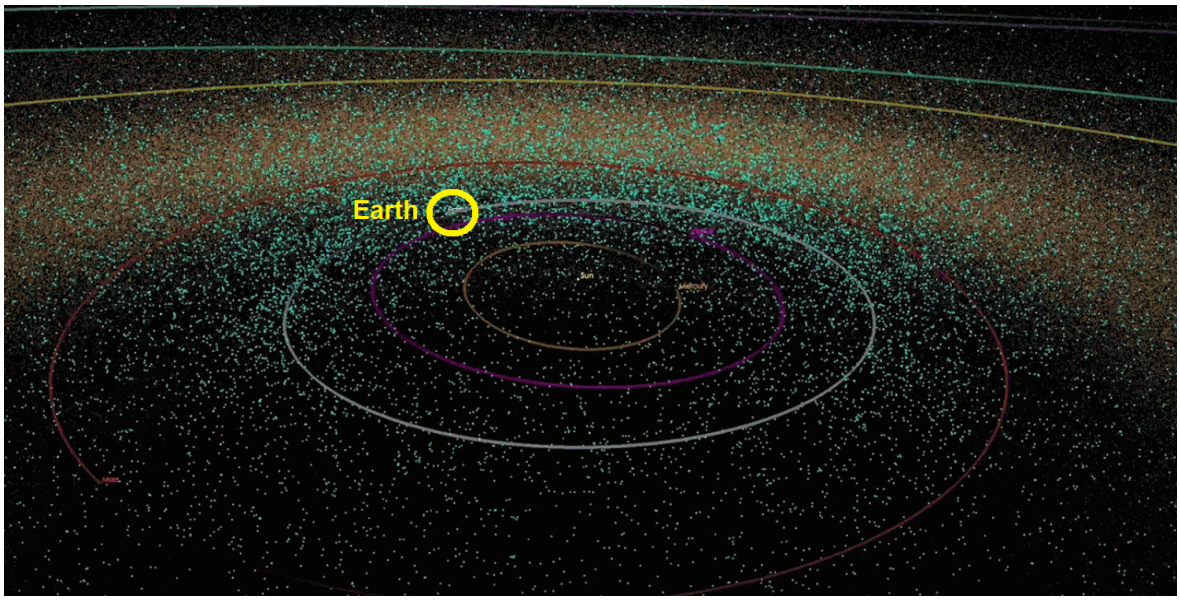


Figure 1.1: *Distribution of NEOs around Earth. This simulation shows the orbit of the Earth, circled in yellow, together with the NEOs around. Credits to:* `https://www.jpl.nasa.gov/news/news.php?feature=7194` [Last visited on 08/11/2019].

1.2 State of the art

It is here presented the meteoroid environment under study understanding what these meteoroids are and from where they come from, and in particular which models are used to simulate their behavior. Then, according to the detection method used in LUMIO mission, an overview of how the impact dynamics evolves and is modeled is performed. Furthermore some missions and programs with aim similar to LUMIO are presented and discussed. In the end an overview about the possible methods to have detection is performed.

1.2.1 Meteoroid environment

NEOs are asteroids or comets whose perihelion is at a distance less than 1.3 AU, whose orbit encounters the Earth's one. These objects are mainly composed by Near-Earth Asteroids (NEAs) and Near-Earth Comets (NECs), and are important to be studied since their impact on Earth could lead to humanitarian crisis, or potentially the extinction of human being. In Figure 1.1 the NEOs are shown in relation to the Earth orbit, while in Figure 1.2 there is an estimation of the discovered and mapped bodies up to now.

Meteoroids are the results of asteroids or comets fragmentation and their size and energy depends both on the body form which they are originated and both from the fragmentation process itself. Once these new bodies are formed, they remains in a cluster near the parent one for thousands of years, as long as their path is not perturbed. When these bodies encounter the Earth and enter the atmosphere, they give rise to phenomena called meteors, which are bright sources striking the sky that leave a trail on their path. If the size of the orbiting meteoroid is enough to permit the meteor to pass through all the atmosphere and arrive at ground, the fragments gives rise to a meteorites, that can be collected and studied. These kind of studies are useful to study the formation of the Solar System, being these object formed at that time, keeping unaltered their environment. Atmosphere therefore protects the Earth

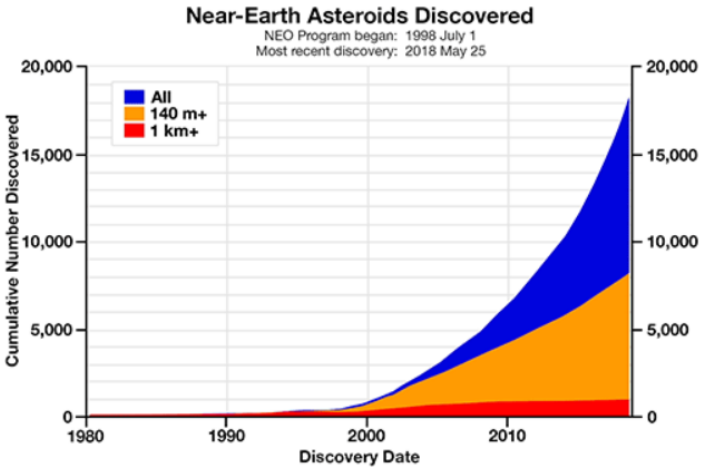


Figure 1.2: *Cumulative distribution of discovered NEAs. Are here shown the discovered bodies up to May 2018; in particular it is shown haw the vast majority of bodies orbiting around the Earth are smaller than a* km. *Credits to:* `https://www.jpl.nasa.gov/news/news.php?feature=7194` [Last visited on 08/11/2019].

soil from the small meteoroids and reduce both the mass and the energy of the greater ones, even if this is not sufficient to avoid potential disasters. Indeed, Earth history has been conditioned by many impacts, whose effects are still visible today. It is supposed that one of them led to the extinction of the dinosaurs, while other huge impacts craters are present in many parts of the world. In 2013, a small meteoroid has impact the Earth at Chelyabinsk, in Russia (Brown et al., 2013), exploding in the sky generating a large airburst shock wave, creating several meteorites, result of the original body fragmentation after the explosion, then founded on ground. Although the meteor did not arrived to the ground since its energy was absorbed by atmosphere, the main damages were due the airburst shock wave generated (Brown et al., 2013).

On the Moon, the absence of an atmosphere allows all the incident meteoroids to impact on the surface without significant energy losses. With noways technologies it is possible to observe these phenomena in different ways, exploiting different physical evidences (Oberst et al., 2012). Result obtained by these observations can be exploited also for the Earth environment. Meteoroids phenomena are usually divided into two different families, one collecting the ones coming from the so called sporadic background and the other from those belonging to meteoroid showers. The latter are phenomena which repeats yearly. Moreover, meteoroids from the same shower have similar geocentric velocity V, which is the entry velocity at Sphere of Influence (SOI); in addition, they show the same radiant direction, meaning that they seem to originate from the same point in the sky.

On the contrary, meteoroids from sporadic background do not show and activity period, have variable geocentric velocities, and do not show a preferential radiant direction.

The velocity at the SOI, typically indicated in the literature as geocentric velocity, ranges between 0 $\mathrm{km\,s^{-1}}$ and a maximum one, which is obtained considering an impactor in a retrograde, highly eccentric orbit around the Sun, when Earth is at perihelion. Impact velocity on the Moon should range between 10.2 $\mathrm{km\,s^{-1}}$ and 73.8 $\mathrm{km\,s^{-1}}$ (Cipriano, 2017). Indeed the impact velocity depends of both

the Heliocentric velocity the meteoroid has, both on the body it is colliding with, according to:

$$V_{imp} = \sqrt{V^2 + 2\frac{\mu}{R}} \tag{1.1}$$

where V is the velocity at the SOI, μ the gravitational mass parameter and R the radius of the body. Although the Moon and Earth environment are similar the are not identical, since Moon and Earth have different masses and therefore they exert different gravitation attraction on meteoroids; moreover they are placed in different points with respect to them, since Moon is orbiting around Earth, and furthermore these one has an atmosphere that is not present on the Moon. Indeed the higher is the mass a body has, the higher is the gravitational attraction it has on the environment. In theory an impact should happen only if the trajectory of the meteoroid would intersect the Earth or Moon path; however, due to the gravitational attraction, that for massive bodies is not negligible, it is possible to define a phenomenon called gravitation focusing, namely an effect due to the gravity thanks to which it is possible to define a fictitious radius $\tilde{R}$, that is bigger than the actual one, that indicates the actual area of probable intersection, defined as:

$$\tilde{R} = R\sqrt{1 + 2\frac{\mu}{V^2 R}} \tag{1.2}$$

Performing the calculation it comes out that this effect is negligible for Moon, while it is not for Earth (Oberst et al., 2012).

Starting from these information, in Merisio (2018) a more in depth analysis is performed directly on the simulation of the environment to which Moon will be subjected during LUMIO mission. This work confirms all the data present in the literature, presenting also a methodology to estimate a consistent random environment for the simulation of showers.

1.2.2 Impact dynamic

When a meteoroid impact with the surface of the Moon, the kinetic energy the body has, is converted into others forms of energy. This conversion generates a seismic wave, excavates a crater, accelerate ejecta particles from the ground and releases radiation, though which it is possible to study the impact. The trade off analysis for LUMIO mission is performed by Cipriano (2017) and it is also present in Topputo et al. (2017), while full coverage of the possible detection methods can be found in Oberst et al. (2012). As the crash happens between two solid objects, a seismic wave is produces as propagation on the impact in the ground; at the same time a crater is left on the impact location on the surface, while the particles of ground that were in that locus are burst and thrown away and at the same time the a plume of ejecta is generated. This last phenomenon is the selected one for LUMIO mission through which the detection will be performed, and this section will be dedicated to explain how to model this event. The impact of a meteoroid with the ground generates an hot ejecta plume Figure 1.3 that rises from the surface emitting electromagnetic radiation, to which is typically associated a black body temperature; the duration of this event can last up to tens of ms, accordingly to the kinetic energy and mass of the impact body. In general the power is not constant during all the duration of the event, since in the first part the ejecta just created are hot while, during this time, the plume tends to cool down. The reduction of the power influences directly the black body temperature since:

$$Q = \sigma T^4 \tag{1.3}$$

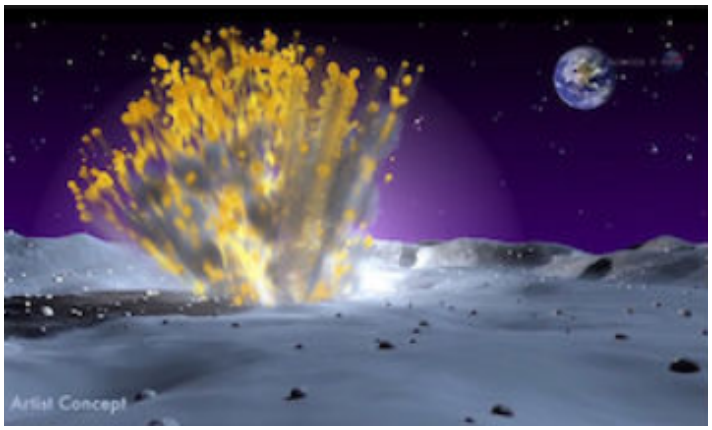


Figure 1.3: *Artistic representation of the impact dynamic on the Moon surface. It is in particular emphasized the radiating plume that is rising. Credits to:* `https://neliota.astro.noa.gr/About/Project` [Last visited on 08/11/2019].

where Q is the power per unit area emitted and σ the Stefan-Boltzmann's constant, and as a consequence also the temperature decreases. This phenomenon is widely used for the detection of these impacts from the Earth by using telescopes looking at the Moon's Near-Side surface. By the way, once the impact is found, it is not easy to retrieve information about the body originate it, since impact dynamics is influenced by many factors, like velocity, angle of impact, mass, initial kinetic energy that are not easy to model. Indeed, while for meteors from Earth it is possible to compute the velocity vector analyzing the passage in two points, (Madiedo et al., 2015a), it is impossible to extract the velocity vector from a single frame containing the impact flash. Nevertheless in this way it is possible to obtain the impact location and estimate with a certain degree of uncertainty the meteoroid shower to which the impactor belongs and associate to the impact the shower characteristic velocity. More information about source determination methods are present in Madiedo et al. (2015b) and Suggs et al. (2014)

An important quantity that is vital to model in the correct way the impact dynamics is the luminous efficiency η. Luminous efficiency is a parameter that is fundamental when trying to obtain in the aftermath the impact kinetic energy. This parameter is the ratio between radiated energy in the visible spectrum E_r and impact kinetic energy KE. Formally:

$$\eta = \frac{E_r}{\mathrm{KE}} \tag{1.4}$$

The issue is that η strongly depends on impact velocity and weakly on other quantities, such as impactor mass, impact angle, meteoroid physical properties, and target physical properties. Many correlations, available in Bellot Rubio et al. (2000) and Swift et al. (2011) have been built linking the luminous efficiency to the impact velocity.

1.2.3 Observation

From the beginning of the observation of the Moon surface to detect impact flashes, several programs have been started, all using Earth-based. In particular in 2006 National Aeronautics and Space Administration (NASA) begin its activity at Marshall Space Flight Center in Alabama[1]. In 2009

[1] `https://www.nasa.gov/centers/marshall/news/lunar/overview.html` [Last visited on 08/11/2019]

J.M. Madiedo begun the Moon Impacts Detection and Analysis System (MIDAS) project in Spain. The most recent one is from ESA, NELIOTA, started in February 2015 at National Observatory of Athens[1]. All these projects have in common the fact that their science is carried out by collecting data by pointing and recording images of the Night-Side of the Moon Near-Side, since otherwise the flash luminosity is covered by the Moon brightness. In particular NELIOTA monitors lunar impact flashes exploiting a novel technique in which the equivalent black body temperature of radiating plume is obtained by means of two independent measurements in different bandwidths of the spectrum; detection are simultaneous, the first is in the visible spectrum while the second in the near-infrared (Bonanos et al., 2018).

This work is not trivial since the image so obtained is full of noise, mainly due to the brightness of the Moon, the Earth atmosphere and the so called Earthshine, namely the illumination arriving from the Sun that the Earth reflects on the Moon. For this reason, in certain conditions, it its not possible to have science; in particular, for Moon illumination higher than 60%, the monitoring is not feasible (Ortiz et al., 2006).

Moreover Earth-based observatories, have to deal also with weather, indeed, if sky is covered, also partially, by clouds, it is hard to detect bodies impacting the Moon.

A first try to detect lunar impact flashes from space will be done by Japan Aerospace Exploration Agency (JAXA) with a 6UCubeSat, named EQUULEUS. Originally, the launch was scheduled in June 2018, but it has been delayed in December 2019. EQUULEUS will be carried by Exploration Mission-1 (Campagnola et al., 2016).

Typically, Earth-based installations employ telescopes to collect light which mount Charged-Coupled Device (CCD) detectors. In particular, when a flash is observed from Earth, the atmosphere and the weather are a big issue, since they reduce the signal quality. Doing the same operation in space it is possible to prevent these phenomena, allowing the detection in every condition; on the other side this gives rise to some issues typical of the space remote sensing, like diffraction in the optics or false signals due to Galactic Cosmic Rays (GCR). In LUMIO mission Phase-*0* design technical report (Topputo et al., 2017) those issues are faced and a proper design for the Payload (P/L) is proposed.

1.2.4 Processing methods

The problem of retrieving information from an image is a widely covered in literature; in particular it is possible to identify two families of algorithms for the detection of finite objects; one called object based and one called pixel based (Hussain et al., 2013; Lu et al., 2004). While the first one tries to map a shape and tries to detect if in the image there is a shape similar to the reference one in order to highlight it from the background, with the second approach the analysis of the content of the image is done by analyzing each pixel the picture is composed by. For this work is considered the pixel based approach, since, according to the requirements of this work, it is the better choice to proceed; furthermore it is not possible to have predict a priori nor the shape nor the size of the expected pixel, and moreover it is not present in every frame, leading to a high probability of false positive since whatever could be considered as a flash. For this reason is here reported more in depth a focus on the categories the pixel based approach is divide into. Both Lu et al. (2004) and Hussain et al. (2013) identifies different categories, which differs each other by the way the image is modified to get the

[1] `https://neliota.astro.noa.gr/` [Last visited on 08/11/2019]

final result; in these references can be found a deep study for each method. Among these categories are present some regarding the direct comparison techniques, whose aim is just to compare images among them in form of matrices, other regarding transformation techniques, that have the detection by looking at the information that can be retrieved by the image via a transformation by applying filters, others regarding machine learning and others that use both the vegetation or the geography to have a proper detection. Comparing the list of these methods with the ones used in NELIOTA, MIDAS and NASA, it comes from that the most used method is to differentiate the acquire image with a reference one and have detection from the result. However all of these programs, once the algorithm has done the detection, the result is always checked by an operator, that confirms or not the accuracy of the software. Another feature in common of the programs stated before, is that all collects videos of the event that is then processed; this allows to have more accuracy on the scientific data due to the high frame rate. In particular at NASA Marshall Space Flight Center the detection is performed by looking at data coming from the Night-Side of the Moon's Near-Side only, since otherwise the amount of noise would cover the signal, by using an adaptive threshold that evaluating the mean value and the standard deviation of the image, check is some pixels are over that value and if so they are counted as flashes (Suggs et al., 2008). In MIDAS software the recorded images are first of all pre-processed in order to reduce the noise that would lead to false detection. A mask that eliminates the parts of the image that do not want to be investigated is placed, and therefore the only detected part is the Night-Side of the Moon, like done by NASA. In the pre-processing are applied filters to mitigate the presence of the atmosphere and of the Earthshine. After that a Moon calibration is performed to pass from the image reference frame to the selenocentric one. The detection is performed in by the software itself, by checking if the pixel intensity is over a user's defined threshold and if so are stored; the threshold is set by the user. Once the detection has succeeded, and the event is confirmed the output is a photometric analysis (Madiedo et al., 2015a). NELIOTA data are first of all calibrated to remove all the noise in the signal and then the detection is performed by setting a weighted in time background function, connected to the Moon phase and illumination, that subtracted to the image created a difference image to which are applied two thresholds that detect the flash. This process is overall automatic, but a final visual check is performed to validate it. (Xilouris et al., 2018; Liakos et al., 2019)

In NELIOTA program after the result from the detection is obtained, a further analysis is performed by virtue of their two channels detection. Indeed exploiting the knowledge of the power irradiated by the source, in this case the flash, and knowing the bandwidth the detector is operating it is possible to recover the black body temperature of the flash by applying the Plank's Law. (Bonanos et al., 2018). A real image from NELIOTA website is shown in Figure 1.4.

A huge difference that has to be taken into account is that all these projects are carried out on ground and therefore both the computational effort and the storage availability are not a problem. For LUMIO mission this is a crucial point since both the computational effort and the memory size are limited and this is a massive constraint to be faced. This because the capability of the computer on-board the spacecraft could not be comparable to the one present on ground and at the same time the memory is limited to the size of the satellite and the downlink of the data cannot be performed together with the science activity since the required power is not feasible.

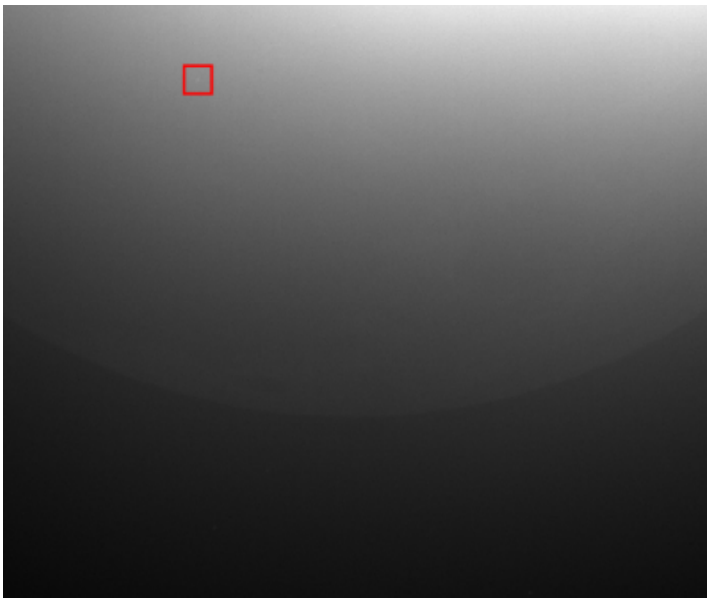


Figure 1.4: *Real image of the detection made by NELIOTA project of an event happened on 03/11/2019. It is clearly visible the high amount of noise present in the frame, mainly due to the Earthshine. The detected impact is highlighted by the red rectangle. Credits to:* `https://neliota.astro.noa.gr/DataAccess/EventDetails/210` [Last visited on 08/11/2019].

1.3 Motivation

Taking into account all the disturbances that affect the detection on Earth, it follows that carrying out the detection of impact flash on the Moon surface in orbit gives back more faithful data than doing the same on Earth. Moreover, according to LUMIO final report (Topputo et al., 2017), the satellite will be placed in an halo orbit around the L_2 Lagrangian Point in the Moon-Earth system, looking the Far-Side; in this way the information obtained can be used as dual to the ones find on the Earth, since the rotation motion of the Moon is equal to its revolution motion around the Earth and therefore the Moon shows always the same face to Earth; at the same time the quality of the measurement is far better than the one obtained on Earth since it is possible to get rid of many of the disturbances sources present for the ground observations. Indeed when we deal with remote sensing around the Moon, all the problems due to the atmosphere con be neglected, and furthermore, being in the Far-Side allows not to consider also Earthshine. Doing so, two of the main sources of noise are neglected, and therefore the output is more accurate. Moreover, the absence of the atmosphere makes no possible the formation of clouds thus the surface is always visible. Another feature characteristic of this situation is that there is not the alternation between day and night; indeed on Earth this kind of science can be performed only during night since during day the atmosphere is illuminated by the Sun and the level of noise if far higher than the signal, while on the Moon this don't happens, and the analysis of the surface can be performed continuously in the science window. This last consideration is particularly useful for this kind of investigation, since not being possible to predict a flash, doing a continuous sensing, allows to maximize the scientific output. By the way, together with these pros, some issues have to be faced, since the fact of being on a satellite introduces some constraints on both the computational effort and on the storage capability. Furthermore, the amount of data taken by performing continuous science will saturate memory in few days, introducing another constraint. In this context is inserted this work;

indeed it is not possible to simply take an already working procedure and to deploy it on LUMIO, since it will likely overcome both the computational effort and the storage. It is important to find an optimized procedure able to maximize the scientific output and minimize the false positive results that would waste memory, but at the same time the algorithm should be fast enough to be able to discover if a flash has been found in the frame in a time compatible with the shooting time.

1.4 Research question

In LUMIO final report (Topputo et al., 2017), a very brief analysis of the algorithm is performed and it is stated that a further analysis has to be carried. For this reason, also in the CDF Study Report performed by ESA (Walker et al., 2018), this topic is not covered. The need of an ad-hoc algorithm for the management of the scientific payload comes from the novelty of the mission itself, and from the challenges to face in situations like the one this mission is facing, that on ground can be easily overcome. In particular in this work the focus in put on the method to have the detection and to post-process the data obtained in order to save storage by collecting only the useful data.

The research question and objective are the following:

Research Question	Given the position, the time duration and the intensity of the impact: What is a method to collect and post-process the data with the higher accuracy and compatible computational efforts?
Research Objective	To develop a novel methodology to detect and post-process data from the analysis of the images taken by LUMIO-Cam.

1.5 Structure of the manuscript

The manuscript is organized as follows:

- Chapter 1: **Chapter 1**. Introduction to the context of the work, analysis of the environment this work is inserted in and discussion about the state of the art.
- Chapter 2: **Chapter 2**. In this chapter it is introduced LUMIO mission. At first it is presented the aim of the mission itself, focusing on the Phase-*0* design and on the CDF performed by ESA. Then a more in depth analysis of the payload is done, in particular about the cam and the processing.
- Chapter 3: **Chapter 3**. In this chapter all the steps developed for the algorithm are covered. Here are shown the flow charts representing the logical process the methodology follows and together are listed also some comments about the choices and simplifications done in order to achieve the result.
- Chapter 4: **Chapter 4**. In this chapter are validated all the methods and functions developed and used in the algorithm and listed in Chapter 3. In particular are faced all the steps performed by the algorithm, both for the detection both for the post-processing and are shown the logical processes that the algorithm follows to get the result.
- Chapter 5: **Chapter 5**. In this chapter are shown four simulations of the methodology developed; data used to perform those simulations have been generate according to the models available

regarding both the NEOs environment and the impacts dynamics. Data so generated have been used either for the detection that for the post-processing.

- Chapter 6: **Chapter 6**. This chapter contains the conclusive observation about the results of this work, together with some suggestions for future developments.

2

LUMIO Mission

In this chapter it is introduced LUMIO mission. At first it is presented the aim of the mission itself, focusing on the Phase-*0* design and on the CDF performed by ESA. Then a more in depth analysis of the payload is done, in particular about the cam and the processing.

2.1 Introduction to LUMIO Mission

LUMIO is a proposed mission presented by Politecnico di Milano (Polimi), Delft University of Technology (TU Delft), École Polytechnique Fédérale de Lausanne (EPFL), Science and Technology AS (S[&]T), Leonardo S.p.A. (Leonardo), and University of Arizona (UoA) as response to "SYSNova: R&D Studies Competition for Innovation No. 4, Lunar Cubesat for Exploration (LUCE)" (SYSNova: R&D Studies Competition for Innovation – No.4., 2016a; SYSNova: R&D Studies Competition for Innovation – No.4., 2016b). It is designed to place a 12UCubeSat on an halo orbit around Earth-Moon L_2 Lagrangian Point. Orbit design has been driven by the aim of permit a continuous full-disk observation of the Moon Far-Side. In Figure 2.1 the science period of LUMIO is shown; the aim is to have complementary information about meteoroids impacts against the Moon surface to the one that are collected on ground. Moreover, due to the shorter distance and better definition of the image the data from LUMIO are expected to increase the quality of the overall information about this phenomenon. LUMIO Phase-*0* design is concluded and can be found in Topputo et al. (2017). This project has been awarded as one of the winners of the LUCE challenge and a CDF study for the Phase-*0* design has been performed by Directorate of Technology, Engineering and Quality (TEC-SF) of ESA, reported in Walker et al. (2018); in this technical report are faced all the issues found in the Phase-*0* design, and some solutions or improvements are proposed together with some recommendations for future works, in particular for the forward step from Phase-*0* to Phase-*A*. The CDF study covers all the design fields covered in the preliminary design of the mission, namely system engineering, mission analysis,

Figure 2.1: *LUMIO science period. Solid blue line represent the period in which the S/C shall be in science mode. As can be seen this period coincides with the period without activity on Earth. Credits to:* Topputo et al. (2017).

ground segment & operations, subsystem, and cost/risk/programmatic. Hereafter are presented and discussed in particular the improvements concerning the P/L, since these variation influence a lot this work and its outcome.

2.2 Overview of LUMIO Mission

This mission is a 12UCubeSat that has to perform science around L_2 Lagrangian Point of the Earth-Moon system, through an optical instrument able to detect flashes in visible and near-infrared spectrum, generated by impacts of meteoroids on the Moon. The optical instrument is the main P/L an is called LUMIO-Cam, whose aim is to continuously collect and process data from the Moon surface. Besides the scientific aim, this mission will be a test for the novel CubeSat technology for further mission in deep space. The launch is scheduled for 2023, likely in conjunction with the Lunar Pathfinder mission. As reported in Topputo et al. (2017), the operative phase shall start on 6 September 2023, at midnight UTC and shall operate for an year, until 6 September 2024 at UTC midnight. Nevertheless, it is also reported that there are opportunities to extend the operative phase of LUMIO

The objectives LUMIO mission has to satisfy are related either to the ones stated form ESA within SYSNova: R&D Studies Competition for Innovation – No.4. (2016b), collected in Table 2.1 either to ones defined by LUMIO team, listed in Table 2.2

2.2.1 Requirements

Here is presented the list of requirements that are related to this work, concerning the Science, the Payload and the Mission requirements; the full list, comprising also the ones related to other subsystems, are available in Topputo et al. (2017). In Table 2.3 the science requirements are reported; these requirements justify the scientific aim of the mission. All these define the environment LUMIO is subjected to and are the framework in which the algorithm has to operate and the aim for which it is developed; SCI.01 is the justification of the mission from the scientific point of view. SCI.04 gives

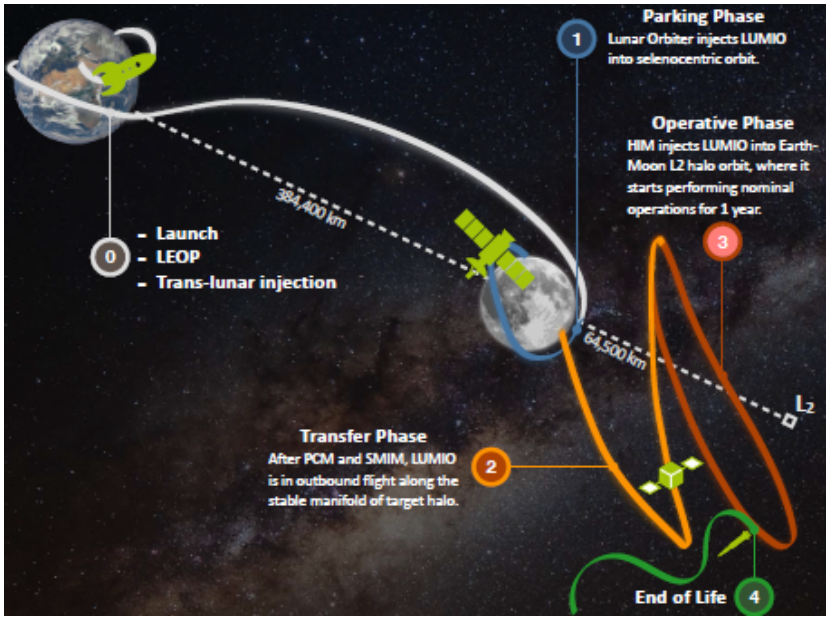


Figure 2.2: *Mission design of LUMIO. It is here shown the whole mission design from the launch from the Earth for the End of Life (EOL) decommission. It is clearly visible the L_2 Lagrangian Point of the Earth-Moon system. Credits to:* Topputo et al. (2017).

Table 2.1: *LUCE top-level mission objectives*.

