Addressing camera sensors faults in Vision-Based Navigation
Simulation and dataset development

Gallon, Riccardo; Schiemenz, Fabian; Menicucci, Alessandra; Gill, Eberhard

DOI
10.1016/j.actaastro.2025.10.071

Publication date
2026

Document Version
Final published version

Published in
Acta Astronautica

Acta Astronautica 239 (2026) 849–872

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research Paper

Addressing camera sensors faults in Vision-Based Navigation: Simulation and dataset development

Riccardo Gallon [a,b,*], Fabian Schiemenz [b], Alessandra Menicucci [a], Eberhard Gill [a]

[a] Department of Space Systems Engineering, Faculty of Aerospace Engineering, Kluyverweg 1, HS Delft, 2629, Netherlands
[b] Airbus Defence and Space GmbH, Claude-Dornier Straße, Immenstaad am Bodensee, 88090, Germany

ARTICLE INFO

Keywords:
Image segmentation
FDIR
Onboard artificial intelligence
Vision-Based Navigation

ABSTRACT

The increasing importance of Vision-Based Navigation (VBN) algorithms in space missions raises numerous challenges in ensuring their reliability and operational robustness. Sensor faults can lead to inaccurate outputs from navigation algorithms or even complete data processing faults, potentially compromising mission objectives. Artificial Intelligence (AI) offers a powerful solution for detecting such faults, overcoming many of the limitations associated with traditional fault detection methods. However, the primary obstacle to the adoption of AI in this context is the lack of sufficient and representative datasets containing faulty image data.

This study addresses these challenges by focusing on an interplanetary exploration mission scenario. A comprehensive analysis of potential fault cases in camera sensors used within the VBN pipeline is presented. The causes and effects of these faults are systematically characterized, including their impact on image quality and navigation algorithm performance. To support this analysis, a simulation framework is introduced to recreate faulty conditions in synthetically generated images, enabling a systematic and controlled reproduction of faulty data. The resulting dataset of fault-injected images provides a valuable tool for training and testing AI-based fault detection algorithms. The final link to the dataset will be added after an embargo period. For peer-reviewers, this private link[1] is available.

1. Introduction

The most promising trends in recent space exploration target ambitious technological innovations and advanced operations, such as Lunar and Martian exploration, in-orbit refuelling and maintenance, and active debris removal. These complex endeavours heavily rely on vision-based navigation (VBN), marking a significant departure from traditional satellite operations. As we venture into these uncharted territories, the reliability and accuracy of visual sensors, including LiDARs and cameras, become paramount to mission success.

In the present work, framed within the project *Astrone KI* [1], the VBN approach is used to perform autonomous relocation in a harsh Small Solar System Body (SSSB) environment, leveraging AI to enhance the VBN task [2,3]. Additionally, the project aims for an AI-augmented FDIR subsystem [4,5], in order to manage potential faults arising in onboard sensors and maintain nominal VBN operations. Since FDIR based on the traditional Packet Utilization Standard (PUS) [6] struggles with the multi-dimensional nature of image data, AI has been proposed as a promising solution for VBN-based FDIR, thanks to its historically-proven effectiveness in dealing with image processing tasks.

However, the peculiarities of the VBN approach require the present analysis to investigate on the nature and source of the possible issues occurring in visual sensors. Depending on the application, different factors can affect commonly employed cameras and LiDARs, such as dust covering the optics after landing, malfunctioning pixels, light reflections, etc. A precise assessment of these faults, up to a proper Failure Modes and Effects Analysis (FMEA, [7]), will mark a key step towards the reliability of VBN systems. Nevertheless, the development of an accurate simulation environment, inclusive of the aforementioned failure modes, is crucial to the integration of AI into spaceflight solutions. The ability to realistically model and generate diverse fault scenarios in visual data is essential for training and validating AI algorithms, ensuring their effectiveness and reliability in the FDIR task.

A first minor contribution of this work resides in an analysis of faults affecting optical sensors which enable the vision-based navigation subsystem of *Astrone KI*, laying the groundwork for the development of

* Corresponding author at: Claude-Dornier Straße, Immenstaad am Bodensee, 88090, Germany.
E-mail address: r.gallon@tudelft.nl (R. Gallon).

[1] The link will be available upon publication. For the time being, readers are kindly requested to contact the corresponding author to obtain access to the dataset.

https://doi.org/10.1016/j.actaastro.2025.10.071
Received 3 July 2025; Received in revised form 26 September 2025; Accepted 28 October 2025
Available online 5 November 2025

the AI-powered FDIR subsystem. Based on this, this work takes initial steps towards the adoption of AI technologies in space applications, whose ultimate goal should be to enable safer, more efficient, and more ambitious missions in our solar system and beyond. As main contribution, a dataset of synthetic faulty images is generated. The dataset incorporated the considered faults with by means of documented implementation approaches, and constitutes a useful tool for training and testing AI algorithms for FDIR.

The structure of this paper is organized as follows. Section 2 reviews relevant work concerning the simulation of camera sensor faults and the generation of datasets for VBN applications. Section 3 introduces the fault cases under consideration, detailing the specific issues each fault imposes on the camera, the underlying causal effects, and their manifestation in the resulting image. Section 4 outlines the dataset generation methodology, beginning with a description of the simulation environment, followed by the image acquisition strategy, the fault injection process, and the generation of corresponding label masks. Finally, Section 5 concludes the paper by summarizing the key achievements and providing an outlook on potential future developments.

2. Related work

Since the introduction of AI-based VBN solutions in the space domain with the Mars Exploration Rover (MER) mission [8], the significance of data for the training and testing of algorithms has become increasingly evident. The requirements of having realistic and representative images has led researchers to address the topic of dataset creation to support their AI developments [9], [10], [11]. Datasets for the specific case of landing and navigation on extra-terrestrial bodies are very limited, primarily because of the reduced number of missions targeting the mentioned scenarios, but also because of the challenges of *labelling*. The process of labelling consists in assigning an image or each of its pixels to a specific class, which will be employed during the AI training to learn what the specific image or pixel represents. Typical examples of classes used in VBN can be landmarks (e.g., rocks, crater), *Sun*, *sky* and *distant landscape*, but also *solar panel*, *satellite body*, *docking port*, depending on the specific VBN task.

While missions such as Hayabusa, Hayabusa 2 and OSIRIS-REx pioneered the exploration of SSSBs and retrieved a gigantic number of images and scientific data during their respective landing phases with their camera sensors [12,13], the effort of labelling every single image or even pixel to be able to use those images for training navigation tasks has not yet received an efficient solution, thus leaving a significant gap in the state of the art. The problem of labelling is far from being solved. Thus, researchers are mostly oriented towards the generation of synthetic datasets which provide a faithful representation of the environment and automatically output labels accordingly. Some synthetic datasets exist in literature to accomplish navigation tasks, including satellite pose estimation, in-orbit rendezvous and landing [9], [10], [11]. Sharma et al. [9] and Park et al. [10] are pioneering works in the field of spacecraft pose estimation, dealing with the problem of navigation around non-cooperative space objects and their relative position estimation. Sharma et al. [9] proposes a synthetic dataset comprehensive of camera images of an actual spacecraft mock-up, which are fused to simulated images of the Earth and the Sun to enhance the realism of the result. For the same pose estimation task, Park et al. [10] proposes an improvement of the SPEED dataset [9], SPEED+, which comprises fully synthetic images and Hardware-in-the-Loop test images taken exploiting a robotic simulation testbed. Lebreton et al. [11] addresses the task of pose estimation in two different scenarios, one is relative pose estimation during rendezvous of non-cooperative objects and the other is self pose estimation during Moon landing. In Lebreton et al. [11], a number of different datasets belonging to the two scenarios is proposed. The synthetic datasets are entirely generated with a high-fidelity image simulator (SurRender, [14]), employing digitalized models of mock-ups of the Envisat satellite. The same mock-up is used to obtain completely experimental images exploiting the TRON facility at DLR [15], which was already utilized in Park et al. [10]. Finally Generative Adversarial Networks (GANs) are leveraged to reproduce the effects of real camera acquisition on the dataset images, trying to bridge the domain gap that real images have with respect to synthetic ones.

As Lebreton et al. [11] is the only attempt reported in the state of the art to obtain a realistic dataset for landing applications, it does not cover the exploration of extra-terrestrial bodies, as in the case of *Astrone KI*. However, the similarities between the terrain of the Moon and any other Small Solar System Body could support the use of a dataset generated on the former in training and testing an AI for navigating on the latter, but no work has been done so far in this field. Recently, NASA has been releasing labelled data from the Opportunity and Spirit rovers from NASA's Mars Exploration Rovers (MER) mission [16], from the Curiosity rover from Mars Science Laboratory (MSL) mission [17, 18] and from the HiRISE sensor on the Mars Reconnaissance Orbiter (MRO) [19]. These datasets comprise images of landmarks on the Mars surface and have their own associated labels, based on which landmark is represented. Consequently, the labelling task has been consistently addressed employing experts from the mentioned missions, leading to the release of various Martian datasets. Targeted tasks include rock detection [20], rock segmentation [21] and landmarks classification and semantic segmentation [22,23].

Differently, no relevant work of dataset generation or even AI application has been found, to the best of the author's knowledge, in the field of FDIR for space-employed camera sensors. Therefore, the present research contributes to an innovative FMEA of camera sensors, mainly leveraging industrial expertise and previous space exploration missions and the respective camera design documents, where available. Cassini [24], Hayabusa2 [13] and OSIRIS-REx [12,25] are examples of interplanetary exploration missions presenting similar traits to *Astrone KI*. While Cassini does not include any landing, Hayabusa2 and OSIRIS-REx actually comprise a landing phase, but they can all contribute to the identification of failure modes which also apply to *Astrone KI*. West et al. [24], Yamada et al. [13], Bos et al. [12,25] describe the design challenges of the camera sensors of the mentioned missions, reporting useful information for the occurring faults. Specifically, all the cameras focus on the analysis of the foreseen broken pixels rate, making considerations on the mission lifetime and the criticality of this fault for the navigation algorithm. Besides, strategies to reduce the occurrence of other fault cases are described. Inner and outer coating of the camera and the lenses, lens hoods and other mission-specific countermeasures can be found in the mentioned literature. Particularly useful is the work done in Cassini (Figure 34 in Porco et al. [26], Figure 18 in West et al. [24]), where the analysis of reported images has highlighted typical fault patterns due to broken pixels and straylight.

Concerning other fault cases occurring in space cameras, the present research had to focus on other domains to retrieve applicable simulation strategies. Chapman et al. [27,28,29,30,31] are a series of works about the occurrence of broken pixels in commercial cameras, that were extensively employed to retrieve a simulation methodology applicable to the space domain. King [32], Maughan [33], Kilgard [34] are pioneering works in the field of straylight simulation, and were employed in the present work to guide the development, as the same methodologies are applicable to the space domain too. Moreover, the straylight field reports the presence of datasets in literature, addressing the flares removal task, targeting images from a variety of real-life scenarios [35,36], [37]. These datasets can be efficiently used for pre-training or testing of specific AI solutions targeting flares removal and employing the dataset proposed in this work.

Other faults presented in this work employ simulation strategies derived from the industrial expertise of the authors, as proper literature describing the fault occurrence and/or the simulation strategy is limited.

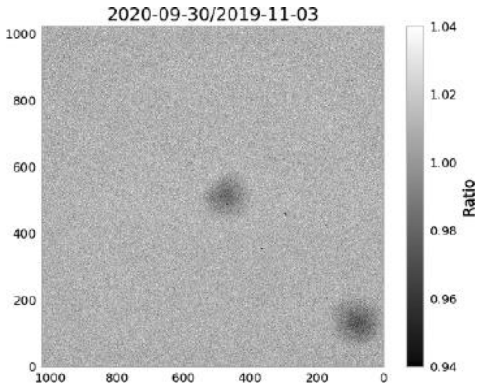

Fig. 1. Magnified view of dust particles deposited on the lens of the telescopic Optical Navigation Camera (ONC-T) of the Hayabusa2 mission. The original image is provided in Yamada et al. [13].

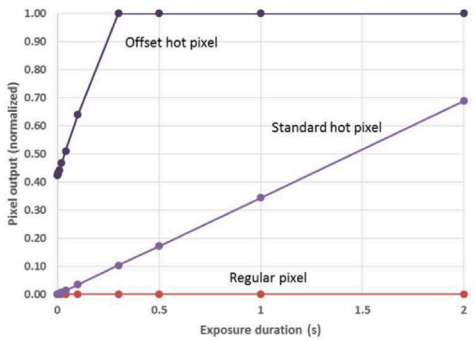

Fig. 2. Comparison of regular and hot pixels sample dark response. The regular pixel keeps its output constant over the Exposure Time, while hot pixels present a linear behaviour [30].

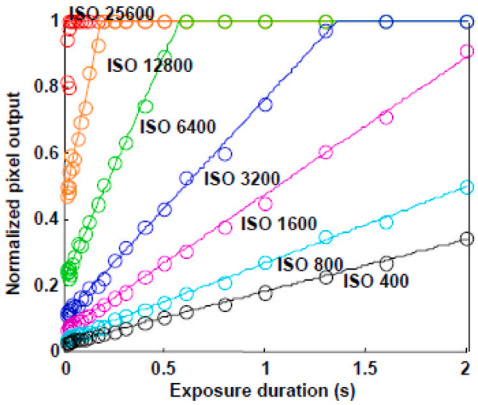

Fig. 3. Dark response of the same hot pixel at varying ISO [28].

3. Considered fault cases

This section presents the results of our investigation into faults affecting Vision-based Navigation systems. The approach to faults identification encompassed a two-steps process. First, industrial expertise of visual sensors engineers was leveraged to identify common failure modes. Second, an extensive literature research was conducted to provide a detailed per-fault analysis and the simulation strategy described in Section 4.4. The identified faults are not classified as critical or non-critical from the mission point of view, as the scope of the survey is rather to shed light on the various failure modes that can occur in camera sensors.

Identified faults are tailored on the *Astrone KI* use-case, meaning that they are analysed in relation to the specific mission scenario. Other interplanetary exploration missions or even with different operational environment (i.e. planetary exploration, Earth Observation) may need additional, modified or reduced failure modes applicability. The main drivers derived from *Astrone KI* are mainly the interplanetary unshielded radiation environment, the prolonged mission lifetime, the fast dynamics and the expected latency of performing onboard procedures commanded from the ground.

The description of each fault presented hereafter is structured as follows: an overview of the fault effects on the final image is given, followed by a description of the causes and effects at component level, which constitute the main driver to the simulation strategy described in Section 4.3.

Dust on optics

The problem of dust on optics is particularly critical for optical systems, because it represents a consistent reduction of the visibility. It is especially present in landing vehicles, where dust may raise from the ground in the landing site. The dust rises by effect of the force exerted by the thrusters and electrostatically sticks to the optical lenses, accumulating in grains. The grains can aggregate in different quantities and acquire different shapes, determining a more or less consistent shadowing of the lens area. Fig. 1 presents a realistic example of the effects of dust grains. As noted by Yamada et al. [13], the visible artifacts in Fig. 1 are potentially dust particles originating from the Hayabusa2 target body. These particles are posited to have adhered to the lenses despite the presence of a baffle. Given that the Hayabusa2 mission involved only a single touchdown on its target body, leading to the dust accumulation illustrated, the operational environment poses a more significant challenge for the Astrone KI mission, which is supposed to execute multiple relocations across the comet's surface.

