Spectral Rendering of Space Objects

Julia Renard
Department of Computer Science
EPFL
Lausanne, Switzerland

Andrew Price
Computer Vision Laboratory
EPFL
Lausanne, Switzerland

Stephan Hellmich
Laboratory of Astrophysics (LASTRO)
EPFL
Lausanne, Switzerland

Jean-Paul Kneib
Laboratory of Astrophysics (LASTRO)
EPFL
Lausanne, Switzerland

Mathieu Salzmann
Computer Vision Laboratory
EPFL
Lausanne, Switzerland

***Abstract*—The long-term vision of the project is to generate synthetic satellite images that are both visually realistic and physically accurate, with respect to material reflectance and geometries. These images would be incorporated into current datasets containing observational data of satellite and space debris after additional processing. Current datasets are already extended with synthetic images but they do not reflect any physical accuracy in shape, size or material. Consequently, improving the generation of this synthetic images would allow to improve current algorithms in space debris detection. This report aims to present the first steps to create synthetic satellite images that are both visually realistic and physically accurate. Using the Mitsuba rendering system, we implement a spectral rendering pipeline to account for the diverse interactions between light and material surfaces based on the specific wavelengths involved. We then use this pipeline to render simple shapes with different material before focusing on the rendering of a more complex object. The Advanced Composite Solar Sail System (ACS3) [1] serves as a representative case to validate our framework. The code is available on GitHub [2].**

I. Introduction

Our research focuses on generating physically informed synthetic images that aim to be visually realistic but also grounded in accurate modeling of light interactions with satellite materials and geometries. By implementing a spectral rendering and leveraging detailed material properties, our goal is to create synthetic images that can complement existing datasets and enable the study of satellite properties under various observational conditions.

This report represents the initial phase of our work. It first focus on building and validating a spectral rendering pipeline. The pipeline comprises three key components: a spectral emitter modeling the Sun, material responses of the surface's objects, and the sensor camera equipped with transmission filters, selecting a certain range of wavelengths.

Additionally, we investigate the limitations of the rendering system when rendering objects at large distances and we examine critical parameters such as material roughness and object geometry, to understand their impact on rendered images and establish a baseline for physical accuracy. Finally, we demonstrate the approach's effectiveness by rendering a more complex object resembling the Advanced Composite Solar Sail System (ACS3).

While this work is just the beginning, it demonstrates the feasibility of generating physically accurate synthetic images and sets the stage for future efforts.

II. Implementation of the Spectral Rendering Pipeline

We start by defining the essential components, specifically tailored for spectral rendering, required to render future scenes. We will use the TELESTO [3] telescope in Geneva as a reference for the observing camera's characteristics.

A. Mitsuba Renderer

For this project, we use Mitsuba [4] , a rendering system developed by the Realistic Graphics Lab at EPFL. A large variety of plugins are available to implement functionality such that materials, light sources and rendering algorithms. It offers support for derivative tracking, LLVM and CUDA compilation and different formats for light simulation such that RGB, spectral or polarized.

B. Source emitter

Given the object's distance from the Sun, we make the following assumptions:

1) There is no obstruction between the Sun and the observed object.

2) The sunrays reach the object perfectly parallel.

We use the Mitsuba emitter plugin `directional`, which implements a "source that radiates a specified power per unit area along a fixed direction", aligning with our requirements. To implement spectral rendering, we use an Adobe PostScript (SPD) file provided by ClearSpace [5] , defining the solar irradiance across wavelengths in the visible spectrum (even though the sun radiates beyond this domain). The sun spectrum in displayed in Fig. 1

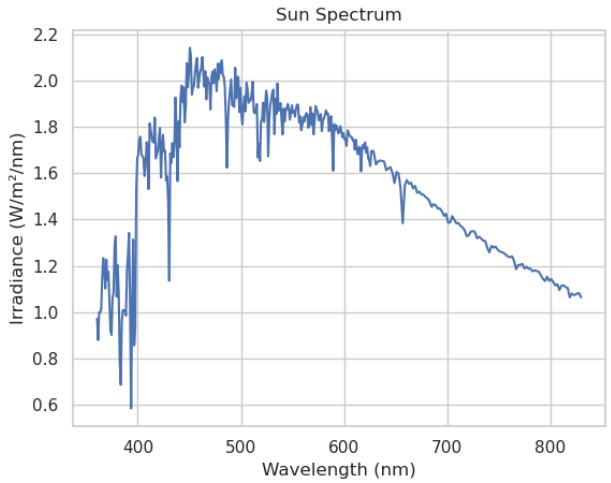

Fig. 1. Sun Spectrum.

C. Sensor

1) The Telesto telescope: The TELESTO is a 60 cm diameter Ritchey-Chretien [6] telescope with a focal ratio of $f/3.8$. It is equipped with a QHY600M Pro CMOS detector [7], providing a field of view of approximately 0.9° x 0.6°. The telescope offers a selection of six Baader-Sloan/SDSS filters.

