Modeling Faults in Spacecraft Imagery for Pose Estimation

Final Report – Spring 2024–25

Imane Jennane
Student ID: 310900
Supervisor: Andrew Lawrence Price
Professor: Mathieu Salzmann
Computer Vision Lab

June 2025

Abstract

This report presents a comprehensive framework for simulating and evaluating faults in spacecraft imagery, with the goal of improving the robustness of computer vision systems used in spacecraft applications. Visual data collected in orbit are often degraded by physical and electronic artifacts such as blooming, lens flare, dark current noise, and readout glitches, which can significantly impair the performance of downstream tasks like pose estimation or object tracking. To address this challenge, we implement a physically informed fault injection pipeline that models these effects using parameterized, image-level transformations grounded in their real-world physical causes. Each fault type is simulated through a dedicated module, allowing for independent control, combination, and scaling of visual degradations.

The simulation framework is designed to integrate seamlessly with synthetic datasets. To validate the realism of the generated artifacts, we perform a fidelity evaluation using the SPEED+ dataset as reference, focusing on the blooming effect due to the availability of real annotated examples. Quantitative evaluation is conducted using the Feature Quality Index (FQI) and histogram correlation, both of which capture the structural and photometric similarity between synthetic and real faulty images.

1 Introduction

Spacecraft imaging is prone to a exposed of physical and transmission-related faults that can degrade visual data. Simulating these artifacts is essential for developing and validating robust vision algorithms. This report presents a systematic framework for identifying, modeling, and evaluating such faults.

2 Literature Overview

The simulation and modeling of faults in spacecraft imagery is a growing area of interest, driven by the increasing reliance on vision-based systems for autonomous navigation, rendezvous, and inspection tasks in space. As spacecraft operate in harsh and unpredictable environments, characterized by extreme illumination conditions, radiation exposure, and sensor limitations, the need for robust image processing pipelines is more pressing than ever. Recent literature addresses this challenge through improvements in synthetic dataset generation, the modeling of image degradations, and the design of fault-tolerant pose estimation algorithms.

Bechini et al. [1, 2] developed a physically-based ray-tracing framework using POV-Ray and PBRT to produce synthetic images of spacecraft under realistic lighting and orbital conditions. Their SPEED and SPEED+ datasets simulate effects such as shadows, glints, and surface reflectance with high fidelity. These datasets have been validated through both qualitative inspection and quantitative metrics, such as histogram correlation and descriptor-based feature matching. Their work emphasizes the need for accurate optical modeling in synthetic datasets to ensure generalization of trained models to real mission data.

Price et al. [3] extended this line of research by applying a CNN-based pose estimation pipeline to real imagery captured during the deployment of the Minerva-II2 rover by Hayabusa2. The deployment images were highly challenging due to low texture, specular highlights, and sensor saturation. To support model training, a synthetic dataset was created that mimicked these visual conditions. Their work underscores the limitations of traditional pose estimation pipelines when faced with real-world visual artifacts and highlights the importance of simulating sensor imperfections, such as blooming and high dynamic range effects, during dataset creation.

In contrast to dataset-focused studies, operational systems such as Northrop Grumman's Mission Extension Vehicle (MEV) [4] demonstrate the real-world challenges and solutions for spacecraft imaging. The MEV uses a combination of visible, long-wave infrared (LWIR), and LiDAR sensors to perform autonomous rendezvous and docking operations in geostationary orbit. The use of multiple sensing modalities helps mitigate the impact of environmental disturbances and sensor noise. While the MEV system is proprietary, its successful deployment highlights the practical importance of fault-tolerant imaging and the need for realistic testing during the development phase.

Finally, the work by Bechini and colleagues on the evaluation of image fidelity [2] introduces metrics such as the Feature Quality Index (FQI), which quantifies structural similarity between real and synthetic images based on ORB descriptor matching. These metrics provide a principled way to assess the realism of fault simulations and have been used in this report to validate the simulated blooming effect.

Together, these contributions outline a trajectory in spacecraft vision research that moves from basic synthetic image generation toward physically grounded, fault-aware simulation. They motivate our approach: to inject realistic faults into synthetic images in a modular way, evaluate them against real-world benchmarks, and provide tools that can be directly integrated into training and testing pipelines for fault-resilient spacecraft vision systems.