ID	Objective	Stakeholder
TLO-01	To perform remote sensing of the lunar surface and measurement of astronomical observations not achievable by past, current, or planned lunar missions	ESA
TLO-02	To demonstrate deployment and autonomous operation of CubeSats in lunar environment, including localization and navigation aspects	ESA
TLO-03	To demonstrate miniaturization of optical instrumentation and associate technology in lunar environment	ESA
TLO-04	To perform inter-satellite link to a larger Lunar Communications Orbiter for relay of data and for TT&C	ESA
TLO-05	To demonstrate CubeSat trajectory control capabilities into lunar environment	ESA
TLO-06	To gain European flight heritage in emplacing and operating assets at Earth-Moon Lagrange points	ESA

Table 2.2: *LUMIO mission objectives*.

ID	Objective
MO.01	To conduct observations of the lunar surface in order to detect meteoroid impacts and characterize their flux, magnitudes, luminous energies, and sizes
MO.02	To complement observations achievable via ground-based assets in terms of space, time, and quality in order to provide a better understanding of the meteoroid environment

Table 2.3: *Science requirements*.

ID	Requirement	Parent ID
SCI.01	The mission shall discover new impacts on the Moon in the range 10^{-4} kton TNT to 10^{-1} kton TNT	MO.01
SCI.02	The mission shall refine cumulative number of impacts on the Moon in the range of 10^{-6} kton TNT to 10^{-4} kton TNT	MO.01
SCI.03	The mission shall allow observing and quantifying of the meteoroid impacts during meteor showers, where 40 impacts per 24-hour observation period are expected	MO.01
SCI.04	The mission shall observe sporadic events with frequencies ranging from 1 impact per year to 4 impacts per 24-hour observation period (approximately 1500 impacts per observation year)	MO.01
SCI.05	The minimum number of detected impacts shall be 240 to provide a statistical data on meteoroid impacts	MO.01
SCI.06	The mission shall perform observations of the lunar Far-Side	MO.02

an idea of the amount of data that in first approximation could be collected. All these data have been proved in Merisio (2018)

Table 2.4 collects the remarkable payloads requirements. Some of these requirements have been improved by CDF (Walker et al., 2018), while others have been optimized and proved in Merisio (2018).

In particular after the CDF PLD.03 has changed, since the new suggested baseline consists of using two wavelength channels from 350 nm to 820 nm, Visible Spectrum channel (VIS) and from 820 nm to 950 nm, Near-Infrared Spectrum channel (NIR); moreover PLD.08 is set in this works as 66 ms, namely 15 Frames Per Second (FPS), accordingly to the integration time provided by the CCD. PLD.05 and PLD.14, have been verified in Merisio (2018) and some of the outputs of this works have been used to have the simulation of the algorithm. PLD.15 set a limit for the total amount of data that can be stored and this set a constraint on the working principle of the algorithm.

Mission requirements are listed in Table 2.5. Although in this work the mission analysis is not covered, it worth to report these requirements since are highly linked with the payload. In particular thanks to MIS.01 and MIS.02 it is possible to state that the Moon is always fully in-sight, and this feature is used in the algorithm to reconstruct the Moon disk edge from the image. In particular MIS.01 takes its rationale by the considerations reported also by Ortiz et al. (2006) that a flash over the lighted part of the Moon surface will have too noise to be detected properly and therefore during that period there will not be science, placing a constraint over the detection capabilities the algorithm has to accomplish. MIS.05 state the duration of the mission, and from this comes the request to have real time detection and to get rid of useless images is emphasized.

In Table 2.6, the payload processor requirements are listed. These set the constraints to which the camera has to comply. Indeed the image rate that is provided depends on the capability of the processor to manage the data. Form PLDPROC.02 comes from the necessity to select the amount of data to be stored accordingly to the scientific relevance they have.

Table 2.4: *Top-level requirements, payload.*

ID	Requirement	Parent ID
...		
PLD.02	The LUMIO payload shall detect the meteoroid impact flashes with energies between 10^{-6} kton TNT and 10^{-1} kton TNT	SCI.01 & SCI.02
PLD.03	The payload shall detect the Meteoroid impact flashes within the EM radiation spectrum range between 450 nm and 890 nm	MO.01
PLD.04	The payload shall have a Field of View sufficient to have the full Moon disk in view	MIS.02
PLD.05	The payload shall have an aperture diameter of 55 mm	
...		
PLD.07	The camera shall operate continuously, during the observation cycle, to register flashes	
PLD.08	The image integration time shall be $\geq$ 30 ms	
...		
PLD.14	The payload processor shall identify flashes with an SNR greater than 5 dB	
PLD.15	The payload processor shall create less than 20 MB of the science data per day	
...		

Table 2.5: *Mission requirements.*

ID	Requirement	Parent ID
MIS.01	The mission shall observe of the lunar disk when the disk illumination is less than 50%	
MIS.02	During operations, the mission shall observe the lunar full disk	
...		
MIS.05	The mission shall have a minimum lifetime of 1 year	SCI.03, SCI.04 & SCI.05
...		

Table 2.6: *Top-level requirements, payload processor.*

ID	Requirement
PLDPROC.01	The payload processor shall receive and process a maximum 15 images per seconds from the payload
PLDPROC.02	The payload processor shall store a maximum of 13 MB of payload data per 29 days period to the COMMS for transmission to Lunar Orbiter
...	

2.3 System

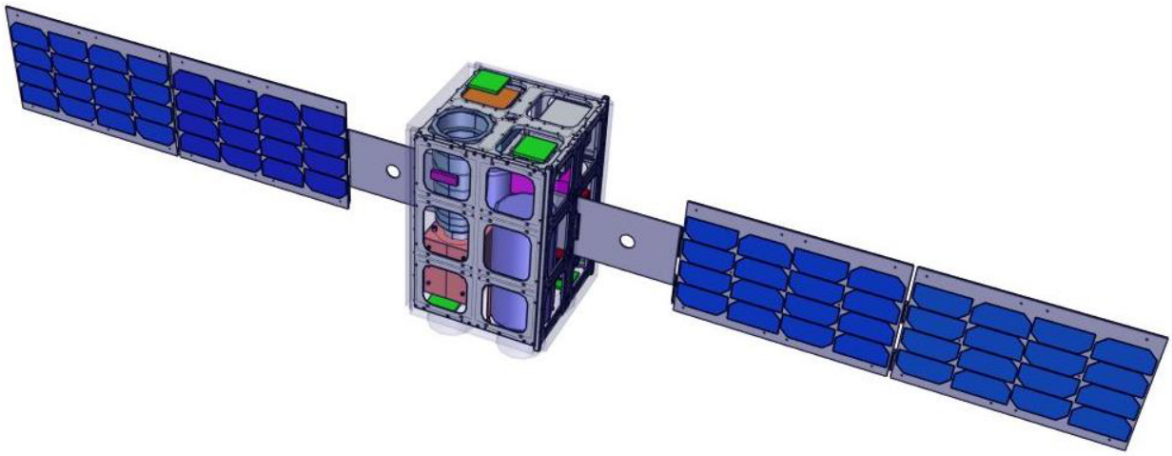


Figure 2.3: *Deployed configuration of LUMIO. Credits to:* Topputo et al. (2017).

LUMIO is a 12UCubeSat (2U×2U×3U), with a dry mass of 20.02 kg and a wet mass of 22.82 kg that will be placed in the L_2 Lagrangian Point of the Moon-Earth system with the aim of studying the Moon Far-Side in order to find flashes via an optical instrument. The launch opportunity is scheduled for 2023 together with the Lunar Pathfinder mission. The most recent information are related to Phase-*0* design (Topputo et al., 2017), and to the CDF (Walker et al., 2018). The main payload carried is the so called LUMIO-Cam, an optical instrument capable of detect either visible either near-infrared radiation, performing science finding impact flashed on the Moon surface. LUMIO-Cam itself is also designed to perform all the navigation and attitude tasks; indeed LUMIO implements autonomous navigation to bring it from the parking orbit where it is deployed by the launcher, to the final halo orbit. Once the operative life is over, it is required to de-commission the systems and to execute an EOL maneuver for safe disposal of the S/C. To accomplish all these operations LUMIO embeds chemical micro-propulsors, while attitude control is performed by means of reaction wheels. Regarding the power management, it is equipped with two Solar Array Drive Assemblys (SADAs) to be deployed after the release by the launcher able to point and follow the Sun by rotating along their axis. According to Topputo et al. (2017), documentation abut this technology is very poor, by the way IMT has recently developed a suitable SADA for CubeSats. As already said S/C shall operate in a quasi-halo orbit about Earth-Moon L_2 Lagrangian Point; many simulation and proposal has been performed in order to find the better orbit for LUMIO. In particular it has been selected the one with Jacobi's constant $C_j = 3.09$ (Topputo et al., 2017), since provide both optimal maneuver cost and robustness against errors.

2.4 LUMIO-Cam

LUMIO-Cam is an optical instrument and it is the payload carried by LUMIO S/C. The cam shall be able to detect low energy impacts also at the maximum operative distance from the Moon and at the same time shall be able to maintain a small Field of View (FOV) enough to look at the Moon disk only. The design has changed passing from Phase-*0* design (Topputo et al., 2017) to the CDF (Walker et al., 2018); at first it was proposed to use a single camera detection strategy so that all the radiation

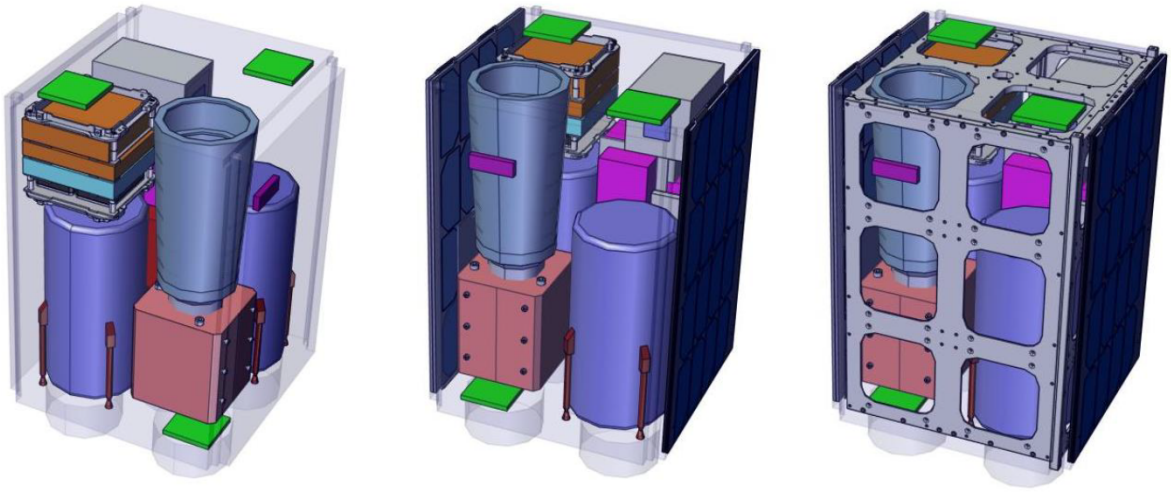


Figure 2.4: *Internal layout of LUMIO. The arrangement of the instrument inside the CubeSat is shown. Credits to:* Walker et al. (2018).

incident would have detected by a single detector sensible to visible radiation wavelength. Doing so, it might happen to have a false detection mainly due to the passage through the CCD detector of a GCR. Indeed it would create a signal that in absence of any further information could be confused with a flash. To avoid this kind of problem it was designed a defocusing lens just before the detector. In a second iteration of the design, coincident with the CDF, the baseline solution has changed, so that now the design is to use two different detection channels, VIS and NIR. The optical design has not changed from the previous solution except for the defocusing lens that is no more used and it is substituted by a dichroic lens, able to split the incoming radiation into two beams according to a threshold wavelength. In this way it is possible to confirm the detection just checking if the flash has been reported from both the channels, otherwise it is considered as a GCR. In the CDF (Walker et al., 2018), a detailed trade off about this topic is reported, and is moreover stated that a full NIR channel, where the black body temperature peak occurs is not feasible. In particular it is reported that a guess solution for further design phases could be to consider for the dichroic a splitting wavelength around 820 nm. According to this last proposed configuration it is possible to exploit the presence of two different channels a different wavelengths by evaluating, as done by NELIOTA (Bonanos et al., 2018), the black body temperature of the radiating plume by using the Plank's Law.

2.4.1 Optics

Optics is composed by 5 lenses whose aim is to focus the incoming rays for the detection. These, in the first iteration were coupled with the defocusing lens, substituted in the latter review by the dichroic lens that has the aim to split the light coming from the Moon surface into two different bands, one up to 820 nm for the VIS and the other over 820 nm for the NIR (Figure 2.6).

From the mechanical point of view the case is aluminum made, while the supporting barrel for the optics are in titanium; in Topputo et al. (2017) the cam is designed to have a short baffle of length 152 mm, while in Walker et al. (2018) this part has been extended up to 160 mm in order to reduce as much as possible disturbances due to strayligths as shown in Figure 2.6.

Regarding the sizing of the baseline solution for the optics, it is possible to starts from the

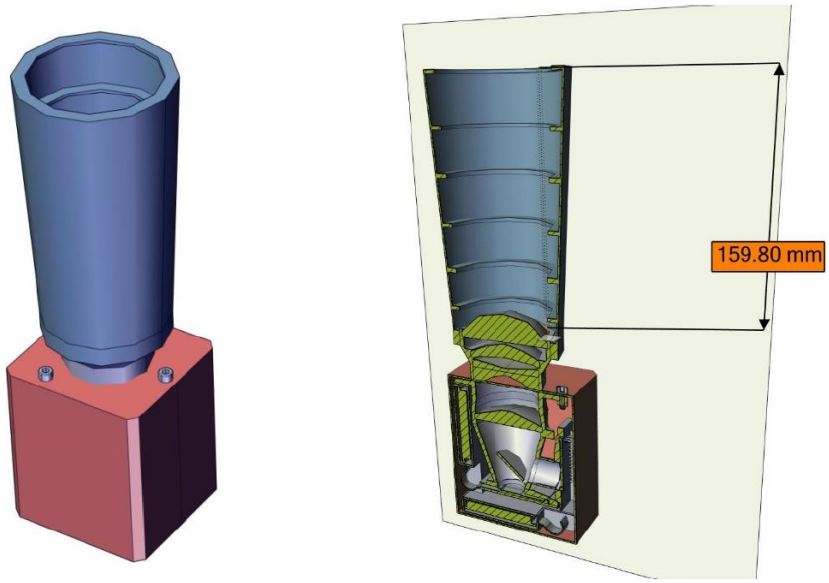
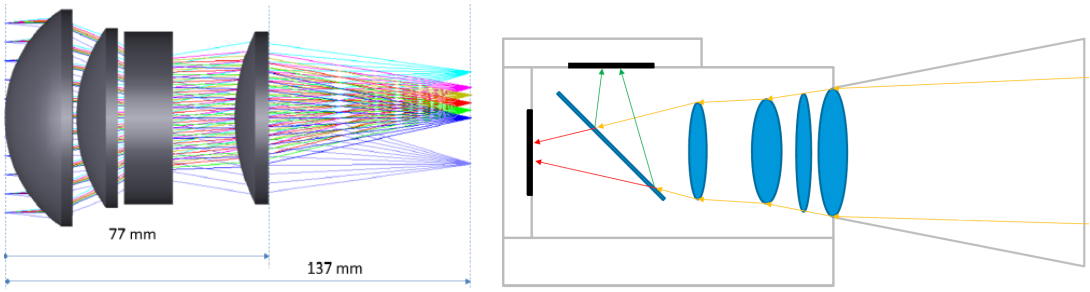


Figure 2.5: *LUMIO-cam configuration. The configuration of the detector is shown. Credits to:* Walker et al. (2018).



(a) *First configuration* (b) *New configuration*

Figure 2.6: *LUMIO-cam optics configuration. On the left the first configuration proposed by Topputo et al. (2017) is shown; the rays comes in from left and are focused by the lenses. In this configuration is used the defocusing lens that spread the radiation. Credits to:* Topputo et al. (2017). *On the right the new baseline configuration is shown; the rays comes in from right and are focused by the lenses until they reach the dichroic that split the radiation into the VIS and NIR channels. Credits to:* Walker et al. (2018).

Table 2.7: *Optics features*.

Parameter	Symbol	Value	Units
Field of View	FOV	6	degrees
Focal Lenght	FL	127	mm
Aperture Diameter	D	55	mm
f-number	f	2.3	-

requirements on the orbit in Table 2.1 and from the trade-off analysis performed by Cipriano (2017) and by Topputo et al. (2017); here it is stated to use an halo orbit around Earth-Moon L_2 Lagrangian Point; in addiction to that it is also stated in Table 2.5 that the camera has to have always in sight the Moon surface. By linking these two requirements it is possible to retrieve the information about the minimum value of the FOV as:

$$\text{FOV}_{\text{min}} = 2\tan^{-1}\left(\frac{R_{\leftmoon}}{d_{\text{min}}}\right) = 5.68^{\circ} \tag{2.1}$$

where $R_{\leftmoon}$ is the Moon's Radius and d_{min} the minimum distance from the surface; as reported in Topputo et al. (2017), the value is rounded to 6°, for possible pointing errors.

For what regards the Focal Length (FL), it can be computed as:

$$\text{FL} = \frac{d_{\text{CCD}}}{2}\tan^{-1}\left(\frac{\text{FOV}}{2}\right) = 127\text{mm} \tag{2.2}$$

where d_{CCD} is the CCD side length. By knowing also that the diameter of the lens is 55 mm, it come form that the f-number is 2.3

2.4.2 Detector

Although the design of the optics have been changed, the detector chosen for the first configuration has been kept also after the CDF. By the way, even though the two channels are deputed to detect different wavelength, it is chosen to use either for VIS, and NIR, the same typology of detector, an Electron Multiplying Charged-Coupled Device (EMCCD) device, characterized by an high sensitivity to faint inputs, whose output is sampled and digitized via an high performance 14-bit Analogue to Digital (A/D) converter, AD9240. The EMCCD is capable of detecting single photon events and to maintain high quality Quantum Efficiency (QE). The current baseline is the CCD201 of E2V L3Vision™. It is a 1024×1024 pixel camera with a readout frequency of 15 MHz. This device has been already tested by the Jet Propulsion Laboratory (JPL) and its maturity is defined as Technology Readiness Level (TRL)-6. In Figure 2.7 is reported the QE, that links how many electrons are formed by each incident photon, digitized by using WebPlotDigitizer[1] and printed using MATLAB®, according to the data found in the datasheet[2]; in particular are highlighted the two different bandwidths the detector is interested in.

[1] `https://automeris.io/WebPlotDigitizer/`[Last visited on 08/11/2019]

[2] `https://www.teledyne-e2v.com/shared/content/resources/File/documents/Imaging%202017/EM%20Sensors/CCD201-20/1491.pdf` [Last visited on 08/11/2019, Version 6, June 2017]

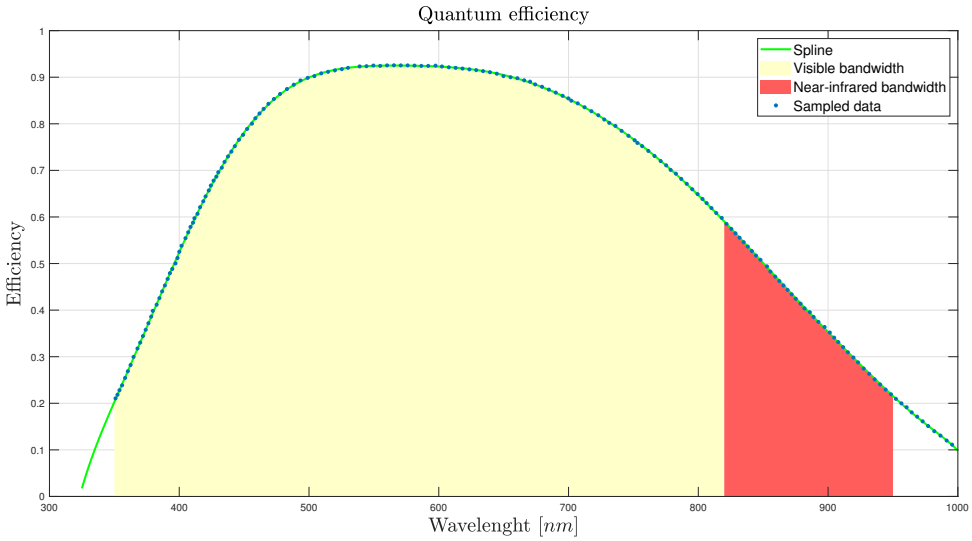


Figure 2.7: *Quantum efficiency of the EMCCD CCD201 of E2V L3Vision™. The blue dots represent the sampled data taken from the datasheet, the green line the spline built with* MATLAB® *an the two colored areas represents the wavelength of interest.*

Table 2.8: *Detector features.*

Parameter	**Symbol**	**Value**
Model	ID	e2v CCD201-20
Image Area	A_{det}	13.3×13.3 mm
Active Pixels	N_{pix}	1024×1024
Pixel size	A_{pix}	13.3×13.3 µm
Exposure Time	Δt_{exp}	66 ms
Low Noise Gain	G	$[1-1000]$
Readout Frequency	f_{rout}	15 MHz
Excess Noise Factor	F	$\sqrt{2}$
Charge Handling Capacity	C_{max}	80 ke$^-$pixel^{-1}
Charge Handling Capacity of Mul. reg.	C_{MR}	780 ke$^-$pixel^{-1}
Amplifier Responsivity	OAR	1.4 µV e^{--1}
Readout Noise @ 1 MHz	$\sigma_{OAR,0}$	< 1 e$^-$rms
Analogue to Digital A/D Converter bit Number	N_{bit}	14
Optics Lens Reduction Factor	ξ	0.5355

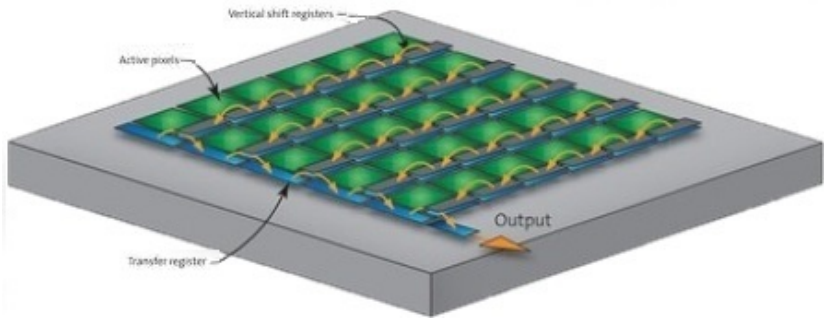


Figure 2.8: *Working principle of a CCD. Each green triangle represent a semiconductor cell (a pixel). The orange arrow indicates the process the data follow to deploy the information to the processor.*

2.4.2.1 CCD

A CCD is a device capable of collecting electromagnetic radiation and to transform it into a signal. Electromagnetic radiation can be considered both as a wave that as a particle, mainly according to the frequency it is considered and this difference leads to different methods of detection. In this case the radiation is considered as made by particles, the photons, whose energy, directly proportional to the frequency, is $E = hf$. Each photon colliding against the CCD releases its energy and creates therefore a signal. A CCD is formed by an array of semiconductors that receiving the energy from the incident photons generate a flux of electrons inside each of them, that will create the output that can be read as voltage, Figure 2.9. Since a semiconductor needs a certain amount of energy to make a electron move, this creates a lower boundary of incident energy (and so the intensity, in this case of the flash) that can be detected. At the same time the amount of energy that each semiconductor can manage is also defined and this creates an upper limit, beyond which even though other photon arrives no electrons can be produced and therefore the produced signal is fixed, and this leads to the saturation of the pixel. The amount of semiconductors cells available in the CCD determines how many pixel it has. In particular LUMIO-Cam having an 1024×1024 pixel array means that is composed by a 1024×1024 array of semiconductors each behaving as stated before. In order to collect the result of a single image, at defined time steps there is a reading phase, in which all the semiconductors are reset to their initial condition, creating the output voltage. The time step is defined by the readout frequency of the camera, that in this case is set 66 ms. Then these operations are replicated every time an image is shoot. The signal extracted from the CCD cannot be used directly in a computer since it is analog, and it has to be converted into a digital signal via an A/D converter. This device taking as input a voltage signal give as output an integer number between 0 and a maximum one that can be computed accordingly to the feature of the A/D itself, a process called quantization; for example for LUMIO, being a 14 bit A/D, the maximum number is 16 383, computed by using the formula $N = 2^{n_{bit}} - 1$. In particular to the maximum number it is associated the condition of saturation while at the minimum one the condition of no signal. All the possible values of intensity are linearly distributed in the middle, taking into account that due to the quantization of the information two very near, but different values in analog signal could be treated as equal in digital world.

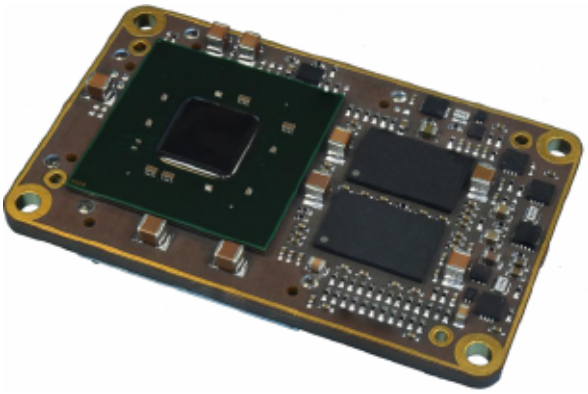


Figure 2.9: *The GomSpace Nanomind Z7000 processor. Credits to:* Topputo et al. (2017).

2.4.3 Processing

The On-Board Payload Data Processing System (OBPDP) is the computer deputed to the processing and the management of the images taken by the camera; it shall work to a frame rate of 15 FPS, with a size of 2 MB per frame (Topputo et al., 2017), leading to a stream of data of about 30 MB s^{-1}. The rationale that has driven the definition of the OBPDP is to either maximize the amount of scientific information that can be collected, but at the same time to minimize both the amount of computations to be performed on-board and to reduce the amount of useless data to be saved. The non-secondary task the OBPDP has to accomplish is to survive in space environment for the whole mission duration (at least one year), and therefore has to be space certified. The choice for the baseline solution of LUMIO team went to a hybrid Floating Point Gate Array (FPGA)/ARM, The GomSpace Nanomind Z7000, already proven in space. It is based on the Xilin Zynq 7030 that has a dual coreARM Cortex A9 (32-bit, 800 MHz) and an FPGA module with 125K logic cells (Topputo et al., 2017). It is designed to receive image from the camera and create the scientific product. The OBPDP has also to control the camera and to send settings to the camera to change its acquisition mode. To keep the communication between these, there is a dedicated high-speed SpaceWire interface between the OBPDP and the camera. This solution has been reviewed in the CDF, and a new processor has been proposed: a Microsemi^{R} SmartFusion2^{TM} FPGA with internal Cortex-M3 with a quad-core Central Processing Unit (CPU) and a Radeaon Graphics Processing Unit (GPU), but at the same time it is stated that the choice between the first and the second alternative has to be done accordingly to the choice of the On-Board Computer (OBC) (Walker et al., 2018)

3

Methodology

In this chapter all the steps developed for the algorithm are covered. Here are shown the flow charts representing the logical process the methodology follows and together are listed also some comments about the choices and simplifications done in order to achieve the result.

3.1 Introduction

Image detection is a large covered topic in literature and a lot of techniques are available, and many of them are summarized in Lu et al. (2004) and Hussain et al. (2013). These methods, also introduced in Chapter 1.2.4, are general, so that are suitable for many applications. Vice versa in Suggs et al. (2008), Madiedo et al. (2015a) and Xilouris et al. (2018) are listed the specific algorithms used in their own programs. Furthermore, have to be taken into account all the constraints and peculiarities that the LUMIO mission provides, so that the methodology can accomplish in the best way the issue. Indeed has to be considered that while on ground the computational effort and the memory size are unlimited, on space this is not true. Moreover for those images taken by Earth an hard pre-process phase has to be done to reduce all the noise that would avoid the correct detection; for these reasons it is not possible to take an already existing methodology and repeat it for this specific application. To select a proper method a trade off analysis has been performed, as shown in Table 3.1. The choice is to use an algorithm that looks at the difference between the image acquired at the current time and compare it to a reference one in order to find if a change has happen or not. In order to get the result for this works some simplifications have been performed, but nevertheless this should not compromise the output obtained. In particular both the presence of strayligth and the presence of stars are neglected. Indeed the simulation of both the effects would be an unuseful complication of the work since a well defined model to simulate them is not available. Regarding the strayligth the new proposed configuration for LUMIO-Cam tries to minimize the disturbance, while concerning the stars, the algorithm is able to

minimize their disturbance by filtering out them.

The algorithm is simulated in MATLAB® (release R2018b), and takes as inputs images of the Moon, searching for impact flashes on the surface and if some are found, it saves as output all the useful data from which it is possible to perform science.

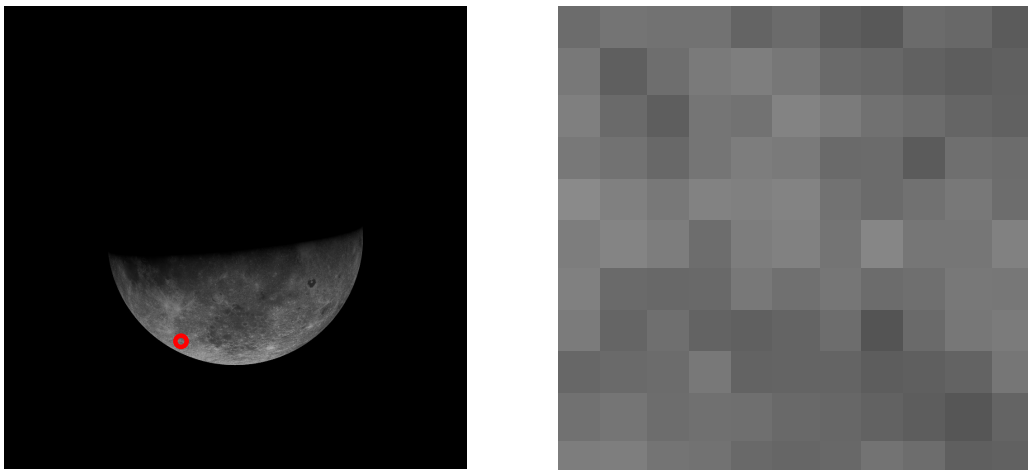
3.1.1 MATLAB®

Since the simulation of the algorithm is performed by using MATLAB®, it worth to point out some characteristics of this environment in which this work has been done. Indeed MATLAB® is a software for numerical computation written in C and created by Mathworks. It is specialized in the management of matrix and arrays and allows to create algorithms and functions to create some programs. In particular a function is an algorithm in MATLAB® language able to take an input from outside, process it and to produce as output some information. According to the way it is developed, it is preferred to manage the information by using a matrix rather than for example by for-loops; this can save a lot of computational effort and time. Another element to save computational time is to preallocate the dimensions of some output vector, where possible, instead of changing the size at each iteration. All these features have been taken into account in the development of the algorithm. MATLAB® has a lot of so called built-in function, namely function already available from the installing package, optimized and produced by Mathworks, that cover a lot of situations. By the way in this work these are not used, since once the algorithm will be deployed on the satellite, those will not be implemented as well, and therefore would be useless. The only built-in function used refers to the Image Processing Toolbox, that will be for sure implemented on the final support, since it is the platform used to manage the image. Some particular features of this toolbox have to be pointed out since they will shape the structure of this work. Indeed whatever image is processed into MATLAB® is transformed into a matrix of whose dimensions depend on the number of pixels the image is composed by, in this case 1024×1024, but the information of each pixel, namely the color, is quantizied by using `uint8`. This means that the intensity of each pixel is represented by integer positive numbers at 8 bit, going therefore from 0 to 255; Each decimal part is rounded to an integer, negative numbers are considered as zero, while number over 255 are considered as 255 (Figure 3.1). In this way this procedure acts as the CCD together with the A/D. Unlike the specifications stated in Topputo et al. (2017), where the A/D is of 14 bit, in this case we have only 8 bit.

