Titan Light Curve Analysis with Python

Alex Medina

University of Central Florida, Department of Physics

Abstract

Aims: The study of light curves and polarimetry is a useful method for understanding atmospheric, and thus surface properties of planetary bodies. Here, I present a light curve analysis of Titan, Saturn's largest moon, serving as an analog for exoplanet research due to it's atmospheric and surface interactions. By analyzing Titan's light curve and polarimetry, I introduce the beginnings of a framework for exoplanet light curve analysis.

Methods: My code employs Python to manipulate and analyze calibrated images. Using packages like skimage and opencv, my algorithm detects the disk of Titan and determines an effective radius – calculating the geometric albedo using theory based off of the Cassini ISS User Guide. Additionally, I employ an IDL Virtual Machine script to produce polarized images to determine the degree of linear polarization and other relevant parameters.

Results: The results showed consistent trends in the brightness phase curves across all filters, naturally decreasing from low to high phase angles, but with dramatic increases at higher phase angles. Additionally, it appeared that higher wavelengths had a much wider range in values for geometric albedo, while polarization phase curves tended the same in a parabolic manner with peaks at around 90 degree phase.

1 Introduction

A light curve is a graphical representation of the brightness of an object as a function of the phase angle. For planets or moons, light curves can reveal a lot about their reflective properties. As a body like Titan orbits its parent planet or a star like the Sun, the angle of illumination changes, causing the observed brightness or reflectivity to vary. These variations mapped into a light curve provide scattering properties or atmospheric composition as different elements reflect differently across different filters.

Of particular interest is polarimetry which involves measuring the polarization of light. Light becomes polarized when it is scattered or reflected by surfaces or particles in atmospheres. Hence, by analyzing the polarization state of light from a celestial object, we can gain information about the size, shape, and composition of the scattering particles.

When applied to Titan, polarimetry helps uncover details about the thick atmosphere of the moon, the presence of clouds, haze layers, and the surface below. Polarimetric measurements can distinguish between different types of atmospheric particles and surface features [1].

With this project, I used Python and its various packages to analyze Titans light curves. With the help of Dr. Robert West, IDL-based scripts were also used to produce polarimetric data, aiming to elucidate the atmospheric and surface properties of the Moon. Through this comprehensive approach, I aim to construct a framework for exoplanet analysis by using Titan as an analog.

This report is organized as follows. In Section 2, I discuss data acquisition and relevant equations, paired with pseudocode, used to calculate geometric albedo and degree of linear polarization of Titan. In Section 3, I present some results in the form of brightness and polarization phase curves with some basic interpretations of the observational results. Finally, in Section 4, I summarize the report, discussion plans moving forward in the next couple months.

2 Methodology

2.1 Data Acquisition

My analysis began by acquiring calibrated images of Titan from the Cassini mission. I began my search on OPUS, Outer Planets Unified Search. For Cassini images, I filtered my search, but some important filters included: Surface Geometry Target Selector: Saturn to Titan; Instrument Name: Cassini ISS; and selecting the appropriate camera and camera filter. After filtering the images, I visually inspected images and selected only those that met a certain visual criteria: images where all of Titan's disk is in the image, no overexposure, data from volumes later than COISS2004, and such that Titan wasn't close to the border as Titan's atmosphere doesn't have a clear border as its atmosphere thins out.

To touch on the camera and filter selection, for the purpose of this report, I only selected images from the NAC, Narrow Angle Camera, as it is less susceptible to stray light at large phase angles than the WAC, Wide Angle Camera [2]. As for filter combinations, I tried to stick with filters that had polarized image counterparts in the NAC. Particularly, the NAC offered filters with polarizer at 0, 60, and 120 degrees while the WAC primarily had polarized filters at 0 and 90 degrees. You can see the filter selection and other relevant information in Table 1 and 2 below.