3 Categorizing Faults

3.1 Physical Mechanisms

Spacecraft imaging systems are exposed to a range of optical, electronic, and environmental disturbances. The order of fault injection in our simulation framework reflects the physical sequence in which such faults naturally arise: from optical distortions during light capture to downstream sensor and digital readout effects. Prioritizing faults this way helps preserve causal consistency and prevents overlaps that would otherwise distort the simulation.

1. Lens Flare. Lens flare is an optical artifact caused by internal reflections within a multi-element lens system. When bright light sources enter the camera off-axis, they reflect between the lens elements and sensor surface, producing ghost patterns or scattered halos. These effects are governed by Fresnel reflections at each air-glass interface:

$$R = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2$$

where n_1 and n_2 are the refractive indices of two adjacent media. Since lens flare forms optically before light reaches the sensor, it is injected first in the simulation pipeline.

2. Blooming. Blooming occurs when the accumulated charge in a pixel exceeds its saturation capacity. For a saturated pixel, the incoming charge Q_{in} surpasses the maximum capacity $Q_{\text{sat}} = C_{\text{pixel}} \cdot V_{\text{sat}}$, causing excess electrons to diffuse into neighboring pixels. This creates bright radial halos or streaks around highly illuminated areas. Blooming is most prominent in CCD or CMOS sensors when imaging the Sun or bright spacecraft surfaces.

3. CCD Saturation Streaks. In CCD sensors, charges are transferred vertically (or horizontally) during readout. When a pixel becomes saturated, the excess charge can leak along the transfer path, producing streaks aligned with the readout direction. This effect is different from blooming, as it originates from the sensor's readout mechanism rather than lateral diffusion.

4. Dark Current Noise. Thermally excited electrons accumulate in sensor pixels even in the absence of light, generating what is known as dark current. The dark current density follows an exponential temperature dependence:

$$I_{\text{dark}} = I_0 \cdot e^{-E_g/(kT)}$$

where I_0 is a pre-exponential factor, E_g is the silicon bandgap energy, k is Boltzmann's constant, and T is the temperature. Over time, this effect introduces a noisy texture in dark regions and contributes to the appearance of hot pixels, especially in aged or uncooled sensors.

5. Vertical Glitch. Vertical glitches simulate abrupt column shifts or duplication due to errors in memory addressing or radiation-induced bit flips. They appear as vertical bands where pixel data have been offset or corrupted. These errors typically occur during digital readout or transmission and disrupt the spatial coherence of the image.

6. Horizontal Glitch. Similar to vertical glitches, horizontal glitches result from row-level readout corruption or buffer misalignment. These manifest as dislocated horizontal stripes and are usually caused by timing mismatches or interference during sensor readout or data packetization.

7. Channel Shift. Channel shift, or chromatic misalignment, occurs when the red, green, and blue channels of an image are not perfectly aligned. One physical cause is chromatic aberration, where different wavelengths of light are refracted by different amounts:

$$n(\lambda) = n_0 + \frac{A}{\lambda^2}, \quad f(\lambda) \propto \frac{1}{n(\lambda)}$$

This wavelength-dependent behavior leads to color fringing, particularly near high-contrast edges. Misalignment can also result from sensor miscalibration or digital post-processing errors.

8. Gaussian Noise. Gaussian noise is added to simulate the cumulative effect of stochastic fluctuations in photon detection, thermal noise, and signal amplification. It is modeled as additive zero-mean Gaussian noise with a tunable variance. While some sources of this noise are physical (e.g., shot noise), others are introduced during analog-to-digital conversion or onboard compression.

Importance of Priority Order. The simulation sequence mimics the natural order in which light and signal interact with an imaging system. Optical artifacts such as flare are added first, since they occur before the light reaches the sensor. Sensor-related effects like blooming and saturation follow, as they depend on charge accumulation. Thermal and electronic noise are applied next, representing background physical processes. Finally, digital artifacts such as glitches and channel shifts are introduced, with Gaussian noise acting as a final pass to capture residual statistical variations. This order ensures physical consistency and prevents artifacts from obscuring or distorting one another.