3.2 Detection chain

In Figure 3.2 the flow chart of the detection algorithm is shown. It is now explained in depth the way it is built and the motivations of the choices done. The trade off analysis shown in Table 3.1 clearly states the method used.

The choice to use the direct comparison method, in particular the image difference, ensures a minor computational effort since the code have neither to perform vectorial transformations on images nor to collect a large amount of data in the memory; in addition this topic covered by the literature and widely used also in the programs that are carried out on ground. Besides, since the algorithm shall work on MATLAB®, either for this simulation, either for the final support, this environment is particularly optimized to operate with matrices and therefore it worth to work with them.



(a) *Full image* (b) *Focus image*

108	116	114	114	100	107	93	88	107	104	91
120	95	110	122	126	119	106	103	97	93	96
127	106	94	117	114	131	123	113	108	101	97
120	114	104	117	125	122	106	107	91	111	108
138	128	120	130	128	131	114	107	113	120	109
125	132	125	109	125	129	116	134	118	118	129
128	106	104	105	120	112	118	108	111	120	119
123	101	111	99	96	100	109	84	106	119	123
103	106	108	120	99	100	100	93	95	99	118
113	117	109	112	111	105	103	98	93	86	100
125	126	116	114	105	102	104	115	109	95	97

(c) *Numerical equivalent of the focus*

Figure 3.1: *Example of image management in* MATLAB®. *On left is shown the image as it is acquired by the software, the red circle is the area that is highlighted on right. This image is made by a* 11×11 *pixel matrix. Those are easily visible as each differs for a gray scale from the other in order to make the image in the full image. On bottom is shown the matrix form in which the* 11×11 *pixel image is represented into* MATLAB®.

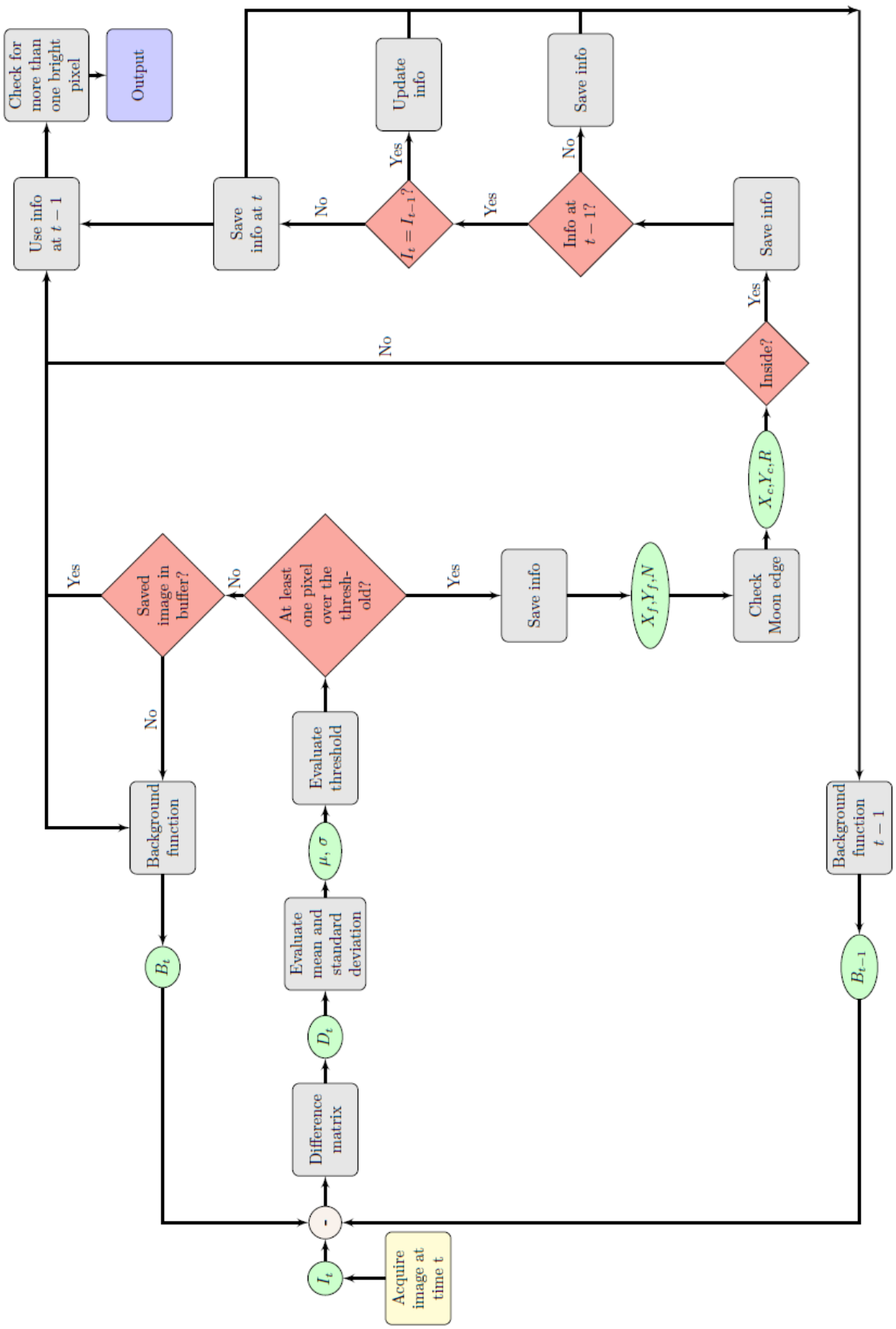


Figure 3.2: *Detection algorithm flowchart.*

Table 3.1: *Solutions trade-off for the detection algorithm.*

Method	Expected computational effort	Time spent	Used fro Earth observations	Implementation
Direct Comparison	Low	< 66 ms	Yes	Matrix operations
Transformation	Medium	$\simeq 66$ ms	No	At least 2 transformations
Classification based	Low	< 66 ms	No	Requires many data
Machine learning	High	≥ 66 ms	No	Requires many data and transformations
GIS	High	> 66 ms	No	No suitable
Spectral analysis	High	> 66 ms	No	Requires transformations

green Good feature yellow Fair feature red Poor feature

3.2.1 Flash detection

It is here covered the process that is designed to choose if in the acquired image there is a potential flash. The input of the whole work is the image at the current time, indicated as I_t. This image enters in the algorithm and is managed in MATLAB® through the MATLAB® Image Processing Toolbox.

Since no pre-processing is required, the difference between the acquired image and one obtained by the definition of background is performed. The subtraction is carried out pixel-per-pixel, namely, since in MATLAB® the images are treated as matrices, cell-by-cell, so that if nothing has changed between the last frame and the background one, the result is 0. This operation is done also by ESA program NELIOTA (Xilouris et al., 2018; Liakos et al., 2019); from this is taken the definition of the background function, written as:

$$B_t = \alpha I_{t-1} + (1-\alpha) B_{t-1} \tag{3.1}$$

where I_{t-1} and B_{t-1} are respectively the image and the background function at previous time. Moreover as made in Xilouris et al. (2018), the value of α is set to be equal to 0.35. This value is simply reported, since should be tuned on the final support since is very dependent on the quality of the images used, and therefore would be unuseful to set up an optimization process for this value by using sample images made with a software on a simulation tool; nevertheless this value can be taken as guess solution for a further optimization.

Looking at the background function definition, it is clear that it takes into account both the last information acquired, but also the complete evolution of the past background functions. Indeed it is important to remember that a real image is always subjected to noise sources, and therefore even if between a frame and the following nothing has happen, very likely they would not be equal, but will be slightly different. For this reason, by using a so written background function it is possible to mitigate all those disturbances, and at the same time taking into account also the change of the Moon's surface. Indeed, as can be seen in Figure 2.1, the science period extend for a long time during which the Moon is illuminated in different ways by the Sun. It is therefore mandatory that the algorithm shall take into

account this situation. It is here important to remember that all the values of the cells the matrix we are dealing with is made, each one representing a pixel, are quantizied by MATLAB® with values in `uint8`; this has to be taken into account since in cases where there is a difference like for example $7-8$, the result will not be -1, but 0.

Once the background function at the current time is defined (B_t), it is possible to evaluate the difference matrix D_t, between the image acquired and the background, defined as $D_t = I_t - B_t$. Here it is exploited the fact that the whole detection is performed with all the values of the cells that are in `uint8`; indeed, for the way D_t is defined, the newest image is always put as subtrahend (as first term in the difference), while B_t is used always as minuend (second term). In this way if a flash happens, it will be for sure in the first image and therefore the resulting number will be for sure positive. Vice versa, if no flash happens the values of the same location pixel in the current image and in the background will be very similar, but not equal due to the presence of noise; by the way exploiting the fact that negative numbers are not supported, all the points that have an intensity also of one unit lower than the corresponding one in the background are set to 0, namely the pixel is black, saving computational effort and reducing a lot the noise the difference image is subjected to. According to Topputo et al. (2017) the noise sources reported in the SNR are due to:

- Noise associated with the impact signal itself;
- Moon surface background noise
- Straylight background noise
- Dark Current
- ReadOut Noise of CCD
- Off-Chip Noise of CCD
- Quantization Noise generated by A/D converter

While some of these are linked to the environment, others are linked to the support through which the detection is performed. Define a model for all these would be an effort out of this work, since definition of the noise level would not add any improvement to the detection algorithm. By the way it is remarkable that by the way the background function is written, it tends to mitigate the effect of the noise, keeping after some iterations only the constant contribution representing the actual intensity of the surface. The same D_t matrix could be build up by using as minuend the previous image at each time step, but in this case the noise would not be mitigated.

When a flash occurs the spike is visible in the first image, and therefore it will be brought also in the background, since the following background function is built using as input also the current image; there is the danger to insert a pixel with an high value in the background function and therefore to have a very high value as minuend, that since we are dealing with `uint8` could cause a non-detection for the following iteration. This is an issue since a flash could last for more than a frame time (for LUMIO-Cam 66 ms), and therefore could happen that by subtracting the new image with the flash in the same pixel location, no detection is performed. The possibility of more than one frame with a flash inside is not so remote an a deep coverage of this topic is performed in Merisio (2018); in particular it is reported that a flash could last up to 3 consecutive frame. Furthermore has to be taken into account that during the integration time the plume cools down, reducing the temperature and therefore the power that is irradiated, and therefore, likely the value of the successive image will be lower that the one of the initial one. To avoid all these issues, it is chosen that if a detection happen, even if it is not

confirmed in the following passages, the background function is not updated, keeping the background constant. Since the impact, as said, should not last more than 3 frames, the background shall be frozen for time less than 0.2 s, a time in which the changes of the background can be easily recovered.

Adopting the strategy of the image difference the problem of stars or objects transiting near the Moon and of the large noise produced by the Moon Day-Side should be solved without the use of a mask to cover the background and half part of the Moon that are not of interest, as suggested by Topputo et al. (2017).

The matrix so obtained is mainly made by zeros and very low values if no flash have occurred, while if an event has happened, there will be also some higher values. Nevertheless a method to state if the value is high or not has to be defined, so that the algorithm could perform by itself the process. Said this it is clear that a threshold cannot be made by a single value and check if each pixel is over or under this one, being the level of noise is time dependent and related to the temperature and to the geometry between Sun-Moon-S/C. It is also remarkable that the output of the difference matrix depends on the background function, that changes in time according the the Moon's phase. It is therefore convenient to set dynamic threshold, able to set by itself a value that is nevertheless optimized at each time step, accordingly to the way the difference matrix is filled. Doing so the threshold would be robust to the variation in time of the background function and to the change in luminosity of the image. Indeed luminosity change will determine an overall increasing of the values, which however affect all the pixels in a constant way. The method used in this work consist on evaluating the mean value μ_{img} of the image and the maximum value of standard deviation σ_{img} of the image and to check for pixel in the image that have values $> \mu_{img} + 5\mu_{img}$, similarly a what is done by Suggs et al. (2008). In this way the threshold value is not fixed, but varies accordingly to the overall value of the image. Doing so the whole constant signal in the difference matrix is take in to account while the standard deviation takes into account the presence of the noise. Also in this case the choice to use the value 5 is due to the fact that it is discovered to be a good value for the simulation performed using MATLAB®, ensuring very good performances. By the way according to the final support that will be used this value can be tuned.

3.2.2 Flash confirmation

Once a detection has happened, a confirmation chain starts. In order to confirm that the detected variation in the image is actually a flash. Indeed, even if the design of the camera is done to have the FOV almost fully occupied by the Moon shape, some part of the image will be nevertheless detecting also the sky around the Moon edge. It could be possible that the detection could happen due to the change in magnitude of a star or due to the passage of a celestial body in the background or due to some noise around the Moon. One solution could be to set up a mask a priori that covers all the area over the edge, while keeping the surface visible. The main problem of this kind of solution is that even if the S/C is orbiting around the L_2 Lagrangian Point, the distance with the Moon will vary and therefore also the dimension of the Moon disk, leading to the need of a self-refining mask, that will require a lot of computational effort. It is chosen to develop for this work a method to recover the Moon edge starting from the image taken as input, as shown in Figure 3.3. Although the noise, it is possible to detect the edge of the surface; indeed, in those points the gradient of values between the pixel and its neighborhood will be very high since some pixels will be bright, due to the lunar environment made mainly by regolith, with a value very near to the maximum one on the almost white

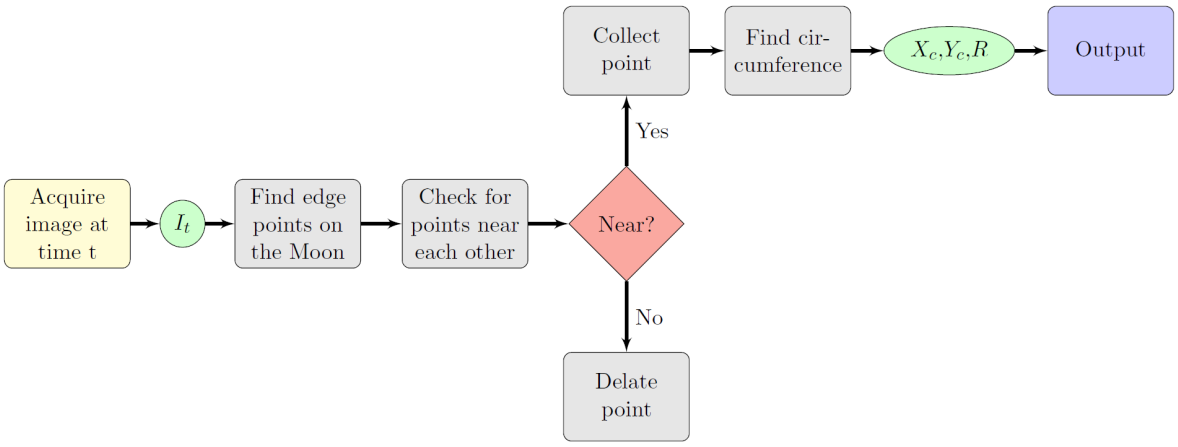


Figure 3.3: *Moon shape algorithm flowchart.*

surface, and the background will be close to 0, being all dark. It is required therefore to find all those near pixels that have the difference of their values higher than a threshold. In this case it is possible to select a priori a value and to use it without a dynamic process that refine it. Indeed even if the location of the lighted portion of the Moon is not always the same, we are interested on finding the difference between two values that, independently from the location are almost the same, due to their definition. In this work this threshold value is set to be equal to 80. Also in this case this number is taken as guess solution to perform this work with a synthetic image created with a computer a simulation in MATLAB®; for further analysis this value has to be changed, mainly due to the fact that it is highly dependent of the number of bits the software works with.

In this way all the cells that have a value higher than 80 with respect to the neighbor are selected. Since the transition from Day-Side to Night-Side of the Moon is not sharp, this method does not affect all the points of the internal edge of the Moon. This task is performed in the algorithm via a for-loop over each row and column, checking at the difference between a cell and its 8 near cells. Also in this case all the intensities of the pixel are treated as `uint8`; therefore the difference can be performed only in one direction, since in the other would return a negative number, that in this situation in counted as 0, avoiding to count the same point two times as for floating points. After this operation a large number of points are found, mainly on the edge of the Moon; nevertheless it is not said that all of them belong to the edge since there can be points either inside either outside the Moon with a difference with the neighborhood higher than 80, for example stars or the detected flash itself. A way to get rid of all these is to check for near points between the one found before. In this case all those points we are not interested in can be deleted. Indeed a star and a flash can be treated in the same way; both would at least occupy 4 pixels in the worst of the cases. Also in this case the algorithm can avoid to consider these as an edge point since there is only another near point. The possibility of getting both a flash that a star occupying more than a pixel is not due to the size, but rather due to the way a CCD is built. Indeed by performing the calculation about the size the area under a single pixel is in the best condition, namely in the point nearer with all the Moon surface in sight it comes to be equal to 11.5 km^2, but due to the fact that a flash could occur under the intersection of two or three pixel and therefore its energy is divided into those. Indeed it is required to check for the distance between the selected point and both the previous and the following one; for the way MATLAB® manage the matrices and the points that are found, if two points are near in the matrix they shall be also near in the image, otherwise are discarded.

After this at least three points are needed in order to find a circumference through which it is modeled the Moon's surface. If the algorithm cannot find three points, since for example the full Moon Far-Side in the night this passage is jumped, the flash detection is kept but in output no information is displayed (in MATLAB® gives `NaN`).

By taking three random points in this last set, it is possible to reconstruct a circle by following the definition of the circumference passing through three points. As first step are evaluated the angular coefficients of the two chords built by the three points.

$$m_1 = \frac{y_2 - y_1}{x_2 - x_1} \tag{3.2}$$

$$m_2 = \frac{y_3 - y_2}{x_3 - x_2} \tag{3.3}$$

being (x_i, y_i) the coordinates of the found points; once get those it is possible to evaluate the angular coefficient of the lines orthogonal to these ones.

$$m_{\perp 1} = -\frac{1}{m_1} \tag{3.4}$$

$$m_{\perp 2} = -\frac{1}{m_2} \tag{3.5}$$

Between all the lines with these angular coefficient, the ones passing through the mean points of the chords are the ones on which belong the centre coordinate; therefore the points are:

$$x_{m1} = \frac{x_2 + x1}{2} \tag{3.6}$$

$$y_{m1} = \frac{y_2 + y1}{2} \tag{3.7}$$

$$x_{m2} = \frac{x_3 + x_2}{2} \tag{3.8}$$

$$y_{m2} = \frac{y_3 + y_2}{2} \tag{3.9}$$

since the coordinates of the centre of the Moon belong to both the lines, finding the interception between them it is possible to recover $\hat{X}_{c\mathbb{C}}$ and $\hat{Y}_{c\mathbb{C}}$

$$\tag{3.10}$$

$$\hat{X}_{c\mathbb{C}} = \frac{y_{m2} - y_{m1} + m_{\perp 1} x_{m1} - m_{\perp 2} x_{m2}}{m_{\perp 1} - m_{o2}} \tag{3.11}$$

$$\hat{Y}_{c\mathbb{C}} = m_{\perp 1}(\hat{X}_{c\mathbb{C}} - x_{m1}) + y_{m1} \tag{3.12}$$

then, since the centre coordinates are known as well as a point on the edge it is possible to evaluate the distance between them and therefore recover also the radius

$$\hat{R}_{\mathbb{C}} = \sqrt{(\hat{X}_{c\mathbb{C}} - x_1)^2 + (\hat{Y}_{c\mathbb{C}} - y_1)^2} \tag{3.13}$$

$$\tag{3.14}$$

An image representing these passage is shown in Figure 3.4. All these quantities are expressed in

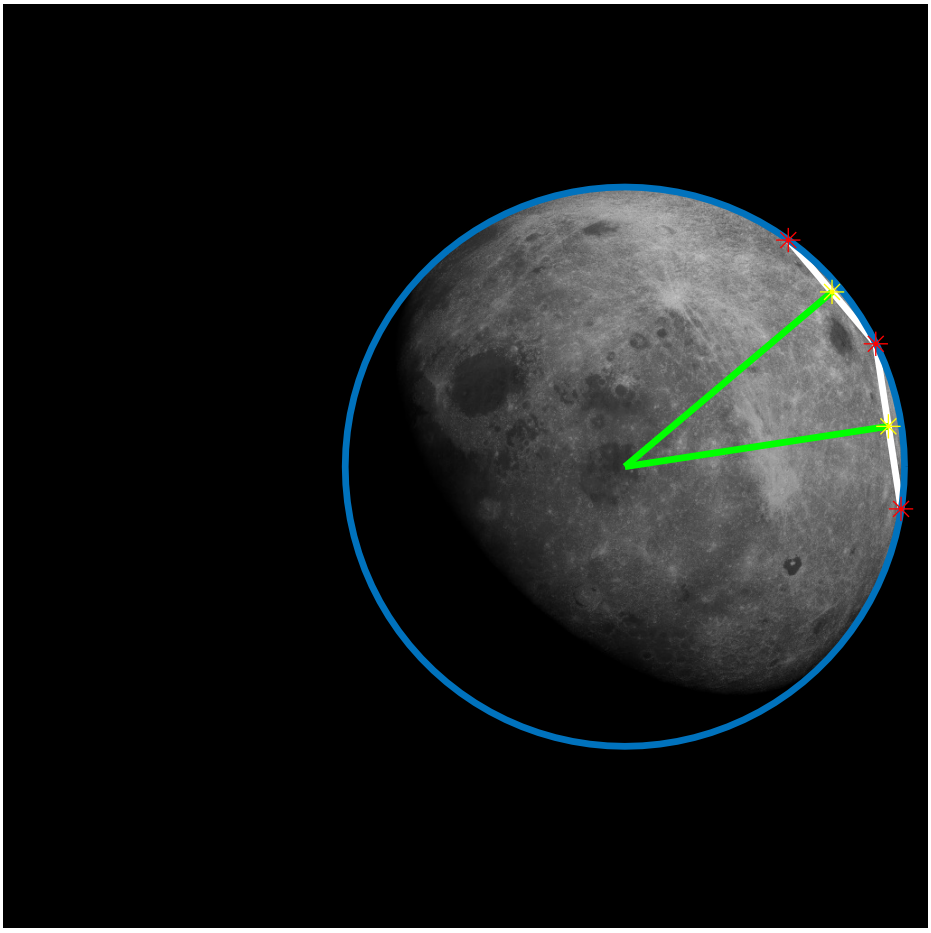


Figure 3.4: *Reconstruction of the Moon shape. In red are shown the three selected points with high gradient with their neighborhood, in white the chords that join these points, in yellow the two mean points of the chords and in red the perpendicular lines to the chords on which the centre lies. In blue is drown the reconstructed Moon edge*.

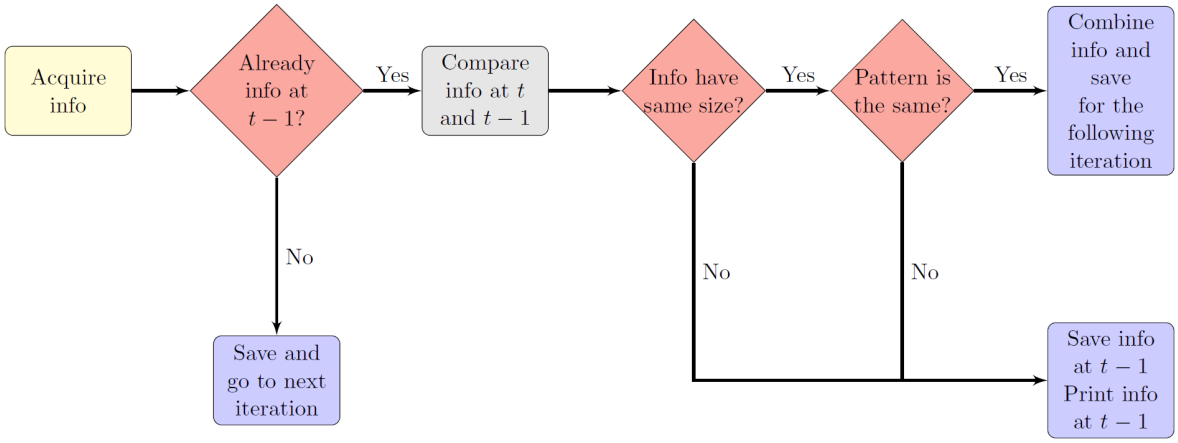


Figure 3.5: *Long lasting algorithm flowchart. It is here shown the flowchart for the algorithm that manage those flash that last more than a frame.*

the image frame; this means that all the length here found are expressed in pixel measure. Knowing the centre coordinates and the radius, it is possible to evaluate the Moon shape. Indeed by evaluating the distance between the flash and the centre of the Moon just retrieved and checking if the distance so computed is less than the value of the radius; if so the flash is for sure inside the Moon, otherwise not.

Once the centre coordinates and the radius are known, it is possible to check if the flash is inside the Moon for all the detected pixel by looking if the distance between the potential flash and centre is less than the radius.

3.2.3 Detection output

Up to now have been discussed the procedure the algorithm follows at each time step to detect if a flash has happened or not. Once a flash has been detected the result has to be printed in a proper manner in order avoid to memorize useless information and to waste memory. Moreover, as said previously, a flash event could last for more than a single frame; therefore it should be unuseful to save information for different frames that represent the same event; hence the algorithm has to recognize multiple frame events and collect all the data into a single information. To accomplish this task, each time a flash is detected in a frame it is checked if there is an already detected flash in the just previous image, as shown in Figure 3.5. If not the information are saved and the same process defined so far is followed in the next iteration, otherwise the current and the previous information are compared. A first check is performed to see it the amount of detected lighted pixel is the same. If it is not the case the previous image is saved in the memory for the further analysis, while the newest one remains in the buffer for the next iteration. If instead the number of information of the old and new images are the same a check over the shape of the pattern is done. Indeed even if the amount of information is the same, maybe the way the lighted pixel are placed is different passing from an image to the following one and therefore are treated in the same way are treated two images with different information. If also the patterns are the same, then the algorithm takes that the flash has last for more than a frame and therefore all the information are put together, saving only one set instead of two. In particular the values representing the intensity of the impact are summed, representing the total amount of energy that is released during the event, that from now on will have a duration multiple of 66 ms, according to the number of frame summed. It could happen that for very faint flashes, passing from a frame to the

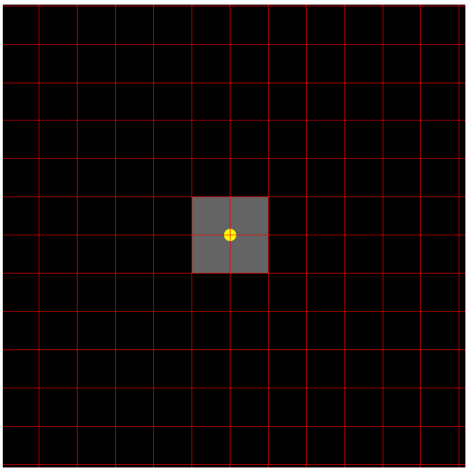


Figure 3.6: *Flash spread over more pixels*.

following the algorithm finds that the patterns or that the amount of information are different since the signal is not over the noise; the image representing the frame just before and after the flash are also saved in the memory to avoid this possible issues.

In this way it is possible to check a posteriori by an operator if the two consecutive patterns are made by a single event or not.

As stated before it could happen that the flash energy is spread over more than one pixel just because the impact occurs over a surface that is under the interception between different pixels. Therefore the image could be spread at least over four pixels (Figure 3.6). Nevertheless it is useful to collect all the data in order to save the information in a more ordered way and to have a proper post-process. This process, explained as flowchart in Figure 3.7, is performed once a full event is detected; this mean that this check is not performed each iterations, but just before to save in the memory the information. At this point the position of all the detected lighted pixels are known. For each pixel is evaluated the distance between itself and all the others; if another pixel nearer than 3 units is found the two are considered as made by the same source and thus the corresponding intensities are summed. Vice versa the pixel cannot fine any other pixel near this is counted as coming from another source, saving another set of information.

3.3 Post-processing chain

Once the information are retrieved by the image it is possible to recover further details; hereafter the method used in this work is described. The information saved up to this point are related to the amount of energy that is collected by the camera (values of the pixel), the coordinates of the flash and the coordinates of the Moon centre and radius in the frame coordinates system.