Broken pixels

Broken pixels are a diffused problem in camera sensors, equally affecting commercial and space-grade cameras. The defects can assume different patterns based on their root cause, leading to limited camera visibility and capability to distinguish details. Moreover, infinite and/or zero value in the pixel output may also be misprocessed in the navigation algorithm, leading to incorrect results or ultimately processing faults.

The problem of defect pixels, and in general of increasing dark currents in CCD and CMOS detectors, is usually faced via mission-dependent calibration procedures, documented in [12,13,24,25]. Furthermore, Bos et al. [12], Bos et al. [25], West et al. [24] report data about the number of broken pixels occurring in OSIRIS-REx and Cassini, showing that they represent no critical issue in the respective use-cases. In Bos et al. [25] it is observed that a concentration of > 3 broken pixels in a close neighbourhood can cause misdetection of objects on the asteroid surface, a condition that applies also to the *Astrone KI* case and deserves investigation.

We differentiate between broken pixels defects caused by the external environment or internal electronic components. The external environment affects pixel defects by means of the space radiation impinging the detector surface. Depending on the specific mission profile, e.g. LEO, GEO or interplanetary orbits can experience significantly different radiation environments [40], determining the number of defective events. Particles impinging a pixel cause a modification in its response to light. Pixels whose properties have been modified by a transient radiation particle are called *Hot Pixels*. In Fig. 2, it is possible to observe different responses of hot pixels as measured in a fully dark environment (*dark response*).

Increasing the Exposure Time, the pixel output will increase and potentially saturate if the impinging radiation has introduced an offset in its response curve. Besides, the two extreme cases of a fully hot and fully dead pixels, called *Salt&Pepper Defect*, can be present,

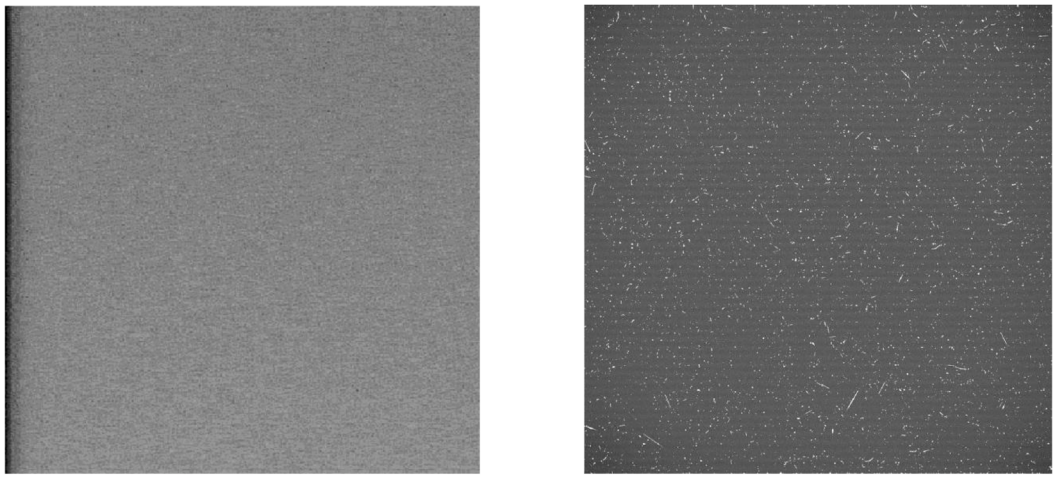

Fig. 4. Broken lines (left) and Broken pixels (right) sample images from Cassini's Imaging Science System (ISS), provided in Porco et al. [26]. The left image depicts ≈ 13 broken lines of pixels due to an electronics failure. The right image depicts single pixels and aggregation in stripes due to the exposure to cosmic rays.

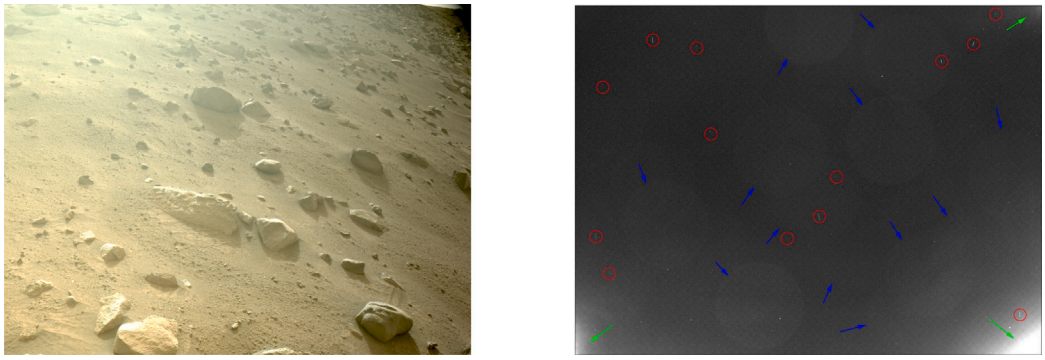

Fig. 5. Solar glare affecting the Navigation Cameras (NavCam) of Perseverance rover in the Mars 2020 mission, retrieved from National Aeronautics and Space Administration [38] (left). Solar flares affecting the Touch And Go Camera System (TAGCAM) of the OSIRIS-REx mission, retrieved from Bos et al. [25] (right). Circular flares in the TAGCAM image are pointed by blue arrows.

Fig. 6. Vignetting effect shown in images from the Navigation Cameras (NavCam) of the Opportunity rover in the Mars Exploration Rovers (MER) mission (left) and Perseverance rover in the Mars 2020 mission (right). Images available at NASA Planetary Data System [38,39].

but they have been observed to be much less frequent than what has been measured because they are often confused with offset hot pixels [41]. This last statement is derived from the works of Leung et al. [41,42], Chapman et al. [27,28], which proved it in smartphone commercial cameras, but has no further evidence in space, where the radiation effect is eventually stronger. This latter feature may lead to increased *Salt&Pepper Defect* rate, but this has not been verified yet. Further studies [30,31] also suggest a correlation between the broken pixel occurrence, empirically showing that the occurrence of a couple of broken pixels in a 3x3 neighbourhood is higher than the probability expected only considering the generalized birthday problem of random processes.

Camera features affecting the broken pixels occurrence include Exposure Time and sensor sensitivity to light (ISO), as these two variables control the amount of light absorbed by the detector. This, in turn, determines the characteristics of the hot pixel in the final image. Based on the response curve in Fig. 2, the response of hot pixels increases with the Exposure Time, eventually leading to increased brightness. Fig. 3 shows the hot pixels correlation with ISO, which determines a generic increase of the pixel output with increasing ISO.

Other pixel defects occurring in camera sensors are related to random faults in the electronics, especially the readout components. *Readout* refers to the process of retrieving the current measure from the single pixels to process the final image. There are different readout mechanisms based on the existing sensors architectures [43]. In space, Charge-coupled Device (CCD) sensors were preferred to Complementary Metal-oxide Semiconductor (CMOS) sensors [12,25] in interplanetary missions for their inherently higher sensitivity [13,24,43], while more recent missions shifted to CMOS detectors which in turn provide lower cost and power consumption [43]. In addition, the two kinds of detectors present different readout mechanisms, which in turn affect the associated failure modes. In CCD sensors charges shift towards the readout component and are read in series, while in CMOS they are read in parallel per row (or column). More details on this two construction choices are out of the scope of this work and can be found in Waltham [43].

An common effect in cameras employing both CCD and CMOS sensors is the spreading of a defect to other pixels than the broken one, which is caused by *demosaicing* algorithms. *Demosaicing* refers to algorithms employed in the camera software responsible of postprocessing the output of the detector, in order to retrieve the colour value of a single pixel from its current output. As the detector is made by different colour filters, responsible of grasping different light colours, the output of the different filters is interpolated to retrieve a final colour for the associated pixel [30]. This interpolation involves neighbouring pixels in a way that whenever a pixel is broken, its defective output will corrupt also the pixels around it. Different algorithms of demosaicing exist, where *bilinear interpolation* is the simplest [44]. Chapman et al. [31] provides an example of how bilinear interpolation spreads the defect of a single pixel in its von Neumann neighbourhood (i.e. top, bottom, right, left pixels).

More defects in common between CCD and CMOS detectors are associated to other components taking part in the readout. Registers, multiplexers and row select transistors [43] regulate the row/columns to read out, selecting the pixel coordinates and shifting them along the process. A defect in these components can generate full row or column defects, which means completely saturated or dead lines of pixels crossing the whole image. These defects can occur during flight operations after an unforeseen fault in the electronics. By construction, Full-frame and Frame-transfer CCD sensors [43] can experience mutually exclusive row or column defects, while Interline-transfer CCD and CMOS [43] can be affected by both row and columns at the same time.

A real image to depict the effects described in this paragraph is provided in Fig. 4.

Straylight

Straylight is a generic term referring to unwanted image artifacts, specifically *flares* and *glares*, caused by internal reflections of light within a camera's system of lenses. Flares are discrete, burst-like textures originating from a light source, whereas glares are a more diffuse spreading of light. These artifacts are unique to the specific camera and lens configuration. In space, straylight can be caused by the reflection of light from sources like the Sun, celestial bodies, or stars. Its primary effect is a degradation of image quality, obscuring objects with bright, coloured artifacts and reducing the camera's ability to resolve them. An example of straylight faults affecting real camera sensors is presented in Fig. 5.

Attenuation measures usually include baffles and inner coating [45], or ultimately the enforcement of operational constraints to limit the Sun phase angle of the camera.

The effects of straylight on camera lenses have been studied in literature across different fields such as cinema, animation and video games. First approaches in straylight simulation employed fixed textures blending with the image [32–34]. Textures were generated *a priori* with respect to the images and could assume shapes of e.g. *rings*, *halos* and *orbs*. They were positioned in the image on a line linking the light source to the centre of the image, while their brightness varied according to the distance from the light source. More recently, Keshmirian [46], Hullin et al. [47], Lee and Eisemann [48] proposed refined methodologies based on physical lens simulation, marking the current standard in the state of the art.

Vignetting

The vignetting effect is a well-known inherent effect of any existing camera system, and it is caused by the light hitting the sensor at different angles, depending in turn on the specific lenses allocation. Lower angles are associated to off-axis sources and longer distances for the light to cover to impinge the detector, determining a reduction of the transferred power. As a consequence, Vignetting causes a generalized darkening of the edges of the image (Fig. 6) which can be described by the *cosine-fourth law* (Eq. (1)) [49].

$$\frac{E_\Theta}{E_0} = \cos^4 \Theta \tag{1}$$

In Eq. (1), E_Θ and E_0 are the on-axis and off-axis illuminance respectively, while Θ is the off-axis angle. The *cosine-fourth* law is valid for *natural vignetting* only, which is present in any camera sensor. Other vignetting effects include *mechanical vignetting* and *optical vignetting* . For the sake of the present work, optical vignetting is not covered in detail, but since it provides an edge-darkening effect as well as natural vignetting, the *cosine-fourth* law is assumed to describe both effects.

AI is well-suited for solving the problem of vignetting correction. While traditional software typically handles this as a post-processing step, AI augmentation can overcome this limitation by allowing corrections to be applied in real time.

Optics degradation

Optics degradation causes a gradual worsening of the optical performances of a system of lenses along the operational lifetime, affecting objects recognition capabilities, details resolution and navigation functionalities in general. This phenomenon is usually associated to environmental factors such as atomic oxygen, UV radiation, outgassing and cross-contamination, charged particles, impact of micrometeoroids and/or space debris and dust deposition [50]. These factors degrade the surface properties of exterior lenses, changing the way light crosses them, and can be mitigate via surface coatings, treatments, or specific materials choice. As these effects can be diverse on final images, in this work a diffused image blurriness is considered as only consequence. A more detailed study of degradation effects can be a promising field of future work.

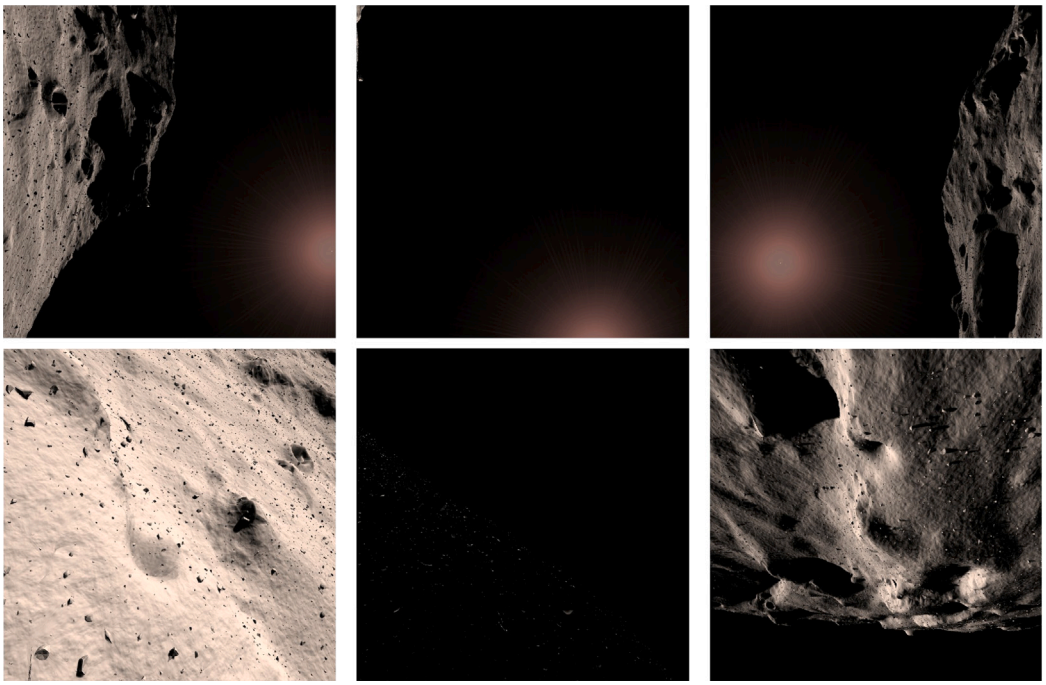

Fig. 7. Sample images from the simulation of the 67P/Churyumov-Gerasimenko comet environment.

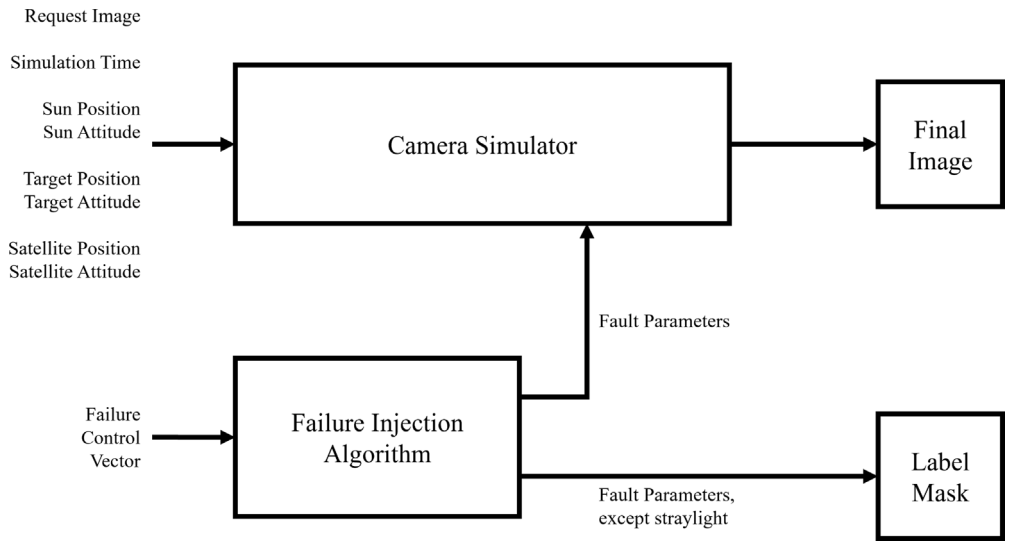

Fig. 8. Overview of the dataset generation environment. Straylight parameters are not employed in the *Label Mask* block, as detailed in Section 4.4.