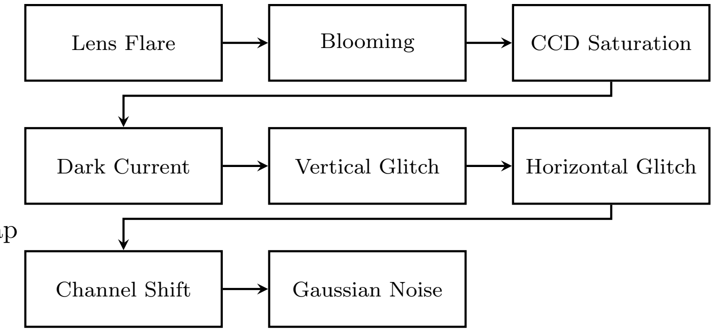


Figure 1: Ordered pipeline of simulated faults, arranged by physical origin from optical to digital stages.

3.2 Simulation Methods

We developed a modular fault injection framework in Python to simulate physically accurate degradations in spacecraft imagery. The simulator operates directly on RGB images and processes them in a causally consistent order that mirrors the real-world imaging pipeline: from optical distortions to sensor-based effects, followed by readout and digital noise.

Each fault type is implemented as a distinct method in the simulation class, enabling selective activation, parameter tuning, and batch processing. All outputs are clipped to valid pixel intensity ranges, and intermediate layers are combined additively to simulate fault superposition.

Lens Flare is injected first, simulating ghost reflections caused by bright light entering the lens. The brightest region in the image is detected and used as the flare source. Semi-transparent circular artifacts ("ghosts") are placed along the vector between the light source and image center. Their sizes, tints, and positions are randomized within configurable bounds to produce a variety of flare appearances.

Blooming is simulated next by detecting overexposed pixels and adding radial glow and light streaks around them. The glow is constructed using blurred circular masks, while streaks radiate outward in random directions. Both components are accumulated on a floating-point flare layer and blended into the original image. Bloom intensity, directionality, and radius are fully tunable.

CCD Saturation is modeled by extending saturated regions into linear streaks aligned with the sensor's readout direction. The method detects the most overexposed regions and applies horizontal or vertical streaks centered on these pixels. Gaussian blur is applied to soften the edges, and the final result is alpha-blended with the input image.

Dark Current Noise simulates the effect of thermal electrons accumulating in the absence of light. The noise level and density of hot pixels grow with the simulated sensor age. Gaussian noise is added with intensity weighted by pixel darkness to reflect higher noise in shadowed regions. Additional faint streaks, representing thermally-induced current leakage, are overlaid diagonally across the image.

Vertical and Horizontal Glitches mimic digital artifacts caused by radiation hits or readout corruption. For vertical glitches, randomly selected columns are displaced up or down; for horizontal glitches, random rows are shifted laterally. The number of glitches and their offsets are randomized, simulating scanline misalignments and jitter.