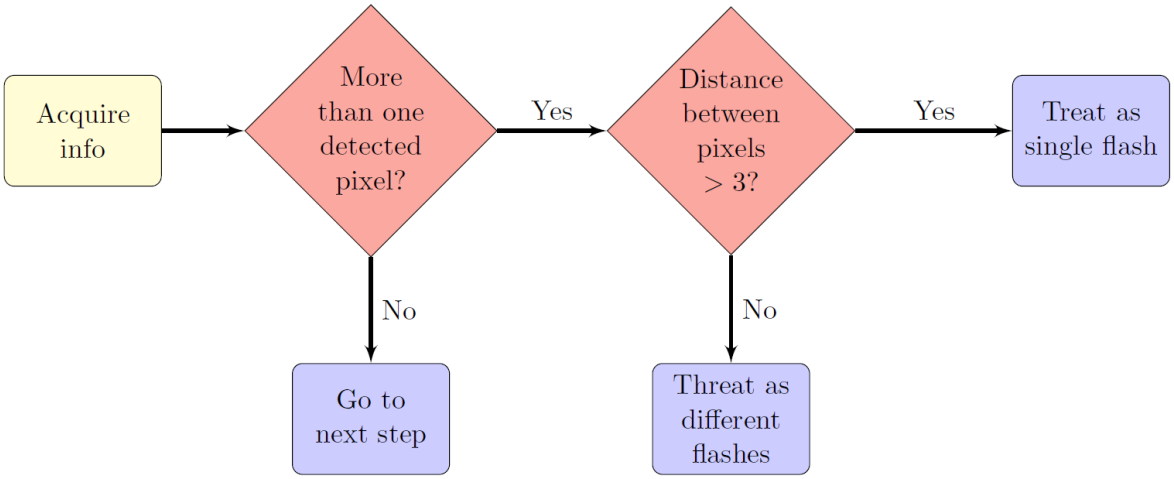


Figure 3.7: *Large flash algorithm flowchart. With this algorithm are managed those flash that are spread into more than a pixel.*

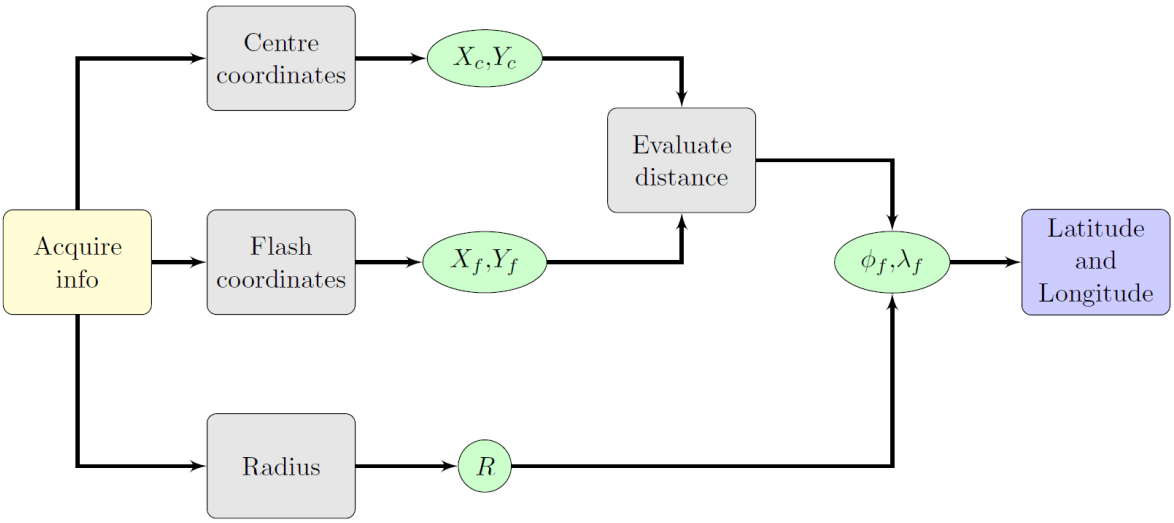


Figure 3.8: *Coordinate evaluation algorithm flowchart.*

3.3.1 Coordinates

A first information that can be found is the latitude ϕ_f and longitude λ_f angles. Those are not the absolute ones, but are referred accordingly to the image reference frame. The procedure followed is printed in Figure 3.8. The absolute ones can be found putting together the these data with the orbital information. Knowing the centre of the Moon coordinates, the radius and the coordinate of the flash it is possible to write:

$$\Delta x = \hat{X}_f - \hat{X}_{c\mathbb{C}} \tag{3.15}$$

$$\Delta y = -(\hat{Y}_f - \hat{Y}_{c\mathbb{C}}) \tag{3.16}$$

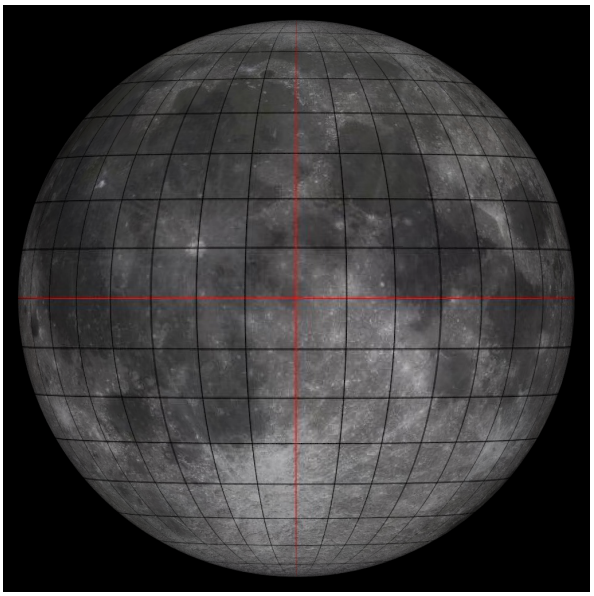


Figure 3.9: *Coordinates on the Moon. In red are shown the reference meridian, in vertical, and the reference equator, in horizontal. In black are shown all the coordinates lines, one each* 10°. *Note that the coordinates are always referred to the centre that depends on the used image.*

Δx and Δy represent the distance in pixel between the flash and the centre. The minus sign for Δy stays to compensate the two reference frames the measurements refers, one in pixels and the other in Cartesian axes

$$\phi_f = \sin^{-1}\left(\frac{\Delta y}{\hat{R}_{\leftmoon}}\right) \tag{3.17}$$

Latitude angle ϕ_f, tells how much the point is North or South from the centre. In the case faced in this work it can be easily recovered via this last formula since the lines will be parallel to the Equator line, being all referred in a relative frame. This don't happen for the longitude angle λ_f, since the lines are not parallel, but instead they all get together near the poles; therefore the distance between two meridians near the equator is far higher than the distance between the same meridians near the pole (Figure 3.9). For this reason is defined that has to take into account this issue:

$$R_{eq} = \hat{R}_{\leftmoon} \sin\left(\pi - |\phi_f|\right) \tag{3.18}$$

This radius represent an equivalent radius that changes its length according to the latitude angle ϕ_f.

$$\lambda_f = \sin^{-1}\left(\frac{\Delta x}{R_{eq}}\right) \tag{3.19}$$

Longitude angle λ_f tells if the location of the flash is West or East with respect to the centre.

(3.20)

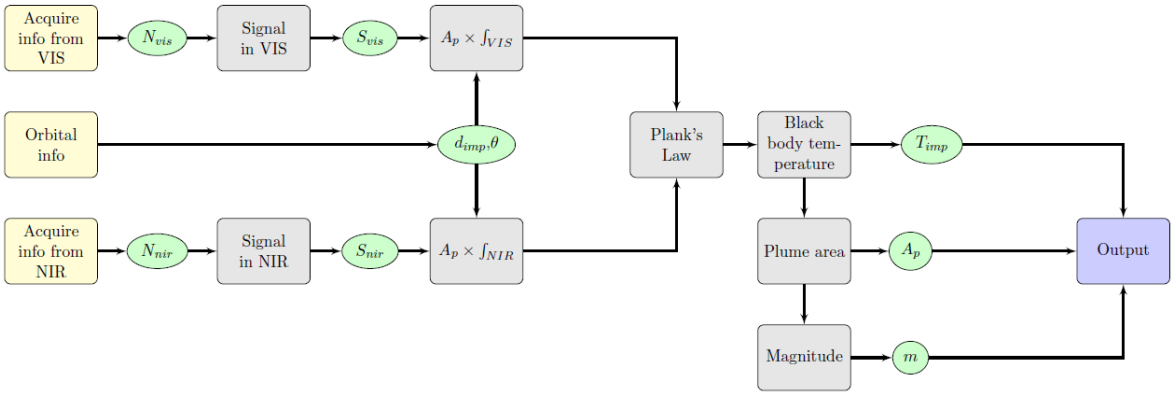


Figure 3.10: *Post-processing algorithm flowchart.*

3.3.2 Physical data

Exploiting the presence of VIS and NIR channels, it is possible to recover many other physical data from the image, as shown in Figure 3.10. The detection is performed only on one channel, and if something is found the same data are collected also in the other one. For this reason the data represent the same phenomenon, but according to different wavelengths. A first information that can be found in the black body temperature of the plume that is generated.

For the way the data are collected in the previous steps, for each impact event will be collected both the total intensity of the lighted pixels, representing the energy released and both the duration of the event in multiples of the integration time 66 ms. Exploiting the Plank's law it possible to link the Spectral emissive power $\mathrm{W\,m^{-2}\,m^{-1}}$ to the temperature

$$L(\lambda, T) = \frac{2\pi hc^2}{\lambda^5 e^{\frac{hc}{\lambda k_B T}} - 1} \tag{3.21}$$

where h is the Plank's constant, c the speed of sound in vacuum, λ the wavelength, k_B the Boltzmann's constant and T the black body temperature.

According to Merisio (2018), the impact signal can be written as:

$$S_{imp} = f_{ld} \frac{D^2 \cos(\vartheta)}{4 f d^2} A_p t \int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, T)\, d\lambda \tag{3.22}$$

where f_{ld} represent the degradation coefficient due to light diffraction and in this work it is taken as constant equal to 0.84, D is the aperture diameter of LUMIO-Cam lens, ϑ represent the angle between LUMIO boresight direction and the impact vector, f the anisotropy degree of light emission and for this work is set equal to 3 (Merisio, 2018), t represent the total duration of the event and A_p the plume area. The quantity:

$$\int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, T)\, d\lambda \tag{3.23}$$

is expressed in $\mathrm{e^-\,m^{-2}\,s^{-1}}$ and represent the amount of electrons that are generated per unit time over all the considered wavelengths.

$N_{e^-}(\lambda, T)$, expressed in $\mathrm{e^-\,m^{-2}\,m^{-1}\,s^{-1}}$ is linked to the Plank's law since:

$$N_{e^-}(\lambda, T) = \frac{L(\lambda, T)\lambda}{hc} QE(\lambda)\xi \tag{3.24}$$

where $L(\lambda, T)$ is the spectral emissive power, h is the Plank's constant, c the speed of sound in vacuum, λ the wavelength, $QE(\lambda)$ the quantum efficiency and ξ the optics lenses reduction factor.

Therefore it is possible to take the information about the total intensity of the detected pixels in one single event, the estimated duration of the event, the location of the flash together with the orbital information and use this to get the black body temperature of the event; indeed S_{imp} can be estimated as:

$$\hat{S}_{imp} = \hat{N} \times \frac{C_{max}}{N_{max}} \tag{3.25}$$

where $\hat{N}$ is the quantized value retrieved by the pixel in the image, C_{max} the saturation value according to the hardware used and N_{max}, equal to 255 for MATLAB® the number of bits of the system; once this is find recover the quantity $A_p \int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda$ as:

$$A_p \int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda = \frac{4 f d^2 S_{imp}}{f_{ld} D^2 \tau \cos(\vartheta)} \tag{3.26}$$

Note that it is possible only to find this quantity since the real area of the plume is far smaller than the pixel area, and the time duration has switched from t to τ. Nevertheless this process has to be performed both for VIS that for NIR channel, setting proper values for λ_1 and λ_2 so that at the end, since the area they are considering is the same (they are looking at the same plume), this make no difference. Once $A_p \int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda$ is known for both the channels, it is possible to find the temperature as:

$$\frac{A_p \int_{\mathrm{VIS}} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda}{A_p \int_{\mathrm{NIR}} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda} = \frac{\int_{\mathrm{VIS}} L(\lambda, \hat{T}_{imp}) \frac{\lambda QE(\lambda)\xi}{hc}\, d\lambda}{\int_{\mathrm{NIR}} L(\lambda, \hat{T}_{imp}) \frac{\lambda QE(\lambda)\xi}{hc}\, d\lambda} \tag{3.27}$$

from which it is possible to solve for the temperature, that will be black body temperature of the event, $\hat{T}_{imp}$

It is therefore possible to evaluate the quantity:

$$\int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda \tag{3.28}$$

Multiplying Eq. (3.26) times this last equation(Eq. (3.28)), both evaluated for a consistent bandwidth (i.e. both in VIS, 320-850 nm) it is possible to evaluate the plume area (A_p). A last information that can be found is the magnitude at satellite distance; as reported in Suggs et al. (2017) and Madiedo et al. (2015b) the power emitted by an impact event can be evaluated as:

$$P = F_0 10^{-\frac{m-m_0}{2.5}} f \pi d^2 \Delta\lambda \tag{3.29}$$

being P the irradiate power from the impact, F_0 a constant representing the reference flux density of a magnitude m_0 source, equal to $1.36949 \times 10^{-10}\ \mathrm{W\,m^{-2}\,m^{-1}}$, m_0 is the reference magnitude of a source with flux density F_0, equal to 21.1, f is the anisotropy degree of light emission, d the distance from the Moon and $\Delta\lambda$ the bandwidth.

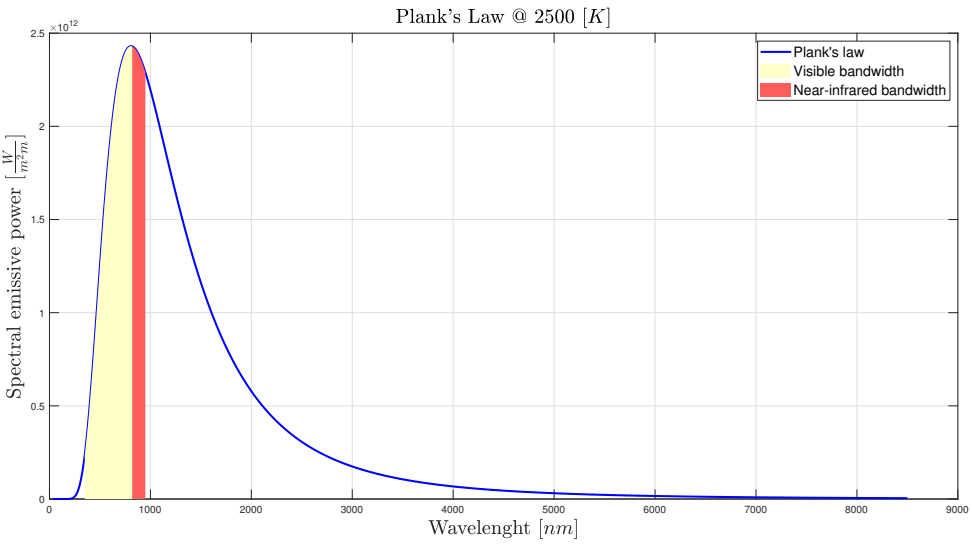


Figure 3.11: *Representation of the Plank's Law at* 2500 K. *Are shown also in two different colors the two bandwidths that are used to find the temperature.*

Solving for the magnitude it becomes:

$$m_{imp} = m_0 - 2.5log_{10}\frac{P_{imp}}{F_0 f \pi d^2 \Delta\lambda} \tag{3.30}$$

where P_{imp} is the power released during the impact.

4

Validation of the algorithm

In this chapter are validated all the methods and functions developed and used in the algorithm and listed in Chapter 3. In particular are faced all the steps performed by the algorithm, both for the detection both for the post-processing and are shown the logical processes that the algorithm follows to get the result.

4.1 Background function

It is here shown the behavior of the background function according to the input given and its evolution in time. The need of a function representing the second term of the image difference is due to the change of the Moon surface during the operative life of the LUMIO mission. Indeed the image seen by the S/C will be subjected to change both due to its orbital motion around the Moon, both due to the motion of this one around the Earth and therefore around the Sun. This will cause both change due to the illumination conditions and due to the distance from the soil. A first guess to accomplish this task can be done by using the previous image acquired at each time step; according to Topputo et al. (2017) and Walker et al. (2018), LUMIO-Cam shall produce image each 66 ms, a time short enough to compare two successive images and to consider them as equal. By the way this is not done in this work since a factor that has to be taken into account is also the noise due to the detection process. Taking a large number of images, with a subject almost constant in time, the set of images will be almost similar between them, except for a random component due to the noise. Therefore, by collecting a large number of information can improve the quality of the detection by reducing the impact of the noise. The function defined as:

$$B_t = \alpha I_{t-1} + (1-\alpha) B_{t-1} \tag{4.1}$$

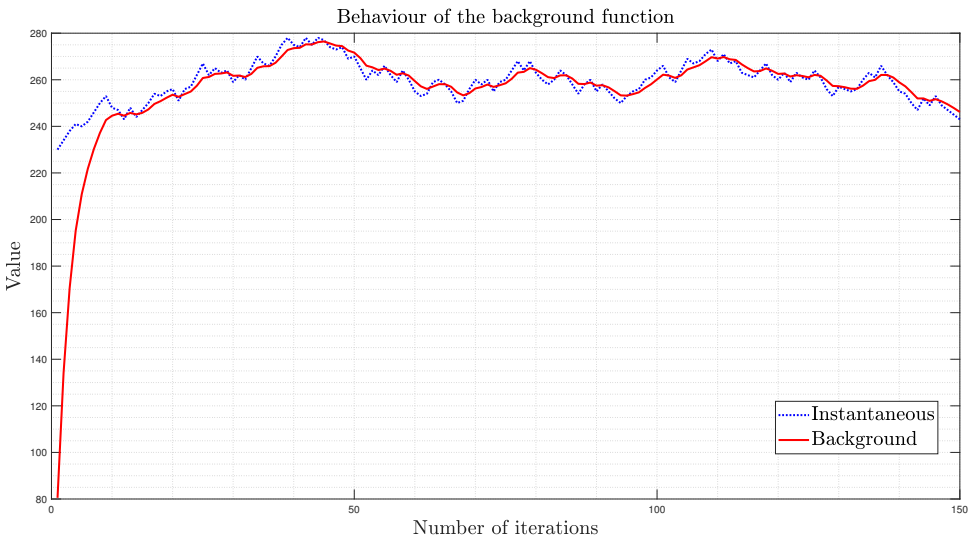


Figure 4.1: *Background function response to random inputs. The dotted blue line shows the input data, while the solid red line the background function response. This simulation is performed over a single pixel.*

can actually perform this effort. The newest information is taken into account without any pre-processing operation, but its weight is less (according to α value) with respect to the other term representing the old background function. In this way all the sharp variations that occur in the scene are not taken into account, since more importance is give to the past background function. Vice versa, if a change is extend for more than a frame, since either the current and the background function have the variation, the change comes to have an higher and higher weight, changing dynamically the background.

In Figure 4.1 it is shown the evolution in time, as the number of iterations past, of the value of a single pixel subjected to random value input, and the background function behavior accordingly to it. Mathematically speaking, the background function behaves like an integrator, all the sharp variation that are given as input are smoothed and the background function follows the trend in a more conservative way. Another feature that has to be pointed out is that the response of the background is always delayed with respect to the input, due to the definition of the function itself. Random instantaneous numbers are used as input to obtain this figure, the only constrain is of being bounded near (within 6 units) the previous value of the input; this choice is due to the fact that variation due to the distance or the illumination of the Moon will be always slower with respect to the acquisition frequency, and therefore the pixel variation from an image to the following will be smooth. It is nevertheless interesting to evaluate how long the transition last for a time varying image; in Figure 4.2 it is reported the response and the error variation of the background function with respect to a fixed value to be reached. As reported, from iteration 0 to 40 the evolution follows the same rules of Figure 4.1, while, after the solid green line, the input is set to be constant to a value (reported by the dashed black line) that has to be reached by the background. A deeper analysis can be performed looking at Figure 4.2b where it is shown the error between the input value and the background. As can be seen, the error goes rapidly to decrease to zero, with a rate dependent of the value of α chosen, with the rationale that the more weight to the newest image is given, the faster the transient will be. On the

other side a too fast decreasing rate would create a too sharp variation in the background. It worth to notice that, after the green line the highest value of error is equal to 2; this mean that between the input value and the number assumed by the background at that iteration the difference is 2 units; this result can be accepted since the level of noise that is assumed during the simulation of this work produces values due to the difference of the images higher than 2.

Looking at the first iteration both in Figure 4.1 that in Figure 4.2, can be seen a large difference between the input value and the one evaluated through the background function. Indeed since its definition, in order to have a correct meaning, the function has to collect a certain number of images before being used. In the simulation for the first iteration has been taken $B_t = 0$; for the second $B_{t-1} = 0$, while from the third on the method is automatically updated, according to the function: so a period of tuning for the background function, during the first instants of the science mode of LUMIO mission, is needed. Nevertheless this phase will last only few frames, less than 10, corresponding to less than 0.66 s. In Figure 4.3 is shown the evolution of the background for an image representing the Moon that has been used for the simulation of this work from the first iteration, frame 1 and t =0.066 s, to the 10^{th} frame at t =0.66 s. As can be seen in the last images the variation with respect to the previous one in almost negligible, which indicated that the background function work properly.

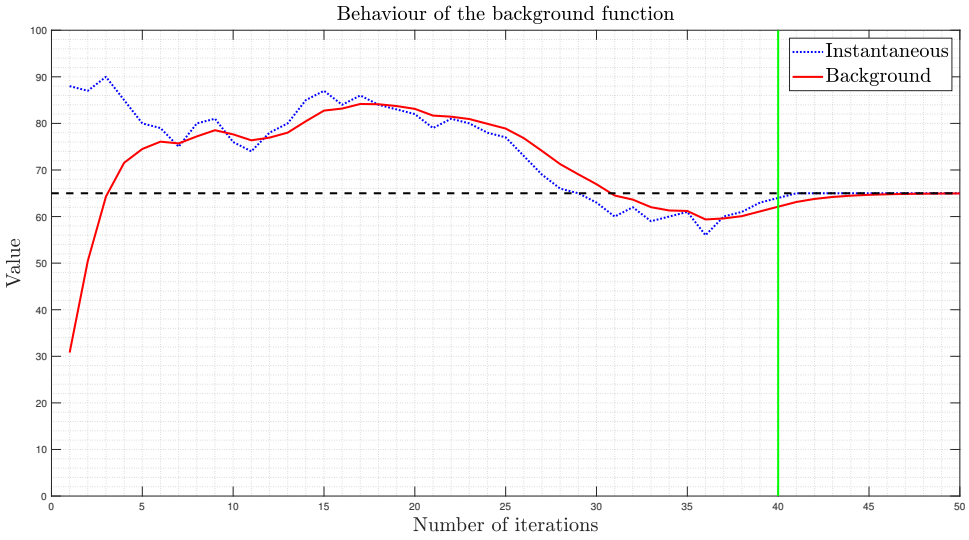
4.2 Difference matrix

The results obtained by the subtraction of the background from the acquired image are here discussed. The difference matrix is defined as:

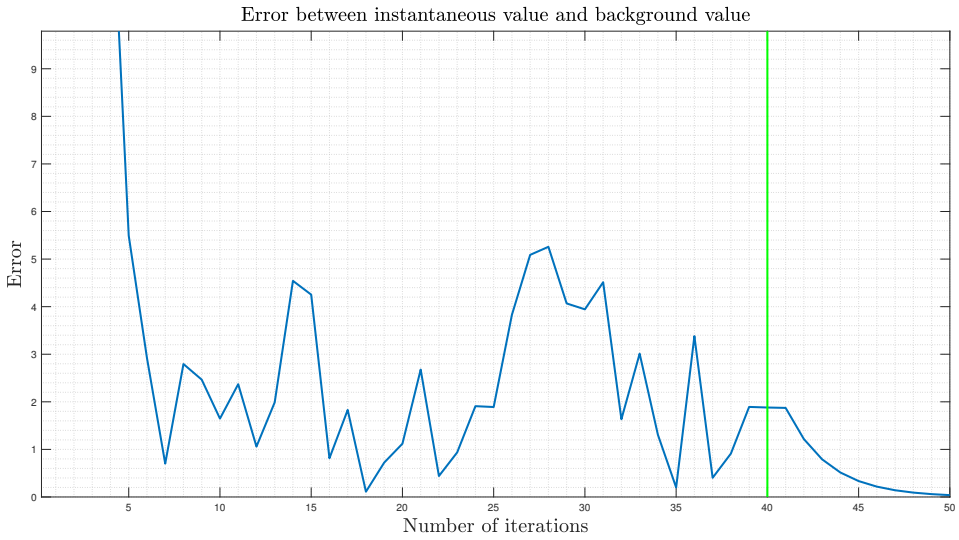
$$D_t = I_t - B_t \tag{4.2}$$

therefore the result is not only dependent from the acquired image, but also from the value of the background used. In particular in Figure 4.4 is reported the evolution of the difference image for the first iterations, up to the tenth, where there is the tuning period of the background. It is here evident that up the 6^{th} image the detection it is impossible since the difference image is almost similar to the acquired one; nevertheless just from the 7^{th} iteration it is possible to get the detection, loosing just less than half a second.

As can be seen, when all works, this function it is almost black, since the images that are subtracted are almost equal, nevertheless the result of the subtraction pixel-per pixel is not zero, since there is a component due to the presence of the noise; in Figure 4.5 it is shown the focus around the detected flash in one frame of the difference image. To make it visible at naked eye the value set for the pixel is of 100 over 255; nevertheless it is more useful to look at the matrix representing the same image shown on the right, where the highest number (101) represent the flash with the noise, and the other set of ones and zeros are the result of the difference between the acquired image and the background function. Those values are due to the noise in both the acquired image and the background. In spite of that, the values that are associated to the noise in the difference function are very limited, never over 3, granting a good performance.

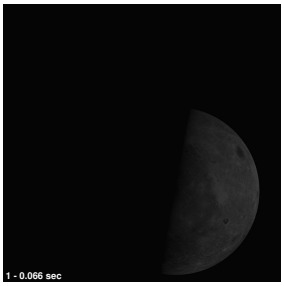


(a) *Response*

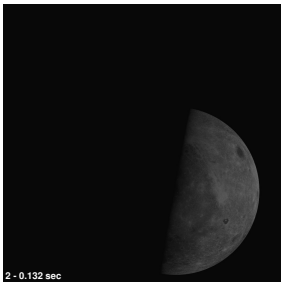


(b) *Error*

Figure 4.2: *Background function response to a setting value. In a) The dotted blue line represent the instantaneous input value, the red solid the background function and the green the moment when the final value (represented by the black dashed line) is forced. In b) the error between the instantaneous and the background function is shown.*

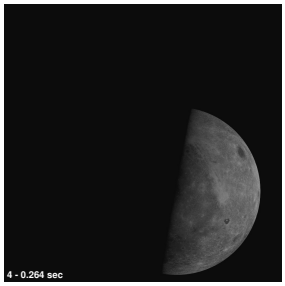


(a) *First iteration* - 0.066 s

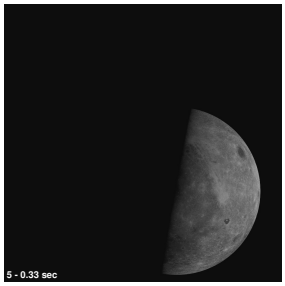


(b) *Second iteration* - 0.132 s

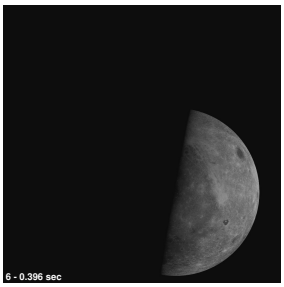
(c) *Third iteration* - 0.198 s



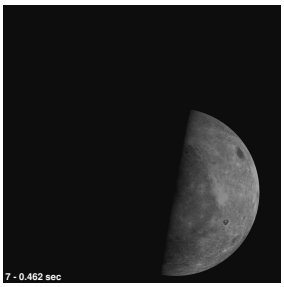
(d) *Fourth iteration* - 0.264 s



(e) *Fifth iteration* - 0.33 s



(f) *Sixth iteration* - 0.396 s



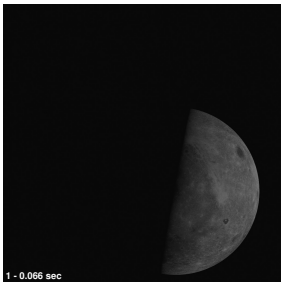
(g) *Seventh iteration* - 0.462 s

(h) *Eighth iteration* - 0.528 s

(i) *Ninth iteration* - 0.594 s

(j) *Tenth iteration* - 0.66 s

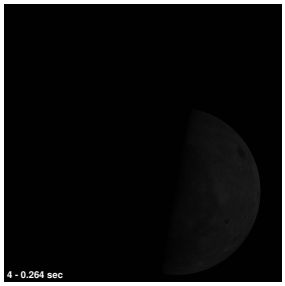
Figure 4.3: *Background function set up phase. Representation of the images created through the background function as the iterations increase from a) to j).*



(a) *First iteration* - 0.066 s

(b) *Second iteration* - 0.132 s

(c) *Third iteration* - 0.198 s

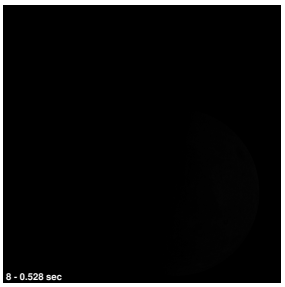


(d) *Fourth iteration* - 0.264 s

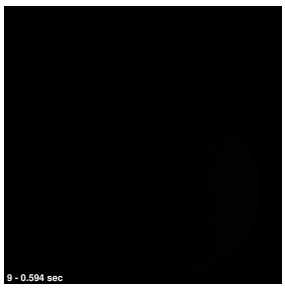
(e) *Fifth iteration* - 0.33 s

(f) *Sixth iteration* - 0.396 s

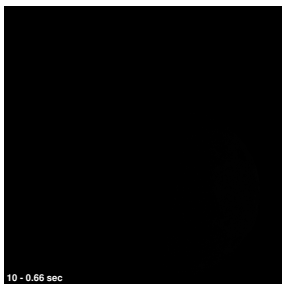
(g) *Seventh iteration* - 0.462 s



(h) *Eighth iteration* - 0.528 s

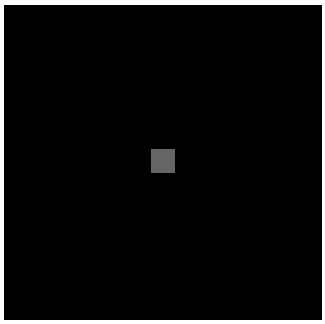


(i) *Ninth iteration* - 0.594 s



(j) *Tenth iteration* - 0.66 s

Figure 4.4: *Difference image during background function set up phase. Representation of the images created through the difference of the acquired image and the background function as the iterations increase from a) to j).*



