AN INTEGRATED FRAMEWORK FOR LIGHT CURVE SIMULATION AND SHAPE INVERSION OF HUMAN-MADE SPACE OBJECTS

by

Liam Robinson

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Aeronautics and Astronautics

School of Aeronautics and Astronautics

West Lafayette, Indiana

December 2023

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Carolin Frueh

School of Aeronautics and Astronautics

Dr. Kenshiro Oguri

School of Aeronautics and Astronautics

Dr. Keith LeGrand

School of Aeronautics and Astronautics

Approved by:

Dr. Carolin Frueh

ACKNOWLEDGMENTS

This work was supported by Katalyst grant number, Boeing grant number, and the National Defense Science and Engineering Graduate Fellowship through the Air Force Office of Scientific Research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF LISTINGS

LIST OF SYMBOLS

I irradiance in $\left[\frac{W}{m^2}\right]$

$\hat{I}$ normalized irradiance in $[W]$

ABSTRACT

PurdueThesis is a LaTeX document class used for master's bypass reports, master's theses, PhD dissertations, and PhD preliminary reports. This template demonstrates how to use PurdueThesis.

1. Introduction

The uncontrolled proliferation of human-made space debris puts space operations at risk. Determining the current state and predicting the future dynamics of space objects is critical for many fields within Space Domain Awareness (SDA) [1]. Estimating the shape of an object helps analysts characterize it, but doing so is difficult as distance and atmospheric turbulence prevents direct imaging [2]. As a result, passive techniques for object characterization often rely on light curves – optical brightness measurements collected over time. Light curves are particularly efficient for the task as they are inexpensive to collect and contain information about the shape, orientation, and material properties of the object that produced them [2], [3]. Solving light curve inversion in a general case would enable robust active debris removal, anomaly detection, and collision avoidance, all of which rely on accurate shape information.

1.0.1 Contributions

The framework detailed in this work contributes to the light curve simulation and shape inversion literature in a few notable areas.

Simulation Advances

A high-fidelity light curve simulator called LightCurveEngine was developed to support inversion algorithm development. The simulator is one to four orders of magnitude faster than ray tracing-based renderers commonly used in literature [3], [4]. It supports variable material properties, a variety of reflection functions, and dynamic solar panel rotation. In concert with the comprehensive background signal model and observer constraints, the engine generates realistic light curves for inactive debris, highly non-convex objects, and actively-controlled satellites.

Shape Inversion Advances for Convex Objects

This work presents a suite of changes that robustify the classical shape inversion algorithm for convex shapes [5]. New resampling and merging steps in the Extended Gaussian Image optimization stage yield sparser and more accurate shapes that are easier to reconstruct. An alternative optimization method for the shape support vector decreases convergence time for highly symmetric objects where the classical optimization algorithm fails

2. Literature

Light curve simulation methods differ between approaches and the object class under study. Kaasalainen and Torppa employ a Lambertian model for convex objects with a facetwise ray tracing scheme for non-convex objects [6]. Fan, Friedman, Kobayashi, and Frueh [2], [7]–[10] use a nearly identical scheme for human-made objects. Allworth et al. developed a ray traced simulator for light curves in Blender, accounting for photorealistic shadowing and motion blur [4], [11]. Many deep learning approaches including Furfaro et al. [12] and Cabrera and Bradley [13], [14] use a simple Lambertian model with no self-shadowing. Linares and Crassidis [15] apply a more specialized approach with a non-Lambertian Bidirectional Reflectance Distribution Function (BRDF) for lighting. McNally et al. [16] use a Phong BRDF without shadowing shadowing, citing computational intensity. Blacketer [17] implemented a Cook-Torrance BRDF for lighting with a plane stacking method for self-shadowing.

Methods for shape inversion fall into three major categories: Extended Gaussian Image (EGI), statistical estimation, and deep learning based methods, each approaching the problem from a different perspective.