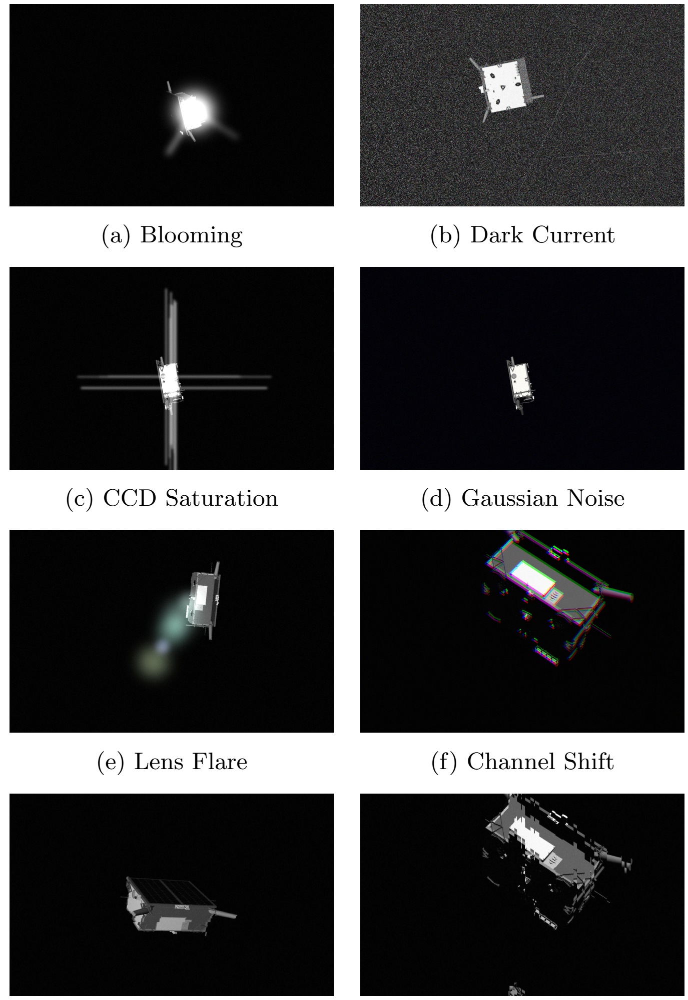
Channel Shift introduces chromatic misregistration by applying independent translations to the red, green, and blue channels. These shifts create visible color fringing, especially near edges, and emulate both chromatic aberration and sensor misalignment. The shift vectors are sampled randomly within a specified range.

Gaussian Noise is applied last to represent residual signal fluctuations and lossy transmission effects. The noise variance is randomly sampled within user-defined bounds and added to the image as zero-mean Gaussian noise. Optionally, strong JPEG compression is applied to simulate degraded onboard encoding.

The modular structure of this simulator allows faults to be composed sequentially or applied independently. Parameters such as noise strength, shift magnitudes, flare intensity, and streak lengths can be adjusted to match different mission scenarios or stress-test specific vision tasks. The framework supports batch processing of image folders and is designed for seamless integration into data augmentation pipelines.

Fault Type	**Domain**	**Key Parameters**
Lens Flare	Optical (lens reflections)	Ghost count, intensity, position vector, RGB tint
Blooming	Sensor overflow	Glow radius, streak count, flare intensity
CCD Saturation	Charge transfer artifacts	Streak length, direction, thickness, blur sigma
Dark Current Noise	Thermal/electronic	Sensor age, base noise, hot pixel rate, streaks
Vertical Glitch	Readout memory error	Max glitch count, vertical offset
Horizontal Glitch	Readout memory error	Line height, horizontal offset
Channel Shift	Sensor misalignment	RGB shift range in x/y
Gaussian Noise	Transmission degradation	Noise variance, optional JPEG compression

Table 1: Summary of implemented spacecraft image faults with their physical origin and adjustable parameters.



(a) Blooming (b) Dark Current (c) CCD Saturation (d) Gaussian Noise (e) Lens Flare (f) Channel Shift (g) Horizontal Glitch (h) Vertical Glitch

Figure 2: Examples of simulated faults in spacecraft imagery. Each subfigure shows a different type of injected degradation.

Below are more detailed simulation algorithms for the two key faults:

Algorithm 1 Simulate Blooming

Input: Image I, random bright spots, parameters for glow radius r_g, streak length l_s, streak width w_s, and intensity I_s
Output: Image I_{bloom} with blooming (glow and streak) effects

Convert image I to grayscale
Initialize empty flare layer F
for each bright spot **do**
    Add radial glow around the spot using blurred circular mask
    Generate several light streaks in random directions using blurred lines
    Add the glow and streaks to the flare layer F
end for
Clip values of F to range $[0, 255]$
Add flare layer F to original image I, then clip the result
Return updated image I_{bloom} with blooming effect

Algorithm 2 Simulate Lens Flare

Input: Image I, brightness threshold T, number of ghosts n, scale range $[r_{\min}, r_{\max}]$, intensity I_g, transparency α
Output: Image I_{flare} with simulated lens flare effects