2) Perspective Sensor: To model the sensor, we use the `perspective` which implements a perspective camera, equivalent to a simple pinhole camera. It has infinite depth of field, meaning that no optical blurring occurs. To define the camera's properties, we must specify either the focal length or the field of view along one axis, with the other axis's field of view automatically determined by Mitsuba. To ensure precise control, we opt to specify the focal length directly, calculated using the following formula:

$$\text{Focal Ratio} = \frac{\text{Focal Length (mm)}}{\text{Aperture (mm)}}$$

The amount of light captured by the telescope is given by the mirror aperture, which is about 60cm. We obtain a focal length of 2280mm. The resolution comprises 60 MP which are distributed as approximately 9400 in the width and 6400 in the height of the image. In practice, we often lower the resolution to allow faster computation but we keep the 3:2 ratio.

3) Sampler & Ray Tracing: As Mitsuba is using ray tracing [8] to render the scene, we need to define a sampler. The camera is sending a certain amount of rays for each pixel. When those rays reach a surface, it is evaluated depending on the material response and the emitter in the scene, to define the intensity of this pixel. In the case of RGB rendering, each pixel is evaluating for three different channels, while in spectral rendering, each pixel is evaluated for each wavelength defined in the filter applied. The sampler constructs light paths to evaluate the flow of light through the scene, simplifying the complexity of the integration process. These integrals are solved numerically by sampling the function at many different positions. We use the `stratified` plugin, which helps reduce sample clumping and ensures uniform coverage. This sampler is more complex than the default "independent" plugin, resulting in a finer rendering. We did not explore the other samplers.

D. Film & Spectral Rendering

We use the `spectral` film plugin. We are provided with six Baader Sloan/SDSS filters, designed to match the spectral bands used by Sloan Digital Sky Survey (SDSS). These filters feature narrow bandwidths and minimal overlap between bands. This ensures clear separation of spectral features making them ideal for observing distant astronomical objects with high precision. Summarized in Tab. I are the wavelengths range in which the filters transmit more than 80% of the light received:

Filter	Transmission Range (nm)
g	[401, 554]
i	[699, 834]
u	[332, 353]
r	[570, 700] - [1240, 1260]
y	[959, 1052]
z_s	[828, 917]

TABLE I
TRANSMISSION RANGE FOR BAADER SLOAN/SDSS FILTERS TABLE

For one image, we obtain n gray-scale spectral rendering, where n correspond to the number of filters. In practice, we only use the first three filters as they select wavelengths in the visible domain and our emitter is defined solely in this domain.

E. Material Rendering

How the texture of an object is perceived depends on how it reflects and refracts light. In our renderer, these properties are controlled by the 'BSDF' (Bidirectional Scattering Distribution Function). Mitsuba provides a wide range of predefined BSDFs ad depicted in Fig. **??**. For this project, we focus on three simple BSDFs commonly used to model materials for satellite construction. These BSDFs will later be applied to our objects to define their material properties during rendering.

1) Diffuse BSDF: To model an object covered by a thin layer of paint, we use the `diffuse` plugin. This represents a simple matte material that scatters light uniformly in all directions.

2) Rough Conductor BSDF: To model metallized material, we use the `roughconductor` plugin. It allows to model material such that aluminium (Al) and chromium (Cr), two materials commonly used in satellite construction and in particular in the ACS3 solar sail, that is our rendering target. This plugin implements a micro-facet scattering model using a distribution of specular facets to simulate surface roughness. The alpha roughness value is between 0 and 1. An alpha value

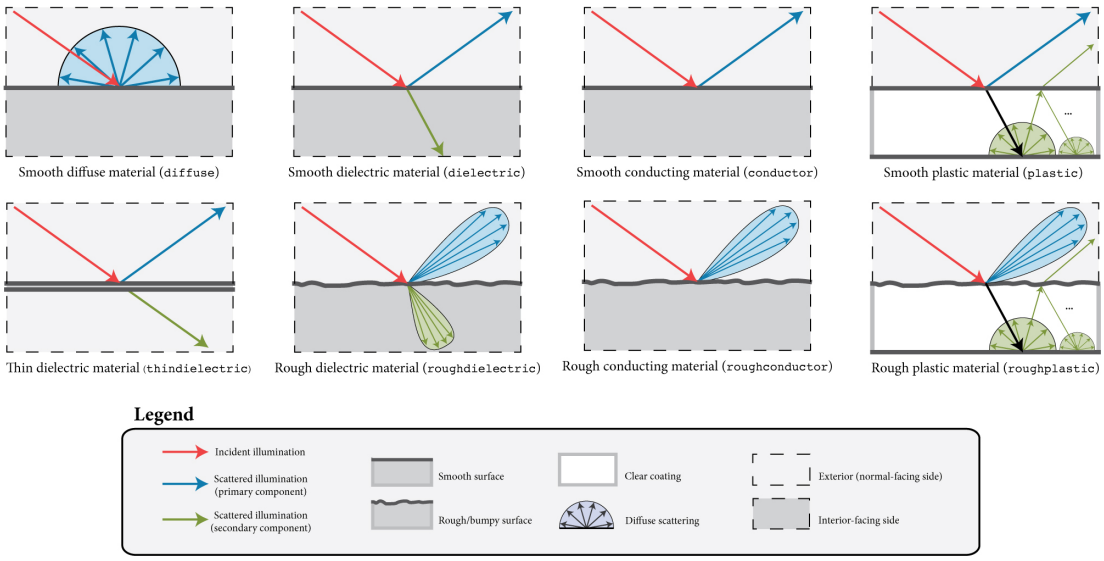

Fig. 2. BSDF Models available in Mitsuba.

close to zero corresponds to a very smooth material and above 0.3 to a rough material.