0	0	1	1	0	0	0	1	1	1	0	1	1
0	1	0	1	1	0	0	0	0	1	0	1	1
0	0	0	1	1	0	0	1	0	1	0	1	0
0	0	0	1	0	1	0	0	0	0	0	1	0
1	0	1	0	0	1	1	0	1	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	1	1	1	101	0	1	0	1	1	1
1	0	0	0	0	0	0	1	1	0	0	0	0
1	0	1	1	1	0	1	0	0	0	0	1	1
0	1	0	1	0	1	0	0	1	0	0	0	0
0	0	1	0	1	0	0	1	0	0	1	0	0
1	0	1	1	0	0	0	1	1	0	0	1	0
1	0	1	1	1	1	0	0	1	1	1	1	0

(a) *Focus on the image* (b) *Focus on the matrix*

Figure 4.5: *Focus around the flash. The focus is performed over the same image, around the same location with the same number of pixels (or cells) considered. On the left the focus is done on the acquired image, while on the right the same focus is performed on the matrix representing the image.*

4.3 Detection

It is here discussed the solution adopted to have the detection of the flashed pixel in the difference image generated at the previous step; the detection is performed by virtue of an adaptive threshold that evaluates mean over the image μ_{img} and the standard deviation of the pixels σ_{img}, setting as a threshold:

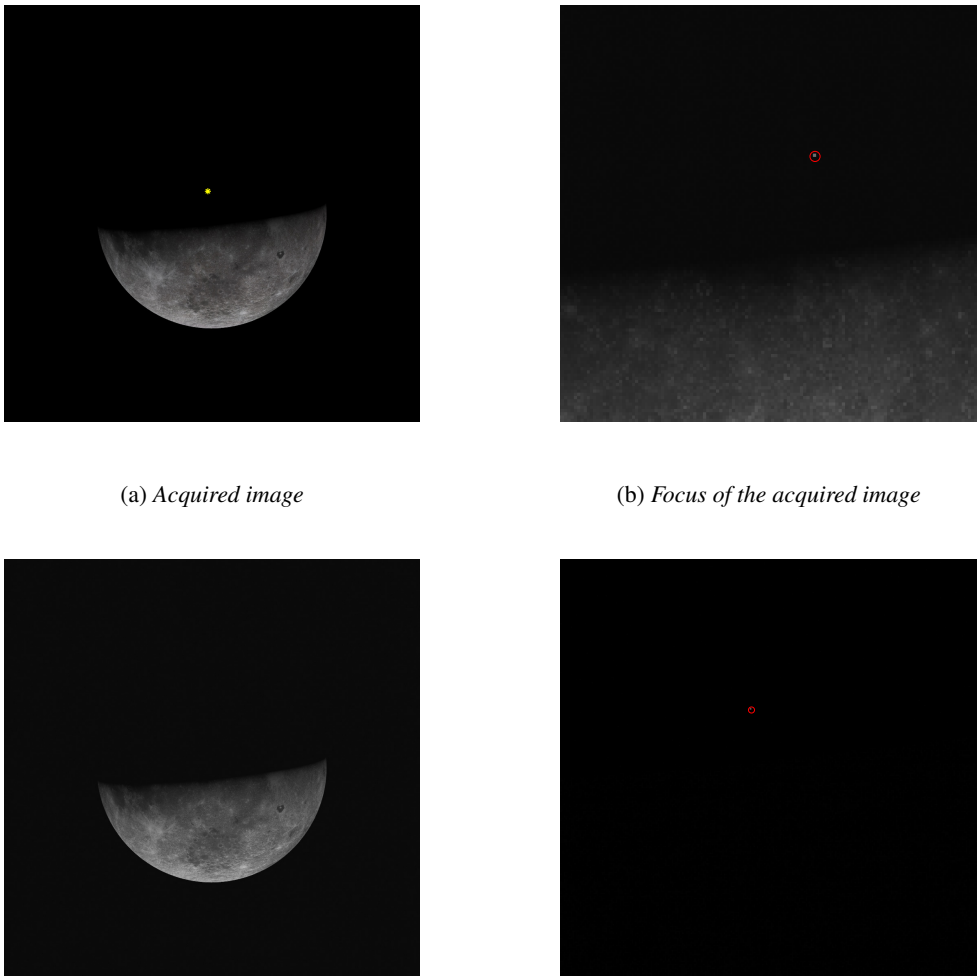
$$\mu_{img} + 5\mu_{img} \tag{4.3}$$

If a pixel with a value higher than this one is found, it will be considered as a flash. The number 5 has been chosen accordingly to the simulation performed in this work, by the way for further works this value should be changed accordingly to the situation, maybe using this choice as guess solution. Clearly if this number decreases, the number of possible false detection increases, while if this number increases the fainter flashes are filtered out. Both the mean and the standard deviation has been evaluated by using MATLAB® buit-in functions. In particular, giving as input a matrix, they are evaluated column-by-column; therefore as μ_{img} has been taken the mean of the means of the columns, while as σ_{img} has been taken the highest value of the standard deviation of the columns.

In order to validate this procedure it has been performed a simulation with 30 different images representing the Moon at different distance and different illumination conditions chosen random for 3000 different case with the flash put in different position and different intensity Table B.1 and the result says that the success chance is about 97.86%. Both the location and the intensities of the pixels have been generated in random condition, since the aim of this was not to simulate a real situation, but the reliability of the detection process. In this table are also shown the values of intensity used for each case. It can be seen that also very low values have been detected; moreover there are some cases in which the flash has been detected in the right way also in the Day-Side of the Moon, indicating a very reliable method to detect the changing pixel.

In Figure 4.6 is reported an example of how this simulation has been carried out;

Figure 4.6a represents the image of the Moon used in the simulation; the yellow spike is the flash location. In Figure 4.6b it is show a focus of the actual image acquired by LUMIO-Cam; the flash is almost invisible at naked eye, even if the value given to the pixel is 165 over 255. Figure 4.6c shows the background retrieved with the background function at that iteration. As can be seen this last image



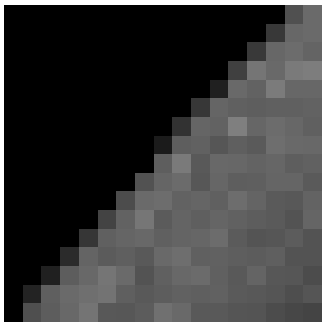
(a) *Acquired image*

(b) *Focus of the acquired image*

(c) *Background function*

(d) *Focus of the difference function*

Figure 4.6: *Detection test images. In a) the detected image is represented; the yellow spike highlights the flash. In b) it is shown the focus of a) around the flash location (white pixel highlighted by the red circle). In c) is represented the background function used and in d) the focus of the delta image around the flash location (white pixel highlighted by the red circle).*



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	106	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	103	106	
0	0	0	0	0	0	0	0	0	0	0	0	0	57	103	108	100	
0	0	0	0	0	0	0	0	0	0	0	0	64	117	105	122	125	
0	0	0	0	0	0	0	0	0	0	0	32	105	98	123	103	103	
0	0	0	0	0	0	0	0	0	0	55	101	96	94	96	96	98	
0	0	0	0	0	0	0	0	0	50	101	98	127	102	108	103	96	
0	0	0	0	0	0	0	0	27	94	97	86	95	90	110	105	104	
0	0	0	0	0	0	0	0	94	122	89	104	103	99	102	94	97	
0	0	0	0	0	0	0	82	101	107	87	102	94	95	101	101	90	
0	0	0	0	0	0	76	106	109	98	101	96	109	100	90	88	99	
0	0	0	0	0	67	98	118	91	100	95	101	97	93	90	83	101	
0	0	0	0	54	96	101	104	96	100	102	108	91	84	86	92	83	
0	0	0	43	100	91	95	89	105	110	96	97	94	92	95	83	83	
0	0	32	93	104	117	106	83	96	104	107	96	89	100	87	86	76	
0	35	95	106	111	115	95	89	93	97	95	97	90	88	85	78	85	
0	79	93	104	116	88	87	92	112	105	89	89	85	84	76	71	68	

(a) *Edge image* (b) *Edge matrix*

Figure 4.7: *Edge of the Moon representation. In the two images it is represented the same scene. On the left it is shown the focus around the Moon edge, as captured by the camera, while on the right the matrix representation of the edge, highlighted by the high gradient of values.*

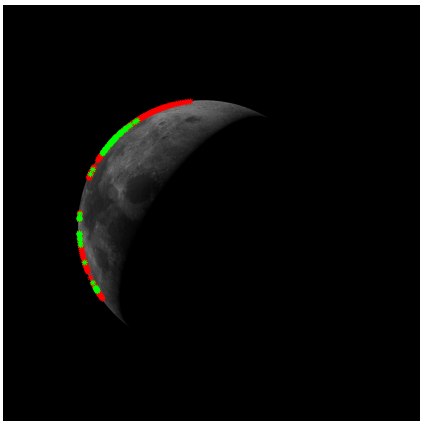
is very similar to the acquired one, except for the flash, that is shown in Figure 4.6d. This image represent a focus of the difference function around the flash location, and can be clearly seen in front of all the black background. Finally the detection by the algorithm is performed by evaluating the mean μ_{img}, resulting equal to 0.0956 and the standard deviation σ_{img}, equal to 5.1725, both evaluated over Figure 4.6d; the threshold is hence $\mu_{img} + 5\mu_{img} = 25.94$, and the only pixel that is detected is the actual one.

4.4 Moon edge reconstruction

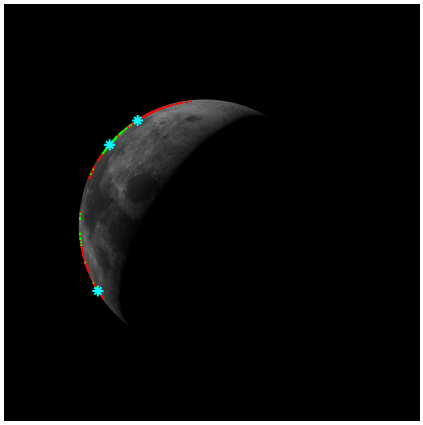
It is here validated the procedure followed by the algorithm in order to find the centre of the Moon starting from the image that is captured by LUMIO-Cam and reconstruct the edge. Since the only input available is the current image, the information can be retrieved only by using this. In particular it is possible to exploit the fact that for the vast majority of the time at least a small portion of the Moon will be illuminated by the Sun, and therefore the contrast with the black background of the space will be high Figure 4.7. For this kind of method there is an issue when the Moon Far-Side is completely in night, since in this case the algorithm fails; to solve this issue if the algorithm cannot find enough points to find the Moon edge the output is set to be `NaN` and this step is avoided. Following Figure 4.8 it is explained in detail the procedure adopted;

First step shown in Figure 4.8a is to select all the points with an high gradient with respect to the neighborhood. The value adopted here has to be set properly according to the support used and to the quantization of the A/D converter, since a too conservative value could lead to a poor number of points and therefore to a wrong detection, meanwhile in the other hand could be taken also points not on the edge. Each red point in figure represent a cell with a value higher than 80 with respect to the neighbor one. Nevertheless not all these points can be used, since there could be the possibility of having inside the edge some points with high difference values among them and therefore a point could be inside. Even if the probability of getting an inner points is very low, a second step is performed to reduce more this possibility. Since the red points are found via a for-loop over all the pixels, those are found and listed going from the upper left to the lower right of the image, row per row. The coordinates (rows and columns) are saved in a vector ordered according to the procedure adopted to scan each pixel on the image. In this way the same position in both the vector will give a row and a column, that put

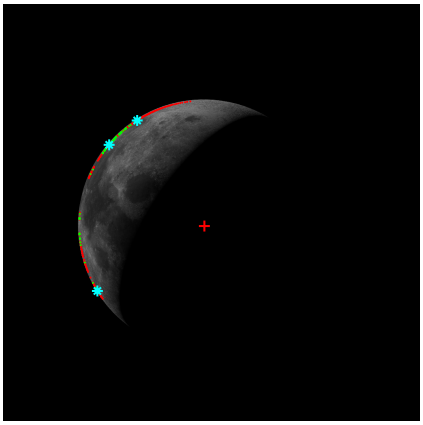
(a) *High gradient points*



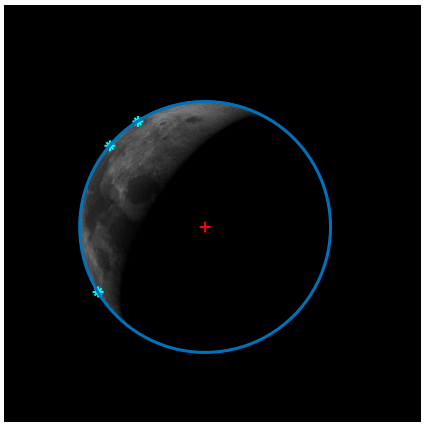
(b) *In green the points that are kept*



(c) *Points to evaluate the circumference*



(d) *Centre of the Moon*



(e) *Circumference*

Figure 4.8: *Moon circle reconstruction procedure. In a) are represented all the selected points. In b) are kept only the green ones. In c) the blue cyan dots represent the three points taken to evaluate the centre and the circle like in d) and e).*

together will recover the coordinate of the pixel that is selected. Hence, for example the far right pixel of the first row will be put near the far left of the second row, but will be very far in the vector from the far right pixel of the second row, in all those are counted. Exploiting this it is possible to evaluate the distance in pixels between consecutive points in the vector (in this case are evaluated the distances between a point, the previous and the following one). As they are listed in the vectors as stated before, the only way of a point to be near (distance between them has to be less than 2 pixels) to both the previous and the following one either in the vector and in the image is to lay on the edge of the Moon and to have no points detected on the same row; doing this the method leads to Figure 4.8b, where green points are those that are kept, while the red ones are the discarded. Among the green points that are for sure near each other and with high probability on the edge of the Moon, three points are chosen, the ones represented in cyan in Figure 4.8c. Those points are taken as the first, the last and the middle points in the vector; indeed the aim is to estimate the circumference to which those points belongs, since through three points pass one and only one circumference, and the farther those points are each other, the better the resolution of the circle it is. The procedure it is shown if Figure 3.4. The result is the coordinate of the centre, depicted as a red cross in Figure 4.8d. Once the coordinate centre are know the distance between this and a point on the edge (for example one of the one used to reconstruct the circle) is the radius. Putting all together it is possible to draw the circle as in Figure 4.8e.

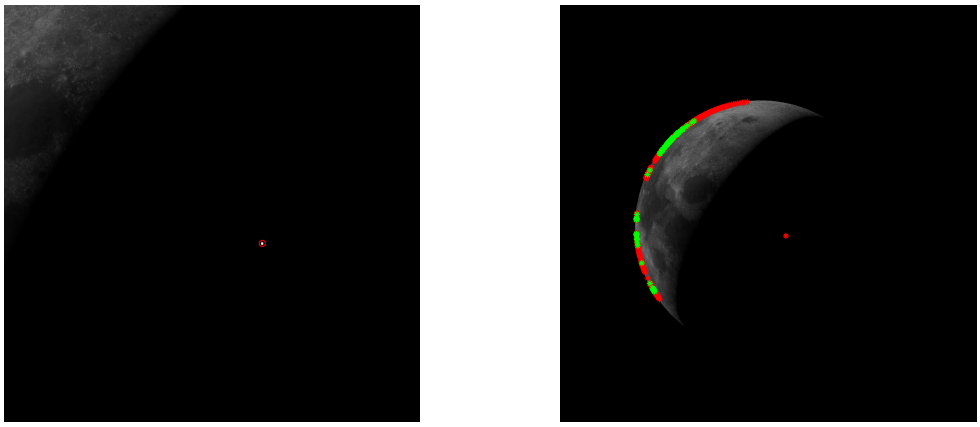
This procedure avoid also to take a flash as edge point, even in the worst condition possible, that is when a flash is spread over four pixels, as shown in Figure 4.9. In the left image is reproduced the flash, that in this case occupies four pixels (the white dot); as stated before this is the limit case due to the fact that the impact location is in correspondence of the intersection of four pixels. Therefore it is impossible to have cases in which three consecutive pixels on the same row or column are lighted. The way the situation is managed is shown in Figure 4.9b; indeed those points are selected in the first check since the difference with the background is higher than 80, but are not kept after the second check since those do not respect the request of being near both in the vector that in the image, and therefore being discarded are depicted in red. This case guarantees that also the other cases are well managed, being for sure the number of lighted pixel lower. Furthermore, since stars in the background can be treated exactly as flash are treat this case ensures that also those are filtered out. In Table B.1, are also shown the results of the evaluation of the centre of the Moon of the detected flash. The detection is always successful, since the detected centre coordinates and radius, even if are not equal to the reference ones, differs from them only of few pixels.

4.5 Flash lasting more than a frame

It is here presented the procedure that is followed to summarize all the data coming from a flash that last for more than a single frame. Due to the fact the integration time of LUMIO-Cam is 66 ms, there could be the possibility of having a flash that last for more than one single frame, mainly for two reasons (Figure 4.10);

- A flash last for more that 66 ms
- A flash last less than 66 ms, but starts later than the integration time

Therefore, it is useful to collect all the information that a single event spreads over two frames to reconstruct properly the dynamics, this step has to be performed in the best way, trying not to ruin the detection information that are taken in input. The rationale behind this procedure is to compare the



(a) *Flash inside the Moon* (b) *Points to evaluate the Moon circle*

Figure 4.9: *Robustness of the Moon edge reconstruction procedure. On the left the flash inside the Moon is represented, highlighted by the red circle, while on the right it is shown that the point is taken into account since has an high gradient, but it is discarded.*

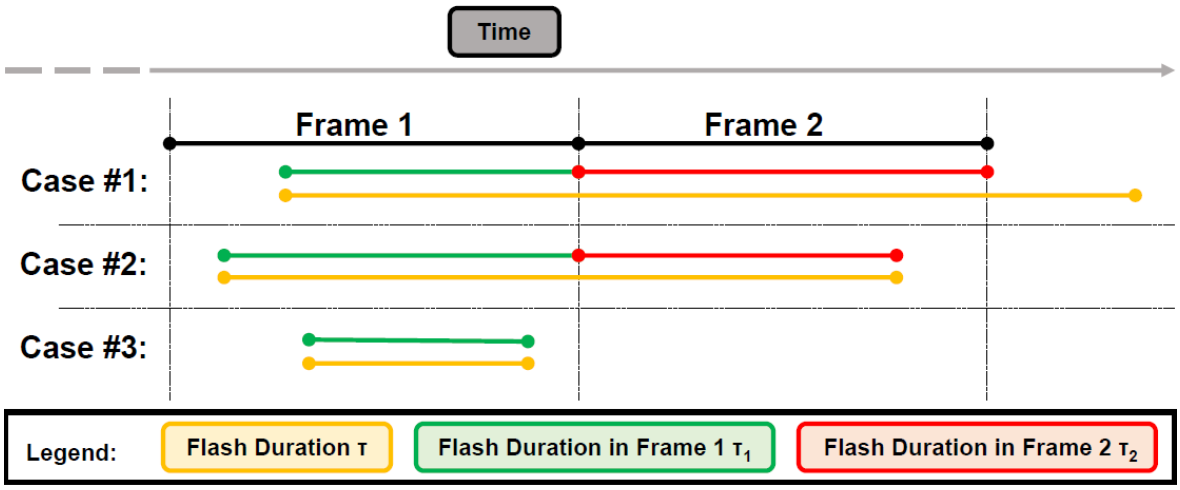


Figure 4.10: *Flash duration possibilities. Schematic representation of the cases in which a flash can happen inside a single or multiple frames. Credits to:* Merisio (2018).

Figure 4.11: *Test reference image. Flash are highlighted by red circles.*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
296.0000	623.0000	255.0000	632.0000	473.0000	296.0000	0.0667
297.0000	624.0000	255.0000	632.0000	473.0000	296.0000	0.0667
486.0000	442.0000	255.0000	632.0000	473.0000	296.0000	0.0667
486.0000	443.0000	255.0000	632.0000	473.0000	296.0000	0.0667

(a) *Information after first frame*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
296.0000	623.0000	129.0000	632.0000	473.0000	296.0000	0.0667
297.0000	624.0000	104.0000	632.0000	473.0000	296.0000	0.0667
486.0000	442.0000	204.0000	632.0000	473.0000	296.0000	0.0667
486.0000	443.0000	104.0000	632.0000	473.0000	296.0000	0.0667

(b) *Information after second frame*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
296.0000	623.0000	384.0000	632.0000	473.0000	296.0000	0.1333
297.0000	624.0000	359.0000	632.0000	473.0000	296.0000	0.1333
486.0000	442.0000	459.0000	632.0000	473.0000	296.0000	0.1333
486.0000	443.0000	359.0000	632.0000	473.0000	296.0000	0.1333

(c) *Information after the multiframe algorithm*

Figure 4.12: *Detection process for events lasting more than a frame. In a) it is represented the output of the detection on the first frame, in b) the output of the detection on the second frame, while in c) the information of a) and b) are combined together.*

information that are found in the image just processed and to compare them with the one that are saved from the step just before. If no information are already present, this step is not performed, data are saved for the next iteration and the algorithm goes on. Vice versa it is here shown the full procedure for a two frames duration flash; this situation can then be extended for all the flash that last for more than a frame. A first check that is performed is that the flash pattern, namely the position of all the lighted pixels, is the same between the two frames. Indeed it could happen that passing from a time step to the following one, the location of the lighted flash change, due to the presence of different showers or due to the cooling of the impact plume. To not have any problem with the algorithm it is stated that if a situation like this happens, the two frames are counted as made by different events and therefore are both saved.

In the case instead the pattern is found to be the same, it is followed the procedure shown in Figure 4.12. Starting from a situation like the one represented in Figure 4.11, where, to be more visible, the detected flashed are highlighted with red circles, supposing that the detection has been already performed, the information that are retrieved from the first frame are shown in Figure 4.12a. In particular the first two columns represent the flash location row and column, respectively, the third the pixel intensity and the last the duration; obviously for each frame the duration is set to be equal to the integration time and therefore to 66 ms.

The second frame (the following one) present the same flash pattern of the previous one, and therefore the printed information in Figure 4.12b are similar. The two patterns are considered as equal since the the two information, Figure 4.12a and Figure 4.12b have the same coordinates that has been found. Even if only a row or a column of Figure 4.12b would have been different from Figure 4.12a,

the two frame would have been considered as different. Nevertheless, since in this situation both the column that the rows are equal, the algorithm recognize that two consecutive frames have the same pattern and therefore the information are put together as shown in Figure 4.12c; has to be noted that it is not required that passing from one frame to the following the intensities of the pixels remains constant. Indeed those will be summed in Figure 4.12c, as well as the time duration.

As already said, this procedure can be extended for n frames with the only constraint of having the same detected pattern for all the frames, with the change with respect to this example of having as information saved from the previous step the synthesis of the $n-1$ frames before. The other way around, this procedure is interrupted as soon as a frame without any detection is founded.

4.6 Flashes spread over more pixels

Here it is shown the procedure followed by the algorithm in a single frame if more than a single lighted pixel is found. Considering Figure 4.11, and supposing this time that the event duration is all represented in one frame, the information entering at this stage are the same shown in Figure 4.12a. The aim here is to collect all the near lighted pixel in order to reduce the amount of the outputs, and to collect all those information that are generated by a single event. Looking at Figure 4.13 there is the focus on the two locations where the lighted pixels are placed; Figure 4.13 is composed by two frames because just looking at the full image is it clear that the location are very far one from the other. Nevertheless in the algorithm this consideration is performed by evaluating the distance between a row of Figure 4.12a with the other until all the rows are not merged or considered as made by different events. Looking at the image the expected output is to have two information representing the two locations with a pixel intensity equivalent to the summation of the one that are merged. This is what actually happens, as show in Figure 4.14; the coordinates of the flash are represented in the first two columns while the third one shows the total value of the pixel intensity, according to the summed pixel per each location. Note that in this case the information about the time is not changed since this procedure is performed once a flash is considered concluded.

Indeed even if an impact event would last for more than one frame, this procedure is performed only after the algorithm has found or an image with a different pattern or an image without a flash, and therefore in the single frame there is the full event information.

4.7 Latitude and longitude

The coordinates are retrieved by the algorithm using the information about the geometry of the Moon found in the image up to this point. The coordinates are referred to the Moon centre that has been found before. The image acquired by LUMIO-Cam is reproduced in Figure 4.15; the flash is highlighted by the yellow spike. From previous steps the centre coordinates and the radius of the Moon in the image reference frame are known, and therefore it is possible to produce Figure 4.16, where the green cross is the centre and the red lines represent the axes aligned with the equator and the prime meridian. In

(a) *First location* (b) *Second location*

Figure 4.13: *Focus over the flash locations. In this case there are two flash location, one represented on the left and the the represented on the right. Both are composed by multiple pixels.*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
296.0000	623.0000	510.0000	632.0000	473.0000	296.0000	0.0667
486.0000	442.0000	510.0000	632.0000	473.0000	296.0000	0.0667

Figure 4.14: *Output after the algorithm has detected near lighted pixels. Note that in this case the information are two, since the impact location are two.*

this reference frame it is possible to evaluate two quantities that in Figure 4.16 are evaluated as:

$$\Delta x = X_f - \hat{X}_{c\mathbb{Q}} \tag{4.4}$$

$$\Delta y = -(Y_f - \hat{Y}_{c\mathbb{Q}}) \tag{4.5}$$

$$\tag{4.6}$$

Once these quantities are known to find latitude ϕ_f and longitude λ_f angles; to evaluate the latitude ϕ_f the step to follow are straightforward since according to the reference frame this angle is proportional to the quantity Δy. Vice versa to retrieve longitude angle λ_f some geometric consideration has to be done, as shown in Figure 4.17. Indeed the latitude angle, since has to be represented on a sphere in $2D$, is dependent both of Δx that on the latitude angle ϕ_f. Note that the quantity R_{eq} is the projection of the radius over the plane parallel to the equator, passing through the impact location. The first step to evaluate the longitude λ_f is shown in Figure 4.17a, and consist of evaluate R_{eq}, namely the distance between the centre and the projection of the flash, by knowing all the other quantitie. The second Figure 4.17b is to evaluate the angle by knowing R_{eq} and Δx; in this image the observer is places in a position view position above the sphere

The angles are printed in degrees with the sign, accordingly to the direction; latitude is positive going North, while Longitude is positive going East, as summarized in Figure 4.18

Figure 4.15: *Reference image to evaluate latitude and longitude. The yellow spike represent the flash location.*

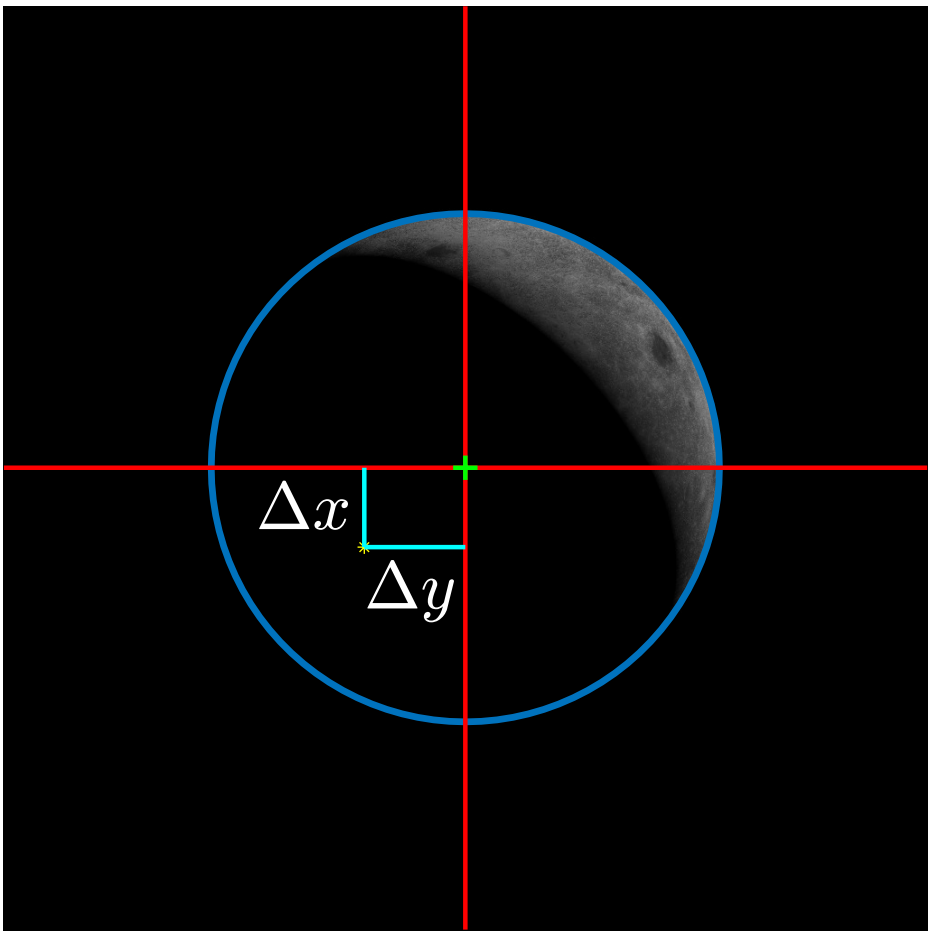
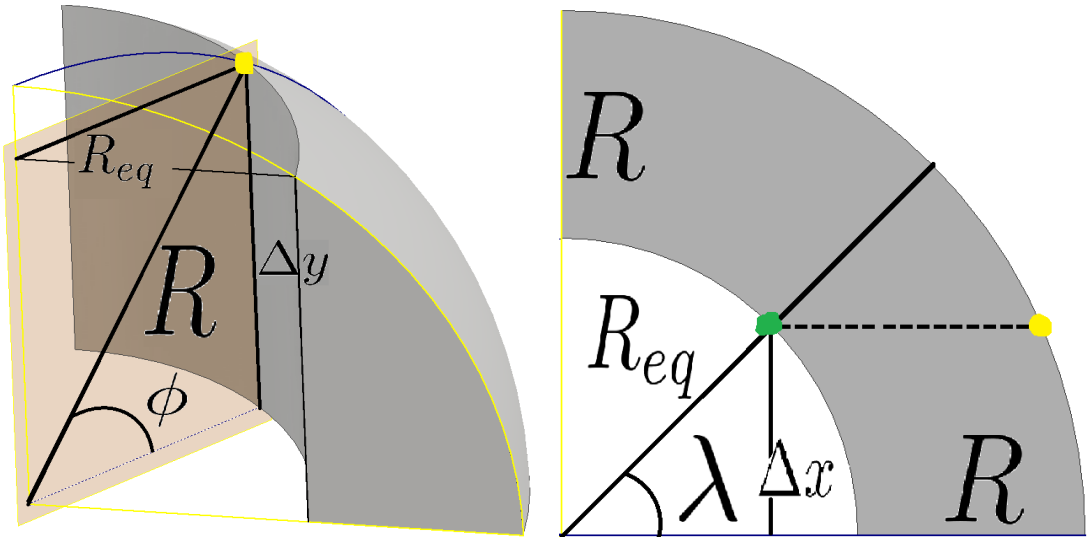