4. Dataset generation

4.1. Simulation environment description

This study selects comet 67P/Churyumov-Gerasimenko for the environment simulation, made feasible by publicly available data collected during the Rosetta mission and provided by the European Space Agency (ESA) [51]. The environmental rendering is executed using *CamSim*, a Camera Simulator developed by ASTOS Solutions GmbH [52]. This simulator is capable of rendering celestial bodies and modelling spacecraft dynamics and sensor performance. Examples of the simulated environment are presented in Fig. 7.

Furthermore, the software incorporates a Simulink interface, schematically depicted in Fig. 8, which functions as the primary simulation infrastructure for this work.

In Fig. 8, the CamSim block is responsible for images generation is located in the top part, while the bottom pipeline is dedicated to the fault injection and simulation mechanism. The *Failure Injection Algorithm* block computes the parameters of the faults simulation algorithm and commands faults injection to CamSim, while it also contributes to the data annotation (*Label Mask* block). CamSim inputs comprise the state vectors of each body involved in the simulation, allowing to command the environment and the spacecraft orientation (i.e., the camera and LiDAR field of view), in order to obtain diverse landscapes and illumination conditions in the final image.

The Target body rendering is divided in patches starting from the original full model [51] and the spacecraft motion is defined within the patch. To this aim, we introduce convenient reference frames, presented in Appendix A. We hereby provide details about the Local-Level frame, as it plays a crucial role in the definition of dataset variables which will be employed in the simulation. The Local-Level frame is

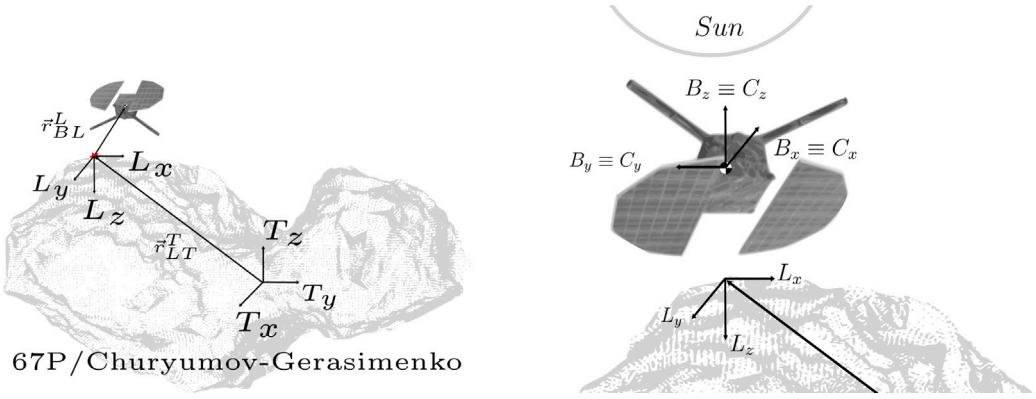

Fig. 9. Reference frames employed in the dataset generation.
Source: Adapted from Olucak et al. [2].

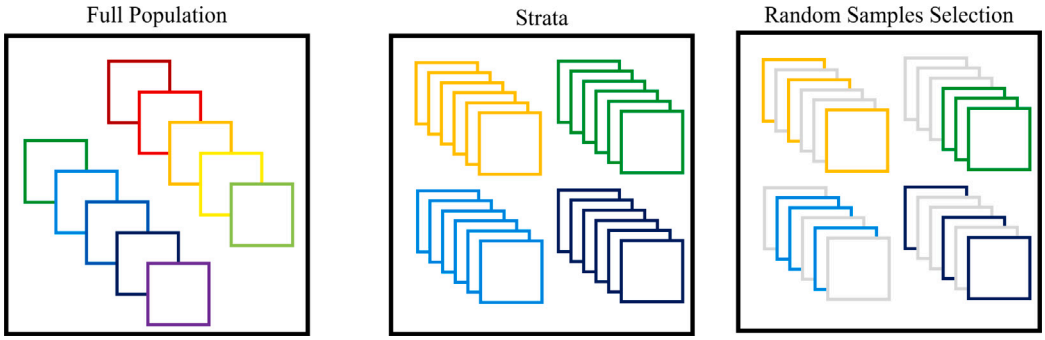

Fig. 10. Stratified sampling technique.

computed locally on the terrain patch, via a separated procedure that slices the patch from the full comet model and interpolates the model points to enhance the resolution. The Local-Level frame coordinates in each point of the enhanced-resolution map are then computed based on the gravity vector. The gravity data are provided together with the global comet model in Preusker et al. [51]. The details of the patching procedure are out of the scope of this paper.

Fig. 9 gives a graphical overview of all the reference frames involved in the simulation. Specifically, $\mathcal{T}$ indicates the Target-Body Centred Target-Body Fixed Frame, $\mathcal{L}$ is the Local-Level Frame, $\mathcal{B}$ and $\mathcal{C}$ are the Body-Fixed and Camera-Fixed Frames respectively.

4.2. Image acquisition

The image acquisition procedure outputs images by setting randomized parameters for the environment, sensor and fault variables. No specific spacecraft trajectory is assumed as input, leading subsequent images to be uncorrelated. This assumption avoids on one side the presence of biases in data derived from the constrained variables values. On the other side, it limits the AI application employing the dataset to process only one image at a time.

First, representative independent variables of the environment and parameters of the camera sensor shall be identified. The definition of the camera sensor in *Astrone KI* is not detailed up to the identification of a specific camera model or set of lenses, hence none of them are employed in the dataset generation. The definition of the optical parameters is kept as generic as possible, aiming to embed in the data the representation of diverse hardware setups, and boost the applicability range of the trained AI.

The camera and the image acquisition parameters employed in the present work are specified in Table 1.

Table 1
Camera and Image Parameters.

Parameter	Value
FoV	65 deg
Resolution	1024 × 1024 pixels
Colour Depth	8 bits per channel
Number of Channels	4 (R, G, B, I)
Values per Channel	256 (0 – 255)
Image Format	RGBI
Colour Range	RGB

Table 2
Boundaries of dataset variables.

	Sun-facing Case	*Sun-averted* Case	Reference Frame
Sun Azimuth ρ	$0 < \rho < 2\pi$	$0 < \rho < 2\pi$	Local-Level, $\mathcal{L}$
Sun Elevation γ	$-\frac{\pi}{6} < \rho < \frac{\pi}{6}$	$-\frac{\pi}{6} < \rho < \frac{\pi}{2}$	Local-Level, $\mathcal{L}$
Spacecraft Roll Angle ψ	$-FoV/2 < \psi < FoV/2$	$FoV/2 < \psi < 2\pi - FoV/2$	Body-Fixed, $\mathcal{B}$
Spacecraft Pitch Angle θ	$-FoV/2 < \psi < FoV/2$	$FoV/2 < \psi < 2\pi - FoV/2$	Body-Fixed, $\mathcal{B}$
Spacecraft Yaw Angle φ	$0 < \varphi < 2\pi$	$0 < \varphi < 2\pi$	Body-Fixed, $\mathcal{B}$
Spacecraft Position, x_B	$-150 < x < 150$ [−]	$-150 < x < 150$ [−]	Local-Level, $\mathcal{L}$
Spacecraft Position, y_B	$-150 < x < 150$ [−]	$-150 < x < 150$ [−]	Local-Level, $\mathcal{L}$
Spacecraft Position, z_B	$0 < x < 100$ [−]	$0 < x < 100$ [−]	Local-Level, $\mathcal{L}$

The environment variables that influence the scenario and must be varied to generate a comprehensive dataset include the relative Sun-comet position, the satellite position on the surface of the comet and

Table 3
Dust on optics fault parameters. The values of covariance are strictly related to the 30 × 30 pixels grid used for generating textures.

Parameter	Effect	Lower bound	Upper bound
Number of Dust Grains per Image	–	30	100
Max of the Gaussian Distribution	Grain darkening	100	200
Covariance of the Gaussian Distribution, σ_{xx}	Grain Shape	3	6
Covariance of the Gaussian Distribution, σ_{yy}	Grain Shape	3	6
Covariance of the Gaussian Distribution, $\sigma_{xy} = \sigma_{yx}$	Grain Shape	0	0

the satellite attitude. These variables are deemed sufficient to express a representative dataset.

Due to the presence of the straylight fault (Section 3), which is bounded to the Sun position, a classic random sampling over the unconstrained boundaries of variation of the dataset variables would not be an efficient strategy to apply. Specifically, this fault occurs when the Sun stays in the camera FoV or in its proximity. Hence, utilizing a classic random sampling would inject an unpredictable number of straylight features in the dataset, corresponding to those image where the Sun randomly occurs in a suitable position. This behaviour would result in an unpredictable balance of the classes in the dataset, ultimately leading to an insufficiently representative number of certain faults with respect to others. Conversely, the other faults listed in Section 3 do not depend on the Sun position, thus can be randomly injected across all the generated images, as detailed in Section 4.3.

Stratified sampling (Fig. 10) has been utilized to ensure uniformity across the fault classes. This methodology involves a two-stage sampling process. In the first stage, the dataset variables are constrained to include instances where the Sun is within specific boundaries of the Camera-Fixed frame. Random sampling is then conducted to capture images at various Sun positions, ensuring the presence of straylight. In the subsequent stage, the camera is directed away from the Sun, and random sampling is again employed to collect the remaining images for the dataset. The number of images acquired in both phases ensures an appropriate percentage of samples belonging to the straylight class.

The simulation setup requires the Sun position and the Target position and attitude, together with the spacecraft attitude, to be defined in the International Celestial Frame ($\mathcal{J}$), while the spacecraft position is defined in the $\mathcal{T}$ frame. The camera sensor is instead attached to the spacecraft, so its orientation is relative to the $\mathcal{B}$ frame. Although the initial Sun state is imposed in the $\mathcal{J}$ frame, its position with respect to the Target body is commanded in the $\mathcal{L}$ frame, employing Azimuth (ρ) and Elevation (γ). The major advantage offered by this reference frame resides in the intuitiveness, allowing to achieve a comprehensive coverage of the illumination conditions in the dataset. Remarkably, for reasons related to the CamSim implementation, the Azimuth and Elevation of the Sun position are translated into the relative Target body attitude, which is then employed in the simulator to obtain the defined illumination. The Sun position and the Target attitude are indeed coupled variables, thus can be used equivalently to the dataset generation purpose. The satellite attitude is set to rotate with respect to the Body-Fixed frame according to the Euler Angles ψ (roll), θ (pitch) and φ (yaw). The Camera-Fixed frame and the Body-Fixed frame are set as coincident as to the purpose of the dataset generation the specific allocation of the camera in the vehicle is irrelevant. The satellite position on the comet surface is varied in the Local-Level frame according to the definition of the single patches.

Table 4
Broken pixels and lines fault parameters, dimensionless.

Type	Parameter	Lower Bound	Upper Bound	Categories
Broken Pixels	Number of Broken Pixels per Image	10	150	–
	Brightness of the Pixel	0	255	–
	Brightness of the Subtended Line (CCD)	0	65	–
	Subtended Line Direction (CCD)	–	–	Up, Down, Right, Left
Broken Lines	Number of Broken Lines	1	5	–
	Entity of the Broken Lines	–	–	Black, White

Table 5
Straylight fault parameters. Flares Position ensures no texture falls outside the image. Flares Radius and Brightness are set empirically to comply with different size and brightness of the textures provided by the software.

Parameter	Lower bound	Upper bound
Number of Injected Flares	1	10
Index of Injected Flares	1	10
Flares Position	0.5	2
Flares Radius	0.05	0.3
Flares Brightness	1.5	2.5

Table 6
Vignetting fault parameters.

Parameter	Effect	Lower bound	Upper bound
On-axis Illuminance	Darkening effect	105	255

Table 7
Optics degradation fault parameters.

Parameter	Effect	Lower bound	Upper bound
Gaussian Filter Size (pixels)	Blur Intensity	3	17

Table 8
Parameters of the Injected faults in Fig. 24.

	Parameter	Value
Dust on Optics	Number of Dust Grains per Image	50
Broken Pixels	Sensor Type	CCD
	Number of Broken Pixels per Image	9
	Subtended Line Direction (CCD)	right
Broken Lines	Number of Broken Lines	5
	Entity of the Broken Lines	3 White, 2 Black
Straylight	Number of Injected Flares	3
	Index of Injected Flares	3, 4, 5
	Flares Position	0.83, 1.00, 1.17
	Flares Radius	0.11, 0.13, 0.16
	Flares Brightness	1.50, 1.50, 1.50
Vignetting	On-axis Illuminance	255
Optics Degradation	Gaussian Filter Size (Pixels)	3

Due to the sampling methodology detailed earlier, it is convenient to start the image generation process by aligning the $\mathcal{L}$, $\mathcal{B}$ and $\mathcal{C}$ frames with their z-axes pointing at the Sun, which means fixing the Sun in the centre of the Camera FoV. This scheme allows a convenient definition of the boundaries of the dataset variables, especially the spacecraft attitude angles, where the Sun pointing corresponds to 0 deg.

The complete dataset generation procedure is detailed hereafter:

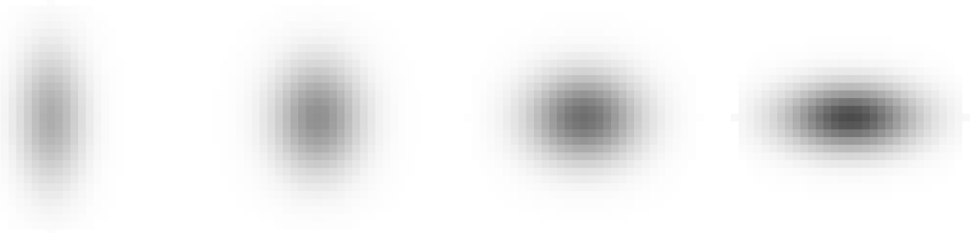

Fig. 11. Sample textures of dust grains with the respective parameters. From left to right: *Max of the Gaussian Distribution* (100, 133, 166, 200); σ_{xx} (3, 4, 5, 6); σ_{yy} (3, 4, 5, 6); σ_{xy} (0, 0, 0, 0).

Table 9
Summary of dataset properties.