Table 1: Summary of Filters and Wavelengths

Filter	**Number of Images**	**Min Wavelength (μm)**	**Max Wavelength (μm)**
`CL1_BL2`	25	0.4252	0.4757
`CL1_CB2`	191	0.7455	0.7555
`CL1_GRN`	204	0.5116	0.6246
`CL1_MT1`	185	0.6171	0.6208
`CL1_MT2`	133	0.7254	0.7295
`CL1_CB1`	102	0.6144	0.6244

Table 2: Summary of Filters and Wavelengths

Filter	**Number of Images**	**Polarized Images**	**Min Wavelength (μm)**	**Max Wavelength (μm)**
`BL2_P0,60,120`	78	26	0.4253	0.4758
`GRN_P0,60,120`	78	26	0.5116	0.6246
`CB1_P0,60,120`	72	24	0.615	0.625

2.2 Prepossessing

The calibrated images from OPUS are in a .IMG extension while polarized images are in a .VIC extension. To deal with the vicar format, or binary image file, I used the python package rms-vicar from the PDS Ring-Moon Systems Node, a package to read and write VICAR image files, with the following algorithm:

```
import vicar
vic = vicar.VicarImage("path/to/file")
image = vic.array2d
```

Once they are in this format, they can be manipulated and used for analysis. The calibrated images were formatted such that each pixel units of I/F. Most of the light off the limb of Titan is light scattered from Titan within the camera and you want to include it to get an accurate result. If you only want to get the integrated reflectivity for the entire image you just need to sum all of the pixels, including pixels not on Titan. If I_j/F is the radiance from an image pixel j [2], then the irradiance from the whole Titan disk at the spacecraft-Titan distance l is:

$$\mathcal{F} = \frac{\sum_j \frac{I_j}{F}\mathcal{S}_{\text{pixel}}}{l^2}. \tag{1}$$

In equation (1), the summation runs over all pixels j within the circle of radius R_{target}, and S_{pixel} is the projected area of a pixel at the distance. F depends on the spacecraft-Titan distance l, but unclear at the moment, hence, we introduce the following equation:

$$\mathcal{F} = \frac{\pi R_{\text{target}}\mathcal{A}_g\Phi(\alpha)}{l^2} \tag{2}$$

Equation (2) provides a definition of Titan's overall reflectance, A_g according to R_{target} appropriate for Titan, and combining it with equation (1), we can derive the following expression:

$$\mathcal{A}_g\Phi(\alpha) = \frac{\sum_j \frac{I_j}{F}\mathcal{S}_{\text{pixel}}}{\pi R_{\text{target}}} \tag{3}$$

Using this relation, and given the parameters like pixel scale, we know that the number of pixels is described in the code and the equations are simply represented by:

```
def geometric_albedo(image, pixel_scale):
    R = 2575 #Radius of Titan in km
    n_pixel = np.pi * R**2 / pixel_scale**2 #number of pixels
    total_image_sum = np.sum(image)
    return total_image_sum / n_pixel
```

However, if you want to get accurate I/F values on the disk of Titan you would need to 'put back' the light that is scattered by the optics. To do that, I needed to deconvolve the image with the camera filter point spread function [2]. This is something that is not not done for calibrated OPUS images, however, it only increases accuracy by about 1 or 2 percent. The error will be greatest at the low phase angles where Titan's disk is 'big,' and least at the highest phase angles where Titan's illuminated pixels don't occupy very much of the image. The PSF files are also in Vicar format. To do this, I used the python package skimage using the Richardson-Lucy deconvolution algorithm to enhance the image clarity and reduce noise and using the deconvolved image for EQ 1 instead of the general image:

```
import skimage
psf_vic = vicar.VicarImage("path/to/PSFfile")
psf_data = psf_vic.data_2d
deconvolved_image = skimage.restoration.richardson_lucy(image,
   psf_data, 30)
```

Additionally, Dr. Robert West provided me with an IDL Virtual Machine script in order to create polarized images. Much of the coding to process images from Cassini and even to perform calibration are in IDL, however, the IDL Virtual Machine does not require a license. For reference, the script, in the form of a .SAV file, runs makepolar.pro from the CISSCAL IDL program. Since I was dealing with the NAC where the polarization filters were in 0, 60, and 120 degrees, the script required finding the 3 images corresponding to the polarization filters, grouping them in a folder, running the script inside the folder, and the output was three images, one with pixels in I/F units, one with pixels with degree of linear polarization (unitless), and one with the angle of linear polarization. For theory on the math behind the polarized image creation refer to the ISS User Guide [2]. Nevertheless, I treated them the same way.