3) Twosided BSDF: By default, no BSDF is applied to the internal face of an object. The `twosided`BSDF allows to apply one BSDF to the external face and a different one to the internal face. This feature is particularly useful for modeling solar sails, as they typically have a planar structure.

III. Validation Spectral Rendering Pipeline

To assess the validity of our spectral rendering implementation, we designed a test that involves creating an object capable of reflecting only a specific range of wavelengths. This is achieved by assigning a filter transmission as its BSDF.

In this example, the object is configured with a diffuse BSDF corresponding to the g-filter, which reflects wavelengths approximately in the range of 401–554 nm, in every direction.

Expected Results:

- Camera filter g + BSDF reflect g: High intensity values.
- Camera filter r + BSDF reflect g: Overlap of ranges $\rightarrow$ Medium to High intensity values.
- Camera filter i + BSDF reflect g: Low to zero intensity values.

The distribution of intensity values for each filter is shown in Fig. 3. This histogram includes only pixels with non-zero reflectance values, as zero-value pixels correspond to the background where no object are rendered. By excluding these background pixels, the focus is placed entirely on the reflectance properties of the object itself.

The first histogram (left) represents the intensity values obtained when the object reflects wavelengths that the camera filter is designed to transmit. The high mean intensity value indicates that the object exhibits strong reflectance within the wavelength range of this specific filter. It suggests compatibility between the object's surface properties and the filter's transmission spectrum.

The second histogram is particularly interesting as it reflects a scenario where the object's reflected wavelengths partially overlap with the transmission range of the camera filter. The mean intensity value is moderate, suggesting that while there is some alignment between the object's reflective properties and the filter's transmission spectrum, the overlap is not complete. This indicates that only a portion of the reflected light is effectively captured by the camera, leading to a medium-level reflectance response.

Finally, the other filters exhibit very low mean reflectance values, which is expected given that their wavelength ranges do not significantly overlap with the object's reflective spectrum. However, these intensity values are not zero because the filters still transmit a very small fraction of other wavelengths, at much lower transmission rates.

The spectral rendering for a cube with diffuse BSDF and rotated along axes x and z is displayed in Fig. **??**.

Note: Similar tests have been conducted for all filters, yielding consistent results. These tests can be reproduced using the code available on GitHub.

To conclude, the results of our spectral rendering seem to align with the expected physical behavior.

IV. Rendering of Simple Shapes with Different BSDFs

Fig. **??** provides an overview of the types of renderings we can produce. The first row displays a diffuse cube rotated along the x and z axes. The second row shows an aluminum cylinder with an alpha value of 0.5. This images correspond to the g filter channel rendering.

V. Distance

Now that we have configured the camera and emitter specifications for a coherent spectral rendering, a key challenge to address is managing the large-scale distances present in the scene we aim to render.

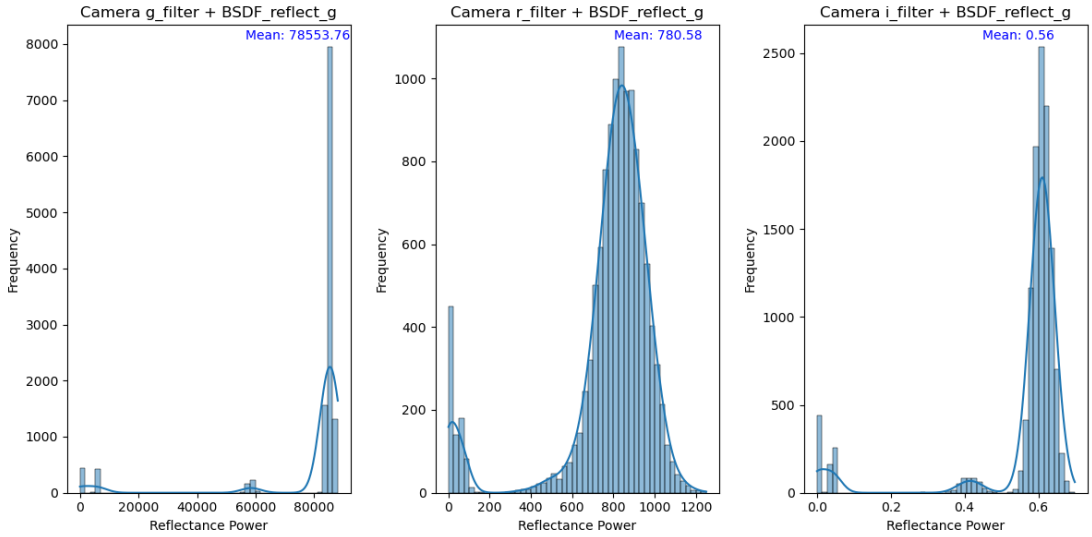

Fig. 3. Histogram analysis of reflectance values across various camera filter and BSDF combinations.