Property	Value
Images	
Number of Images	5000
Classes ID	0–6
Images Resolution	1024 × 1024
Number of Channels	3 (RGB)
Colour Bit Depth	8 bits per channel
Masks	
Number of Masks per image	4
Number of binary labels per image	2
Masks Resolution	1024 × 1024
Number of Channels	1 (binary)

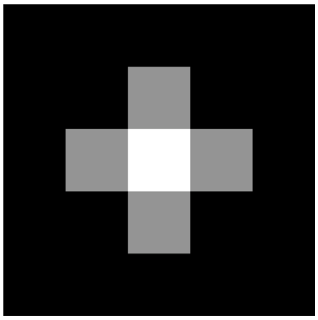

Fig. 12. Sample texture of a single broken pixel.

1. Generate a random Target attitude to fix the illumination conditions.
2. Retrieve the attitude of the spacecraft imposing the alignment of the camera with the Sun direction.
3. Randomize the spacecraft attitude to keep the Sun in proximity of your camera FoV → Acquire image.
4. Randomize the spacecraft attitude to keep the Sun outside your camera FoV → Acquire image.

Step 3 and 4 of the procedure differentiate by the boundaries of the variables, set to obtain Sun-facing or Sun-averted orientations. All the possible illumination conditions and attitude angles of the spacecraft are efficiently covered, ensuring sufficient variety of scenarios framed by the camera.

Table 2 summarizes the dataset variables and their boundaries in the two stages of sampling.

The Sun elevation in the *Sun-facing* phase presents narrow bounds and implies that the Sun always stays close to the horizon of the landscape seen by the *Astrone KI* camera. This choice aims to obtain a balanced percentage of images depicting only the sky and images where the terrain is also visible. A wider choice of the γ interval would lead to increased percentage of sky-only images, penalizing to the ones where the terrain is present. The choice of the specific values has been done empirically as there is no need for an accurate computation, as long as both sky-only and terrain images present statistically significant samples. In fact, both cases are significant from an anomaly detection point of view, especially sky-only images which cover those fault cases causing loss of nominal attitude and off-axis camera pointing.

4.3. Faults injection

The mechanism of faults injection in the whole dataset aims to obtain a balanced dataset including sufficiently representative image samples per fault. This requirement is key to make sure that all the faults will be correctly learnt by an AI employing the dataset. However, further balancing depends on the task that the algorithm accomplishes, and is not in the scope of this work.

The present dataset generation method allows to set the percentage of faulty over nominal image samples, where the faulty samples contain uniformly distributed fault cases. This approach enables sufficient freedom to be used in many different tasks. For instance, setting a percentage to 0.5 yields a balanced dataset for binary (i.e. faulty, not faulty) classification, while a percentage of *1/(# fault classes)* can be more suitable to multi-class classification.

The fault injection is regulated by a matrix defined once at the beginning of the whole generation process, containing flags for commanding the occurrence of the single faults in each of the generated image samples. This matrix has size *# fault classes* × *number of samples* and is generated based on these two parameters. The matrix generation does not include the straylight fault, which is appended as last row and it is constructed as shown in Eq. (2):

$$\begin{cases} 1, & \text{for } i = 1 : n_{\text{straylight}} \\ 0, & \text{for } i = n_{\text{straylight}} + 1 : n_{\text{tot}} \end{cases} \tag{2}$$

where n_{tot} is the total number of samples to generate, while $n_{straylight}$ is the number of samples commanded to face the Sun and derives from the specified nominal-to-faulty samples percentage. In the resulting dataset, the first $n_{straylight}$ images include a straylight fault, while the other faults are randomly distributed. This specific construction descends from the stratified sampling technique (Section 4.2), which requires the straylight fault to be simulated first in the generation procedure. The methodology exposed for the straylight fault is generalizable to any other fault related to the Sun presence.

4.4. Faults simulation

This section details the simulation strategy of the faults identified in Section 3. Each faults is correlated with the detailed implementation and relevant parameters. CamSim-specific features are utilized for this scope, specifically:

- Pixel Overwriting: the simulator allows to define textures of user-defined pixel values that are merged with the images at colour channel-level. This features allows fine-grained modification of the images at pixel-level, enabling the injection of faulty artifacts of various sizes.

Table 10
Applicability of fault cases based on common mission objectives.

		Classes			
		Earth observation	Landing	SSSB exploration	Interplanetary
Faults	Dust on Optics	N	Y	Y	N
	Broken Pixels	N[c]	N	Y	Y
	Broken Lines	Y	Y[a]	Y	Y
	Straylight	Y	N	Y	Y
	Vignetting	Y	Y	Y	Y
	Optics Degradation	N[b]	N	Y	Y

[a] Faults causing Broken Lines can also occur in the cruising phase.
[b] Given that sufficiently high-quality parts, shielding and ultimately correction algorithms are employed.
[c] Carefully adapt the radiation environment.

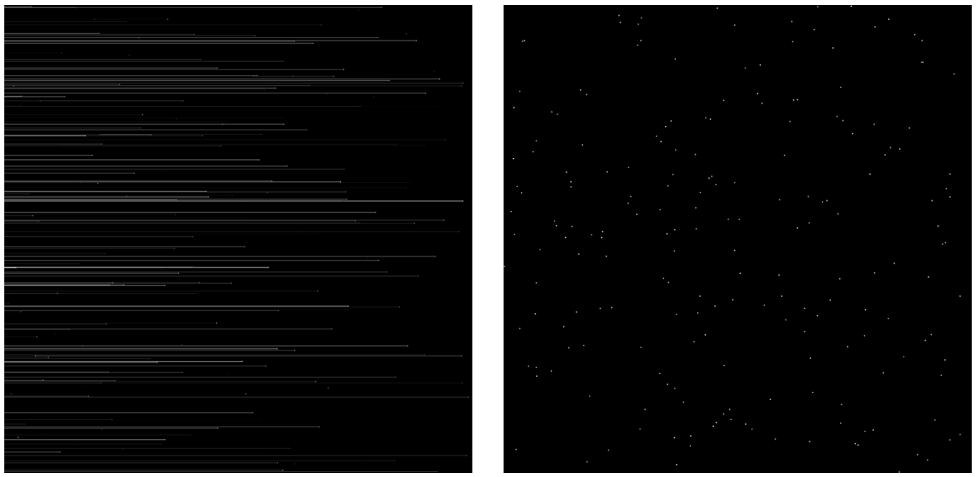

Fig. 13. Sample textures of CCD (left) and CMOS (right) Broken Pixels. Textures represented in black background for visibility reasons. For comparison with a realistic broken pixels occurrence, see Fig. 4 (right).

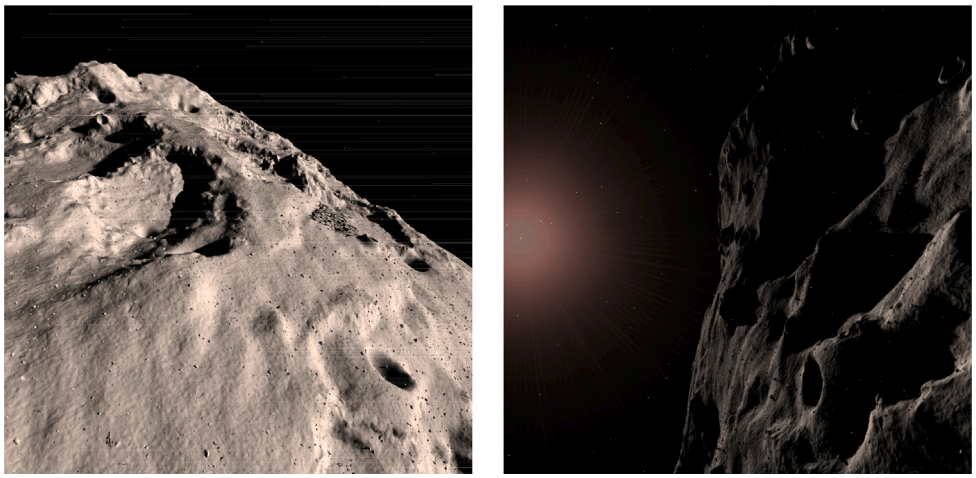

Fig. 14. Sample images with CCD (left) and CMOS (right) Broken Pixels.

- Gaussian Blurriness: the simulator allows Gaussian filtering of images with user-specified kernel size, providing blurriness effects of variable intensity.
- Straylight simulation: the simulator implements realistic rendering of user-defined textures representing solar flares and glares in the simulated images, as described in King [32].

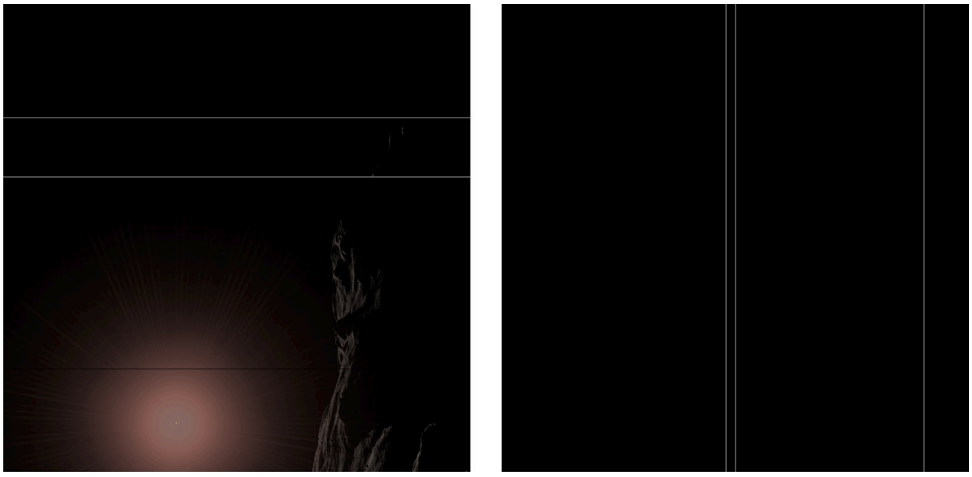

Fig. 15. Sample images with lines of Broken Pixels. For comparison with a realistic Broken Pixels occurrence, see Fig. 4 (left).

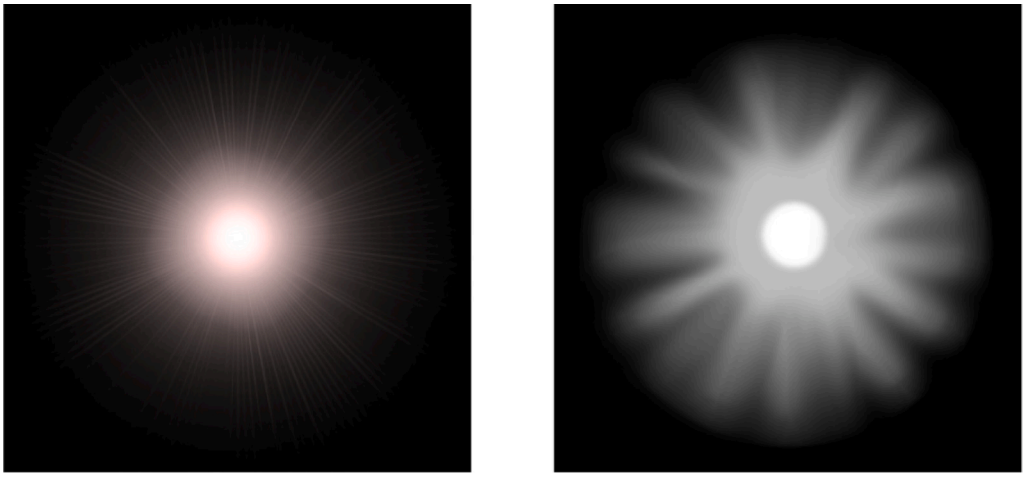

Fig. 16. Injected glares textures.

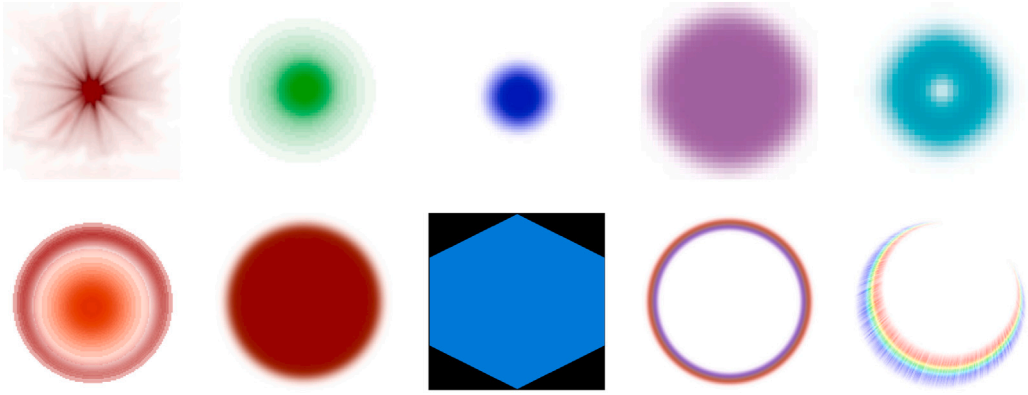

Fig. 17. Injected Flares textures, divided in *Close* (top) and *Far* (bottom).

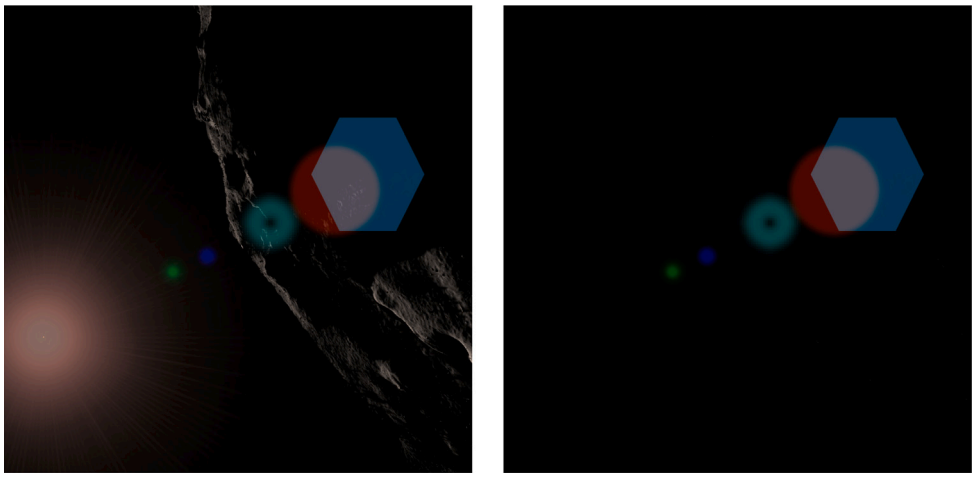

Fig. 18. Sample image affected by straylight (left) and the injected texture (right).

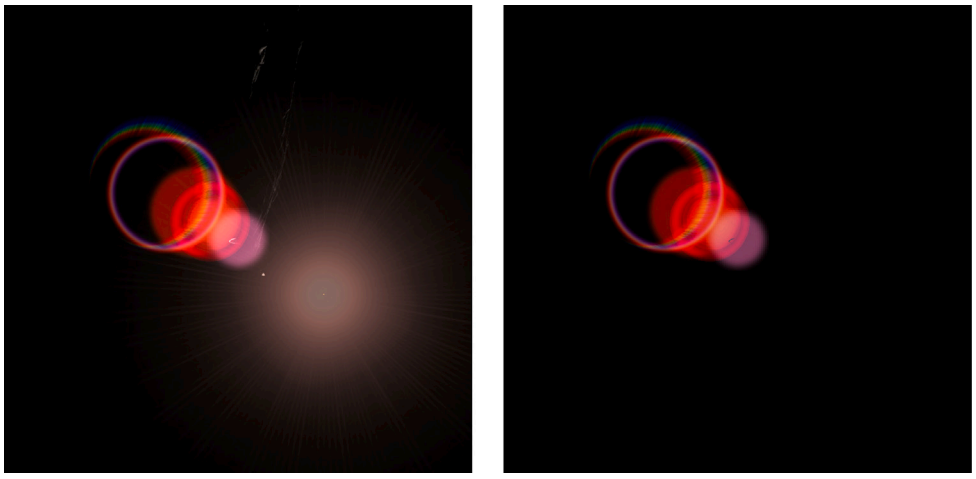

Fig. 19. Sample image affected by straylight (left) and the injected texture (right).