Convert image I to grayscale
Apply threshold T to select the brightest spot as flare source
Compute vector from flare source to image center
Initialize empty flare layer F
for each ghost (1 to n) **do**
    Determine ghost position along the vector
    Randomly select radius within $[r_{\min}, r_{\max}]$
    if ghost position is out of image bounds **then**
        continue
    end if
    Draw a blurred circular ghost at the position
    Apply slight random RGB tint and transparency α
    Add tinted ghost to flare layer F
end for
Clip values of F to range $[0, 255]$
Add F to image I and clip result
Return updated image I_{flare} with lens flare effect

4 Fidelity Evaluation

To assess the realism and impact of simulated faults on spacecraft imagery, we employed a suite of evaluation metrics that quantify both structural degradation and perceptual consistency. These include: the **Feature Quality Index (FQI)**, **Histogram-Based Bhattacharyya Distance**, and the **Shadow Preservation Index (SPI)**. Each metric addresses different aspects of fidelity and reflects the types of artifacts most critical in space navigation and computer vision applications.

The reference images used were extracted from the `sunlamp/` folder of the SPEED+ dataset [5], which contains 2791 real 8-bit monochrome JPEG images with a resolution of 1920×1200 pixels. These were acquired under controlled lighting conditions simulating solar illumination, and are among the few publicly available datasets capturing real blooming effects.

4.1 Feature Quality Index

The Feature Quality Index measures the degradation of structural image content, particularly in terms of keypoints relevant for navigation, pose estimation, or object detection. It is based on ORB (Oriented FAST and Rotated BRIEF) descriptors, which are invariant to rotation and noise.

The FQI is computed as:

$$\text{FQI} = 1 - \frac{1}{256} \cdot \frac{1}{N} \sum_{i=1}^{N} \text{Hamming}(\mathbf{d}_i^{\text{ref}}, \mathbf{d}_i^{\text{faulty}})$$

where N is the number of valid descriptor matches, and $\mathbf{d}_i$ are the ORB descriptors from the reference and fault-injected images. A higher FQI indicates better structural preservation.

This metric is especially relevant for artifacts such as blooming, CCD saturation, or Gaussian noise, which may obscure image structure and reduce keypoint matchability.

4.2 Shadow Preservation Index (SPI)

Although not directly used in our quantitative validation, the **Shadow Index** remains a valuable tool in space imaging where shadows are semantically important. SPI measures the consistency of shadow regions in faulted images versus reference ones.

It is computed by comparing shadow masks obtained through Otsu thresholding:

$$\text{SPI} = 1 - \frac{\left\| 1_{S_{\text{ref}} \neq S_{\text{faulty}}} \right\|_1}{\left\| 1_{S_{\text{ref}}=1} \right\|_1}$$

where S are binary masks representing shadow regions. A higher SPI means more accurate shadow preservation.

In this study, SPI was not used due to sensitivity to lighting conditions and segmentation thresholds, which made it less reliable across different fault types. However, it remains a promising candidate for future evaluation under more controlled illumination scenarios.

4.3 Histogram Similarity (Bhattacharyya Distance)

The histogram similarity metric evaluates the global distribution of pixel intensities between original and faulted images. We use the Bhattacharyya distance:

$$D_B(H_1, H_2) = -\ln\left(\sum_i \sqrt{H_1(i) \cdot H_2(i)}\right)$$

where H_1 and H_2 are normalized intensity histograms. A lower D_B indicates a closer match in global tone and intensity spread.

This is particularly useful for faults such as lens flare, dark current noise, or saturation, which introduce broad photometric shifts rather than localized distortions.

4.4 Performance Metrics

Each metric contributes uniquely:

- **FQI** evaluates structural integrity and feature matchability.
- **Bhattacharyya Distance** captures global intensity shifts.
- **SPI** (when applicable) assesses the semantic accuracy of shadow regions.

This multi-metric evaluation ensures both photometric realism and functional usability of fault-injected images for downstream tasks such as navigation and pose estimation.

Why Shadow Index Was Not Implemented

Although the Shadow Index (SI) is a robust quantitative metric introduced in [2] for comparing simulated and real images under varying illumination, it was not applied in this study due to the lack of strictly aligned image pairs. The SI requires a one-to-one correspondence between synthetic and real images under near-identical geometric and lighting conditions in order to evaluate the preservation of shadow structure. However, in our case, blooming artifacts were simulated on synthetic images for which no precisely aligned real counterpart with matching lighting and viewpoint exists. As a result, applying the SI would yield misleading or inconclusive results. We therefore relied instead on metrics that do not require pixel-wise registration, such as histogram correlation and the Feature Quality Index (FQI), which operate on localized patches and evaluate structural and statistical similarity in a less restrictive manner.