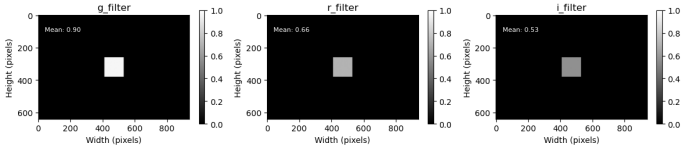

Fig. 4. Spectral Rendering obtained for a diffuse cube.

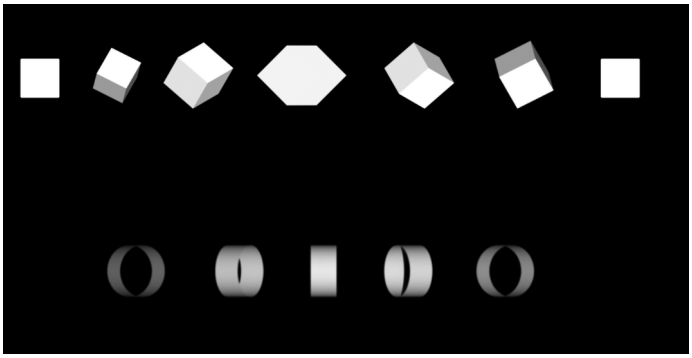

Fig. 5. Rendering of simple shapes using G filter.

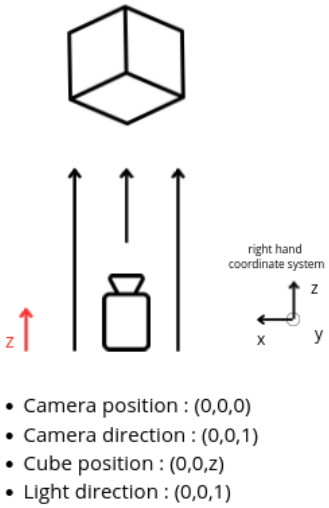

Fig. 6. Components schema for testing distance limitations.

Satellites are located hundreds to thousands of kilometers away from the observer or sensor, which are orders of magnitude beyond the typical rendering scenarios for most software. Mitsuba is primarily optimized for rendering scenes at scales meter-scales. This inherent focus raises questions about its effectiveness and accuracy when simulating phenomena at much larger scales. Mitsuba utilizes numerical methods and approximations that are well-suited for small to medium-sized scenes. However, at satellite scales, these methods may encounter precision errors, unexpected artifacts, or reduced realism.

To test this we will implement two basic scenario. In the first one we gradually increase the distance between the camera and the object. In the second, we gradually scale the object. The Fig. 6 depicts the configuration of the test. The result for a diffuse cube are depicted in Fig. 7.

We recommend to use the second method, as it is able to render objects at distance up to 100km. At this distance, the image is not resolved but it is expected. At real orbital altitudes (650–950 km), resolving the object is challenging. Typically, the object appears as a smudge. However, this smudge should retain the desired spectral properties corresponding to the object.

These results should be taken with caution as we obtained inconsistent results, sometimes, when using other BSDF than the diffuse one.

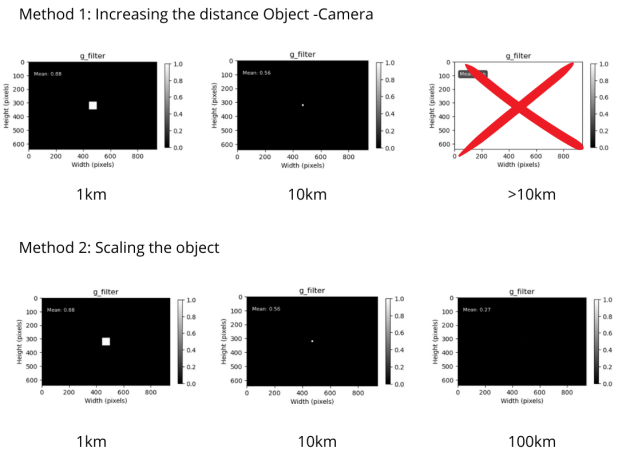

Fig. 7. Results using two different methods for testing distance limitations.

VI. Investigation of Roughness Parameters and Object Geometries

We observed NaN or distorted renderings at certain rotation angles and for certain alpha values. We aim to have a more precise idea of the impact of the alpha value on rendering and analyze more precisely the conditions under which ill-rendering occur, considering the BSDF, shape, and angle of rotation. The tests are done by putting the object at a distance of 1km.

A. Influence of Alpha in Reflectance for Conductor Material

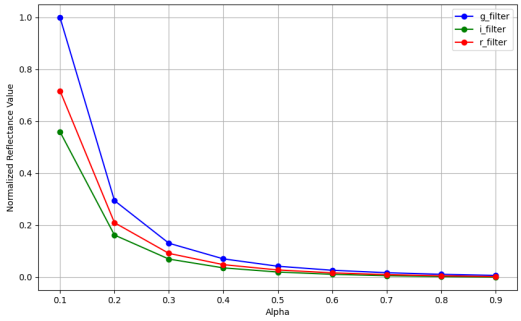

Fig. 8. Influence of alpha on Aluminium reflectance for different filters.