Dust on optics

The modelling of Dust on Optics consists in the modelling of a texture to reproduce the typical artifacts of dust grains depositing on the lens surface. The texture is overlapped to the Intensity channel of the single image and the artifacts in the texture are meant to cause an Intensity drop that resembles the dust deposition effect. Single grains are modelled as Gaussian bivariate distributions, approximating the brightness drop in the area where the grain sticks to the lens. The distribution is generated on a squared grid of $30x30$ pixels and spans all the possible brightness channel values (Table 1). The corners of the grid are removed to make the blending with the background image smoother. The covariance matrix of the distribution and the peak value are randomized to mimic different grain shapes. The amount of rendered grains is set empirically to obtain a non-negligible darkening effect on the image and a statistically significant population of faults. Fig. 11 provides a few examples of dust grain textures that are rendered in the dataset, while Table 3 provides the parameters to generate them. The Covariance values presented in Table 3 were selected empirically to encompass the diverse circular and elliptical morphologies characteristic of dust grains observed in realistic scenarios. To substantiate this approach, the generated textures from our simulation strategy (Fig. 11) demonstrably resemble the artifacts documented during the Hayabusa2 mission (Fig. 1). Nevertheless, the specific Covariance values utilized to achieve these shapes are tailored for the 30×30 pixel grid used for pre-rendering textures and the 8-bit intensity channel depth. Consequently, these values would require adaptation to simulate larger grains or more intricate shapes, as necessitated by experimental requirements. In other words, applying the same covariance values to generate larger textures would yield unrealistic grain morphologies.

Broken pixels

As for Dust on Optics, the simulation of defect pixels makes use of pre-defined randomized textures to inject the faults onto the simulated image. Recalling the discussion about broken pixels at Section 3, these defects can be related to issues of the sensor (i.e. impinging radiation) or the readout electronics. The two categories can occur both in the same image as they are uncorrelated. Instead, faults in CCD and CMOS sensors implies mutually-exclusive realizations of defects pixels, as the architecture of the sensor is fixed.

The simulation of the single broken pixels does not change per sensor type considered, as the effect of a radiation impinging the detector causes a shift in the current output for both (Fig. 2). Therefore, the texture used to simulate defect pixels is the same for CCD and CMOS sensors (Fig. 12).

The number of simulated defects in the pixels depends on the specific radiation environment in which the mission operates. For an

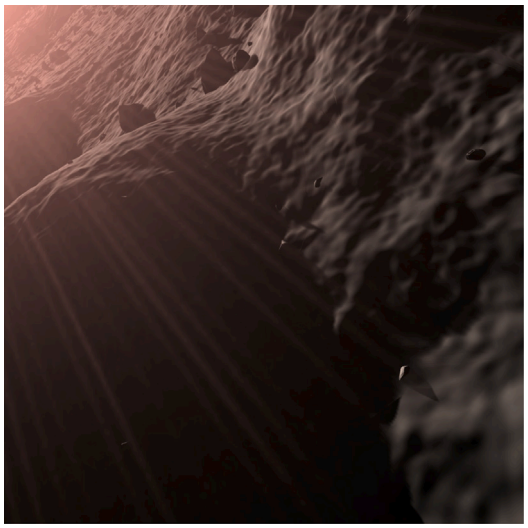

Fig. 20. Sample Image affected by a strong solar glare. For comparison with a realistic image, see Fig. 5 (left).

interplanetary trajectory as the one of the *Astrone KI* mission scenario, Solar Particles and Galactic Cosmic Rays (GCR) will strike the spacecraft undisturbed due to the lack of atmosphere and magnetic field. In addition, the lifetime of *Astrone KI* is not fixed yet but it is a crucial variable to determine the number of defective events that are expected to occur on the camera detector. As a simulation strategy, the number of broken pixels is set empirically to obtain statistically significant samples in the dataset, without making considerations on the influencing variables. This approach is deemed appropriate if the dataset is employed in a classification or segmentation task, as the one targeted by the present work.

In order to simulate the dependency of a defect pixel response from the Exposure Time and ISO (Figs. 2 and 3), the texture at Fig. 12 is injected with a randomized Intensity value (i.e. brightness). Indeed, on the same detector it is possible to find defect pixels with differently shifted outputs, based on the power transmitted by the impinging GCR. Different shifts will result in various brightness levels as they are read-out by the electronics during the image acquisition. The neighbourhood pixels affected by the defect, well visible in Fig. 12, are introduced to keep into account demosaicing effects. The neighbourhood Intensity channel value is computed by linearly interpolating the Intensity of the defect pixel and the adjacent working one on the opposite side.

As already stated in Section 3, broken lines subtended to the respective defect pixels (in CCD detectors) derive their brightness from the respective pixel's offset. In order to simulate this effect, the lines are injected at a randomized level of Intensity between 0 (hot pixel without offset) and 0.4 × Intensity of the generator hot pixel. The direction of the lines in the final image is related to the readout mechanism position with respect to the surface of the detector. This means on one side that lines can have four different directions (top, bottom, right, left), on the other side that the direction is fixed within the same image. In other words, as the electronics is fixed by construction of the camera, all the generated lines will head towards the same direction of readout. In the simulation, a flag in introduced per image to randomize this feature. Figs. 13 and 14 respectively provide an example of injected texture and full rendered image, including faults from both kinds of sensors considered.

Concerning broken lines deriving from faults of the electronics, the number of possible lines is empirically set to mimic the occurrence of such an issue in realistic onboard electronics. This occurrence is related to contingent factors and a robust and expensive design can be useful to reduce it to an acceptable level. The broken lines are simulated with randomized direction and Intensity values (Fig. 15). The direction can be vertical or horizontal, depending on the readout components position with respect to the detector. To simulate different readout settings, a flag is introduced to command the direction of the lines. This flag also takes into account whether CCD or CMOS broken pixels are injected and adapts the injection of the lines accordingly (Section 3). Finally, the Intensity value of the pixels composing the broken lines is randomized to assume values 0 or 255, providing the two cases of dead or saturated pixels.

Hence, the derived taxonomy of faults includes CCD and CMOS broken pixels (Fig. 14), broken lines (Fig. 15 and mixed cases where both faults occur simultaneously.

Finally, Table 4 provides simulation parameters for the Broken Pixels and Lines fault and their respective values.

Straylight

The simulation strategy for straylight adopts the approach described in King [32] of rendering pre-defined textures onto the final image. Compared to the physically-based methods detailed in Keshmirian [46], Hullin et al. [47], Lee and Eisemann [48], this approach provide less-realistic images. However, the focus of this work is on accommodating a broad range of lens geometries rather than on providing sufficient realism. Texture-based simulation provides the advantage of incorporating representative textures from various scenarios and camera setups, achieving extensive data coverage. Furthermore, it supports effective testing of specific scenarios or sensors without defining the sensor's physical model a priori. This is notably advantageous in the targeted FDIR domain, as it enables fast preliminary analyses and trade-offs. Additionally, the dataset can be potentially used for AI models pre-training, ensuring that the latter are later fine-tuned with more accurately modelled data, possibly freezing sensor parameters.

The textures employed for glares and flares injection are presented in Figs. 16 and 17. They are embedded in the CamSim software and can be injected upon user command. Although it has been stated that the random textures injection is the baseline for the present approach, a specific adjustment was made to enhance the realism of the simulation. Based on the graphical results of Hullin et al. [47], Keshmirian [46], Lee and Eisemann [48], it is assumed that the rendered flares follow an increasing-size pattern as the artifacts occur further from the light source. Moreover, closer textures to the light source present *denser* layouts (i.e. orbs), while the further they are, the more diverse shapes they acquire (i.e. rings, polygons). To reproduce this behaviour in the present work without assuming a specific lens layout, the textures are split in two groups, *close* and *far*. The separation between the two groups is always kept along the parameters randomization process described later in this paragraph.

According to King [32], the rendered flares shall lie on the line connecting the centre of the image and the light source, i.e. the Sun. A 2D reference frame is defined along this line, where the light source lies at 0 and the centre of the image lies at 1 (dimensionless units). The flares positions can also assume values outside the (0, 1) interval, considering that for positions $\gg 1$ and $\ll 0$ flares can fall outside the image frame, sacrificing the visibility (i.e. resulting in no noteworthy impact during processing). Additionally, the size and brightness of the flares are expressed in dimensionless units and are essential for regulating the respective characteristics. The values for size and brightness serve as multiplication factors to derive the final size and brightness after rendering, based on the original texture values.

The simulation approach includes as a first step a random selection of which flares to inject among the ones in Fig. 17. Then, a random value of position, radius and brightness is assigned to each of the selected flares, that are finally rendered onto the image. While randomizing the position parameters, it is ensured that textures from the *close* group (top part of Fig. 17) are rendered closer to the light source than the ones coming from the *far* group (bottom part of Fig. 17). Boundaries for the flares parameters are listed in Table 5.

Figs. 18 and 19 provide examples of images affected by straylight, while Fig. 20 shows solar glare injection.

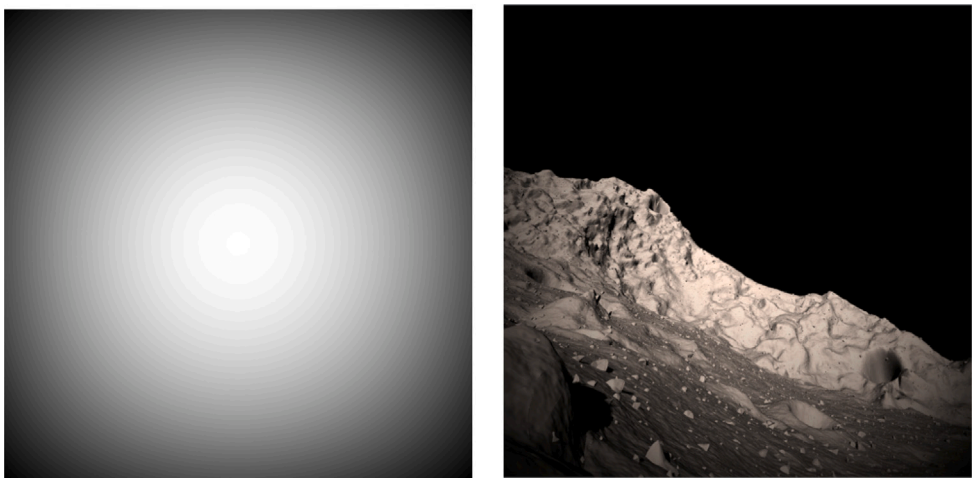

Fig. 21. Sample texture (left) and sample image for Vignetting Effect (right). On-axis illuminance is set to 255. For a comparison with realistic vignetting effect, see Fig. 6 from the NAVCAM of the Opportunity and Perseverance rovers.

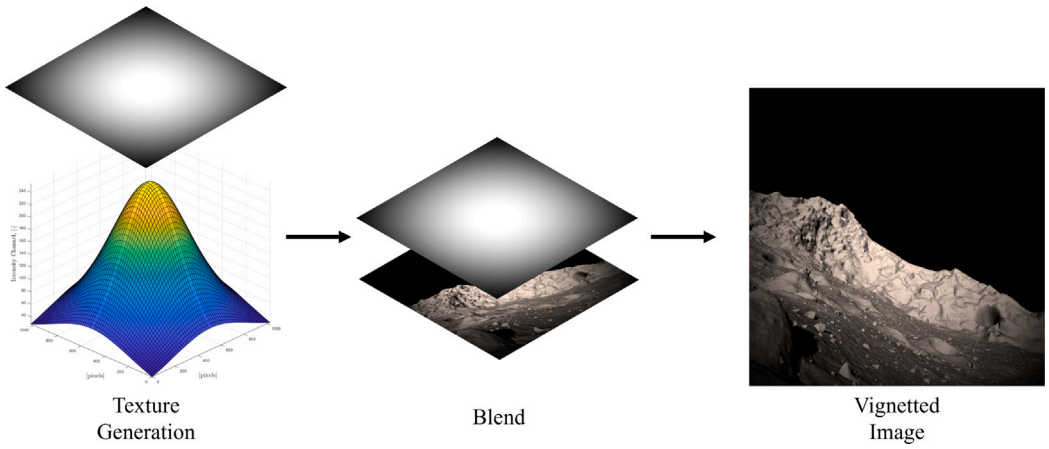

Fig. 22. Simulation pipeline of the vignetting effect. The texture generation step represents the sampled Gaussian curve and the resulting representation on the intensity channel.

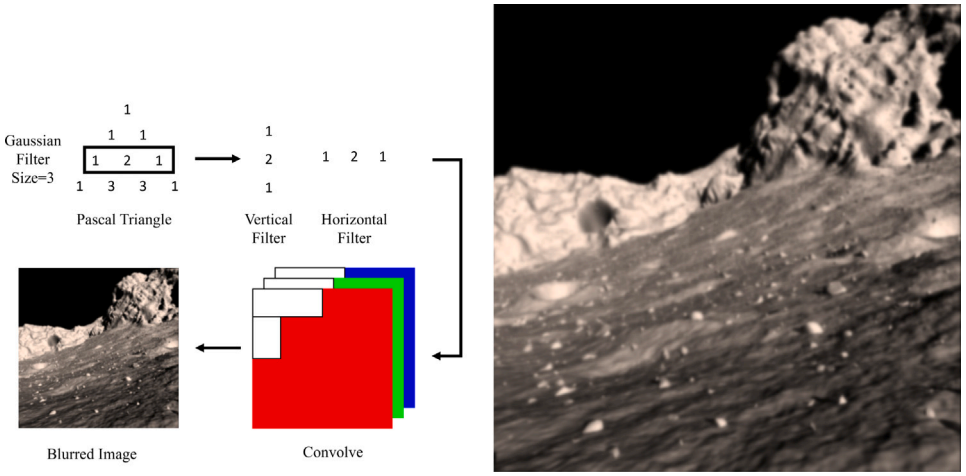

Fig. 23. Optics degradation simulation pipeline (left) and sample image with the injected fault.