2.3 Circle Detection and Effective Radius

First, the polarization for each pixel was calculated by taking ratios in the images using different polarizers. Accroding to Dr. Robert West, that process has no knowledge of which pixels are on Titan and which are not, so the entire image has polarization values. The values away from Titan are a combination of light spread beyond the limb from the point spread function, and noise in the CCD and possibly also some stray light if the camera is pointing close to Saturn or the Sun. So, to interpret the polarization image I needed to restrict my analysis to the disk of Titan, including pixels beyond the terminator since Titan's extended atmosphere is lit by the sun over a large region beyond the geometric terminator. For the same reason, values in the theta image should be ignored if they are not on the target.

For the I/F image it is not so critical since the intensity drops to near zero sufficiently far from the target, and you want to include intensity beyond the limb. So the circle detection isn't necessary unless you do a deconvolution, to pick up light put there by the point spread function if you want to integrate light from the entire disk. I did it both ways.

Simply, to only consider pixels within the disk of Titan in the image, I used the python package, opencv. Using the Hough Circle Transform provided by opencv, the disk, or 'circle' of Titan would be detected and provide coordinates for the center of the circle.' However, to not mess with the image structure, I used a copy of the image, and used the observed coordinates on the original image as such:

```python
def image_center(image):
    image_copy = image.copy()
    gray_image = cv2.normalize(image_copy)
    detected_circle = cv2.HoughCircles(gray_image)
    if detected_circles:
        detected_circles = np.round(detected_circles[0, :])
        for (x, y, r) in detected_circles:
            return (x, y)
```

After finding the center coordinates, it was important to find an effective radius for Titan as its borders are not clear. To do so, I plotted the total brightness of a circle centered on the disk of titan for various radii and returned the corresponding radius as the flux begins to plateau.

```python
def radii_fluxes_from_image(image, center):
    x, y = center
    image_copy = image.copy()
```

```
    gray_image = cv2.normalize(image_copy)
    radii = range(1, 1000, 5)
    flux = []
    for r in radii:
        mask = np.zeros_like(gray_image)
        cv2.circle(mask, (x, y), r)
        masked = cv2.bitwise_and(gray_image)
        flux.append(np.sum(masked))
    return radii, flux
```

```
def effective_radius(radii, flux):
    for i in range(1, len(flux)):
        if flux[i] - flux[i-1] < 0.01 * flux[i-1]:
            return radii[i]
    return radii[-1]
```

This defined an effective radius and this I was able to create a mask to only sum the values of the images and modifies the code of EQ 1 as:

```
def geometric_albedo_disk(image, center, radius, pixel_scale):
    mask = np.zeros_like(image)
    cv2.circle(mask, center, radius)
    R = 2575
    total_image_sum = np.sum(image * mask)
    n_pixel = np.pi * R**2 / pixel_scale**2
    return total_image_sum / n_pixel
```

Where for unitless degree of linear polarization:

```
def degree_of_linear(image, center, radius):
    mask = np.zeros_like(image)
    cv2.circle(mask, center, radius))
    total_image_sum = np.sum(image * mask)
    n_pixel = np.sum(mask)
    return total_image_sum / n_pixel
```

3 Results and Basic Interpretations

3.1 Data Compilation

To complete the analysis, for each image, I found the corresponding pixelscale in the form of pixel resolution in km/pixel and the phase angle. This information was found in the metadata .CSV file after selecting the appropriate metadata filters. Again, for more information on the filters, refer to Table 1.