Figure 4.16: *Schematic representation of* Δx *and* Δy*. Red lines represent the fundamental axes to which the coordinates are referred to.*



(a) *First step, evaluate R_{eq}*

(b) *Second step, evaluate λ_f. IN this case the sphere is seen from above*

Figure 4.17: *Validation procedure for the evaluation of the longitude.*

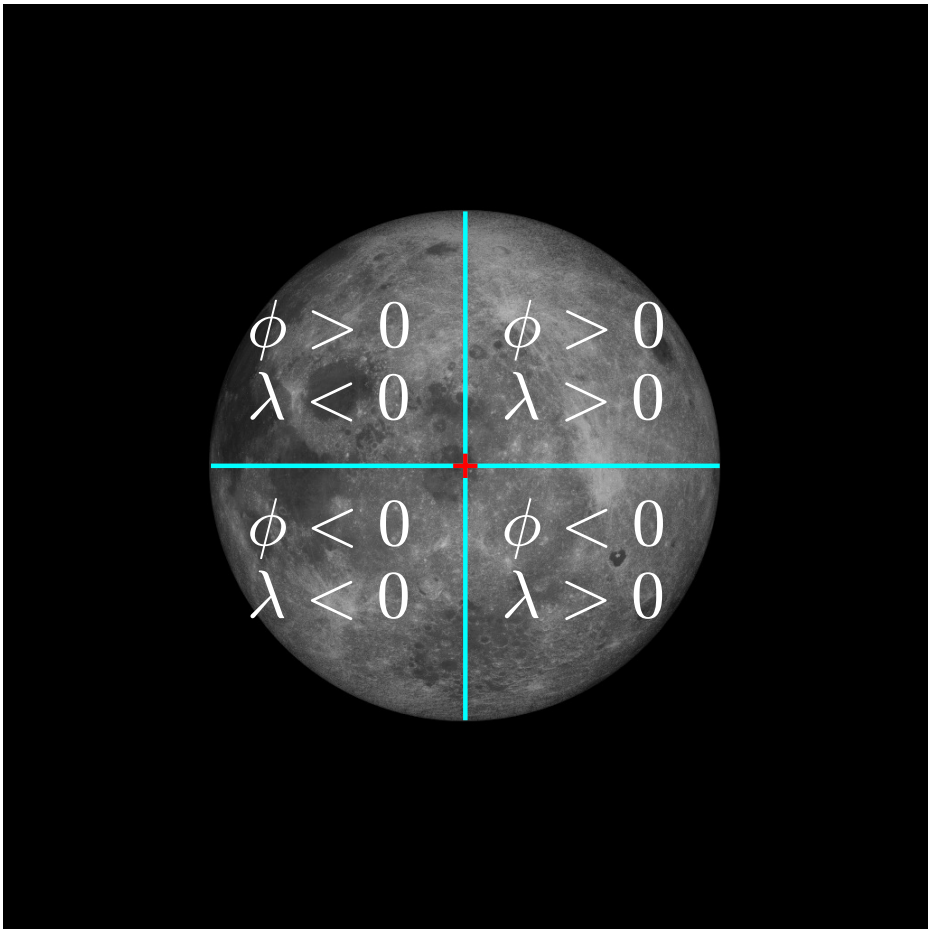


Figure 4.18: *Coordinate angle sign convention.*

Table 4.1: *Data used to simulate the post-process phase.*

Parameter	Symbol	Value	Motivation
Degradation coefficient due to light diffraction	f_{ld}	0.84	LUMIO-Cam model
Aperture diameter	D	55 mm	LUMIO-Cam model
LUMIO-Cam boresight-impact vector angle	ϑ	0°	Random
Anisotropy degree of light emission	f	3	LUMIO-Cam model
Impact distance	d	36000 km	Random
Plume area	A_p	25 m^2	Random
Impact duration	t	50 ms	Random
Impact black body temperature	T_{imp}	2500 K	Random

4.8 Temperature and magnitude

The steps done to find the black body temperature and the magnitude of the impact are here validated; this task is accomplished by virtue of a simulation, by setting as input a black body temperature and by checking that the output given by the algorithm is consistent with the data used. The first step is to generate a signal representing a possible impact of a meteoroid on the surface. This is performed by using the formula:

$$S_{imp} = f_{ld}\frac{D^2\cos(\vartheta)}{4fd^2}A_p t\int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, T_{imp})\, d\lambda \tag{4.7}$$

Since the aim of this simulation is to validate the algorithm to reconstruct the temperature from the image, all the physical and orbital variables can be taken random; in Table 4.1 are listed all the values used.

Integrating this equation either for VIS, with λ_1 =350 nm and λ_2 =820 nm, and for NIR, with λ_1 =820 nm and λ_2 =950 nm the impact signal for both channels are retrieved (S_{imp}^{VIS}=5.0044×10^4 e^-, S_{imp}^{NIR}=2.5320×10^4 e^-); these signals are then used directly of the frame by converting this on a quantizied value as:

$$\hat{N} = \left\lfloor \frac{N_{max}\times S_{imp}}{C_{max}} \right\rfloor \tag{4.8}$$

The results obtained are, for VIS, N_{VIS}=159, and for NIR, N_{NIR}=80.

Assuming that all the detection chain has been concluded, these two represent the total intensities of the pixels, and therefore represent the energy of the impact. The impact signal can be retrieved for both channels, as:

$$\hat{S}_{imp} = \hat{N}\times\frac{C_{max}}{S_{imp}} \tag{4.9}$$

obtaining S_{imp}^{VIS}=4.9882×10^4 e^-, S_{imp}^{NIR}=2.5098×10^4 e^-. As can be seen the these values are not equal to the previous one, since a quantization process has been performed when the image has been acquired. Performing therefore the inverse process it is possible to evaluate the black body temperature of the event. The first quantity that can be evaluated is $A_p\int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, T)\, d\lambda$, both for VIS and NIR, as:

$$A_p\int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda = \frac{4fd^2\hat{S}_{imp}}{f_{ld}D^2\tau\cos(\vartheta)} \tag{4.10}$$

Indeed on the right-hand side all the quantities are known, and listed in Table 4.1. Nevertheless it is

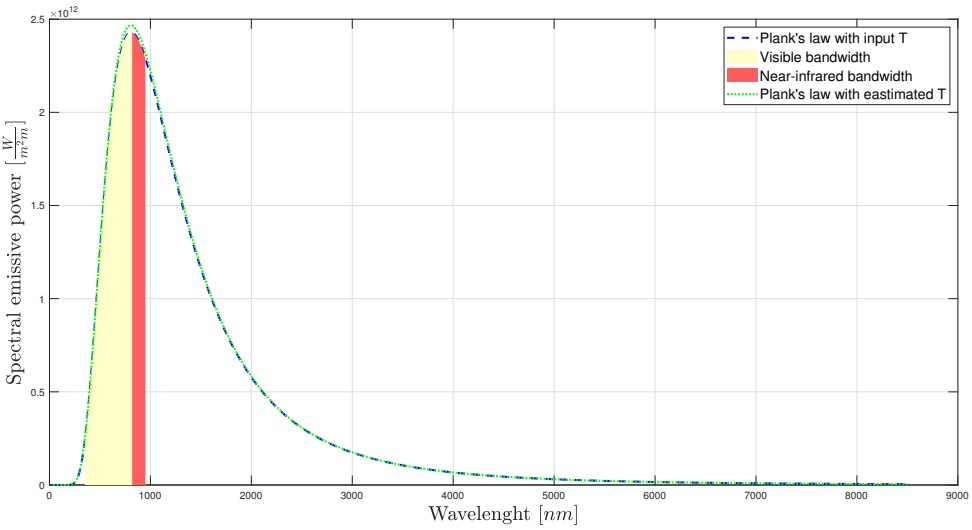


Figure 4.19: *Plank Law of both the input and output temperature. Dashed blue line represent the input temperature, while green dotted line is the output one. The two area corresponds to the VIS and NIR.*

important to point out that the duration of the event t is changed in τ since now it is set to be equal to 66 ms, since the information is retrieved by the frame and all the time instants will be multiple of the integration time. Known the quantity on the left-hand side for both VIS and NIR, the black body temperature $\hat{T}_{imp}$ can be found applying the Plank's Law for both the channels, knowing that the ration of the left-hand sides it is equal to the ratio of the right-hand sides, as

$$\frac{A_p \int_{\text{VIS}} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda}{A_p \int_{\text{NIR}} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda} = \frac{\int_{\text{VIS}} L(\lambda, \hat{T}_{imp}) \frac{\lambda QE(\lambda)\xi}{hc}\, d\lambda}{\int_{\text{NIR}} L(\lambda, \hat{T}_{imp}) \frac{\lambda QE(\lambda)\xi}{hc}\, d\lambda} \tag{4.11}$$

Obtaining a value for $\hat{T}_{imp}$=2507.6 K. Also in this case the value is different, but near to the one used as input, due to the way this last value has been estimated. In Figure 4.19 are shown the Plank's curve of both the input and the estimate temperature. The error between the input and the output temperature it is evaluated to be under the 3%.

Using the quantity $A_p \int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda$ for one channel, it is also possible to evaluate the plume area ($\hat{A}_p$). Indeed this data cannot be estimated by the image due to the fact that the area under a single pixel is far higher also than the biggest plume area. Multiplying therefore this quantity times $\int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda$, choosing λ_1 and λ_2 accordingly to the channel chosen, and $\hat{T}_{imp}$ the one just estimated, the value of $\hat{A}_p$ found is equal to 18.21 m^2 (in this simulation evaluated by using VIS). In this case even if the size is quite different with respect to the initial value, the order of magnitude is comparable, therefore the result can be acceptable; the difference with respect to the input value is mainly due to the fact that as duration of the flash has been taken the integration time, since in principle no other information are available. This observation is confirmed by the fact that using as impact duration time 0.05 ms also in this case the result would be 24.92 m^2, which differs less than the 3% from the input.

Last information that can be estimated is the magnitude at the satellite distance; first of all can be

evaluated the black body power of the impact as:

$$P_{imp} = \hat{A}_p \int_{\lambda_1}^{\lambda_2} \frac{2\pi hc^2}{\lambda^5 e^{\frac{hc}{\lambda k_B \hat{T}_{imp}}} - 1} \, d\lambda \tag{4.12}$$

using both $\hat{A}_p$ and $\hat{T}_{imp}$ just retrieved, obtaining P_{imp}=3 0901d6 W Finally the apparent magnitude m_{imp} can be evaluated by:

$$m_{imp} = m_0 - 2.5 log_{10} \frac{P_{imp}}{F_0 f \pi d^2 |\lambda_2 - \lambda_1|} \tag{4.13}$$

getting in this case a value equal to 4.6, corresponding to a faint star that can be seen at naked eye by a man on the Earth.

5

Performance simulation

In this chapter are shown four simulations of the methodology developed; data used to perform those simulations have been generate according to the models available regarding both the NEOs environment and the impacts dynamics. Data so generated have been used either for the detection that for the post-processing.

5.1 Real data simulation

Real data for the simulation are generated according to the methodology developed in Merisio (2018); In particular by starting from the impact velocity, and the kinetic energy of the impactor it is possible to estimate all the quantities that are needed to create a consistent environment to create an impact dynamics and by this a set of images to have the simulation of the algorithm.

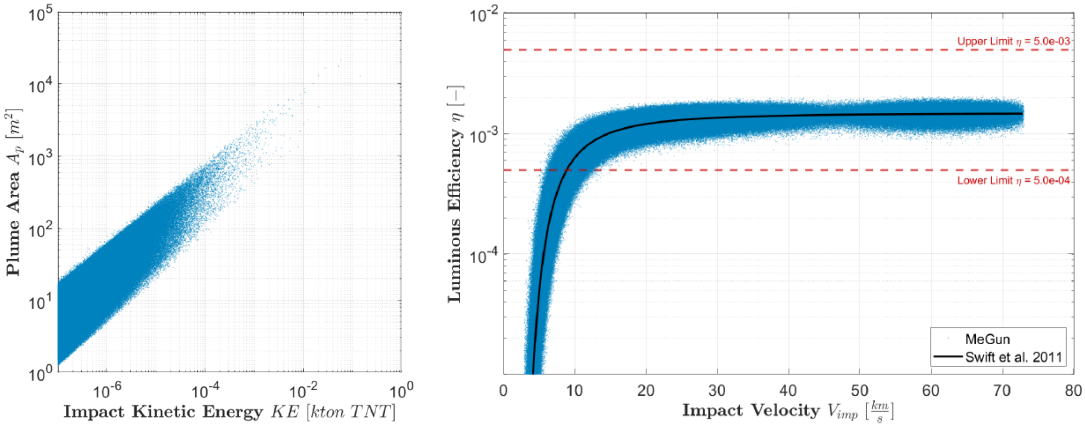
The starting point is the generation of a random value of the kinetic energy; by using the graphs represented in Figure 5.1, the values of both the kinetic energy of the impactor that the impact velocity can be estimated by sampling[1] all the values represented on the x-axis. In this way it is not possible only to find the range between the values are, but also the probability of a single value to happen, since a more probable value will be more sampled. As can be seen in Figure 5.1a., the values of the kinetic energy are expressed in kton TNT, and therefore have to be converted into Joule [J] to be properly used.

Once the value of the impact velocity is know it is possible to recover the magnitude of the orbital velocity (the magnitude) as:

$$V = \sqrt{{V_{imp}}^2 - 2\frac{\mu_{\leftmoon}}{R_{\leftmoon}} - 2\frac{\mu_{\oplus}}{d_{\oplus\leftmoon}}} \tag{5.1}$$

where $\mu_{\leftmoon}$ is the Moon's gravitational mass parameter, $\mu_{\oplus}$ the Earth's gravitational mass parameter

[1] through WebPlotDigitizer `https://automeris.io/WebPlotDigitizer/` [Last visited on 23/11/2019]



(a) *Image used to estimate the Kinetic Energy*

(b) *Image used to estimate the Impact Velocity*

Figure 5.1: *Reference images for data generation. On the left it is shown the graph representing the distribution of the Plume Area A_p with respect to the Kinetic Energy KE, while on the right the relation between the Luminous Efficiency η with respect to the Impact Velocity V_{imp}. Credits to:* Merisio (2018).

and $d_{\oplus\leftmoon}$ the mean distance between the Moon and the Earth. By definition, the kinetic energy can be written as:

$$KE = \frac{1}{2} m {V_{imp}}^2 \tag{5.2}$$

with KE the kinetic energy from which it is possible to recover the value of the mass m

As already introduced, when two solid bodies impact each other, not all the energy that is generated is converted in luminous energy, since other phenomena take place together with this; by the way considering only the light emitted by the source, it seems that a certain amount of energy has gone lost, a quantity that is estimated by virtue of the Luminous Efficiency η. According to the definition given by Swift et al. (2011), this can be written as:

$$\eta = Ce^{-\frac{V_c^2}{{V_{imp}}^2}} \tag{5.3}$$

with C and V_c generated by selecting random values over a normal distribution produced according to μ_C =1.5 $\times 10^{-3}$ and $\sigma_C = 0.1\mu_C$, and μ_{V_c} =9.3 km s^{-1} $\sigma_{V_c} = 0.1\mu_{V_c}$ km s^{-1}.

The bulk density of the Moon ($\rho_{\leftmoon}$) is taken as constant and equal to 1500 kg m^{-3}, while the one of the impactor body (ρ_i) is generated picking a random value in a Gaussian distribution with mean 1800 kg m^{-3} and standard deviation 300 kg m^{-3}.

Another data that is generated randomly through a Gaussian distribution is the impact angle γ, that represent the angle between the ground horizon and the impact vector, so that if the impact is vertical the angle would be 90°. In literature this value is assumed to be 45° (Cipriano, 2017; Madiedo et al., 2015b); by the way it is chosen to use a Gaussian distribution with mean 15° and standard deviation 15°, constraining the limit value not going over 90° and under 20°.

Once those quantities are known, the diameter of the crater made due to the meteoroid impact can be retrieved as (Bouley et al., 2012):

$$D = 0.25 {\rho_i}^{0.167} {\rho_{\leftmoon}}^{-0.5} g^{-0.165} KE^{0.29} \sin(\gamma)^{\frac{1}{3}} \tag{5.4}$$

This law is valid for impact with diameter up to 100 m; vice versa for higher impact diameters it is suggested to use the formula (Ortiz et al., 2015):

$$R = \frac{10.14}{1.2} G^{-0.17} \left(\frac{3m}{4\pi\rho_i} \right)^{\frac{0.83}{3}} V^{0.34} \sin(\gamma)^{\frac{1}{3}} \tag{5.5}$$

where ρ_i is the meteoroid density, g the standard acceleration of gravity, γ the impact angle with respect to the horizon, G is the gravity acceleration in units of Earth's gravity acceleration.

The impact of the meteoroid with the Moon surface created not only the crater, but also a plume of ejecta that rises and its size can be estimated as in Bouley et al. (2012):

$$A_p = \beta \pi R^2 \tag{5.6}$$

with beta a coefficient that is picked by a uniform distribution bounded between 4 and 25.

To estimate the time duration of the impact, and thus of the flash, it is used the law taken from Bouley et al. (2012):

$$t = Ae^{-bm} \tag{5.7}$$

where A and b are two constants, and m is the magnitude. A and b are evaluated selecting two random values in a normal distribution, with μ_A =77.6 s and $\sigma_A = 0.1\mu_A$ s and μ_b =0.94 and σ_b =0.06, while m is evaluated by using data so far available. The magnitude is recalled as:

$$m = \frac{1}{b + 0.4ln(10)} ln\left(\frac{k1}{k2} \right) \tag{5.8}$$

where $k1$ and $k2$ are evaluated as:

$$k1 = F_0 10^{\frac{m_0}{2.5}} f \pi d^2_{\oplus \leftmoon} \Delta\lambda_{vis} \tag{5.9}$$

$$k2 = \frac{5\eta KE}{A(1 - e^{-5})} \tag{5.10}$$

$$\tag{5.11}$$

The choice to use as distance the Earth-Moon and as bandwidth the visible is due to the validation limits the law to evaluate t has. Indeed this law has been developed for visible flash visible on the Earth and therefore the validity is this one.

Knowing all these quantities the black body temperature of the impact can be evaluated as:

$$A_p \int_{\lambda_1}^{\lambda_2} L(\lambda, T_{imp})\, d\lambda - \frac{\eta KE}{t} = 0 \tag{5.12}$$

This procedure has up to now simulated the physical problem, generating a random scenario. Now this data has to be inserted in the image and from this have to be detected. To accomplish this task it is used the definition of the signal used in the SNR:

$$S_{imp} = f_{ld} \frac{D^2 \cos(\vartheta)}{4fd^2} A_p t \int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, T_{imp})\, d\lambda \tag{5.13}$$

$$N = \left\lfloor \frac{N_{max} \times S_{imp}}{C_{max}} \right\rfloor \tag{5.14}$$

This value is quantizied over 255, the limit of the 8bit to find the value of the flash. In the case the value obtained is over 255, the flash is considered saturated and the value remains 255. The choice to take the floor of the computed value respect the fact that the energy to have a signal in the pixels forming the CCD is quantizied. Since a flash could last for more than 66 ms, the signal is evaluated for each frame, accordingly to the duration of the flash

5.2 Simulation framework

The simulation of the algorithm has been performed by using MATLAB® as simulation tool; The first step is the generation of the impact environment and of the signal to be inserted in the image, through the procedure just explained in Chapter5.1. The second step simulates the methodology developed for this work, performing all the steps that have been explained in Chapter 3. As input it is given the an image of the Moon, the flash location and the pixel intensity, according to the step before, and the time duration of the flash. The output of the detection phase it is then used to simulate the post-processing, through which are estimated the temperature, the area of the plume and the magnitude of the flash at S/C distance. The simulation of the detection is performed by restricting the science period over 10 images, one for each time step, for a total time of simulation of 0.66 s; among them it is selected an image to be modified accordingly to the input of the simulation (therefore if a flash last for more than 1 frames, also the successive frames are modified). Since in this time the Moon environment do not change, the image used from the first iteration to the tenth is always the same. Furthermore to solve the problem of the initialization of the background function, an initial loop is performed.

5.3 Raw image generator

Images of the Moon are generated by using POV-Ray (v.3.7), a ray tracer software able to generate high fidelity images. Those images are given as input for the simulation. This software takes as input a texture of the Moon, which reproduce with high accuracy its soil. By changing the parameters regarding distance of the S/C from the surface, position of the Sun and position of the Moon in the FOV, it is possible to create a set of different images to have the simulation. Those images are generated without any noise, and therefore a first modification is to add noise, modeled as random values pattern that modifies the intensity of the image.

5.4 Numerical examples

Here are reported four scenarios each representing a possible event. All the data are generated in a consistent way, accordingly to the models available, and the detection and post-processing is performed according to the methodology that has been developed for this work. As reference distances for the simulation has been taken the maximum and minimum distance of the S/C from the lunar surface, namely 33 340 km and 85 117 km (Merisio, 2018). For simplicity has been taken that the flash event starts in the same instant of the integration time. This do not happens always, since it could be that the

(a) *Image of the Moon at* 33 340 km *from the surface* (b) *Image of the Moon at* 85 117 km *from the surface*

Figure 5.2: *Reference images used in the simulation. Both the images have been generated through POV-Ray. On the left it is shown the Moon at the nearest point of the orbit, while on the right at the farthest point. The illumination conditions are generated by POV-Ray as well.*

impact event happens when the integration time has already started. Nevertheless the results that a situation like this gives are not different from the ones that are here presented. In Figure 5.2 are shown the images of the Moon used to perform the simulation.

5.4.1 Sample scenario 1

Environment generation This scenario simulates an impact that happens when the S/C is in its nearest position on the halo orbit around L_2 Lagrangian Point of the Earth-Moon system; the used image if the one represented in Figure 5.2a and in Table 5.1 are listed the distance from the surface and the angle between the boresight direction and the flash direction in LUMIO-Cam reference frame used in this simulation.

Table 5.1: *Orbital data of sample scenario 1.*

Parameter	Symbol	Value
LUMIO-Moon distance	d	33 340 km
Flash-boresight angle	ϑ	1.88°

According the the methodology presented in Chapter 5.1, are also generated the physical data representing the impact of the meteoroid with the Moon surface, collected in Table 5.1. Putting these information together it is possible to modify the raw image created using POV-Ray and insert the flash in the correct location, with the correct intensity; the value corresponding to the lighted pixel is retrieved using Eq. (5.13), and the signal that is produced is equal to 6.4235×10^4 e^-. To transform this information in a number that can be handle by the computer it is used Eq. (5.14), getting the actual value that has to be used in the simulation, in this case equal to 204.

Table 5.2: *Environmental data of sample scenario 1.*

Parameter	Symbol	Value	Origin
Kinetic energy	KE	2.46×10^8 J	Random
Impact angle	γ	64.99°	Random
Impact velocity	V_{imp}	8.1959 km s^{-1}	Random
Velocity	V	7.7109 km s^{-1}	Eq. (5.1)
Meteoroid density	ρ_i	1800 kg m^{-3}	Random
Meteoroid mass	m	7.3245 kg	Eq. (5.2)
Luminous efficiency	η	3.1409×10^{-4}	Eq. (5.3)
Crater diameter	D	5.44 m	Eq. (5.4)
Plume area	A_p	232.7526 m^2	Eq. (5.6)
Magnitude (at Earth distance)	$m_\oplus$	8.56	Eq. (5.8)
Impact duration	t	0.0420 s	Eq. (5.7)
Impact temperature	T_{imp}	2019.7 K	Eq. (5.12)

Detection simulation In Table 5.3 are summarized the inputs given to the simulation tool for the algorithm developed for this work. Note in particular that all the coordinates are given in a reference frame assuming an $x-y$ plane set with the origin of the top left of the image and the axes arrows going down and right. The way the inputs are written here ($[X\ X]$) is to indicate that the flash it is here spread over two pixels, and therefore also its intensity is spread over two pixels. Note also that the total value of the pixels used inside the simulation is slightly different with respect to the one generated before, this due to the presence of the noise.

Table 5.3: *Input data of sample scenario 1.*

Parameter	Symbol	Value
flash x-coordinate	X_f	[610 610]
flash y-coordinate	Y_f	[214 215]
Pixel value	N	[100 100]

In Figure 5.3 it is shown the flash location, highlighted through a red circle. The detection steps are shown in Figure 5.4. In particular Figure 5.4a represent the output of the detection algorithm just after the iteration in which the flash has been detected. As the value of t that has been generated is less than the integration time, the flash is expected to last for only one frame. Indeed in the successive iteration, as represented in Figure 5.4b the near lighted pixels are merged together and the output is produced.

In Table 5.4 it is represented the output given by the simulation of the algorithm for this scenario; note that the column and row information are given in the matrix reference frame, since the image is read as matrix by the algorithm. The coordinates of the centre and the radius are evaluated through the algorithm that uses the image to find the boundaries pixels and the values are expressed as coordinates in the $x-y$ plane introduced before. The value of the radius it is here expressed in pixels units. The

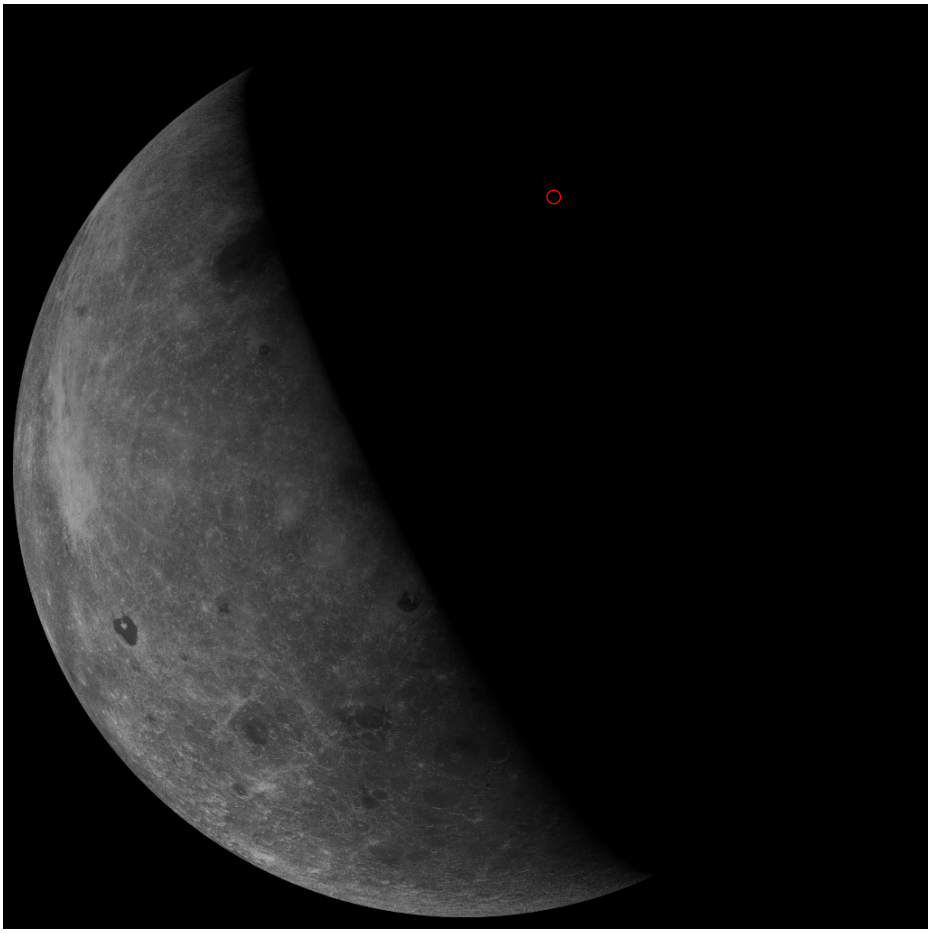


Figure 5.3: *Moon at sample scenario 1. The red circle represent the location in which has been inserted the flash.*

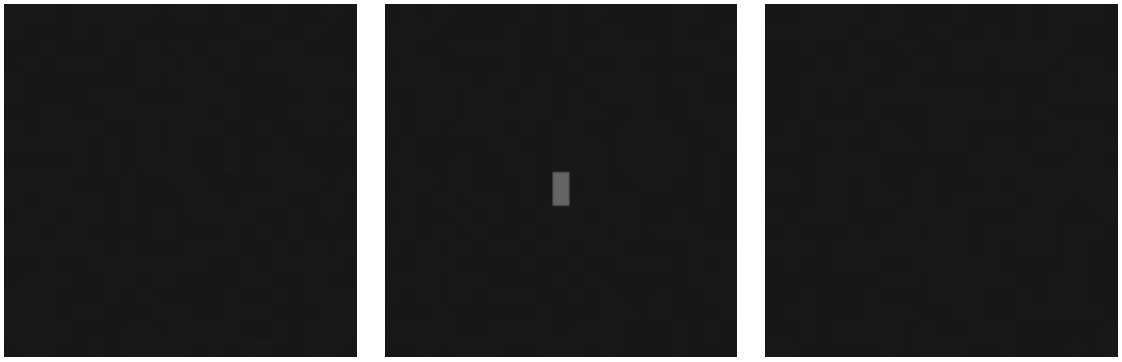
Row	Column	Intensity	Xcentre	Ycentre	Radius	time
214.0000	610.0000	100.0000	511.0000	513.0000	499.0000	0.0667
215.0000	610.0000	100.0000	511.0000	513.0000	499.0000	0.0667

(a) *First output, each representing a detected pixel*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
214.0000	610.0000	200.0000	511.0000	513.0000	499.0000	0.0667

(b) *Output after the check for near flash*

Figure 5.4: *Sample scenario 1 algorithm process. In image a) it is represented the first output of the algorithm just after the detection phase. In image b) the information represented previously are merged since the flash are near located. Note that the intensity has been summed, while the time step is always the same (one frame).*



(a) *Image just before the impact* (b) *Image of the impact* (c) *Image just after the impact*

Figure 5.5: *Output images of sample scenario 1. Those images are focus around the flash location of the output images of the simulation; the positions represented are the same in all the three image. On the left the situation just after the impact, of the right just after and in the center the impact.*

time is always set as an integer multiple of the integration time. Together with those information are also saved the images representing the Moon in the instant with the detection, and the ones just before and just after (Figure 5.5). Regarding the information about latitude and longitude, those values are consistent with Figure 5.3.