FQI and Histogram Evaluation of Blooming Fault

To evaluate the realism of the simulated blooming artifact, we designed a validation procedure based on structural and photometric similarity, without requiring paired image data. First, we selected approximately 10% of generated images containing bright regions above an arbitrary intensity threshold of 250. These candidate images were processed using the blooming simulator.

To isolate the fault region for evaluation, we applied a contour-based cropping method that extracts the bounding box around the brightest blooming zone in each image. The same process was applied to a dataset of real spacecraft images exhibiting natural blooming. To ensure a fair comparison, both real and generated crops were resized to a common shape through center-cropping, enabling patch-wise evaluation.

We then computed the Feature Quality Index (FQI) and histogram correlation for each cropped pair. The FQI measured the similarity of structural features using ORB descriptors, while histogram correlation assessed global photometric similarity within the blooming region.

Among all real images, we retained the one yielding the highest combined FQI and histogram score for each generated image. This enabled a meaningful, localized assessment of realism despite the absence of globally registered image pairs. Images that did not meet a predefined similarity threshold were flagged and excluded from further use in the dataset.

This process allowed us to obtain objective validation results focused specifically on the blooming region, mitigating the bias that full-image comparison would introduce in the absence of strict geometric alignment.

5 Results and Discussion

5.1 Validation results

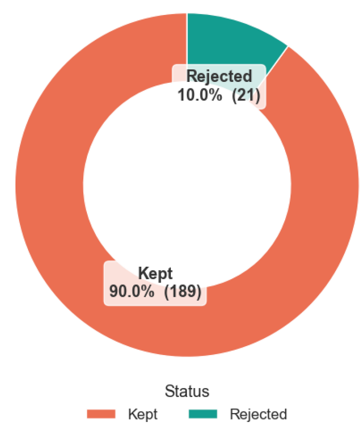


Figure 3: Validation outcome distribution. 90% of generated blooming images passed the FQI and histogram-based similarity criteria.

The results in Figure 3 and Figure 4 demonstrate the effectiveness of the blooming simulation and validation pipeline.

Out of 210 images evaluated, 189 (90%) were retained after passing both the Feature Quality Index (FQI) and histogram similarity thresholds (respectively 0.80 & 0.90). This high acceptance rate indicates that the simulated blooming artifacts are, in most cases, structurally and photometrically close to real examples extracted from the SPEED+ dataset.

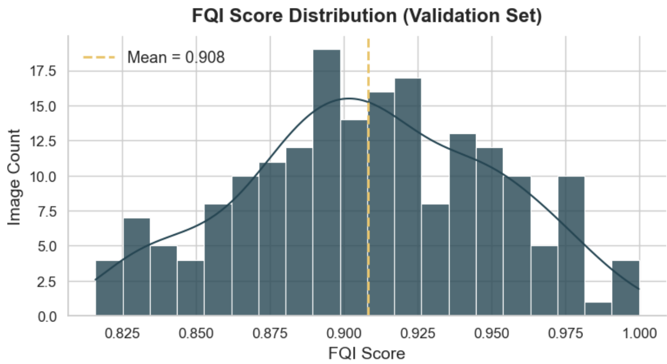


Figure 4: FQI score distribution across validated blooming samples. The mean FQI score is 0.908, indicating strong structural similarity to real blooming artifacts.

Figure 4 shows the distribution of FQI scores across all samples. The scores are tightly clustered around a mean of 0.908 and a median of 0.910, with a standard deviation of 0.042. The presence of a long tail on the lower end highlights a small subset of generated images that failed to preserve enough keypoint-level structure, with the lowest FQI being 0.816.