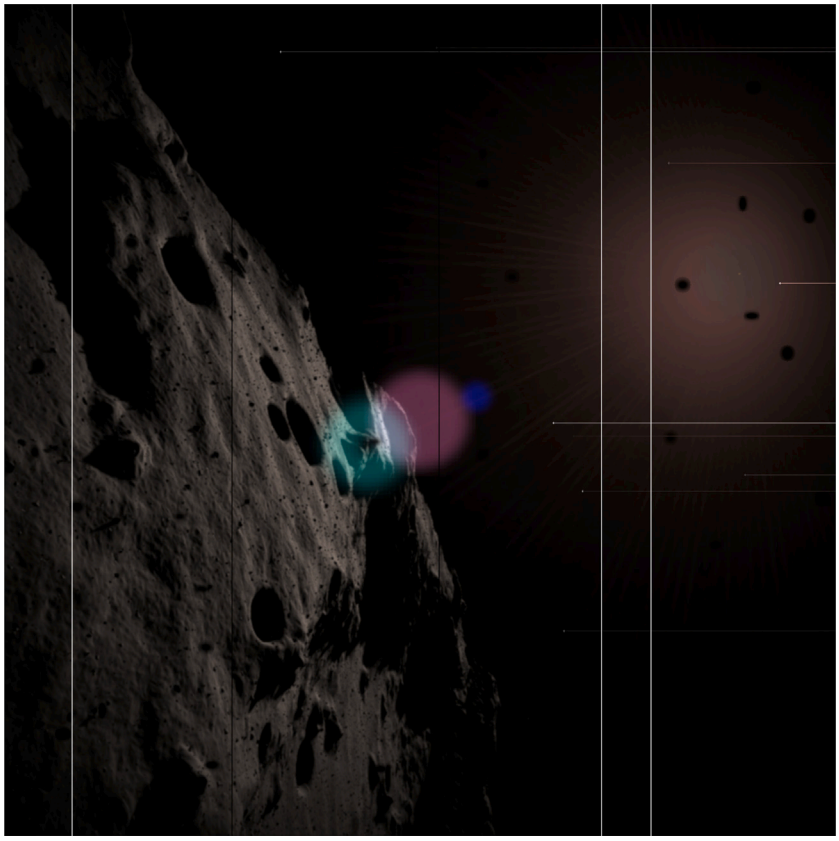

Fig. 24. Sample image full rendering.

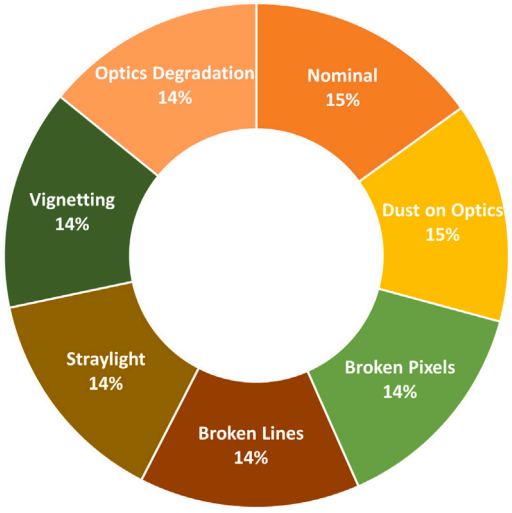

Fig. 25. Summary of the classes distribution across the dataset.

Vignetting

The vignetting effect is simulated in the present work using a monochromatic texture generated to provide the brightness decay effect described in Eq. (1) in Section 3, depicted in Fig. 21 (left). The texture has the size of the acquired image and it assigns to each pixel an Intensity value based on the aforementioned law. The angle in Eq. (1) is computed using the pixel coordinates, while the intensity of the vignetting effect is given by setting the on-axis illuminance (E_0) to an arbitrary value. This value is randomized to provide diverse darkening scenarios. The on-axis illuminance is related to the value of the Intensity channel in the centre of the image and its fixed as a texture parameter. Since the decay law is fixed, the Intensity value in the centre pixel also establishes the intensity of the vignetting effect. The lower boundary for this parameter was empirically set to avoid that extremely subtle vignetting becomes statistically confusable with other sources of intensity variation, such as minor illumination gradients within the scene itself. Indeed, to prevent the model from learning spurious correlations, the minimum vignetting intensity was set to a level where its characteristic radial fall-off constitutes the dominant low-frequency signal in the image, ensuring that the model learns to identify the optical aberration itself rather than confounding artifacts. This aspect was verified by visual inspection, also including corner cases of particularly dark images. Visual inspection also constitutes a more conservative approach with respect to data distribution analysis, making it easier for an algorithm learning from these data to grasp correct features. An example of the texture applied to the sample image and the produced effect is provided in Fig. 21, while Fig. 22 summarizes the simulation pipeline.

Table 6 provides boundaries and explanation of Vignetting parameters employed in the dataset generation.

Optics degradation

We consider distributed blurriness as the only visible optics degradation effect. Established methodology to obtain this effect is convolving the image with a Gaussian filter [53]. The Gaussian Filter is parametrized by a vector of custom-defined size, which commands the number of pixels involved in the convolution operation, and in turn the entity of the blurriness effect. The parameters of the Gaussian filter

are the coefficients of the ith row of the Pascal triangle, as explained in [54]. We consider a Gaussian filter from a shape of three pixels to a shape of seventeen (See Table 7). This choice is dictated by the available range in CamSim. Besides, it is coherent with realistic blurriness occurrence, which can have arbitrary intensity over time.

The simulation pipeline to obtain the blurred effect and a sample image generated with the very same pipeline are depicted in Fig. 23

4.5. Labelling

The labelling of a dataset is strongly related to the targeted task while training and testing the AI on the dataset itself. In this work, supervised learning is assumed, but no specific task. Examples of tasks suitable to VBN comprise but are not limited to direct image classification and segmentation. Both of them aim at the recognition of specific artifacts of the observed terrain to be used for e.g. pose estimation or trajectory planning in the downstream navigation algorithm. The difference resides in the fact that classification assigns a class to a specific image, while segmentation marks groups of pixels belonging to the same class within one image. Examples of datasets for classification in the frame of VBN include Zhao et al. [16] and Lu and Wagstaff [18], which provide images of specific Mars landmarks such as rocks, boulders, Sun, sky, rover parts, etc. The size of the images in these datasets shall be sufficiently small to include one specific landmark in each image, avoiding cases of equivocal classification between two different classes.

In this work, binary segmentation masks are generated per fault, meaning that each image presents a number of labelling masks equal to the number of fault classes in the dataset. The masks can be employed to perform both classification or segmentation, including binary or multi-class classification and binary or multi-class segmentation. Also, it is possible to address the faults incrementally, according to the needs, exploiting the separation of the masks. A binary mask labelling all the faults in the same image is also included, in case the targeted task is segmenting out failed areas instead than recognizing the specific fault. The labelling strategy is straightforward for all the fault cases, except for straylight. Indeed, for those fault cases characterized by a pre-computed texture (Dust on Optics, Broken Pixels and Lines, Vignetting) or an index commanding the injection (Optics Degradation), both the texture and the index can be used as a label themselves. For the straylight, the simulation infrastructure does not output the texture of the injected flares, thus it needs *a posteriori* retrieval. To do this, a post-processing step is carried out after the main one, including only the images facing the Sun (Table 2), to obtain the same images without rendered flares. Subtracting the new images from the ones obtained in the nominal generation procedure allows to isolate the flares from the rest of the image and obtain the desired masks.

Fig. 24 shows an example of a full rendered image. The parameters of the injected faults for this picture are provided in Table 8. The image counterpart with no rendered straylight and the rest of the masks are provided in Appendix B.

4.6. Dataset specifications

A recap of the principal dataset features is presented in this section: Fig. 25 depicts the distribution of faulty classes across the dataset, while Table 9 provides numerical values for specific data properties.

5. Conclusion and future work

In this study, a novel methodology for the simulation of faults occurring in typical Vision-based Navigation systems is presented. The research focuses on common issues that can be experienced in the scenario of interplanetary exploration such as the *Astrone KI* one, providing a comprehensive analysis of the faults occurrence and effects in image data. Nevertheless, generalization is expected to any mission employing visual sensors with diverse needs.

A novel method is proposed to simulate the identified faults, employing cross-domain studies and methodologies tailored to the specific space application. The study proposes a level of detail of the simulation that is deemed sufficient for the data to be representative of the relative fault scenario, but it could easily be expanded in the future according to the needs.

Finally, a dataset is derived simulating together 5000 images with injected fault cases. The dataset simulates the environmental condition of the comet 67P/Churyumov-Gerasimenko, capturing a wide range of scenarios and illumination levels, marking a valuable resource for developing and testing AI algorithms in VBN systems. This dataset sets a significant step in literature with its unique contribution, and aims at triggering the scientific research towards the development of reliable and safe autonomous system for spacecraft navigation.

Looking ahead, the natural follow-up of the present work is the employment of the dataset in a practical use-case of VBN where it can be used for training and testing an AI to perform anomaly detection. Such an idea can be logically applied to the *Astrone KI* system, where an AI-based anomaly detection step would enhance the robustness of the downstream VBN pipeline, allowing it to work safely without fearing failed samples that can cause issues in the processing. Moreover, the authors encourage wide use of the dataset across various mission types, with the expectation that it will generalize effectively. To this purpose, Table 10 presents an overview of different space mission classes, suggesting the applicability of the faults analysed in the present work.

CRediT authorship contribution statement

Riccardo Gallon: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Fabian Schiemenz:** Writing – review & editing, Supervision. **Alessandra Menicucci:** Writing – review & editing, Supervision. **Eberhard Gill:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The results presented in this paper have been achieved by the project Astrone - Increasing the Mobility of Small Body Probes, which has received funding from the German Federal Ministry for Economic Affairs and Energy (BMWi) under funding number “50 RA 2130A”. The consortium consists of Airbus Defence and Space GmbH, Astos Solutions GmbH, Institute of Automation (Technische Universität Dresden) and Institute of Flight Mechanics and Controls (Universität Stuttgart). Responsibility of the publication contents is with the publishing author.

Appendix A. Reference frames

Table A.11 details the reference frames employed in the present study (Section 4).

Appendix B. Sample masks

Based on Figs. 24, B.26 present its counterpart without straylight. The retrieved straylight mask is shown in Fig. B.27. B.28 shows the Dust on Optics mask. Figs. B.29 and B.30 provide the masks associated to Broken Pixels and Broken Lines. Finally, Fig. B.31 aggregates all masks in a unified one.

Table A.11
Reference frames summary table.

Reference frame	Origin	Axes
International Celestial, $\mathcal{J}$	Solar System's Barycentre	***x-axis:*** The intersection of equatorial and ecliptic planes (vernal equinox) at 12:00 Terrestrial Time on 1 January 2000, 12:00:00, ***z-axis:*** Normal to the equator at 12:00 Terrestrial Time on 1 January 2000, 12:00:00 in the direction of the north pole, ***y-axis:*** Completes the right hand rule.
Target-Body Centred Target-Body Fixed, $\mathcal{T}$	CoM of the target body	***x-axis:*** Towards the axis with the least MoI, ***y-axis:*** Completes the right hand rule, ***z-axis:*** Towards the axis with the highest MoI.
Local-Level, $\mathcal{L}$	User-defined on the target surface	***z-axis:*** Aligned with the effective gravity vector, ***x-axis:*** Points in north direction and is perpendicular to the z-axis, ***y-axis:*** Completes the right-hand system.
Body-Fixed, $\mathcal{B}$	CoM of the spacecraft	***z-axis:*** Downwards directed, perpendicular to the horizontal bottom panel, ***y-axis:*** Perpendicular to the z-axis in the direction of one solar panel, ***x-axis:*** Completes the right hand rule.
Camera-Fixed, $\mathcal{C}$	CoM of the spacecraft	***z-axis:*** Same as the Body Fixed frame, ***y-axis:*** Same as the Body Fixed frame, ***x-axis:*** Same as the Body Fixed frame.

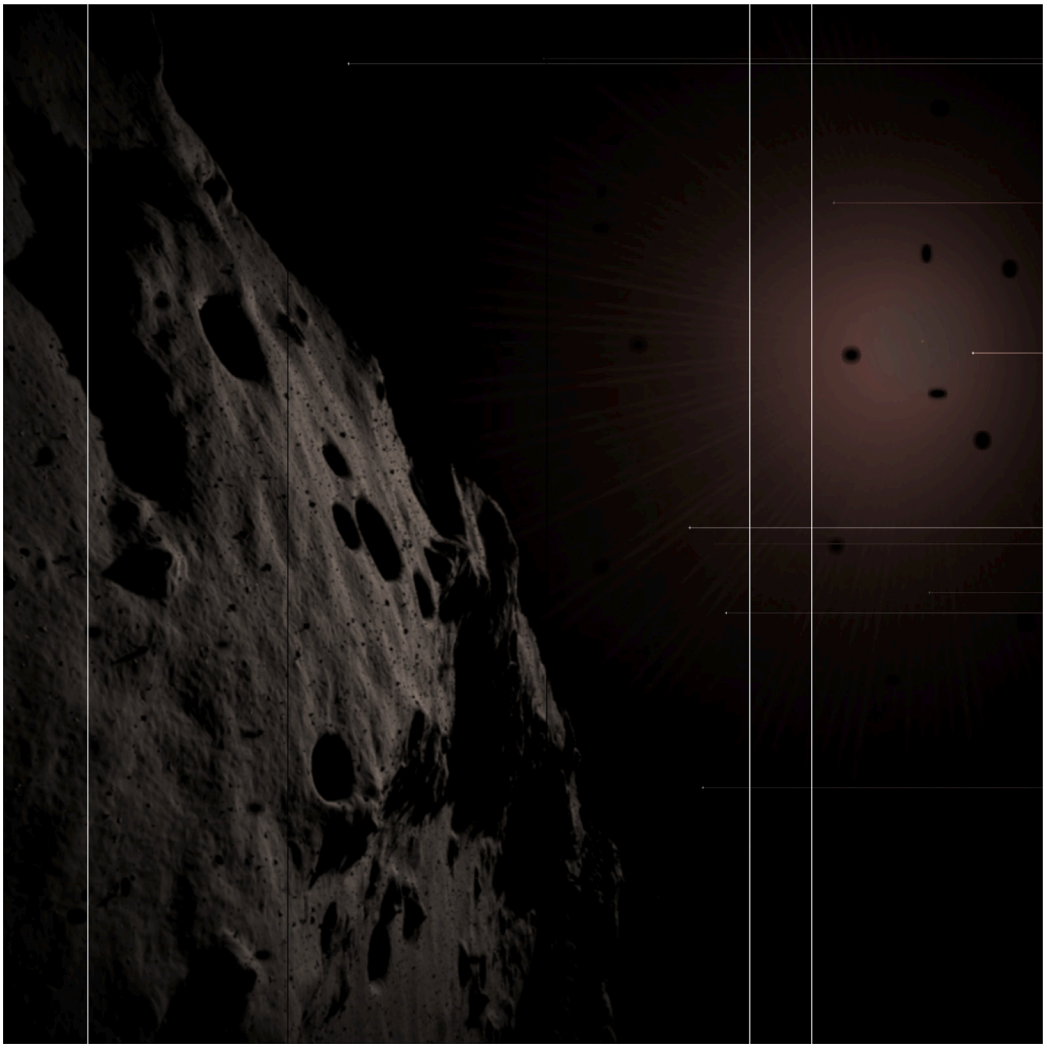

Fig. B.26. Fig. 24 without rendered straylight.