By increasing the alpha value, we simulate a rougher surface, and we aim to explore how this alteration affects the material's reflectance and if it behaves accordingly to physics properties.

As illustrated in Fig. 8, a reduction in the alpha value, corresponding to a smoother surface, results in an increase in reflectance. This phenomenon can be attributed to the enhanced ability of smooth surfaces to reflect light efficiently with minimal scattering caused by surface irregularities. Conversely, as surface roughness increases, reflectance decreases significantly due to the diffuse scattering of light, which reduces direct specular reflections. These observations align with the physical principles governing material reflectance

The Mitsuba rendering documentation notes that alpha values exceeding 0.7 are generally unrealistic in terms of real-world materials, as they imply an excessively rough surface that does not correspond to typical material properties.

The Fig. 8 also shows the specific wavelength range that is most strongly reflected by aluminium. Notably, aluminium reflects light most effectively in the wavelength range of approximately [401, 570] nm, corresponding to the g filter, described in Tab.I. It is worth noting that this filter spans the broadest range of wavelengths within the visible spectrum among all available filters, which may contribute to its dominance.

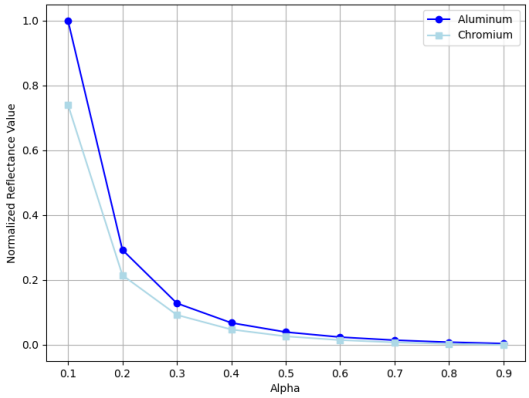

Fig. 9. Influence of alpha on reflectance for Aluminium and Chromium.

By comparing the two conductor materials of interest in Fig 9, we observe that both behave similarly as alpha increases. However, aluminum exhibits higher reflectance than chromium, although it undergoes a more abrupt decrease in reflectance at smaller alpha values.

B. Influence of Rotation over Reflectance

In the following test, we apply rotations along both the X and Z axes as it creates more interesting scenarios (in terms of light reflectance) compared to rotation around a single axis.

1) Rotation effect in function of different filters: As shown in Fig. 10, a change in orientation has unsurprisingly a noticeable impact on the reflectance values. However, the object, a cube with a diffuse material, exhibits consistent behavior under rotation, regardless of the applied filters. The only variation is in the reflectance levels specific to each filter.

2) Rotation effect in function of different materials: The graph in Fig. 11 illustrates distinct reflectance behaviors for diffuse, aluminum, and chromium materials as the angle increases. The alpha value for rough conductor material is

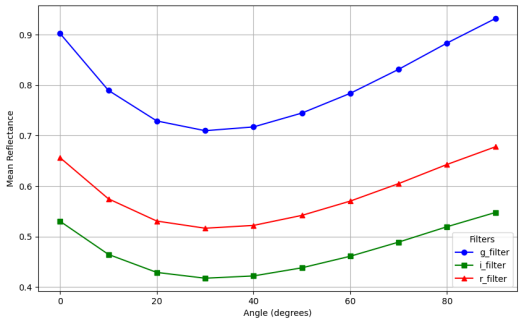

Fig. 10. Rotation effect on reflectance for different filters for a diffuse cube.

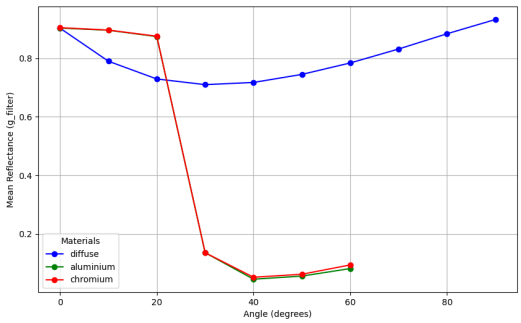

Fig. 11. Rotation Effect on Reflectance for Different Materials.

set to 0.1. Diffuse material exhibits a gradual change in reflectance, maintaining higher values across the angular range. In contrast, aluminum and chromium display sharp declines in reflectance at smaller angles, stabilizing at lower values beyond 40 degrees. We notice however, that a rotation beyond 60 degrees leads to NaN rendering. This behavior is likely attributed to the high reflectance of the material, which can result in extreme values, either very high or very low, when refracted at certain angles.

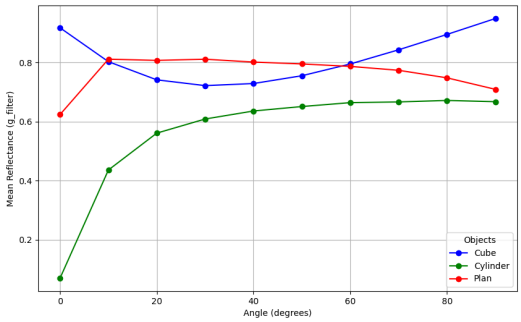

Fig. 12. Rotation Effect on Reflectance for Different Shapes.