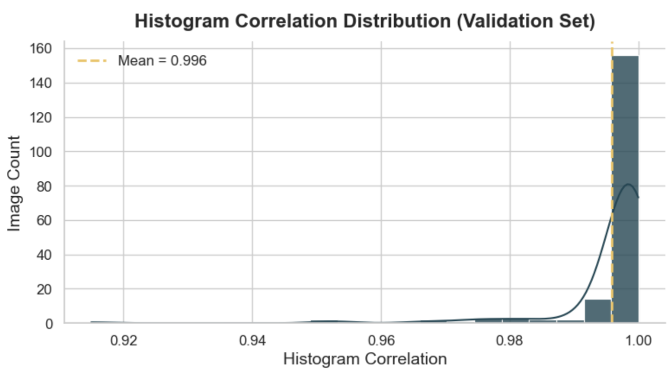


Figure 5: Histogram correlation distribution across validated blooming samples. The mean score is 0.996, indicating strong structural similarity to real blooming artifacts.

The histogram similarity in Figure 5 (mean correlation: 0.996) further confirms that the synthetic blooming closely matches the tonal distribution of real blooming regions. This suggests that the simulation does not introduce significant global intensity artifacts beyond the expected flare-like effect.

These results validate the fidelity of the blooming simulation method and confirm its suitability for augmenting training datasets in spacecraft vision pipelines. Moreover, the region-focused comparison strategy (cropping around the blooming area) was effective at producing objective metrics in the absence of globally aligned reference data.

5.2 Metric sensitivity analysis

To validate the sensitivity and discriminative power of our comparison metrics, we performed an additional control experiment: comparing real blooming regions to their synthetic counterparts *before* blooming was applied.

In this setup, we reused the same cropped region (as defined by the blooming mask) from the original non-bloomed synthetic image, and attempted to match it with the most similar real blooming patch, using both the FQI and histogram similarity criteria. This mirrors the main validation pipeline, but replaces the simulated blooming with its unaltered version.

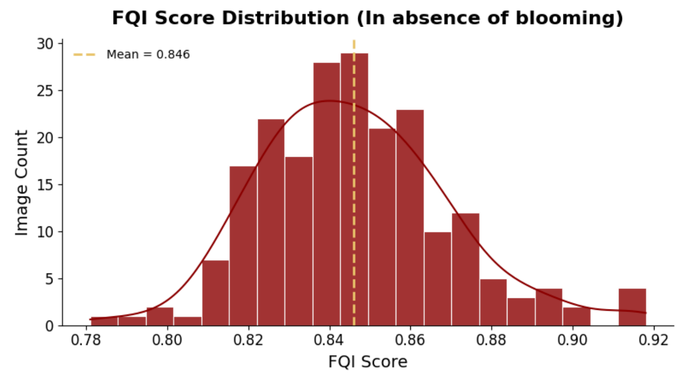


Figure 6: FQI score distribution when comparing non-bloomed synthetic crops to real blooming regions. Mean FQI = 0.846.

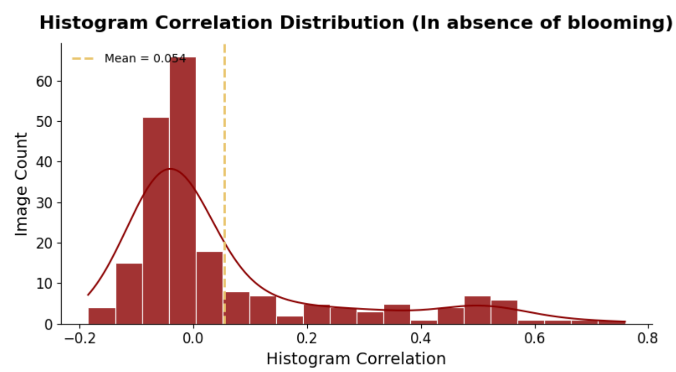


Figure 7: Histogram correlation distribution when comparing non-bloomed synthetic crops to real blooming regions. Mean correlation = 0.054.

Figures 6 and 7 show a marked drop in both structural and photometric similarity metrics when the blooming effect is not present. The FQI distribution is slightly shifted leftward, with a mean of 0.846 (vs. 0.908 in the validated blooming samples), indicating a lowest keypoint correspondence between clean and real-bloomed crops. Similarly, histogram correlation drops significantly to an average of 0.054, confirming tonal discrepancies.