Direct light curve inversion with the EGI uses a series of optimization problems to fit a convex shape to measurements. These methods were pioneered by Kaasalainen and Torppa for asteroids in [6] with simultaneous attitude inversion in [6]. While natural space objects like asteroids are largely convex, nearly all human-made space objects are highly non-convex, highlighting the need for a robust inversion scheme for both convex and non-convex space objects. The work of Kaasalainen et al. on asteroids was extended by Chng et al. [18] to find globally optimal spin pole and area vector solutions. Calef et al. [19] were early adopters of Kaasalainen and Torppa's EGI methods for human-made objects, focusing on multispectrum measurements. Bradley and Axelrad [14] applied EGI methods to recover convex approximations of representative GEO objects. Fan and Frueh [2], [3], [20] used the EGI with a multi-hypothesis scheme to recover human-made object shapes with measurement noise. Friedman [8], [21] quantified the observability of EGI inversion to inform sensor tasking schemes. Cabrera et al. [13] studied the effects of area regularization on Fan and Friedman's methods to achieve more accurate reconstructions.

A second approach leverages statistical estimation to retrieve shape information. Linares et al. [22] applied an unscented Kalman filter to estimate attitude and convex shape simultaneously, representing shape with vertex displacement on a sphere. Linares et al. [23] used a Multiple-Model Adaptive Estimation (MMAE) algorithm to predict the truth geometry and attitude by comparing observations with a bank of reference objects. Linares and Crassidis [15] used an an Adaptive Hamiltonian Markov Chain Monte Carlo scheme to estimate shape and other characteristics simultaneously.

A third approach relies on deep learning. Linares and Furfaro [24] used a deep convolutional neural network to classify novel light curves as rocket bodies, payloads, or debris. Furfaro et al. [12] used similar methods classify novel light curves into four truth object classes. Kerr et al. [25] adapted the architecture developed by Furfaro et al. to classify object shape and size in an extended training set. McNally et al. [16] use AI and differential approaches to identify satellites from simulated and real light curves. Allworth et al. [11] applied transfer learning to simulated and real measurements to classify object type.

Various other unique methods have been applied to the light curve shape inversion problem. Hall et al. [26] investigated methods for independently solving shape parameters in isolation without attitude information. Fulcoly et al. [27] used measurements from different sensor locations to determine shape under various attitude profiles. Yanagisawa and Kurosaki [28] fit an analytical light curve model for a tri-axial ellipsoid to derive the shape and attitude profile of a Cosmos rocket body. Kobayashi applied compressed sensing to recover shape information from light curves by taking advantage of shadowing geometry [9], [29].

Shape inversion for non-convex objects — mainly asteroids — has been studied by others in the past. Durech and Kaasalainen [30] determined a relationship between concavity size and the minimum solar phase angle where self-shadowing impacts the light curve. Viikinkoski et al. [31] investigated recovering large concavities from adaptive optics imagery, noting the fundamental non-uniqueness of any solution. They discuss how a single large concavity may produce identical scattering behavior to multiple smaller concave features [31]. Cabrera et al. [13] studied convex solutions for non-convex objects, concluding that the convex fit diverges from the true shape as the relative concavity size increases.

We approach the shape inversion problem with the foundational EGI optimization and object reconstruction methods of [2], [6]. The EGI optimization processes of [2], [6], [13] are improved using novel resampling and merging steps. These improvements circumvent the need for the regularization terms explored by Cabrera et al. [13]. We also address the reconstruction scaling issues present in Fan's work [2] with an objective function proposed by Ikeuchi et al. [32] in place of Little's [33]. The support optimization procedure is accelerated and strengthened with a preconditioning term proposed by Nicolet et al. [34], enabling the rapid reconstruction of more detailed convex objects than previously feasible.

Our approach has a number of general advantages. We do not require any *a priori* information about the truth geometry. Thus, unlike MMAE methods [23], we do not require a bank of reference models to recover shape information. Unlike deep learning methods, our method does not rely on the diversity of a training set to achieve realistic results [12], [25]. Our light curve simulation method improves on the facetwise ray traced shadows of [2], [6], [10] with shadow mapping, increasing shadow fidelity per unit computation time.

2.1 Coordinate Systems

2.1.1 International Terrestrial Reference Frame

The most intuitive Earth-centered reference frame is Earth-centered Earth-fixed (ECEF). An ECEF frame has its origin at the center of mass of the Earth and its axes fixed in the crust. The fundamental plane of the frame is defined to be the equator — defining the z-axis through Earth's instantaneous spin axis, and the reference direction through the intersection of the prime meridian and the equator — defining the x-axis. Completing the right-handed system the y-axis yields a reference frame that remains fixed, neglecting continental drift and other pesky (but sufficiently negligible) realities.