3) Rotation effect depending on Different Shapes: The graph in Fig. 12 compares the mean reflectance values for three object geometries, a cube, a cylinder, and a plane, across varying angles, using a diffuse material. The cylinder (green curve) exhibits a sharp increase in reflectance at smaller angles before stabilizing. This behavior is expected, as the camera initially captures the bottom of the cylinder, which is hollow at its center, resulting in minimal visible surface area (primarily the perimeter). In contrast, the plane (red curve) shows relatively consistent reflectance across all angles, with only minor variations. We can see that different shapes exhibit distinct patterns of reflectance variation when rotated, which is expected.

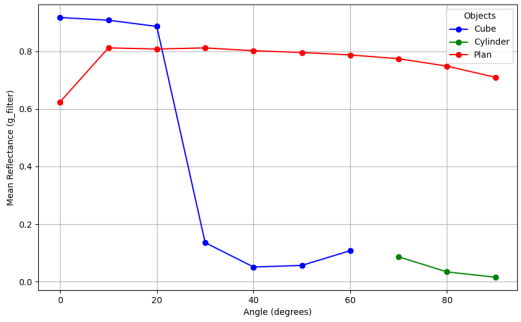

Fig. 13. Rotation Effect on Reflectance for Different Shapes in Aluminium (alpha: 0.1).

Fig. 13 depicts a similar graph in Fig. 12 but this time it compares different shapes of smooth aluminium objects (alpha value of 0.1). The cube (blue curve) shows a sharp decline in reflectance beyond 20 degrees, reaching a minimum around 40 degrees, and then exhibits a slight recovery at higher angles before being rendered as NaN. In contrast, the plane (red curve) remains relatively stable, with a high reflectance value that decreases slightly as the angle increases. However, the cylinder is deeply affected by the aluminium material as it results in a NaN rendering for almost all angles. By increasing the value of alpha, making the aluminum surface rougher, we eliminate the NaN rendering issue. This confirms that the problem was caused by the physical reflectance properties of smooth aluminum, which reflects light in a single direction, resulting in a behavior similar to a mirror in space (thus not visible) and potentially leading to precision and numerical problems. These graphes give us a rough idea of when we can expect a NaN rendering when rotating the object depending on its shape and BSDF.

VII. Advanced Composite Solar Sail System Rendering (ACS3)

This section provides the details regarding the specific implementation for the rendering of the ACS3 [9]. This satellite is a small spacecraft that uses sunlight radiation pressure for propulsion, eliminating the need for traditional rocket propellant. It was launched on April 23rd , 2024. Its mission is to demonstrate the deployment and operation of a solar sail using lightweight composite materials in space. This mission

aims to validate solar sail technology for future deep space exploration by testing its maneuverability and performance.

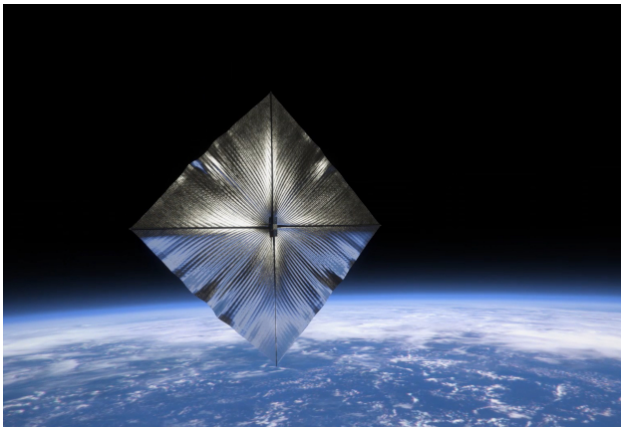

Fig. 14. ACS3 Satellite.
Credits: Sailing the Cosmic Seas: NASA's Solar Sail Odyssey. (Photo Internet reproduction)

A. Shape & Materials

The ACS3 consists of four triangular quadrants, each with a 9-meter outer edge and a 1-meter inboard cutout. The ACS3 solar sail membrane quadrants are made from 2.0 µm polyethylene naphthalate (PEN) plastic film. One side is coated with a 100 nm aluminum layer to reflect solar photons, while the other side has a 15 nm chromium layer to enhance thermal emissivity. We model the material's characteristics using a two-sided BSDF, applying an aluminum BSDF on one side and a chromium BSDF on the other. Due to the satellite's thin structure, we can assume its surface is smooth, as thin materials typically undergo precise manufacturing to ensure minimal surface irregularities, crucial for performance in aerospace applications. A smooth aluminium surface is a flat surface as a microscopic level, allowing to reflect light in a specular manner, meaning light is reflected in a single direction. The material appears shiny and bright. For our satellite, we select an alpha value of 0.1, indicating a smooth material that is less reflective than a mirror.