For future work, increasing the FQI threshold from 0.8 to 0.9 could improve specificity. More importantly, these results highlight the necessity of using both metrics in combination: histogram correlation captures tonal characteristics that FQI may miss, ensuring that retained images are structurally and photometrically faithful to real blooming phenomena.

These findings reinforce the validity of the selected dual-metric strategy: neither metric alone is fully reliable, but together they provide a robust decision mechanism.

5.3 Limitations

A central limitation of this study is the absence of publicly available spacecraft datasets containing pixel-level annotations for specific visual faults. While datasets like SPEED+ offer a valuable benchmark for pose estimation and include realistic lighting and geometry, they do not provide detailed fault labels or segmentation masks for artifacts such as blooming, lens flare, or sensor noise.

As a result, our quantitative fidelity evaluation was constrained to the blooming effect, for which a subset of real images exhibiting natural blooming was available. Even in this case, validation relied on indirect comparison via structural and photometric similarity rather than direct ground-truth alignment. For other faults, such as lens flare, channel shifts, and readout glitches, evaluation was necessarily qualitative, based on visual inspection and domain knowledge.

This lack of reference data limits the scope of objective benchmarking and impedes the development of automated detection or correction pipelines. Addressing this gap will require either synthetic datasets that include labeled fault maps or, ideally, collaboration with space agencies and industry to gain access to annotated imagery from operational missions.

6 Team Contributions

During this project, I developped a comprehensive fault injection module for the data loader, aimed at simulating a variety of imaging artifacts commonly observed in space-based cameras. These included optical effects (e.g., lens flare, chromatic misalignment), sensor-level faults (e.g., blooming, dark current, saturation streaks), and digital readout or transmission issues (e.g., vertical/horizontal glitches, Gaussian noise).

The goal of this component was to help generate more realistic training data, which is now used to train and evaluate object detection models under degraded imaging conditions. This work complements Andrew's efforts by providing the faulty input data needed to test model robustness, and it forms an essential part of the end-to-end simulation and analysis pipeline.

7 Conclusion and Future Work

This work presented a physically grounded fault simulation framework for spacecraft imagery. We modeled a variety of common faults, ranging from optical distortions to sensor-level and readout errors, and implemented a modular system to inject these degradations into synthetic images. Our methodology respects the causal ordering of image formation, from light capture through sensor response to digital encoding. The framework supports batch processing, parameter control, and integration into machine learning pipelines.

To evaluate fidelity, we adopted both structural and statistical metrics, including the Feature Quality Index (FQI) and histogram correlation, and applied them to localized regions affected by blooming. While the absence of paired ground-truth imagery restricted validation to a subset of faults, our pipeline demonstrated strong visual and metric-based correspondence to real-world effects.

Looking ahead, several extensions are possible. First, implementing a probabilistic fault injection scheme would allow more diverse and realistic degradation patterns across large datasets. Second, integrating fault simulation into real-time data augmentation during training could improve the robustness of vision models. Finally, expanding the validation pipeline with new datasets or collaborative annotation efforts would significantly enhance the generalizability and benchmarking power of this approach.

This project lays a foundation for fault-aware learning systems in orbital vision applications and highlights the value of physically interpretable simulations in bridging the domain gap between synthetic and real space imagery.

References

[1] M. Bechini, M. Lavagna, and P. Lunghi, "Dataset generation and validation for spacecraft pose estimation via monocular images processing," *Acta Astronautica*, vol. 204, pp. 358–369, 2023.

[2] M. Bechini, P. Lunghi, and M. Lavagna, "Spacecraft pose estimation via monocular image processing: Dataset generation and validation," in *9th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, 2022.

[3] A. Price and K. Yoshida, "A monocular pose estimation case study: The hayabusa2 minerva-ii2 deployment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021.

[4] M. Pyrak and J. Anderson, "Performance of northrop grumman's mission extension vehicle (mev) rpo imagers at geo," in *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2021.

[5] E. S. Agency, "Pose estimation challenge dataset (speed+)," https://kelvins.esa.int/pose-estimation-2021/data/, 2021, accessed: 2025-06-03.