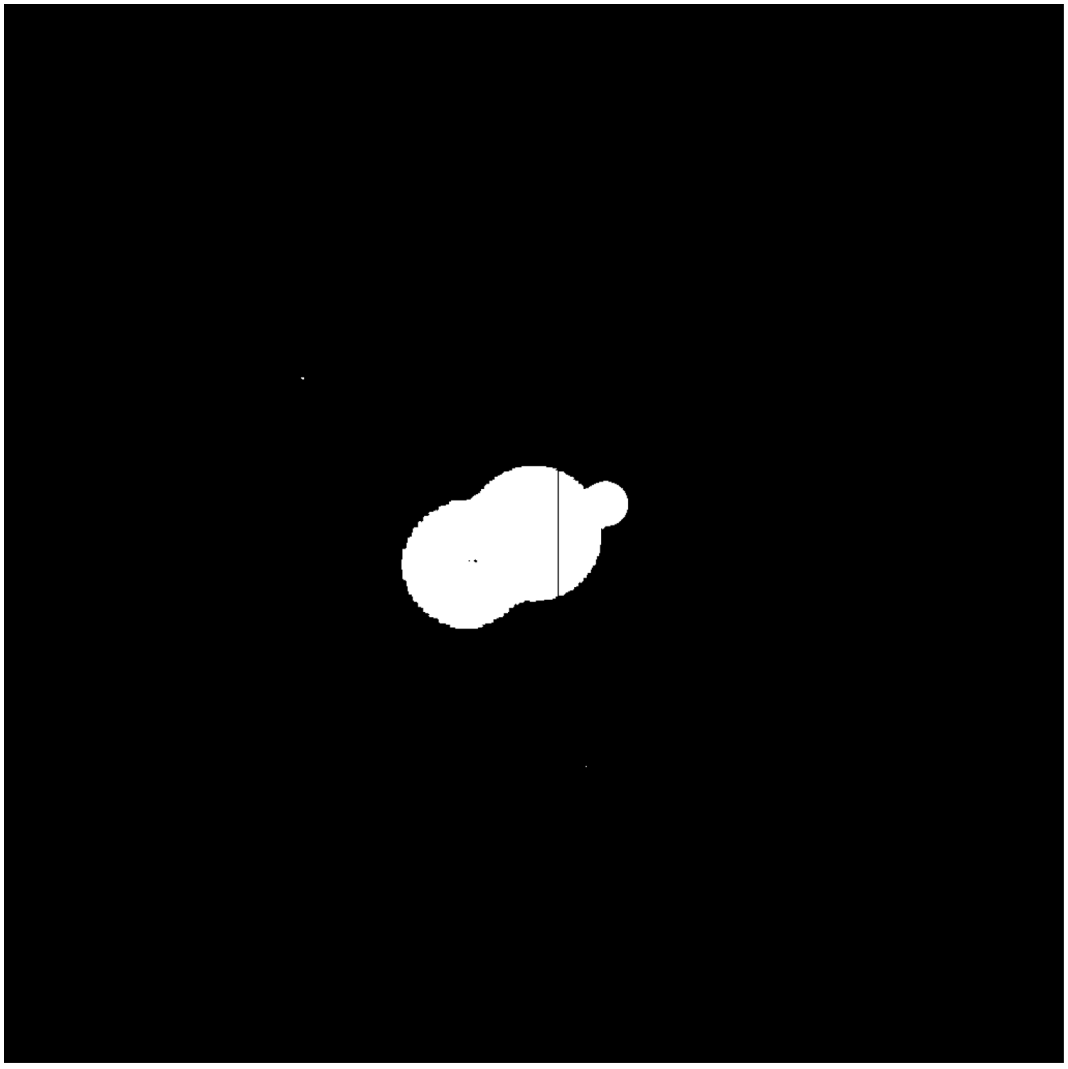

Fig. B.27. Straylight mask. Note that the black line intersecting the flare textures appears black because it is a Broken Line, so it is not part of the Straylight label.

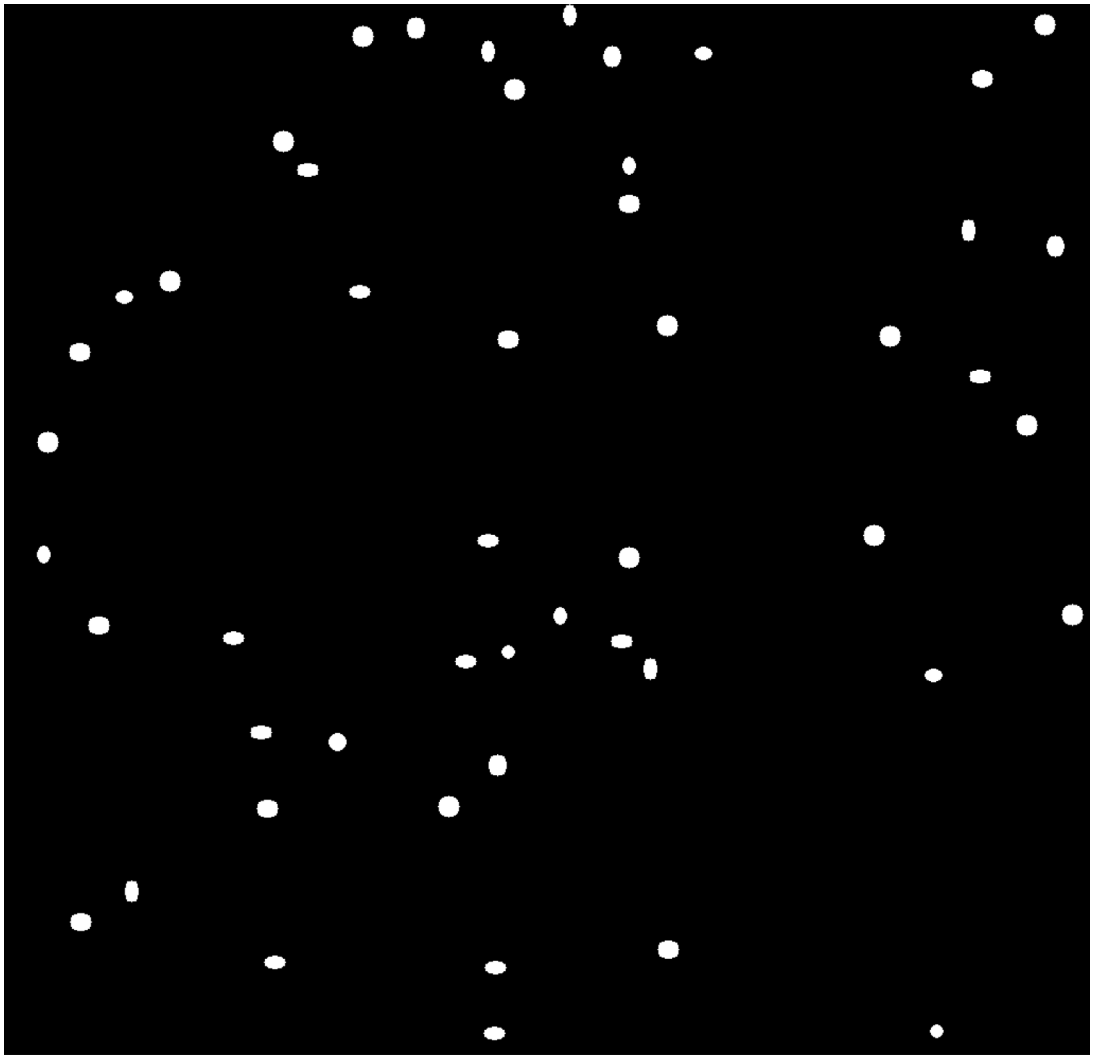

Fig. B.28. Dust on optics mask.

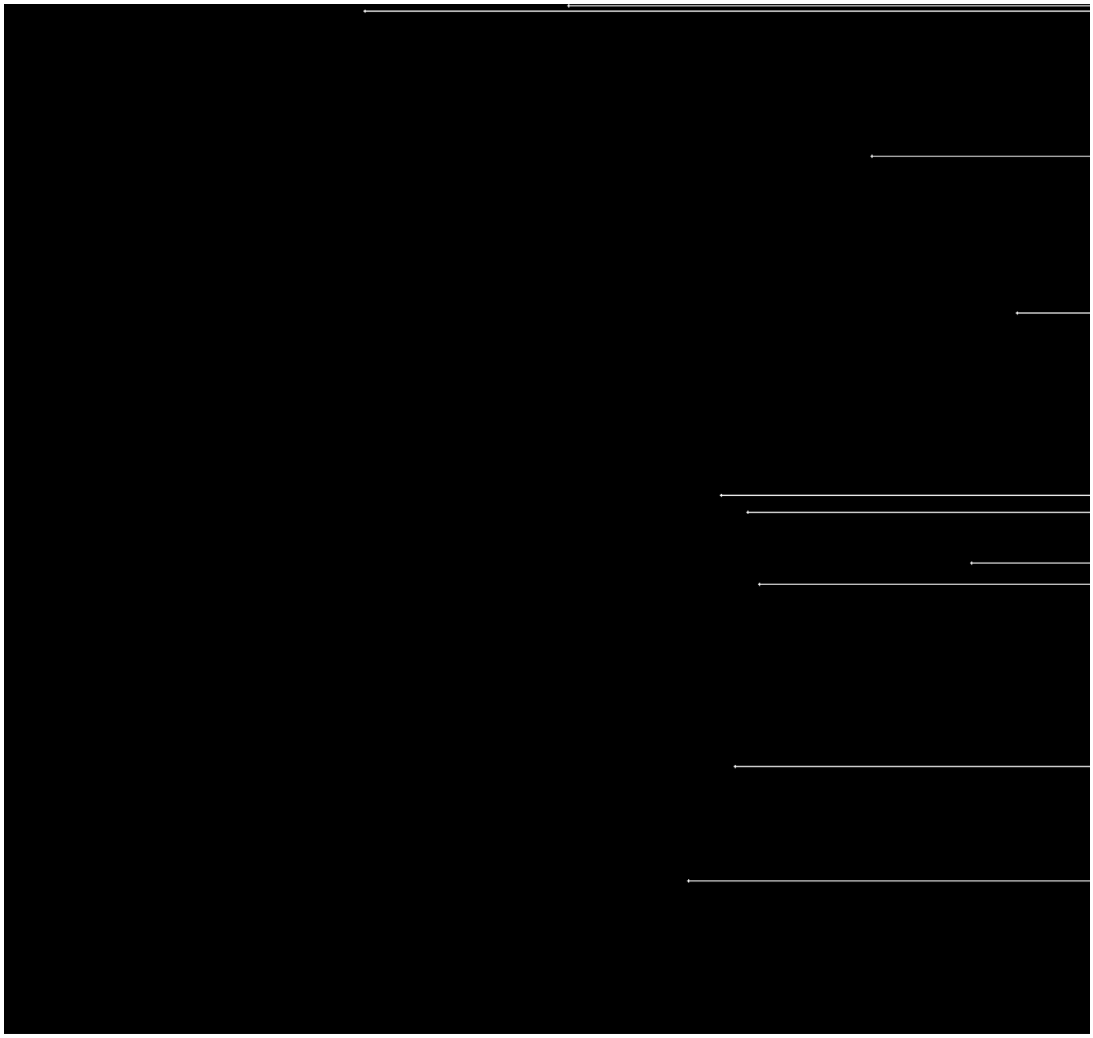

Fig. B.29. Broken pixels mask.

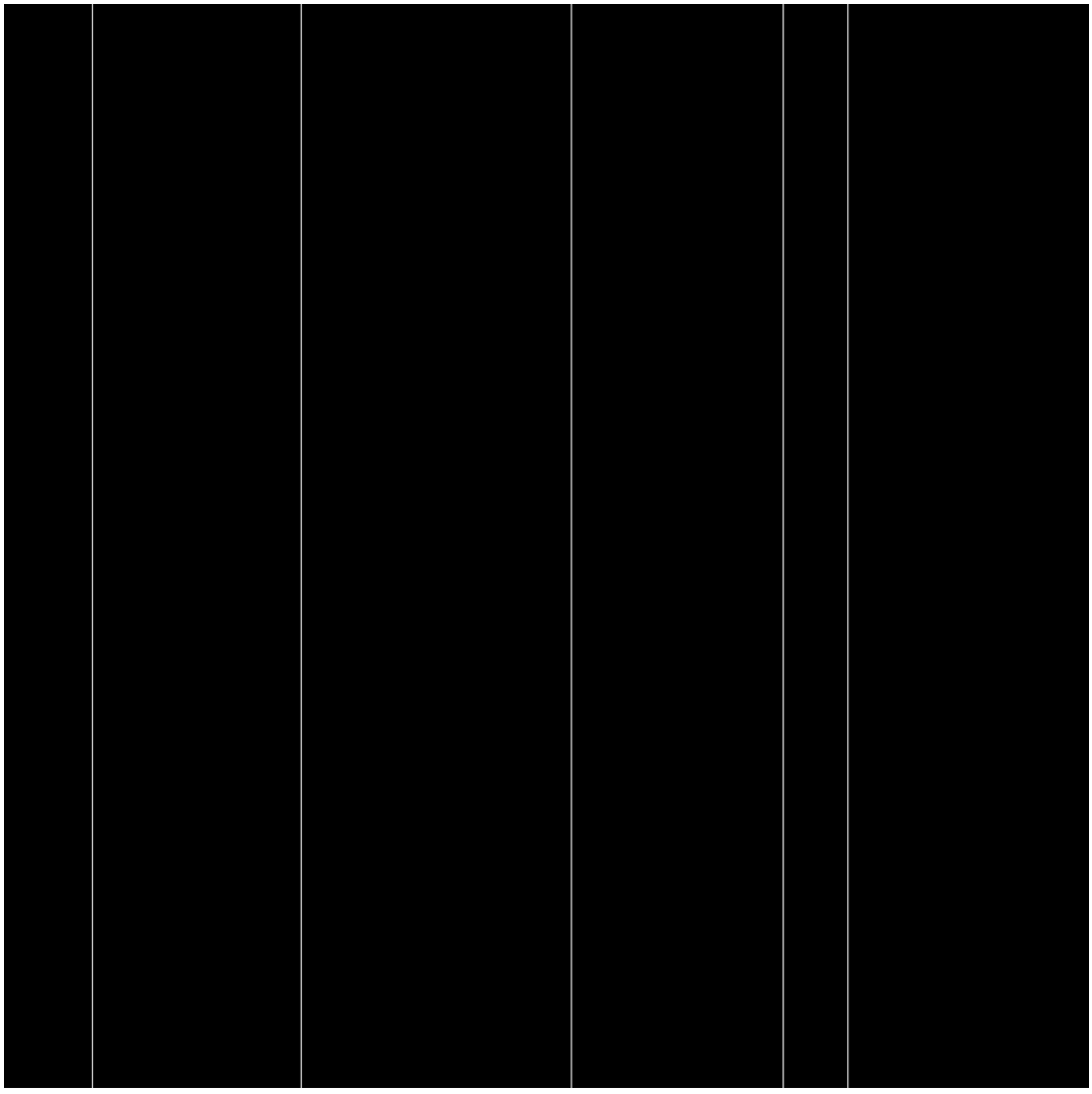

Fig. B.30. Broken lines mask.