Table 5.4: *Output data of sample scenario 1.*

Parameter	Symbol	Value
Flash row	$\hat{R}$	214
Flash column	$\hat{C}$	610
Pixel value	N_{VIS}	200
Moon centre (x)	$\hat{X}_{c\mathbb{C}}$	511
Moon centre (y)	$\hat{Y}_{c\mathbb{C}}$	513
Moon radius	$\hat{R}_{\mathbb{C}}$	499
time	τ	0.066 s
Latitude	ϕ_f	36.8125°
Longitude	λ_f	14.3480°

Post-processing simulation All the step performed so far are executed on the image captured by the VIS channel. The information from the NIR channel in this are taken directly as data. Indeed in reality those data are by knowing the coordinates of the flash in VIS; since the two images are made facing the same scene, the impact location is the same and therefore once a channel is known, also the other is known. NIR are here generated accordingly to the physical data in Table 5.2, but integrating the signal over the near-infrared bandwidth as:

$$S_{imp} = f_{ld} \frac{D^2 \cos(\vartheta)}{4 f d^2} A_p t \int_{\lambda_1}^{\lambda_2} N_{e^-}(\lambda, T_{imp})\, d\lambda \tag{5.15}$$

And once is obtained the value in e^- is quantized as:

$$N = \left\lfloor \frac{N_{max} \times S_{imp}}{C_{max}} \right\rfloor \tag{5.16}$$

Putting all the information that have been retrieved by the image together with the one available from the telemetry (the orbit data), it is possible to post-process the data to find physical data about the phenomenon; the only unknown at this stage is the angle ϑ, that in the simulation is known, since it is needed to generate the data, but in the real application this value is not known a priori. Nevertheless this angle can be evaluated as:

$$\Delta = \sqrt{(\hat{X}_f - C_x)^2 + (\hat{Y}_f - C_y)^2} \tag{5.17}$$

where C_x and C_y are the coordinates of the centre in the cam reference frame both always equal to 512. Knowing the radius of the Moon both in the picture reference frame that the real one, and knowing also the distance from the Moon (from the telemetry)

$$\delta = R_{\leftmoon} \frac{\Delta}{\hat{R}_{\leftmoon}} \tag{5.18}$$

where $R_{\leftmoon}$ is the physical radius of the Moon and $\hat{R}_{\leftmoon}$ the one retrieved in the algorithm; the angle ϑ is therefore evaluated as:

$$\vartheta = \tan^{-1}\left(\frac{\delta}{d}\right) \tag{5.19}$$

$$\tag{5.20}$$

All the other values can be found using:

$$S_{imp}^{VIS} = \frac{N_{VIS} \times C_{max}}{N_{max}} \tag{5.21}$$

$$S_{imp}^{NIR} = \frac{N_{NIR} \times C_{max}}{N_{max}} \tag{5.22}$$

$$\frac{A_p \int_{\text{VIS}} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda}{A_p \int_{\text{NIR}} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda} = \frac{\int_{\text{VIS}} L(\lambda, \hat{T}_{imp}) \frac{\lambda QE(\lambda)\xi}{hc}\, d\lambda}{\int_{\text{NIR}} L(\lambda, \hat{T}_{imp}) \frac{\lambda QE(\lambda)\xi}{hc}\, d\lambda} \tag{5.23}$$

$$\hat{A}_p = A_p \int_{350\text{nm}}^{820\text{nm}} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda \times \int_{350\text{nm}}^{820\text{nm}} N_{e^-}(\lambda, \hat{T}_{imp})\, d\lambda \tag{5.24}$$

$$m_{imp} = m_0 - 2.5 log_{10} \frac{P_{imp}}{F_0 f \pi d^2 |\lambda_2 - \lambda_1|} \tag{5.25}$$

Output data from post-processing algorithm are listed in Table 5.5. Here can be seen that the value of the impact temperature that is retrieved is very near to the one that it is used in Table 5.2, while regarding the plume area estimation the value is different, but the order of magnitude is nevertheless correct. The magnitude is higher than the one found in Table 5.2, since in this case the source is nearer, corresponding to a faint star in the sky seen from the Earth.

Table 5.5: *Post-processing data of sample scenario 1.*

Parameter	Symbol	Value	Origin
NIR pixel value	N_{NIR}	154	Eq. (5.16)
Signal in VIS	S_{imp}^{VIS}	6.2745×10^4 e$^-$	Eq. (5.21)
Signal in NIR	S_{imp}^{NIR}	4.8314×10^4 e$^-$	Eq. (5.22)
Impact temperature	$\hat{T}_{imp}$	2000.7 K	Eq. (5.23)
Plume area	$\hat{A}_p$	159.32 m^2	Eq. (5.24)
Magnitude at S/C distance	m_{imp}	4.3638	Eq. (5.25)

5.4.2 Sample scenario 2

Environment generation This scenario is simulated in the nearest point as the previous scenario. Therefore also in this case the used image is the one represented in Figure 5.2a. In Table 5.6 are listed the orbital data used for this simulation.

Table 5.6: *Orbital data of sample scenario 2.*

Parameter	Symbol	Value
LUMIO-Moon distance	d	33 340 km
Flash-boresight angle	ϑ	2.72°

Following the same steps used before, and explained in Chapter 5.1, new physical values are created, shown in Table 5.7. The value of the signal that is generated is equal to 1.1192×10^5 e$^-$; since it is simulated in this scenario that all the signal is collected by a single pixel, this value is over the handling capacity of a single pixel, and therefore it is saturated. This is visible also applying Eq. (5.14), since the found value is equal to 255.

Table 5.7: *Environmental data of sample scenario 2.*

Parameter	Symbol	Value	Derivation
Kinetic energy	KE	1.0931×10^8 J	Random
Impact angle	γ	45.51°	Random
Impact velocity	V_{imp}	30.7731 km s^{-1}	Random
Velocity	V	30.6475 km s^{-1}	Eq. (5.1)
Meteoroid density	ρ_i	1696.8 kg m^{-3}	Random
Meteoroid mass	m	0.2309 kg	Eq. (5.2)
Luminous efficiency	η	0.0013	Eq. (5.3)
Crater diameter	D	3.9339 m	Eq. (5.4)
Plume area	A_p	267.3926 m^2	Eq. (5.6)
Magnitude (at Earth distance)	$m_\oplus$	8.1646	Eq. (5.8)
Impact duration	t	0.0520 s	Eq. (5.7)
Impact temperature	T_{imp}	2061.6 K	Eq. (5.12)

Detection simulation In Table 5.8 are collected all the input data used in the simulation. In particular here the values of both the x and y coordinates of the pixels are made by only an element since the flash is captured only by a pixel. Since the pixel is saturated the intensity read from the program is equal to the one that is generated.

Table 5.8: *Input data of sample scenario 2.*

Parameter	Symbol	Value
flash x-coordinate	X_f	775
flash y-coordinate	Y_f	883
Pixel value	N	255

Figure 5.6 shows the flash location, with a red circle to highlight the position. According to the input given the flash will occupy only one pixel, and moreover, according to the value of t that it is used, all the event shall last for a frame only. This is what actually happens as shown in Figure 5.7. The only information here shown are related to the iteration in which the flash occurs. No further rearrangements has to be performed on these data since there are nor flash spread over more pixels nor flash that last for more than a frame.

In Table 5.9 are listed the output of the simulation; the same notes made for the previous example for the way the data are produces is still valid. Latitude and longitude values are consistent with the situation represented in Figure 5.6, and furthermore the minus sing in latitude is coherent to what summarized in Figure 4.18, since the flash is placed in the southern hemisphere. The images in output together with the information are shown in Figure 5.8.

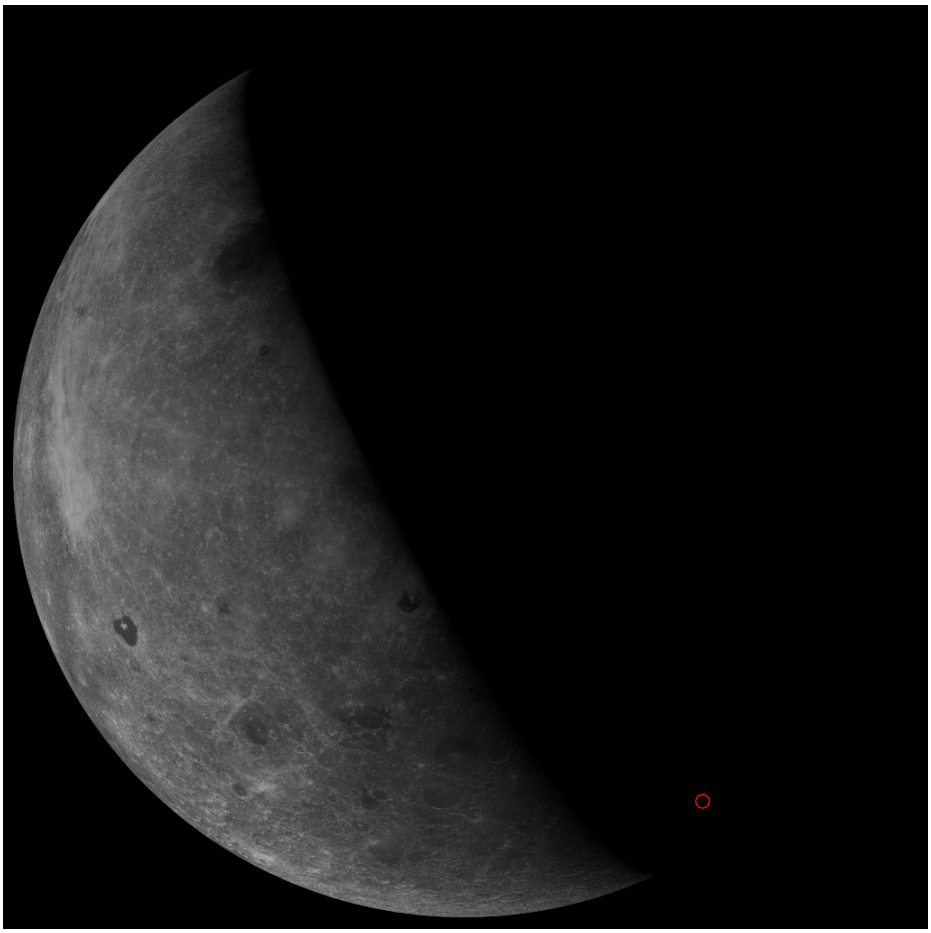
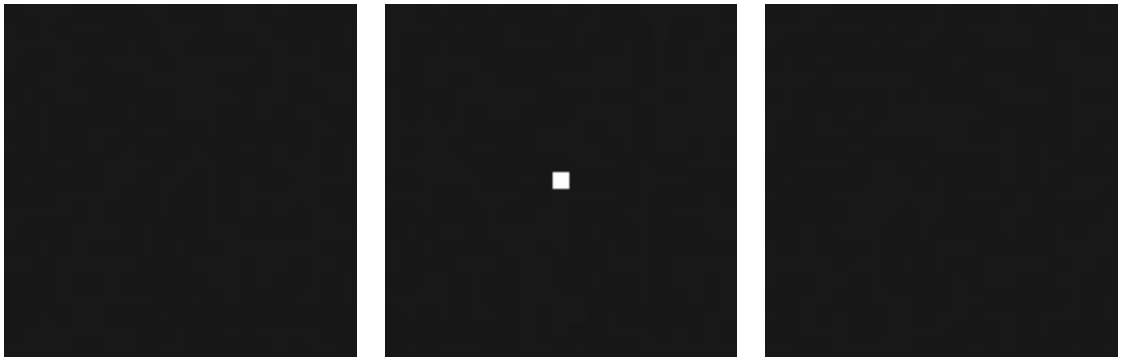


Figure 5.6: *Moon at sample scenario 2. The red circle represent the location in which has been inserted the flash.*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
883.0000	775.0000	255.0000	512.0000	513.0000	500.0000	0.0667

Figure 5.7: *Sample scenario 2 algorithm process. Since the flash is fully contained in one pixel and lasts less than a frame, the information generated just after the detection are also the one that are given as output.*



(a) *Image just before the impact* (b) *Image of the impact* (c) *Image just after the impact*

Figure 5.8: *Output images of sample scenario 2. Those images are focus around the flash location of the output images of the simulation; the positions represented are the same in all the three image. On the left the situation just after the impact, of the right just after and in the center the impact. Note that the pixel with the flash is fully white since the pixel is saturated.*

Table 5.9: *Output data of sample scenario 2.*

Parameter	Symbol	Value
Flash row	$\hat{R}$	883
Flash column	$\hat{C}$	775
Pixel value	N_{VIS}	255
Moon centre (x)	$\hat{X}_{c\mathbb{C}}$	512
Moon centre (y)	$\hat{Y}_{c\mathbb{C}}$	513
Moon radius	$\hat{R}_{\mathbb{C}}$	500
time	τ	0.066 s
Latitude	ϕ_f	$-47.8579°$
Longitude	λ_f	$51.7678°$

Post-processing simulation Also for this example the value for the NIR channel are simulated with Eq. (5.15) and Eq. (5.16), accordingly to the physical data in Table 5.7 and used without the detection for the reasons explained before. Also for the NIR channel the signal comes to be saturated as happened for the VIS. Through Eq. (5.20) it is evaluated the value for the angle ϑ; putting the information together with the one coming from the telemetry the post-processing is performed. In Table 5.10 are reported the values that have been obtained. The data obtained are conditioned by the fact that the pixels are saturated. First of all can be seen by the values that are estimated about the number of electron that hits the CCD. The value that is generated as input is in the order of 10^5, while the maximum number that can be taken in this process is the one corresponding to saturation, equal to 8×10^3. Therefore the temperature that is retrieved is underestimated with respect to the actual one, while the plume area is overestimated. The magnitude value depends of the power retrieved through the Plank's Law, that is strongly related to the temperature. The value obtained for the magnitude is therefore underestimated and a proper value should be far nearer to 3, corresponding to an high visible flash.

Table 5.10: *Post-processing data of sample scenario 2.*

Parameter	Symbol	Value	Derivation
NIR pixel value	N_{NIR}	255	Eq. (5.16)
Signal inVIS	S_{imp}^{VIS}	80 000 e^-	Eq. (5.21)
Signal inNIR	S_{imp}^{NIR}	80 000 e^-	Eq. (5.22)
Impact temperature	$\hat{T}_{imp}$	1753.8 K	Eq. (5.23)
Plume area	$\hat{A}_p$	833.77 m^2	Eq. (5.24)
Magnitude at S/C distance	m_{imp}	4.0995	Eq. (5.25)

5.4.3 Sample scenario 3

Environment generation This scenario simulates an impact that happens when the S/C is in its farthest position on the halo orbit around L_2 Lagrangian Point of the Earth-Moon system; the used image if the one represented in Figure 5.2b and in Table 5.11 are listed the distance from the surface and the angle between the boresight direction and the flash direction in LUMIO-Cam reference frame used in this simulation.

Table 5.11: *Orbital data of sample scenario 3.*

Parameter	Symbol	Value
LUMIO-Moon distance	d	85 117 km
Flash-boresight angle	ϑ	3.14°

Physical data used in this simulation are listed in Table 5.11, produced with the methodology in Chapter 5.1; in this situation the flash lasts for more than a frame (since t=0.160 30 s) and therefore are are available not only a value for both S_{imp} that for N, but one each frame; in the first, the signal evaluated by Eq. (5.13) is equal to 3.4747×10^4 e^- and the corresponding value evaluated through Eq. (5.14) is 110, in the second frame the signal is 3.4747×10^4 e^- and N is 110, while for the third S_{imp}=1.4067×10^4 e^- and N=44. The quantities representing the first two frames are equal since for all the impact duration the temperature of the plume is kept constant as done in Merisio (2018).

Table 5.12: *Environmental data of sample scenario 3.*

Parameter	Symbol	Value	Derivation
Kinetic energy	KE	4.9861×10^8 J	Random
Impact angle	γ	54.59°	Random
Impact velocity	V_{imp}	33.7102 km s^{-1}	Random
Velocity	V	33.5955 km s^{-1}	Eq. (5.1)
Meteoroid density	ρ_i	1604.6 kg m^{-3}	Random
Meteoroid mass	m	0.8775 kg	Eq. (5.2)
Luminous efficiency	η	0.0013	Eq. (5.3)
Crater diameter	D	6.3268 m	Eq. (5.4)
Plume area	A_p	565.8969 m^2	Eq. (5.6)
Magnitude (at Earth distance)	$m_\oplus$	7.7157	Eq. (5.8)
Impact duration	t	0.1603 s	Eq. (5.7)
Impact temperature	T_{imp}	2001.8 K	Eq. (5.12)

Detection simulation In Table 5.13 are shown the input given to the simulation. Note that the as coordinates are given only one input for x and one input for y, while for the values of the pixels are given three inputs. This means that the frame with the flash are three, each with an intensity related to the value corresponding in the input vector, accordingly to the notation ($[X;X;X]$), indicating that the first value corresponds to the first frame, the second to the second and the third to the third; by the way all the flashes are placed in the same location for all the frames, since have the same coordinates in the $x-y$ frame. Also in this case the values of the pixels are subjected to noise, and therefore are slightly different from the one generated.

Table 5.13: *Input data of sample scenario 3.*

Parameter	Symbol	Value
flash x-coordinate	X_f	672
flash y-coordinate	Y_f	380
Pixel value	N	$[112;109;42]$

Figure 5.9 highlighted with a red circle it is shown the flash position; since the position not change during time, this image is valid for all the impact frames. Since t higher than the integration time, the impact will be spread over different frames. This is what actually happens as shown in Figure 5.10, where are shown the processes made by the algorithm step-by-step. In particular in Figure 5.10a it is shown the output just after the first frame with the flash inside. Indeed the value retrieved is equal to the first input and also the time is equal to a single frame. Figure 5.10b shows the output of the following time step, when it is detected the second flash. Also in this case the value is coherent with the input and also the time step is the one corresponding to the single flash. At this step the algorithm finds two consecutive information and therefore these are compared. Since have the same impact positions, it is produced Figure 5.10c, where the pixel intensities are summed and the time step is

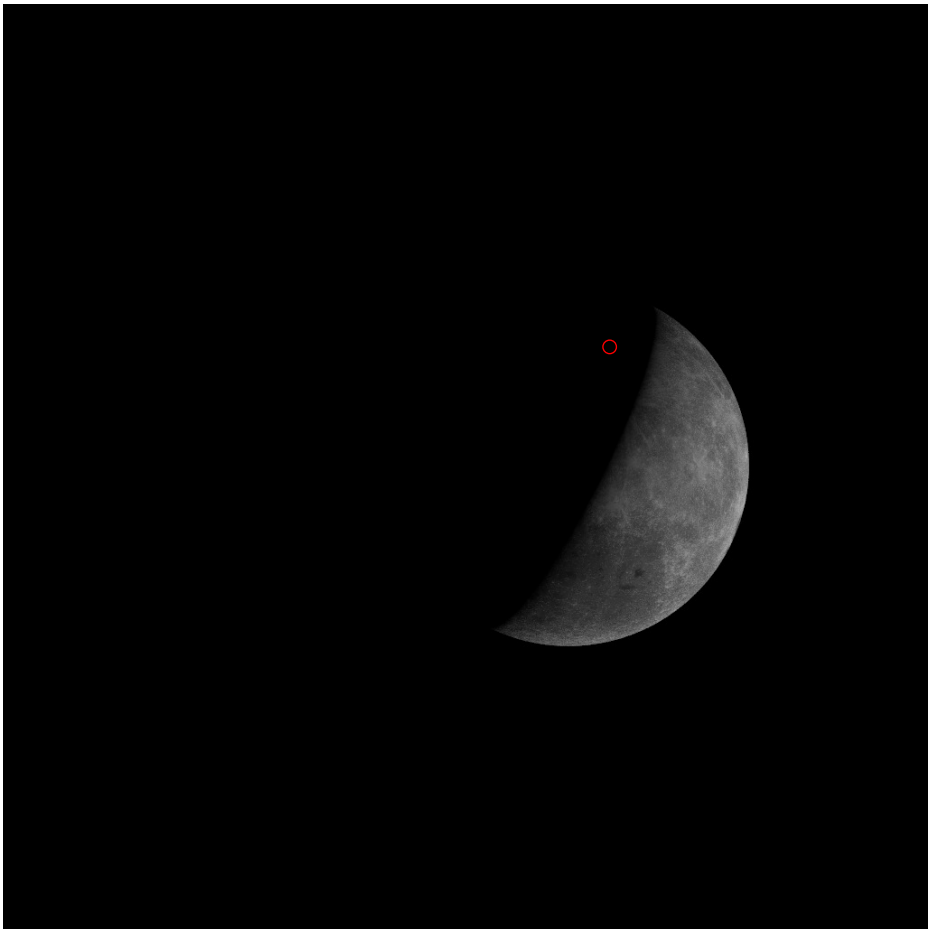


Figure 5.9: *Moon at sample scenario 3. The red circle represent the location in which has been inserted the flash.*

doubled. In Figure 5.10d it is shown the last frame with the flash (the third) with its detected pixel value and the time step representing a single frame. Also in this case the algorithm, since has already information in the buffer, check if the impact location is comparable, and, since in this case it is, the information are merged in Figure 5.10e, that is the actual output. Note that in this last image the time step is equal to three times the integration time and the value of the pixel intensity is the summation of the single intensities of the frames. Looking at the values corresponding to the centre of the Moon and to the radius, those are slightly different from an iteration to the following (1 or 2 pixels) due to the fact the boundary it is estimated; nevertheless this kind of error is not an issue.

In Table 5.14 are listed the output of the simulation. Note that the information of the time is now changed with respect to output of the other examples, due to the longer duration flash. The images in output together with the information are shown in Figure 5.11; since the algorithm has collected the information of the flash spread into three frames over a single one, in Figure 5.11b it is shown a reference image of the impact, that by the way do not represent the actual value of the total intensity.

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
380.0000	672.0000	112.0000	628.0000	513.0000	199.0000	0.0667

(a) *Output after the detection of the first frame only*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
380.0000	672.0000	109.0000	629.0000	513.0000	198.0000	0.0667

(b) *Output after the detection of the second frame only*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
380.0000	672.0000	221.0000	629.0000	513.0000	198.0000	0.1333

(c) *Output putting together information of a) and b)*

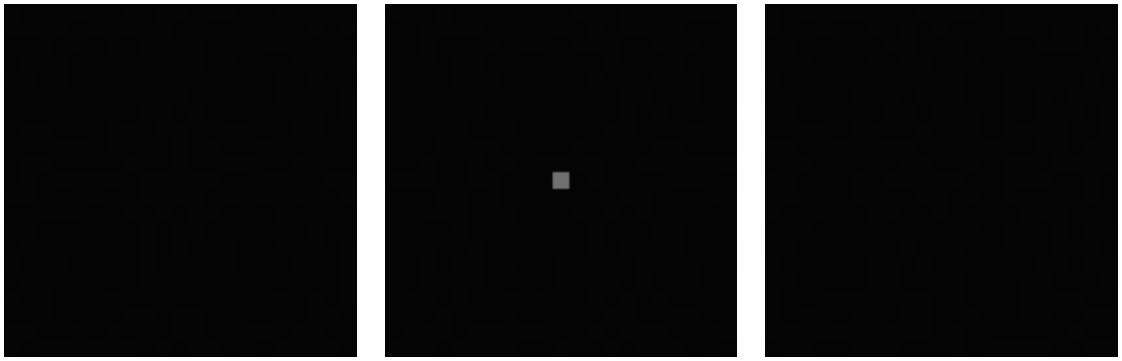
Row	Column	Intensity	Xcentre	Ycentre	Radius	time
380.0000	672.0000	42.0000	627.0000	512.0000	200.0000	0.0667

(d) *Output after the detection of the third frame only*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
380.0000	672.0000	263.0000	627.0000	512.0000	200.0000	0.2000

(e) *Output putting together information of c) and d)*

Figure 5.10: *Sample scenario 3 algorithm process. Figures are listed from top to bottom according to the temporal sequence the algorithm operates: Image a) represent the output just after the detection in the first frame; image b) shows the output just after the detection on the second frame. In image c) it is represented the process that puts together information shown in a) and b). In particular the intensities are summed, and the time is doubles. In d) the information just after the third frame detection are shown. In the last image e) all the previous information are summed, getting the total intensity and the total integration time.*



(a) *New configuration*

(b) *First configuration mettere che è vista da sopra*

(c) *First configuration mettere che è vista da sopra*

Figure 5.11: *Output images of sample scenario 3. Those images are focus around the flash location of the output images of the simulation; the positions represented are the same in all the three image. On the left the situation just after the impact, of the right just after and in the center the impact; this image do not represent the whole intensity, but shows only the impact location.*

Table 5.14: *Output data of sample scenario 3.*

Parameter	Symbol	Value
Flash row	$\hat{R}$	380
Flash column	$\hat{C}$	672
Pixel value	N_{VIS}	263
Moon centre (x)	$\hat{X}_{c\leftmoon}$	627
Moon centre (y)	$\hat{Y}_{c\leftmoon}$	512
Moon radius	$\hat{R}_{\leftmoon}$	200
time	τ	0.2 s
Latitude	ϕ_f	42.1997°
Longitude	λ_f	17.0470°

Post-processing simulation From here on the fact that the flash has lasted for more than a pixel is no more an issue and the post-processing is performed in the same way it is done for a single frame pixel, since it is displayed in output the total quantity of the pixel intensity representing the total amount of energy. The value of the NIR channel is found by applying at first Eq. (5.15) and then Eq. (5.16), according to data listed in Table 5.12. ϑ angle is found by applying Eq. (5.20), obtaining data listed in Table 5.15. The retrieved value of the temperature is very accurate, the plume area is in the same order of magnitude of the input one and the value of the magnitude represent a almost faint flash.

Table 5.15: *Post-processing data of sample scenario 3.*

Parameter	Symbol	Value	Derivation
NIR pixel value	N_{NIR}	202	Eq. (5.16)
Signal in VIS	S_{imp}^{VIS}	8.2510×10^4 e$^-$	Eq. (5.21)
Signal in NIR	S_{imp}^{NIR}	6.3373×10^4 e$^-$	Eq. (5.22)
Impact temperature	$\hat{T}_{imp}$	2003.3 K	Eq. (5.23)
Plume area	$\hat{A}_p$	449.1146 m^2	Eq. (5.24)
Magnitude at S/C distance	m_{imp}	5.2596	Eq. (5.25)

5.4.4 Sample scenario 4

Environment generation In this scenario is simulated an impact happening when the S/C is at its farthest position on the orbit; the used image if the one represented in Figure 5.2b and in Table 5.16 are shown the data related to the orbit for this simulation.

Table 5.16: *Orbital data of sample scenario 4.*

Parameter	Symbol	Value
LUMIO-Moon distance	d	85 117 km
Flash-boresight angle	ϑ	1.91°

Physical data are collected in Table 5.16, obtained as stated in Chapter 5.1; through Eq. (5.13) it it evaluated the signal equal to 2.5571×10^4 e^-, corresponding to a value to be inserted in the image, evaluated via Eq. (5.14), of 81.

Table 5.17: *Environmental data of sample scenario 4.*

Parameter	Symbol	Value	Derivation
Kinetic energy	KE	2.2155×10^8 J	Random
Impact angle	γ	40.42°	Random
Impact velocity	V_{imp}	15.1345 km s^{-1}	Random
Velocity	V	14.8774 km s^{-1}	Eq. (5.1)
Meteoroid density	ρ_i	1800 kg m^{-3}	Random
Meteoroid mass	m	1.9345 kg	Eq. (5.2)
Luminous efficiency	η	0.0011	Eq. (5.3)
Crater diameter	D	4.7232 m	Eq. (5.4)
Plume area	A_p	157.6933 m^2	Eq. (5.6)
Magnitude (at Earth distance)	$m_\oplus$	7.6330	Eq. (5.8)
Impact duration	t	0.0538 s	Eq. (5.7)
Impact temperature	T_{imp}	2264.8 K	Eq. (5.12)

Detection simulation Table 5.18 shows the input given to the simulation. The value given to the pixel is slightly bigger than the one obtained by when data have been produced due to the noise.

Table 5.18: *Input data of sample scenario 4.*

Parameter	Symbol	Value
flash x-coordinate	X_f	611
flash y-coordinate	Y_f	591
Pixel value	N	84

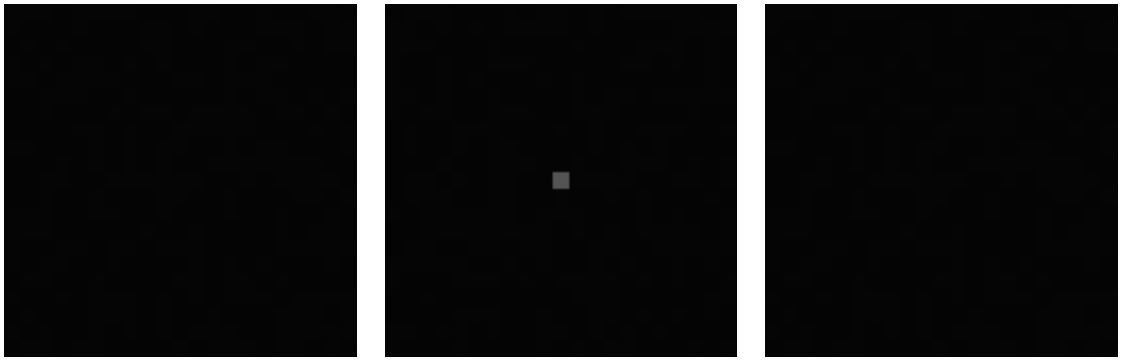
In Figure 5.12 the red circle shows the flash position; the value of t is lower than therefore the impact will on a single frame, as shown in Figure 5.13

Table 5.19 lists the output of the simulation. The value is coherent with the one given as input; note that the evaluated values for the coordinates respect both Figure 5.18 and the convention introduced in Figure 4.18, since the flash is in the south-west position.