3.2 Brightness Phase Curves

The measurements are strictly from Cassini/ISS images and normalized to a Titan normalized radius of R = 2575 km.

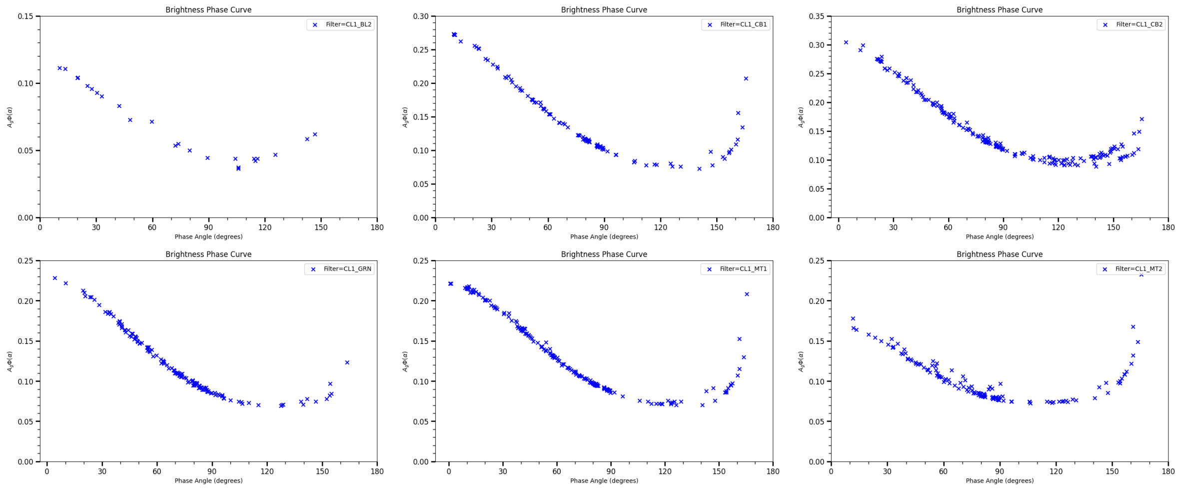

Figure 1: Brightness phase curves of Titan from 0 to 180 degrees phase angle for different filters. Brightness decreases from 0 to 90 degrees and plateaus between 90 and 130 degrees, especially in the MT2 filter (0.7254 μm). Brightness increases again from 130 to 170 degrees, with the MT1 filter (0.6171 μm) showing higher brightness at larger phase angles due to Titan's atmosphere and sunlight scattering.

From phase 0 to about 90 degrees, the curves follow an expected pattern: at 0 degrees, Titan is fully illuminated where at 90 degrees, it is only half illuminated, hence, visually, it makes sense that the brightness would decrease almost linearly. Keeping in mind, however, that it is difficult to get data points for complete illumination.

Uniquely, at phase angles between 90 to anyway from 120 to 130, depending on the filter selection, the curves begin to plateau despite diminishing illumination of Titan. Most noticeably in the MT2 filter, or 0.7254 micron wavelength.

At larger phase angles, from 130 up to about 170, as complete dark, phase angle 180 is difficult to observe, the brightness begins to increase, and at a much faster rate than when it was decreasing in the first half of the phase angles. Notably, the MT1 filter, 0.6171 microns, has a brightness at higher phase angles comparable and higher than full illumination.

The overall trend is that it appears that Titan's higher phase angles, or at less visual illumination, will appear to outshine its lower phase angles, closer to full illumination, at various wavelengths. This behavior is unique to Titan in our solar system, and is caused by its extended atmosphere and the efficient forward scattering of sunlight by its atmospheric haze [3].

3.3 Polarization Phase Curves

As in the brightness phase curves section, the measurements are strictly from Cassini/ISS images and normalized to a Titan normalized radius of R = 2575 km.

For the polarization phase curves, particular to the NAC, I was able to produced a set of polarized images to calculate degree of linear polarization, but this limited behavior only from about 10 degree phase angle to 150 degrees.