B. Triangular Meshing & Delaunay Algorithm

Using the `Mesh` Mitsuba class, we manually create the ACS3 shape in 3D. We begin by creating a trapezoidal shape defined by four coordinates. When meshing this object, we obtain a triangular mesh consisting of two triangles. To improve the realism of the rendering, we add points inside the trapezoid to increase the number of triangles. This is achieved by creating a grid, with adjustable point spacing to control the mesh's granularity. Using the four vertices of the trapezoid, we select the grid points within its area including the border. Once the points are defined, we apply the Delaunay algorithm to generate the faces that form the final triangular mesh. Finally, we apply rotations to the trapezoid and save it in a Polygon File Format (PLY) file. Our final object is modeled by four PLY files, each containing one panel of the solar sail.

Also to be noted:

- This process is performed in 2D due to the planar nature of the solar sail. After applying the Delaunay algorithm, we introduce a third axis, setting all coordinates along it to zero, as the Delaunay algorithm cannot be applied to an object with all-zero third-axis coordinates. This should be kept in mind for potential future 3D generalizations.
- Due to computational limitations, it was not possible to render a single PLY file containing the four trapezoids, which would have been more convenient for manipulation. However, the code for this has been implemented.

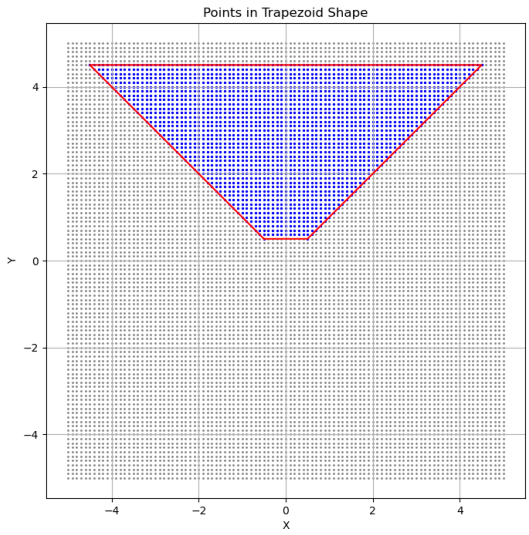

Fig. 15. Point augmentation for one trapezoidal panel.

In Fig 15, the grid granularity is set to 101x101 with 0.1 spacing.

C. Delaunay Algorithm

The Delaunay algorithm is used to generate a triangular mesh from a set of points in a plane. The algorithm works by iteratively refining the mesh, flipping edges where necessary to satisfy the Delaunay criterion. The Delaunay criterion states that for any triangle in the mesh, the circle that passes through its three vertices (the circumcircle) should not contain any other points from the set (vertices excluded). This algorithm seeks to maximize the minimum angle of all the triangles in the mesh, resulting in a more stable and visually uniform structure.

In Fig. 16, the resulting mesh is shown after augmenting the points with a grid of size 11x11 and a spacing of 1. In practice, we use a 1001x100 grid with a spacing of 0.01, but the resulting mesh would be too small to be visualized.

D. Fold Creation

This section presents the experiments for creating folds on the surface of the solar sails. While the techniques remain relatively basic, it is a first step towards more realism.

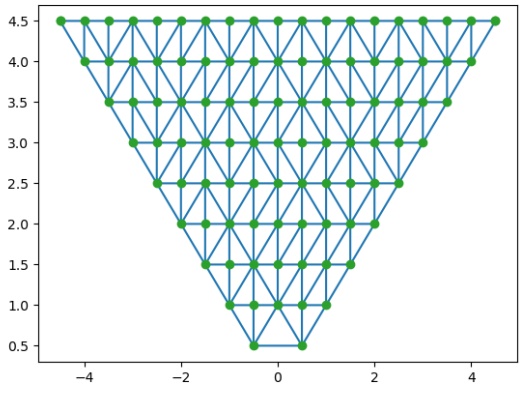

Fig. 16. Triangulation mesh for one trapezoidal panel.

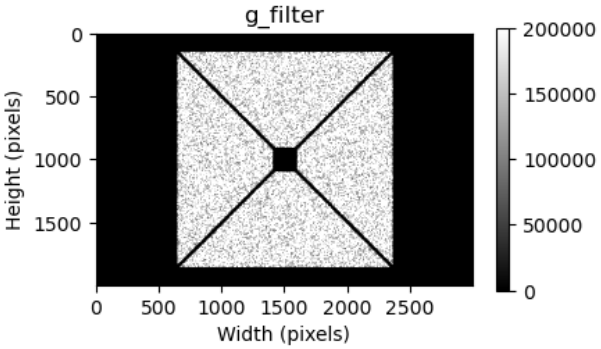

Fig. 17. ACS3 with Random Folding.

1) Random Noise Generation: The solar sail is defined along the x and y axes, with z-axis coordinates set to zero initially. Uniform random values within a specified range [-0.01, 0.01[are generated to create subtle depth variations, resulting in small, irregular facets on the surface. Figure 17 illustrates the resulting rendering.