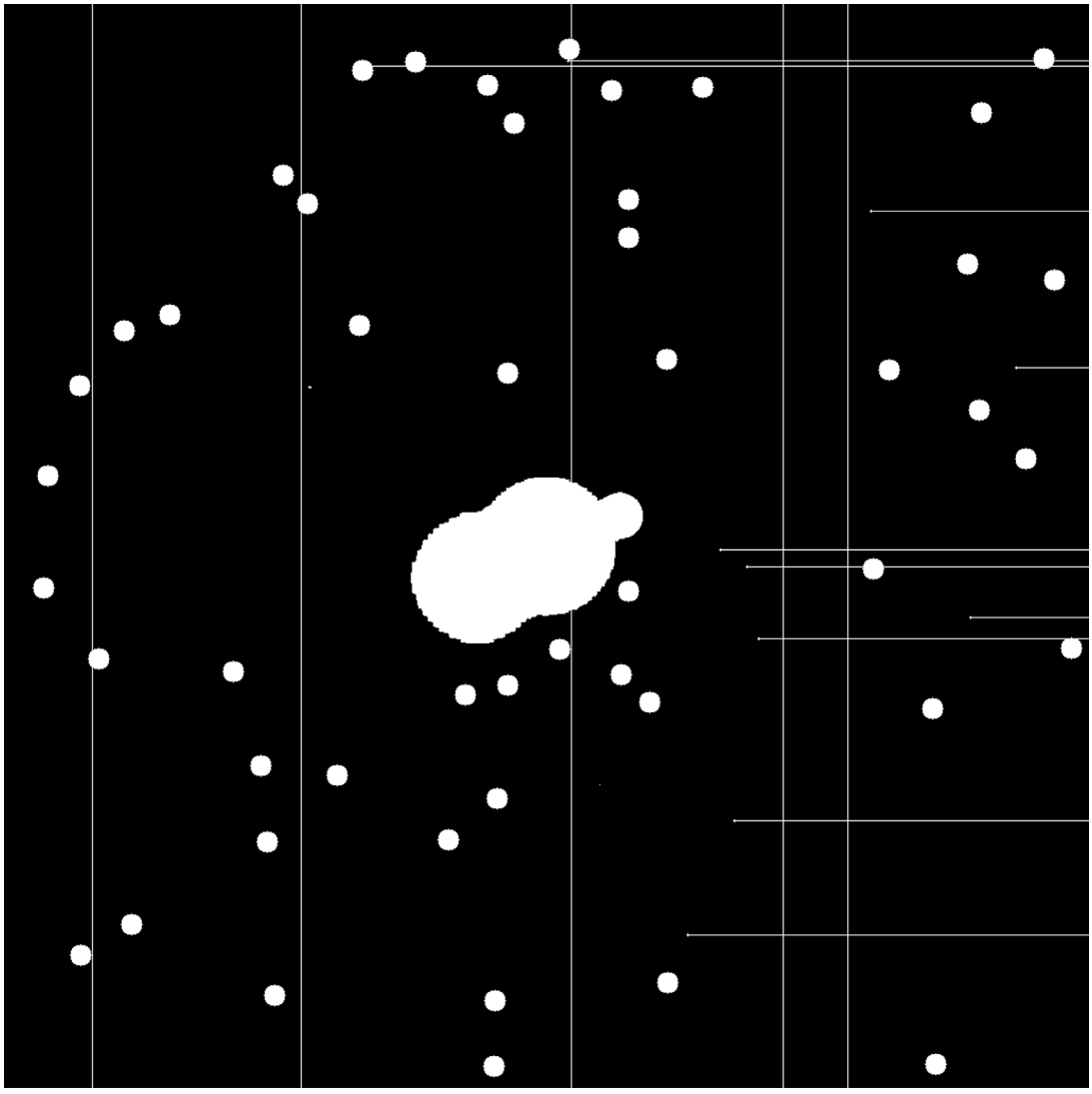

Fig. B.31. Sample mask full faults.

References

[1] M. Martin, E. Caroselli, J. Olucak, S. Busi, W. Fichter, B. Liu, A. Liesch, P. Suwinski, V. Chernykh, K. Janschek, Pioneering the small bodies frontiers: The key enabling technologies for autonomous precise mobilitys, in: 12 th International Conference on Guidance, Navigation & Control Systems, GNC, 2023.

[2] J. Olucak, S. Zähringer, F. Schimpf, W. Fichter, Sensor based motion planner for small solar system body exploration, in: AIAA SCITECH 2023 Forum, 2023.

[3] A. Liesch, P. Suwinski, B. Liu, V. Chernykh, K. Janschek, Ai-based lidar/camera data fusion to enable high-resolution 3d surface reconstruction for autonomous asteroid exploration mission, in: AAS/AIAA Astrodynamics Specialist Conference, 2023.

[4] R. Gallon, F. Schiemenz, A. Krstova, A. Menicucci, E. Gill, Machine learning-based vs deep learning-based anomaly detection in multivariate time series for spacecraft attitude sensors, 2024a, arXiv preprint arXiv:2409.17841.

[5] R. Gallon, F. Schiemenz, A. Menicucci, E. Gill, Convolutional neural network design and evaluation for real-time multivariate time series fault detection in spacecraft attitude sensors, 2024b, arXiv preprint arXiv:2410.09126.

[6] E. C. for Space Standardization, Telemetry and telecommand packet utilization, in: Standard ECSS-E-ST-70-41C, European Space Agency, 2016.

[7] ECSS-Q-ST-30-02C, in: Failure Modes, Effects (and Criticality) Analysis (FMEA/FMECA), Technical Report ECSS-Q-ST-30-02C, European Space Agency, 2009.

[8] M. Maimone, A. Johnson, Y. Cheng, R. Willson, L. Matthies, Autonomous navigation results from the mars exploration rover (mer) mission, in: Experimental Robotics IX: The 9th International Symposium on Experimental Robotics, Springer, 2006, pp. 3–13.

[9] S. Sharma, T.H. Park, S. D'Amico, Spacecraft pose estimation dataset (speed), 2019, https://purl.stanford.edu/dz692fn7184, Stanford Digital Repository.

[10] T.H. Park, M. Märtens, G. Lecuyer, D. Izzo, S. D'Amico, Speed+: Next-generation dataset for spacecraft pose estimation across domain gap, in: 2022 IEEE Aerospace Conference, AERO, IEEE, 2022, pp. 1–15.

[11] J. Lebreton, I. Ahrns, R. Brochard, C. Haskamp, M.L. Goff, N. Menga, N. Ollagnier, R. Regele, F. Capolupo, M. Casasco, Training datasets generation for machine learning: Application to vision based navigation, in: Proceedings of the ESA SPAICE Conference 2024, 2024.

[12] B. Bos, M. Ravine, M. Caplinger, J. Schaffner, J. Ladewig, R. Olds, C. Norman, D. Huish, M. Hughes, S. Anderson, et al., Touch and go camera system (tagcams) for the osiris-rex asteroid sample return mission, Space Sci. Rev. 214 (2018) 1–23.

[13] M. Yamada, T. Kouyama, K. Yumoto, E. Tatsumi, N. Takaki, Y. Yokota, T. Morota, N. Sakatani, M. Hayakawa, M. Matsuoka, et al., Inflight calibration of the optical navigation camera for the extended mission phase of hayabusa2, Earth Planets Space 75 (2023) 36.

[14] R. Brochard, J. Lebreton, C. Robin, K. Kanani, G. Jonniaux, A. Masson, N. Despré, A. Berjaoui, Scientific image rendering for space scenes with the surrender software, in: The 69th International Astronautical Congress, IAC, 2018.

[15] H. Krüger, S. Theil, Tron-hardware-in-the-loop test facility for lunar descent and landing optical navigation, IFAC Proc. Vol. 43 (2010) 265–270.

[16] B. Zhao, S. Cole, S. Lu, Mer opportunity and spirit rovers pancam images labeled data set, 2020, https://zenodo.org/records/4302760.

[17] K. Wagstaff, Y. Lu, A. Stanboli, K. Grimes, T. Gowda, J. Padams, Deep mars: Cnn classification of mars imagery for the pds imaging atlas, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[18] S. Lu, K.L. Wagstaff, Msl curiosity rover images with science and engineering classes, 2020, https://zenodo.org/records/4033453.

[19] G. Doran, E. Dunkel, S. Lu, K. Wagstaff, Mars orbital image (hirise) labeled data set, 2020, https://zenodo.org/records/4002935.

[20] X. Xiao, M. Yao, H. Liu, J. Wang, L. Zhang, Y. Fu, A kernel-based multi-featured rock modeling and detection framework for a mars rover, IEEE Trans. Neural Netw. Learn. Syst. 34 (2021) 3335–3344.

[21] H. Liu, M. Yao, X. Xiao, Y. Xiong, Rockformer: A u-shaped transformer network for martian rock segmentation, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–16.

[22] R.M. Swan, D. Atha, H.A. Leopold, M. Gildner, S. Oij, C. Chiu, M. Ono, Ai4mars: A dataset for terrain-aware autonomous driving on mars, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1982–1991.

[23] Y. Xiong, X. Xiao, M. Yao, H. Cui, Y. Fu, Light4mars: A lightweight transformer model for semantic segmentation on unstructured environment like mars, ISPRS J. Photogramm. Remote Sens. 214 (2024) 167–178.

[24] R. West, B. Knowles, E. Birath, S. Charnoz, D. Di Nino, M. Hedman, P. Helfenstein, A. McEwen, J. Perry, C. Porco, et al., In-flight calibration of the cassini imaging science sub-system cameras, Planet. Space Sci. 58 (2010) 1475–1488.

[25] B. Bos, D. Nelson, J. Pelgrift, A. Liounis, D. Doelling, C. Norman, R. Olds, C. May, R. Witherspoon, E. Church, et al., In-flight calibration and performance of the osiris-rex touch and go camera system (tagcams), Space Sci. Rev. 216 (2020) 1–52.

[26] C.C. Porco, R.A. West, S. Squyres, A. McEwen, P. Thomas, C.D. Murray, A. Delgenio, A.P. Ingersoll, T.V. Johnson, G. Neukum, et al., Cassini imaging science: Instrument characteristics and anticipated scientific investigations at saturn, Cassini-Huygens Mission.: Orbiter Remote. Sens. Investig. (2004) 363–497.

[27] G.H. Chapman, J. Leung, A. Namburete, I. Koren, Z. Koren, Predicting pixel defect rates based on image sensor parameters, in: 2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, IEEE, 2011, pp. 408–416.

[28] G.H. Chapman, R. Thomas, Z. Koren, I. Koren, Empirical formula for rates of hot pixel defects based on pixel size, sensor area, and iso, in: Sensors, Cameras, and Systems for Industrial and Scientific Applications XIV, vol. 8659, SPIE, 2013, pp. 119–129.

[29] G.H. Chapman, R. Thomas, K.J. Meneses, P. Purbakht, I. Koren, Z. Koren, Exploring hot pixel characteristics for 7 to 1.3 micron pixels, Electron. Imaging 30 (2018) 1–6.

[30] G.H. Chapman, K.J.C.S. Meneses, I. Koren, Z. Koren, Image degradation due to interacting adjacent hot pixels, in: 2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT, IEEE, 2022, pp. 1–6.

[31] G.H. Chapman, K.J.C.S. Menes, L.-Y. Wu, I. Koren, Z. Koren, Image degradation in time due to interacting hot pixels, in: 2023 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT, IEEE, 2023, pp. 1–6.

[32] Y. King, Game Programming Gems 2, Charles River Media, 2001.

[33] C. Maughan, Texture masking for faster lens flare, Game Program. Gems 2 (2001) 474–480.

[34] M. Kilgard, Fast opengl-rendering of lens flares, 2000, https://web.archive.org/web/20241105162020/https://www.opengl.org/archives/resources/features/KilgardTechniques/LensFlare//.

[35] Y. Wu, Q. He, T. Xue, R. Garg, J. Chen, A. Veeraraghavan, J.T. Barron, How to train neural networks for flare removal, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2239–2247.

[36] Y. Dai, C. Li, S. Zhou, R. Feng, C.C. Loy, Flare7k: A phenomenological nighttime flare removal dataset, in: Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 3926–3937.

[37] Y. Dai, C. Li, S. Zhou, R. Feng, Y. Luo, C.C. Loy, Flare7k++: Mixing synthetic and real datasets for nighttime flare removal and beyond, 2023, arXiv preprint arXiv:2306.04236.

[38] National Aeronautics and Space Administration, Nasa planetary data system, 2018, https://pds-imaging.jpl.nasa.gov/volumes/mars2020.html, (Accessed 23 September 2025).

[39] National Aeronautics and Space Administration, Nasa planetary data system, 2018, https://pds-imaging.jpl.nasa.gov/volumes/mer.html, (Accessed 23 September 2025).

[40] I. Jun, H. Garrett, W. Kim, Y. Zheng, S.F. Fung, C. Corti, N. Ganushkina, J. Guo, A review on radiation environment pathways to impacts: Radiation effects, relevant empirical environment models, and future needs, Adv. Space Res. (2024) All Open Access, Hybrid Gold Open Access.

[41] J. Leung, J. Dudas, G.H. Chapman, I. Koren, Z. Koren, Quantitative analysis of in-field defects in image sensor arrays, in: 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, DFT 2007, IEEE, 2007, pp. 526–534.

[42] J. Leung, G.H. Chapman, Z. Koren, I. Koren, Statistical identification and analysis of defect development in digital imagers, in: Digital Photography V, vol. 7250, SPIE, 2009, pp. 272–283.

[43] N. Waltham, Ccd and cmos sensors, in: Observing Photons in Space: A Guide to Experimental Space Astronomy, 2013, pp. 423–442.

[44] P. Longere, X. Zhang, P.B. Delahunt, D.H. Brainard, Perceptual assessment of demosaicing algorithm performance, Proc. IEEE 90 (2002) 123–132.

[45] R. Jaumann, N. Schmitz, A. Koncz, H. Michaelis, S. Schroeder, S. Mottola, F. Trauthan, H. Hoffmann, T. Roatsch, D. Jobs, et al., The camera of the mascot asteroid lander on board hayabusa 2, Space Sci. Rev. 208 (2017) 375–400.

[46] A. Keshmirian, A Physically-Based Approach for Lens Flare Simulation, University of California, San Diego, 2008.

[47] M. Hullin, E. Eisemann, H. Seidel, S. Lee, Physically-based real-time lens flare rendering, ACM trans. graph. 30 (2011).

[48] S. Lee, E. Eisemann, Practical real-time lens-flare rendering, in: Computer Graphics Forum, vol. 32, Wiley Online Library, 2013, pp. 1–6.

[49] D.A. Kerr, Derivation of the cosine fourth law for falloff of illuminance across a camera image, 2007, https://web.archive.org/web/20241105161618/http://dougkerr.net/Pumpkin/articles/Cosine_Fourth_Falloff.pdf.

[50] D. Garoli, L.V. Rodriguez De Marcos, J.I. Larruquert, A.J. Corso, R. Proietti Zaccaria, M.G. Pelizzo, Mirrors for space telescopes: Degradation issues, Appl. Sci. 10 (2020) 7538.

[51] F. Preusker, F. Scholten, K.-D. Matz, T. Roatsch, S. Hviid, S. Mottola, J. Knollenberg, E. Kührt, M. Pajola, N. Oklay, et al., The global meter-level shape model of comet 67p/Churyumov-Gerasimenko, Astron. Astrophys. 607 (2017) L1.
[52] J. Eggert, S. Weikert, I. Kossev, Imaging sensor emulation and dynamics simulation for pil/hil, in: Simulation and EGSE for Space Programmes, 2015, Unpublished Proceedings.
[53] R.C. Gonzalez, Digital Image Processing, Pearson, 2022.
[54] M. Aubury, W. Luk, Binomial filters, J. VLSI Signal Process. Syst. Signal Image Video Technol. 12 (1996) 35–50.