2.1.2 Right Ascension and Declination

Right ascension and declination, often shortened to RA/Dec, are useful angles from describing the angular position of an object on the celestial sphere from the perspective of an observer. Right ascension is defined as the angle of the observation projected onto the inertial $x-y$ plane, measured counterclockwise from inertial $\hat{x}$, represented by α. Declination is the angle from the $x-y$ plane to the observation with positive values above the $x-y$ plane (closer to inertial z) and negative values below. Declination is represented by δ. Given a unit vector direction $\hat{v} = [x, y, z]^T$ in inertial space, we can compute RA/Dec via Eq 2.1.

$$\begin{bmatrix} \alpha \\ \delta \end{bmatrix} = \begin{bmatrix} \text{atan2}(y, x) \\ \text{atan2}(z, \sqrt{(x^2 + y^2)}) \end{bmatrix} \tag{2.1}$$

2.1.3 Azimuth and Elevation

Azimuth and elevation, often shortened to Az/El, are similar angular quantities to right ascension and declination, but instead of being based on the inertial sphere, they are referenced to an arbitrary reference frame. For a telescope making observations of an object, the local East-North-Up (ENU) frame may be used. For a satellite star tracker, star azimuth and elevation might be reported in the satellite body frame. In either case, Eq 2.1 can be repurposed in terms of Az/El, where $\hat{v} = [x, y, z]^T$ is expressed in the frame of interest.

$$\begin{bmatrix} Az \\ El \end{bmatrix} = \begin{bmatrix} \text{atan2}(y, x) \\ \text{atan2}(z, \sqrt{(x^2 + y^2)}) \end{bmatrix} \tag{2.2}$$

Note that Eq 2.2 references azimuth to the x-axis, proceeding in the counterclockwise direction. Often, this reference axis and direction may be changed depending on the reference frame being used. For example, ground station observations may be referenced to local North — the second axis of the ENU system — proceeding clockwise. This would require the substitution $Az' = \frac{\pi}{2} - Az$. Notice that this substitution leads to Az' leaking outside the domain of $[0, 2\pi)$. This is not an issue for later coordinate transformations, but may be undesirable for plots. Wrapping the result back to the standard azimuth range via $Az_{wrapped} = \text{mod}(Az, 2\pi)$ is a sufficient fix.

2.2 Time Systems

2.2.1 Julian Date

Most tasks in space domain awareness and astrodynamics more generally are easier when using a continuous time system. For that reason, we adopt the Julian date. This quantity is defined is the number of days elapsed since January 1, 4713 B.C., at 12:00 [35]. Given a date timestamp of the form D/M/Y h:m:s between the years of 1900 and 2100, we can compute the Julian date via Eq 2.3.

$$JD = 376Y - \text{floor}\left[\frac{7Y + 7\cdot\text{floor}\left(\frac{M+9}{12}\right)}{4}\right] + \text{floor}\left(\frac{275M}{9}\right) + d + 1721013.5 + \frac{\frac{\left(\frac{s}{60}+60\right)}{60} + h}{24} \tag{2.3}$$

Another useful quantity for later time and coordinate system calculations is the number of Julian centuries since a particular epoch. In particular, we will often use the J2000.0 epoch, resulting in Eq 2.4 [35].

$$T = \frac{JD - 2451545.0}{36535} \tag{2.4}$$

Often we need to be more specific about the time scale being used in Eq 2.4. For example, if we wanted the number of Julian centuries for an input date in UT1, we would compute T_{UT1} using JD_{UT1}, which is in turn a function a date timestamp expressed in UT1.

2.2.2 Solar and Sidereal Time

Due to Earth's orbit around the Sun, we need to differentiate between solar and sidereal time. A solar day is defined as the time required for the Sun to pass and return to an observer's meridian — a line of constant longitude extending from the geographic south pole to the geographic north pole [35]. By contrast, a sidereal day is the time required for the stars to complete a revolution around an observer's meridian.