2) Sine Ondulation:

E. ACS3 Sinusoïdal Folding

The second implementation introduces a sinusoidal wave along the z-axis to create folds. These folds are intentionally unevenly spaced, adding irregularity to the structure. A fold amplitude parameter controls the height of the folds, while a base frequency parameter specifies the number of folds across the trapezoid's dimensions. The parameters used to generate Fig. **??** include a fold amplitude of 0.01, a base frequency of 7, and no random noise.

1) Sine Ondulation and Random Noise:

F. ACS3 Sinusoïdal Folding and Random Noise

Finally, the third implementation combines elements of the two previous approaches. The result is displayed in the Fig. 19. The parameters used to generate it include a fold amplitude of 0.01, a base frequency of 7, and a random noise of 0.005.

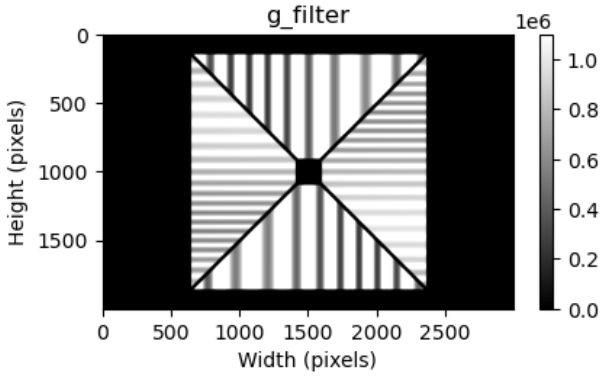

Fig. 18. ACS3 with Sinusoïdal Folds and Varying Frequency.

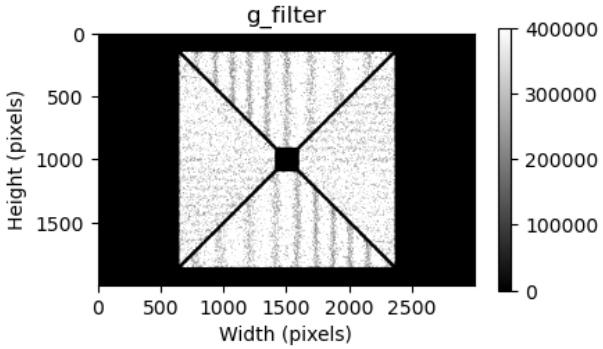

Fig. 19. ACS3 with Sinusoïdal Folds, Varying Frequency and Noise.

A wide range of results can be achieved by fine-tuning the combination of parameters, including amplitude and frequency. This area offers significant potential for further exploration and refinement.

VIII. Conclusion

In conclusion, our study demonstrates the efficacy of Mitsuba in generating realistic spectral renderings for satellite imaging. The validation tests show that the spectral rendering accurately reflects expected physical behaviors. The exploration of distance limitations reveals that Mitsuba can handle objects up to 100 km away using the TELESTO sensor's specifications. However, inconsistent renderings can be observed, depending on the BSDF used. Additionally, the analysis of material properties, particularly the roughness parameter (alpha) for metals such as aluminum and chromium, confirms that smoother surfaces produce higher reflectance, consistent with the principles of physical optics. This effect can lead to precision issues when values become either too small or too large. Lastly, our rotational tests highlight the sensitivity of reflectance to changes in orientation and material properties, particularly in complex shapes and conductor materials, reinforcing the importance of these parameters in achieving accurate renderings. The ACS3 rendering provides a compelling example of how our methods can be applied to real-world satellite models. However, the current folding

techniques, including random and sinusoidal folds, could be improved to better mimic the complexities of real solar sail structures. Future work would involve developing more sophisticated folding algorithms and understanding understand more deeply the behavior of the mitsuba renderer when putting object far away, why and under which conditions distortion or Nan rendering occur. Additionally, for a more comprehensive generalization, we suggest moving the emitter to simulate different lighting conditions. This adjustment would enable a more accurate representation of how satellites reflect light under various observational geometries, enhancing the realism and applicability of the synthetic images.

IX. Acknowledgment

Gratitude is extended to Andrew Price and Stephan Hellmich for their great support, providing resources and valuable guidance throughout the project.

References

[1] Advanced composite solar sail system (acs3). https://www.nasa.gov/mission/acs3/ (accessed 07/01/2025).
[2] Github repository. https://github.com/EPFL-Space-Center/SLiC.
[3] Characteristics of telesto. https://plone.unige.ch/astrodome/history%2C%20documentation/manuels/optical-information/characteristics-of-telesto (accessed 07/01/2025).
[4] Wenzel Jakob et al. Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org.
[5] Clearspace. https://clearspace.today/ (accessed 07/01/2025).
[6] Ritchey-chretien telescope. https://fr.wikipedia.org/wiki/T%C3%A9lescope_Ritchey-Chr%C3%A9tien (accessed 07/01/2025).
[7] QHYCCD camera. Qhy600ph series. https://www.qhyccd.com/astronomical-camera-qhy600/ (accessed 07/01/2025).
[8] Ray tracing. https://fr.wikipedia.org/wiki/Ray_tracing (accessed 07/01/2025).
[9] William K. Wilkie. *Overview of the NASA Advanced Composite Solar Sail System (ACS3) Technology Demonstration Project*.