Figure 5.12: *Moon at sample scenario 4. The red circle represent the location in which has been inserted the flash.*

Row	Column	Intensity	Xcentre	Ycentre	Radius	time
591.0000	611.0000	84.0000	627.0000	513.0000	200.0000	0.0667

Figure 5.13: *Sample scenario 4 algorithm process. Since the flash is fully contained in one pixel and lasts less than a frame, the information generated just after the detection are also the one that are given as output.*



(a) *New configuration* (b) *First configuration mettere che è vista da sopra* (c) *First configuration mettere che è vista da sopra*

Figure 5.14: *Output images of sample scenario 4. Those images are focus around the flash location of the output images of the simulation; the positions represented are the same in all the three image. On the left the situation just after the impact, of the right just after and in the center the impact.*

Table 5.19: *Output data of sample scenario 4.*

Parameter	Symbol	Value
Flash row	$\hat{R}$	591
Flash column	$\hat{C}$	611
Pixel value	N_{VIS}	84
Moon centre (x)	$\hat{X}_{c\mathbb{C}}$	627
Moon centre (y)	$\hat{Y}_{c\mathbb{C}}$	513
Moon radius	$\hat{R}_{\mathbb{C}}$	200
time	τ	0.066 s
Latitude	ϕ_f	$-23.1998°$
Longitude	λ_f	$-5.6762°$

Post-processing simulation Applying Eq. (5.15) and Eq. (5.16), the value corresponding to NIR channel is found. Using Eq. (5.20), the angle ϑ is evaluated and the post-process is performed, shown in Table 5.20.

Table 5.20: *Post-processing data of sample scenario 4.*

Parameter	Symbol	Value	Derivation
NIR pixel value	N_{NIR}	49	Eq. (5.16)
Signal in VIS	S_{imp}^{VIS}	2.6353×10^4 e$^-$	Eq. (5.21)
Signal in NIR	S_{imp}^{NIR}	1.5373×10^4 e$^-$	Eq. (5.22)
Impact temperature	$\hat{T}_{imp}$	2314.7 K	Eq. (5.23)
Plume area	$\hat{A}_p$	108.7535 m^2	Eq. (5.24)
Magnitude at S/C distance	m_{imp}	5.3012	Eq. (5.25)

5.4.5 Computational effort

All these simulation have been performed on MATLAB® (release R2018b), run on a Hp Envy with Intel Core i7-6500U @ 2.50 GHz CPU and 16 GB of RAM. In Table 5.21 are listed all the data obtained. The yellow cells represent the detected frame, and the number in brackets the actual time for the detection, during those iterations.

Table 5.21: *Computational effort of the passages for each scenario.*

	Scenario 1	**Scenario 2**	**Scenario 3**	**Scenario 4**
Frame 1	0.030 s	0.033 s	0.037 s	0.024 s
Frame 2	0.029 s	0.034 s	0.024 s	0.037 s
Frame 3	0.027 s	0.027 s	0.029 s	0.028 s
Frame 4	0.063 s/(0.033 s)†	0.063 s/(0.021 s)†	0.065 s/(0.020 s)†	0.065 s/(0.024 s)†
Frame 5	0.031 s	0.032 s	0.057 s/(0.020 s)†	0.029 s
Frame 6	0.032 s	0.049 s	0.056 s/(0.027 s)†	0.029 s
Frame 7	0.032 s	0.027 s	0.029 s	0.026 s
Frame 8	0.034 s	0.029 s	0.032 s	0.027 s
Frame 9	0.029 s	0.031 s	0.033 s	0.027 s
Frame 10	0.029 s	0.032 s	0.025 s	0.025 s
Post-processing	0.15 s	0.13 s	0.11 s	0.12 s

† Inside brackets the time to have the detection itself, the time outside takes into account also the processes to evaluate the centre of the Moon, to check for the multiple flash inside a single frame, to check for flash in multiple frames and, if it is the last iteration with a flash detected, the evaluation of latitude and longitude.

6

Final remarks

This chapter contains the conclusive observation about the results of this work, together with some suggestions for future developments.

6.1 Considerations about the simulations results

Here are made some considerations about the output of the simulation performed in Chapter 5. Looking at Table 5.21, it is shown that the algorithm works properly, since the time required to have the detection, as required, is always less than the integration time. It worth to be notice that the time spent during the actual detection phase, namely when each pixel is checked to verify if is under or over the threshold, do not change on passing from the frames in which the flash is present to the ones in which it is not. Therefore the difference in time that is highlighted in the steps with the flash in the frame is due to the passages performed after the detection to validate it. In particular analyzing the algorithm more in depth it is shown that the main source of this large amount of time is the reconstruction of the Moon edge. By the way, according to the simulations performed, the execution time of the simulation is fast enough to be acceptable. Nevertheless the main request of this work is satisfied, since it was possible to develop a methodology able to find an impact flash from a set of images given as input.

For what regards the output given by the detection algorithm is consistent with the inputs used and to the conventions that has been declared. In particular the way the flashes that occupies more than one pixels or more than one frame are managed is very satisfying. Furthermore the evaluation of the latitude and longitude angle is very accurate, taking into account the way they are retrieved. The information displayed as output encounter the expectation according to the environment that has been simulated. Each step performed by the algorithm works as expected. It is moreover satisfying the way the methodology overcomes also difficult situations in which the pattern of flashes changes from a frame to the following.

Regarding the post-processing, the estimation of the temperature is very satisfying, since in normal conditions, namely without saturation in the pixel, the error with respect to the input value is less than the 5%; the error could not ever be reduced to zero, due to the technological limitations that have to be faced, first of all the fact that the time step that has to be used is always an integer multiple of 66 ms. As the flash event is more and more coincident with the integration time, the more the result will be accurate. Regarding the estimation of the area of the plume, the error with respect to the one used as input comes to be higher. Nevertheless the vast majority of the times the order of magnitude is always correctly retrieved.

6.2 Conclusions

The main request for this thesis works was to evaluate the feasibility of detecting impact flash on the Moon surface by a S/C. The methodology developed can actually accomplish this task respecting also the requirements stated by LUMIO team (Table 2.3). The scientific product that is given as output by the detection algorithm is compatible to the expected results according to environment that has been used to perform the simulations. One of the major problems that has been faced was to fit all the detection chain inside the integration time boundaries. This requirement has been accomplished on the simulations performed in MATLAB®, and this encourages that the detection can be carried out also on the final support. Regarding the post-processing algorithm, form the simulation results, the working principle has been validated, and the result can be considered acceptable. In particular the result over the temperatures are very confident, while the one on the area of the plume less, but nevertheless it is only an estimation of the amount of material that is contained in the ejecta. Looking on the results in Table 5.21, it is clear that the post-processing, according to the way it is structured up to now, cannot be performed in real time like the detection since the time required is far higher than the one available during the integration time. Therefore this process has to be performed out of the science phase or at ground; anyway, not performing post-processing as soon the data are retrieved is not a problem. Indeed the information that are found by this process do not produce any improvement on the detection, but on the other side produce an increase on the amount of data to be saved; moreover, doing this a posteriori admits to have a more wide vision of the phenomenon and to evaluate a full meteoroid shower. Therefore, even if the post-processing has to be carried on at ground and do not interest directly the S/C it was important to develop a methodology able to interpreter the output of the detection algorithm and to produce from this a scientific output representing the full phenomenon.

6.3 Recommendations for future works

On passing from the development and testing on MATLAB® to the actual support to be put on LUMIO, the methodology developed so far can be modified introducing several improvements. Since the simulations have been performed on the computer, the results here obtained are only a first step for further improvements, and in particular the steps here performed have to be adapted to the need of the real application. Furthermore this methodology is developed according to the latest framework available about LUMIO mission, namely according to Merisio (2018), Phase-*0* design (Topputo et al., 2017) and on the CDF report from ESA (Walker et al., 2018). The simulation carried out in Chapter 5 is performed according to the available models both about the environment and the impact

dynamics. The methodology has to be updated if some of this references is changed. On passing to the final support a series of tuning campaign for all the weighting coefficient has to be performed, like for α inside the background function, using the values in this work as guess solutions. Another detail to take care is the quantization of the information acquired by LUMIO-Cam accordingly to the number of bits the A/D converter works, since for example the value to reconstruct the Moon shape has to be changed according to this. A last remark regards the use of the algorithm to reconstruct the Moon edge. This is the most time consuming process inside the methodology. Many improvements can be performed to speed up this process, like reducing the number of points to be taken, leading however to a less accurate result. A solution to reduce the impact of this process could be to create a mask once that, since the orbit is known a priori, can predict the shape of the Moon circle at each point, refining it at each time step, so that it is possible to have already some guess points on the image near the actual edge ones, avoiding to take a lot of points each frame.

Appendices

A

Constants

All the constant used are summarized in Table A.1

Table A.1: *Constants used.*

Symbol	Description	Value	Reference
$\mu_{☾}$	Gravitational mass parameter of Moon.	$4902.8001\ \mathrm{km^3\,s^{-2}}$	SPICE[†]
$\mu_{\oplus}$	Gravitational mass parameter of Earth.	$398\,600.4354\ \mathrm{km^3\,s^{-2}}$	SPICE[†]
c	Speed of light in vacuum.	$299\,792\,458\ \mathrm{m\,s^{-1}}$	NIST[⋆]
$d_{\oplus☾}$	Reference distance between Earth and Moon.	$385\,000.6\ \mathrm{km}$	Murphy (2013)
F_0	Reference flux density of a magnitude m_0 source.	$1.369\,49 \times 10^{-10}\ \mathrm{W\,m^{-2}\,m^{-1}}$	Bessell (1979)
g	Standard acceleration of gravity of the Moon.	$1.67\ \mathrm{m\,s^{-2}}$	NIST[⋆]
h	Plank's constant.	$6.626\,070\,040 \times 10^{-34}\ \mathrm{kg\,m^2\,s^{-1}}$	NIST[⋆]
k_B	Boltzmann's constant.	$1.380\,648\,52 \times 10^{-23}\ \mathrm{kg\,m^2\,s^{-2}\,K^{-1}}$	NIST[⋆]
kton TNT	Kiloton of TNT.	$1.184 \times 10^{12}\ \mathrm{J}$	NIST[⋆]
m_0	Reference magnitude of a source with flux density F_0.	21.1	Bessell (1979)
$R_{☾}$	Moon's radius.	$1737.1\ \mathrm{km}$	SPICE[†]

[†] `https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/info/mostused.html#D` [Last visited on 27/11/2019]
[⋆] `https://physics.nist.gov/cuu/Constants/index.html` [Last visited on 27/11/2019]

B

Validation of the detection algorithm

Here, in Table B.1, is presented a part of the of the validation results of the detection algorithm. Full table is available at `https://github.com/degliandrea/Thesis.git`.

Table B.1: *Results of the validation of the detection.*

Iter	$\hat{X}_f$	$\hat{Y}_f$	X_f	Y_f	$CK^{\dagger}_{X_f}$	$CK^{\dagger}_{Y_f}$	N[§]	In/Out[⋆]	$\hat{X}_{c_{\mathbb{C}}}$	$CK^{\dagger}_{X_{c_{\mathbb{C}}}}$	$\hat{Y}_{c_{\mathbb{C}}}$	$CK^{\dagger}_{Y_{c_{\mathbb{C}}}}$	$\hat{R}_{\mathbb{C}}$	$CK^{\dagger}_{R_{\mathbb{C}}}$	X_c	Y_c	R
1	422	644	422	644	1	1	55	0	316	1	342	1	308	1	316	342	308
2	229	70	229	70	1	1	38	0	444	1	337	1	308	1	443	337	309
3	704	698	704	698	1	1	115	0	444	1	337	1	308	1	443	337	309
4	436	840	436	840	1	1	210	0	511	1	513	1	284	1	513	512	283
5	416	951	416	951	1	1	179	0	516	1	512	1	280	1	513	512	283
6	82	966	82	966	1	1	233	0	497	1	548	1	312	1	497	547	311
7[¶]	NaN	NaN	532	442	0	0	24	NaN	NaN	0	NaN	0	NaN	0	516	511	230
8	393	822	393	822	1	1	75	0	600	1	315	1	304	1	600	315	304
9	638	659	638	659	1	1	15	1	622	1	476	1	306	1	622	476	306
10	267	559	267	559	1	1	84	0	637	1	690	1	309	1	638	690	308
11	549	284	549	284	1	1	90	1	392	1	425	1	309	1	392	425	309
12	47	802	47	802	1	1	220	0	444	1	337	1	308	1	443	337	309
13	928	61	928	61	1	1	20	0	523	1	522	1	27	1	523	522	27
14	902	242	902	242	1	1	22	0	516	1	512	1	280	1	513	512	283
15	728	144	728	144	1	1	12	0	512	1	513	1	283	1	513	512	283
16	300	296	300	296	1	1	133	0	495	1	545	1	309	1	494	544	310
17	887	337	887	337	1	1	204	1	600	1	314	1	303	1	600	315	304
18	565	619	565	619	1	1	55	0	442	1	332	1	308	1	441	332	308
19	1015	238	1015	238	1	1	110	0	610	1	480	1	309	1	610	480	309
20	1016	1014	1016	1014	1	1	160	0	689	1	513	1	309	1	689	513	310
21	135	444	135	444	1	1	128	0	691	1	512	1	309	1	689	513	310
22	803	1013	803	1013	1	1	160	0	392	1	426	1	309	1	392	425	309

Continue on the next page

Table B.1 – *Continue from previous page*

Iter	$\hat{X}_f$	$\hat{Y}_f$	X_f	Y_f	$CK^{\dagger}_{X_f}$	$CK^{\dagger}_{Y_f}$	$N^{\S}$	In/Out*	$\hat{X}_{c_{\mathbb{C}}}$	$CK^{\dagger}_{X_{c_{\mathbb{C}}}}$	$\hat{Y}_{c_{\mathbb{C}}}$	$CK^{\dagger}_{Y_{c_{\mathbb{C}}}}$	$\hat{R}_{\mathbb{C}}$	$CK^{\dagger}_{R_{\mathbb{C}}}$	X_c	Y_c	R
23	866	587	866	587	1	1	32	0	392	1	487	1	307	1	390	486	308
24	1014	651	1014	651	1	1	73	0	583	1	407	1	308	1	582	407	309
25	1008	1004	1008	1004	1	1	21	0	376	1	409	1	312	1	375	410	312
26	719	801	719	801	1	1	77	0	600	1	315	1	304	1	600	315	304
27	767	724	767	724	1	1	187	0	316	1	342	1	308	1	316	342	308
28	358	678	358	678	1	1	45	1	513	1	509	1	280	1	512	511	282
29	367	864	367	864	1	1	78	0	637	1	690	1	308	1	638	690	308
30	195	733	195	733	1	1	250	0	394	1	421	1	313	1	392	425	309
31	295	735	295	735	1	1	56	0	442	1	332	1	308	1	441	332	308
32	61	669	61	669	1	1	45	0	607	1	473	1	302	1	607	473	302
33	840	882	840	882	1	1	59	0	511	1	514	1	285	1	512	511	282
34	188	567	188	567	1	1	200	1	494	1	544	1	310	1	494	544	310
35	128	327	128	327	1	1	173	1	318	1	340	1	310	1	318	340	310
36	151	747	151	747	1	1	230	0	392	1	487	1	307	1	390	486	308
37	846	326	846	326	1	1	189	0	335	1	691	1	308	1	335	691	308
38	7	499	7	499	1	1	103	0	610	1	480	1	309	1	610	480	309
39	395	151	395	151	1	1	146	0	389	1	486	1	309	1	390	486	308
40	851	44	851	44	1	1	213	0	610	1	480	1	309	1	610	480	309
41	980	520	980	520	1	1	174	0	511	0	514	1	224	0	516	511	230
42	502	589	502	589	1	1	220	0	316	1	342	1	308	1	318	340	310
43	890	600	890	600	1	1	180	0	443	1	337	1	309	1	443	337	309
44	989	102	989	102	1	1	203	0	371	1	387	1	310	1	370	388	309
45	980	88	980	88	1	1	71	0	389	1	486	1	309	1	390	486	308

Continue on the next page

Table B.1 – *Continue from previous page*

Iter	$\hat{X}_f$	$\hat{Y}_f$	X_f	Y_f	$CK^{\dagger}_{X_f}$	$CK^{\dagger}_{Y_f}$	N[§]	In/Out[⋆]	$\hat{X}_{c_{\mathbb{C}}}$	$CK^{\dagger}_{X_{c_{\mathbb{C}}}}$	$\hat{Y}_{c_{\mathbb{C}}}$	$CK^{\dagger}_{Y_{c_{\mathbb{C}}}}$	$\hat{R}_{\mathbb{C}}$	$CK^{\dagger}_{R_{\mathbb{C}}}$	X_c	Y_c	R
46	231	703	231	703	1	1	6	0	583	1	407	1	308	1	582	407	309
47	68	916	68	916	1	1	172	0	442	1	332	1	308	1	441	332	308
48	935	650	935	650	1	1	21	0	600	1	315	1	304	1	600	315	304
49	555	179	555	179	1	1	212	0	390	1	486	1	308	1	390	486	308
50	471	767	471	767	1	1	192	0	583	1	407	1	308	1	582	407	309
51[¶]	NaN	NaN	555	57	0	0	248	NaN	NaN	0	NaN	0	NaN	0	442	331	307
52	159	247	159	247	1	1	120	0	389	1	486	1	309	1	390	486	308
53	960	915	960	915	1	1	175	0	318	1	340	1	310	1	316	342	308
54	996	401	996	401	1	1	245	0	316	1	342	1	308	1	318	340	310
55	115	683	115	683	1	1	55	0	511	1	516	1	280	1	512	515	281
56	1013	894	1013	894	1	1	252	0	387	1	481	0	304	0	391	488	311
57	300	180	300	180	1	1	226	0	512	1	513	1	283	1	513	512	283
58	62	311	62	311	1	1	159	0	526	1	518	1	32	0	523	522	27
59	197	864	197	864	1	1	7	0	637	1	690	1	309	1	638	690	308
60	328	644	328	644	1	1	124	0	689	1	513	1	309	1	689	513	310
61[¶]	NaN	NaN	520	225	0	0	221	NaN	NaN	0	NaN	0	NaN	0	610	480	309
62	783	751	783	751	1	1	246	0	443	1	337	1	309	1	443	337	309
63	732	393	732	393	1	1	128	0	392	1	425	1	309	1	392	425	309
64	752	699	752	699	1	1	68	1	607	1	473	1	302	1	607	473	302
65	838	116	838	116	1	1	107	0	443	1	337	1	309	1	443	337	309
66	492	707	492	707	1	1	68	1	513	1	512	1	283	1	513	512	283
67	866	901	866	901	1	1	232	0	387	1	481	0	304	0	391	488	311
68	887	833	887	833	1	1	45	0	511	1	515	1	281	1	512	515	281

Continue on the next page

Table B.1 – *Continue from previous page*

Iter	$\hat{\mathbf{X}}_\mathbf{f}$	$\hat{\mathbf{Y}}_\mathbf{f}$	$\mathbf{X_f}$	$\mathbf{Y_f}$	$\mathbf{CK}^{\dagger}_{\mathbf{X_f}}$	$\mathbf{CK}^{\dagger}_{\mathbf{Y_f}}$	$\mathbf{N}^{\S}$	$\mathbf{In/Out}^{\star}$	$\hat{\mathbf{X}}_{\mathbf{c}_\mathbb{C}}$	$\mathbf{CK}^{\dagger}_{\mathbf{X}_{\mathbf{c}_\mathbb{C}}}$	$\hat{\mathbf{Y}}_{\mathbf{c}_\mathbb{C}}$	$\mathbf{CK}^{\dagger}_{\mathbf{Y}_{\mathbf{c}_\mathbb{C}}}$	$\hat{\mathbf{R}}_\mathbb{C}$	$\mathbf{CK}^{\dagger}_{\mathbf{R}_\mathbb{C}}$	$\mathbf{X}_c$	$\mathbf{Y}_c$	$\mathbf{R}$
69	236	223	236	223	1	1	151	0	601	1	705	1	309	1	600	704	310
70	514	417	514	417	1	1	198	1	513	1	512	1	283	1	513	512	283
71	1022	685	1022	685	1	1	151	0	318	1	340	1	310	1	316	342	308
72	875	542	875	542	1	1	195	0	582	1	407	1	309	1	582	407	309
73	502	442	502	442	1	1	88	1	369	1	392	1	306	1	370	388	309
74	678	923	678	923	1	1	244	0	336	1	688	1	311	1	335	691	308
75	303	805	303	805	1	1	127	0	691	1	512	1	308	1	689	513	310
76	567	333	567	333	1	1	29	1	512	1	511	1	282	1	512	511	282
77[¶]	NaN	NaN	602	287	0	0	245	NaN	NaN	0	NaN	0	NaN	0	512	511	282
78	84	736	84	736	1	1	153	0	497	1	549	1	313	1	497	547	311
79	841	724	841	724	1	1	105	0	622	1	476	1	306	1	622	476	306
80	925	1000	925	1000	1	1	123	0	497	1	547	1	311	1	497	547	311
81	890	699	890	699	1	1	52	0	582	1	407	1	309	1	582	407	309
82	436	429	436	429	1	1	241	1	441	1	332	1	308	1	441	332	308
83	83	457	83	457	1	1	139	0	514	1	512	1	281	1	513	512	283
84	956	631	956	631	1	1	238	0	444	1	337	1	308	1	443	337	309
85	496	982	496	982	1	1	183	0	637	1	690	1	309	1	638	690	308
86	324	695	324	695	1	1	220	1	495	1	543	1	308	1	497	547	311
87	941	423	941	423	1	1	231	0	600	1	315	1	304	1	600	315	304
88	194	160	194	160	1	1	239	0	390	1	486	1	308	1	390	486	308
89	458	921	458	921	1	1	77	0	316	1	342	1	308	1	316	342	308
90[+]	NaN	NaN	978	136	0	0	1	NaN	NaN	0	NaN	0	NaN	0	390	486	308
91	162	636	162	636	1	1	105	0	622	1	476	1	306	1	622	476	306

Continue on the next page

Table B.1 – *Continue from previous page*

Iter	$\hat{X}_f$	$\hat{Y}_f$	X_f	Y_f	$CK^{\dagger}_{X_f}$	$CK^{\dagger}_{Y_f}$	N[§]	In/Out[⋆]	$\hat{X}_{c☾}$	$CK^{\dagger}_{X_{c☾}}$	$\hat{Y}_{c☾}$	$CK^{\dagger}_{Y_{c☾}}$	$\hat{R}_{☾}$	$CK^{\dagger}_{R_{☾}}$	X_c	Y_c	R
92	647	726	647	726	1	1	184	1	494	1	545	1	308	1	494	545	308
93	828	797	828	797	1	1	178	0	691	1	512	1	309	1	689	513	310
94	727	520	727	520	1	1	98	0	389	1	483	0	306	0	391	488	311
95	740	457	740	457	1	1	106	0	387	1	481	0	304	0	391	488	311
96	824	552	824	552	1	1	142	0	516	1	511	1	230	1	516	511	230
97[+]	NaN	NaN	155	359	0	0	1	NaN	NaN	0	NaN	0	NaN	0	392	425	309
98	45	893	45	893	1	1	111	0	369	1	392	1	306	1	370	388	309
99[‡]	NaN	NaN	506	314	0	0	104	NaN	NaN	0	NaN	0	NaN	0	516	511	230
100	348	862	348	862	1	1	190	0	390	1	486	1	308	1	390	486	308
101	456	567	456	567	1	1	73	1	392	1	425	1	309	1	392	425	309

[†] **CK** stands for **Check**. If 1 the two quantities compared era equal, if 0 not.

[⋆] Checks if the flash inside or outside the Moon; if in the result is 1, if out 0.

[§] The flash intensity is intended.

[¶] Data are set to NaN since the flash is in the Day-side of the Moon, but the flash has been detected.

[+] The detection did not succeeded. No flash is detected since the signal is too faint.

[‡] The detection did not succeeded. No flash is detected since the lighted pixel is in the Day-side of the Moon.

References

Bellot Rubio L., Ortiz J., and Sada P. (2000). "Luminous efficiency in hypervelocity impacts from the 1999 lunar Leonids". In: *Astrophysical Journal Letters* 542, pp. L65–L68 (cit. on p. 5).

Bessell M. S. (1979). "UBVRI photometry. II - The Cousins VRI system, its temperature and absolute flux calibration, and relevance for two-dimensional photometry". In: *Publications of the Astronomical Society of the Pacific* 91, p. 589 (cit. on p. 89).

Bonanos A. Z., Avdellidou C., Liakos A., Xilouris E. M., Dapergolas A., Koschny D., Bellas-Velidis I., Boumis P., Charmandaris V., Fytsilis A., and Maroussis A. (2018). "NELIOTA: First temperature measurement of lunar impact flashes". In: *Astronomy & Astrophysics* 612, A76 (cit. on pp. 6, 7, 17).

Bouley S., Baratoux D., Vaubaillon J., Mocquet A., Feuvre M. L., Colas F., Benkhaldoun Z., Daassou A., Sabil M., and Lognonné P. (2012). "Power and duration of impact flashes on the Moon: Implication for the cause of radiation". In: *Icarus* 218, pp. 115 –124 (cit. on pp. 63, 64).

Brown P. G., Assink J. D., Astiz L., Blaauw R., Boslough M. B., Borovička J., et al. (2013). "A 500-kiloton airburst over Chelyabinsk and an enhanced hazard from small impactors". In: *Nature* 503, pp. 238–241 (cit. on p. 3).

Campagnola S., Ozaki N., Oguri K., Verspieren Q., Kakihara K., Yanagida K., Funase R., Yam C. H., Ferella L., Yamaguchi T., et al. (2016). "Mission Analysis for EQUULEUS, JAXA's Earth-Moon Libration Orbit Cubesat". In: *Proceedings of the 67th International Astronautical Congress* (cit. on p. 6).

Cipriano A. M. (2017). *Orbit Design of a Lunar Meteoroid Impact Flashes Observer (M. Sc. Thesis)* (cit. on pp. 3, 4, 19, 63).

Hussain M., Chen D., Cheng A., Wei H., and Stanley D. (2013). "Change detection from remotely sensed images: From pixel-based to object-based approaches". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 80, pp. 91 –106 (cit. on pp. 6, 23).

Liakos A., Bonanos A., Xilouris E., Bellas-Velidis I., Boumis P., Charmandaris V., Dapergolas A., Fytsilis A., Maroussis A., Koschny D., Moissl R., and Navarro V. (2019). *NELIOTA Lunar Impact Flash Detection and Event Validation* (cit. on pp. 7, 27).

Lu D., Mausel P., Brondízio E., and Moran E. (2004). "Change detection techniques". In: *International Journal of Remote Sensing* 25, pp. 2365–2401 (cit. on pp. 6, 23).

Madiedo J. M., Ortiz J. L., Morales N., and Cabrera-Caño J. (2015a). "MIDAS: Software for the detection and analysis of lunar impact flashes". In: *Planetary and Space Science* 111, pp. 105 –115 (cit. on pp. 5, 7, 23).

Madiedo J. M., Ortiz J. L., Organero F., Ana-Hernàndez L., Fonseca F., Morales N., and Cabrera-Caño J. (2015b). "Analysis of Moon impact flashes detected during the 2012 and 2013 Perseids". In: *Astronomy & Astrophysics* 577, A118 (cit. on pp. 5, 38, 63).

Merisio G. (2018). *Payload, Orbit, and Environment Simulation for LUMIO Mission Coverage Analysis (M. Sc. Thesis)* (cit. on pp. 4, 14, 28, 37, 51, 62, 63, 65, 75, 85).

Murphy T. W. (2013). "Lunar laser ranging: the millimeter challenge". In: *Reports on Progress in Physics* 76, p. 076901 (cit. on p. 89).

Oberst J., Christou A., Suggs R., Moser D., Daubar I., McEwen A., et al. (2012). "The present-day flux of large meteoroids on the lunar surface - A synthesis of models and observational techniques". In: *Planetary and Space Science* 74, pp. 179–193 (cit. on pp. 3, 4).

Ortiz J. L., Aceituno F. J., Quesada J. A., Aceituno J., Fernández M., Santos Sanz P., Trigo Rodríguez J. M., Llorca J., Martín Torres F. J., Montañés Rodríguez P., et al. (2006). "Detection of sporadic impact flashes on the Moon: Implications for the luminous efficiency of hypervelocity impacts and derived terrestrial impact rates". In: *Icarus* 184, pp. 319–326 (cit. on pp. 6, 14).

Ortiz J. L., Madiedo J. M., Morales N., Santos-Sanz P., and Aceituno F. J. (2015). "Lunar impact flashes from Geminids: analysis of luminous efficiencies and the flux of large meteoroids on Earth". In: *Monthly Notices of the Royal Astronomical Society* 454, pp. 344–352 (cit. on p. 64).

Suggs R. M., Moser D. E., Cooke W. J., and Suggs R. J. (2014). "The flux of kilogram-sized meteoroids from lunar impact monitoring". In: *Icarus* 238, pp. 23–36 (cit. on pp. 1, 5).

Suggs R., Ehlert S., and Moser D. (2017). "A comparison of radiometric calibration techniques for lunar impact flashes". In: *Planetary and Space Science* 143, pp. 225 –229 (cit. on p. 38).

Suggs R. M., Cooke W. J., Suggs R. J., Swift W. R., and Hollon N. (2008). "The NASA Lunar Impact Monitoring Program". In: *Advances in Meteoroid and Meteor Science*, pp. 293–298 (cit. on pp. 7, 23, 29).

Swift W., Moser D., Suggs R., and Cook W. (2011). "An Exponential Luminous Efficiency Model for Hypervelocity Impact into Regolith". In: *Meteoroids: The Smallest Solar System Bodies*, pp. 125–241 (cit. on pp. 5, 63).

SYSNova: R&D Studies Competition for Innovation – No.4. (2016a). *Lunar Cubesats for Exploration (LUCE). Invitation to Tender AO/1-8643/16/NL/GLC/as* (cit. on p. 11).

SYSNova: R&D Studies Competition for Innovation – No.4. (2016b). *Lunar Cubesats for Exploration (LUCE). Statement of Work, Issue 1, Rev 0, TEC-SY/84/2016/SOW/RW* (cit. on pp. 11, 12).

Topputo F., Massari M., Biggs J., Di Lizia P., Dei Tos D., et al. (2017). *LUMIO: Lunar Meteoroid Impacts Observer* (cit. on pp. 1, 4, 6, 8, 9, 11–13, 16–19, 22, 24, 28, 29, 40, 85).

Walker R. et al. (2018). *CDF Study Report of LUMIO*. Tech. rep. (ESTEC, ESA, Netherlands) (cit. on pp. 1, 9, 11, 14, 16–18, 22, 40, 85).

Xilouris E. M., Bonanos A. Z., Bellas-Velidis I., Boumis P., Dapergolas A., Maroussis A., Liakos A., Alikakos I., Charmandaris V., Dimou G., Fytsilis A., Kelley M., Koschny D., Navarro V., Tsiganis K., and Tsinganos K. (2018). "NELIOTA: The wide-field, high-cadence, lunar monitoring system at the prime focus of the Kryoneri telescope". In: *Astronomy & Astrophysics* 619, A141 (cit. on pp. 7, 23, 27).