Generally, across all filter demonstrations, it appears that the sunlight reflected by Titan's atmosphere is strongly polarized at phases near quadrature.

Notably, at BL2 0.4253 wavelength, maximum degree of linear polarization was about 0.60, while for CB1 0.615 wavelength, it reaches a maximum of 0.45, and GRN 0.5116 wavelength reaches about 0.50. Just deducing from the numbers, it appears that at higher wavelengths the

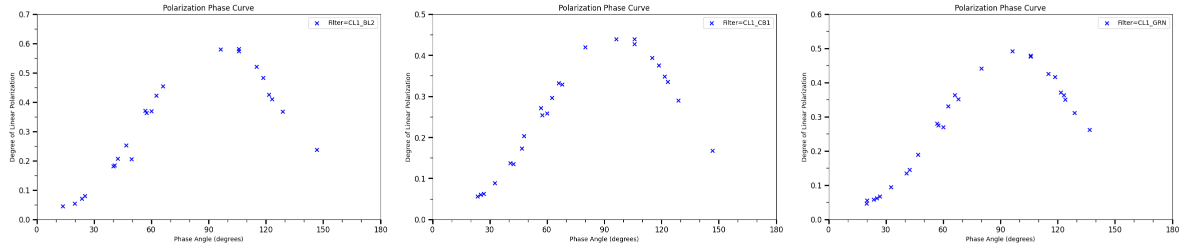

Figure 2: Polarization phase curves of Titan for different filters. The maximum degree of linear polarization is 0.60 at 0.4253 μm (BL2), 0.50 at 0.5116 μm (GRN), and 0.45 at 0.615 μm (CB1), with stronger polarization near quadrature phases.

maximum degree of linear polarization decreases while for lower wavelengths, the maximum degree of linear polarization increases. Of course, with no actual model at the moment, it is difficult to come up with definitive interpretations.

4 Conclusions and Future Work

For this report, I created light and polarimetric curves for Titan using IDL packages and an IDL script to create polarized images of Titan from Cassini. The results demonstrated trends such as the expected decrease in brightness from phase angles 0 to 90 degrees. However, there was an unexpected plateau from 90 to 130 degrees, and at larger phase angles 130 to 170 degrees, the brightness increased again, with the MT1 filter showing higher brightness at larger phase angles. It appears that from 170 to 180, it may even trend higher than the brightness at full illumination. This behavior is attributed to Titan's extended atmosphere and the efficient forward scattering of sunlight by its atmospheric haze.

For the polarization phase curves, my code revealed a strong polarization of sunlight reflected by Titan's atmosphere near quadrature phases. The maximum degree of linear polarization varied across different filters, with lower wavelengths showing higher degrees of polarization.

For the purpose of this report, I stuck with using only images from the NAC, however, I would like to expand my data set to using the WAC images. However, I will focus on wavelengths that are scientifically relevant, yet that may still require an analysis across all available filters.

Additionally, with the help of Dr. Theodora Karalidi, the goal is to implement a radiative transfer model to allow for a more detailed interpretation of the observational results. By integrating RTMs with our observational data, we aim to refine our understanding of the atmospheric composition, particle sizes, and surface reflectivity of Titan. Ultimately, with the goal of using Titan as an analog for exoplanets, and expand this framework for future analysis of exoplanet atmospheres.

References

[1] J. Veverka, "Titan: Polarimetric evidence for an optically thick atmosphere?" Icarus, 18, 657-650, 1973. `https://doi.org/10.1016/0019-1035(73)90069-9`.

[2] B. Knowles, "Cassini Imaging Science Subsystem (ISS). Data User's Guide. Cassini Imaging Central Laboratory for Operations (CICLOPS)." 2014.

[3] A. Garcia Munoz, P. Lavvas, and R.A. West, " Titan Brighter at Twilight than in Daylight." Nature Astronomy, 1, 0114, 2017. `https://arxiv.org/pdf/1704.07460`.