We can compute the sidereal time in seconds via Eq 2.5.

$$\theta_{GMST} = 67310.54841 + (3.15576\text{e}{+}09 + 8640184.812866)T_{UT1} + 0.093104T_{UT1}^2 - 6.2 \cdot 10^{-6}T_{UT1}^3 \tag{2.5}$$

2.2.3 Time Scales

There are many slightly different scales for measuring the passage of time. What follows is a minimal treatment of each. For a more comprehensive overview, see the descriptions in Vallado that this section draws from [35]. International Atomic Time (TAI) is based on measurements from atomic clocks. Universal Time (UT0) is derived directly from observations of the apparent position of the stars. UT1 is derived from UT0 by adjusting for polar motion. UT1 is offset from TAI by $\Delta UT1$, which is a dynamic quantity that must be continually observed. Universal Coordinated Time (UTC) is a truncation of UT1 that uses an integer number of leap seconds ΔAT to stay within 0.9 seconds of TAI. Terrestrial Time (TT) is defined by a constant offset of 32.184 seconds from TAI and preceeding at the same rate as TAI. Global Positioning System (GPS) time is also defined by a constant offset of

19.0 seconds from TAI and preceeds at the same rate as TAI. These time scale relations are summarized in Eq 2.6.

$$
\begin{aligned}
UTC &= UT1 - \Delta UT1 \\
TAI &= UTC + \Delta AT \\
TT &= TAI + 32.184^s \\
TAI &= GPS + 19.0^s
\end{aligned}
\tag{2.6}
$$

Figure 2.1 shows the evolution of UTC, UT1, and TT with respect to TAI. Notice that $\Delta UT1$ continually changes while ΔAT is always an integer.

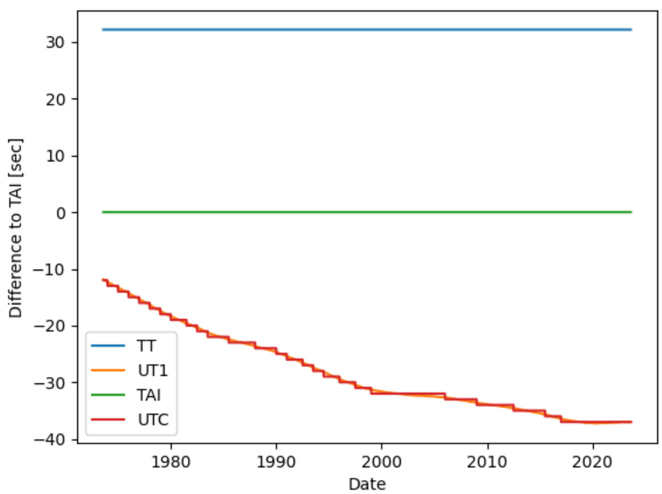

Figure 2.1. Time scales relative to TAI

3. Orbits

3.1 Relative Two Body Dynamics

Given a state vector $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ we can compute the state derivative in the relative two body problem with Eq 3.1.

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ -\frac{\mu x}{(x^2+y^2+z^2)^{1.5}} \\ -\frac{\mu y}{(x^2+y^2+z^2)^{1.5}} \\ -\frac{\mu z}{(x^2+y^2+z^2)^{1.5}} \end{bmatrix} \tag{3.1}$$

4. Attitude

4.1 Attitude Representations

When we talk about the orientation — also known as attitude — of a rigid body in three dimensions, that orientation is always implicitly understood to be relative to some other reference frame. The orientation of a book might be expressed using a frame fixed in the table it sits on. If that same book was sitting in an empty void, we would have no way to talk — or even think — about its orientation. Orientation itself is a three-dimensional quantity. Consider a coordinate system fixed in a rigid object and a second reference frame in which we want to express the orientation of the object. For convenience, we will call the frame fixed in the object the body frame, and the second frame the world frame. Any effective attitude representation must let us express the directions of all three body axes in terms of the world frame basis vectors. This raises an important question: how many numbers do we need to express an object's attitude? We can express the direction of any unit vector with two numbers — the azimuth and elevation of that vector. Naïvely, we might extrapolate from this to conclude that we will need six numbers to express an orientation. Because the basis vectors form an orthonormal set $\left\{\hat{b}_1, \hat{b}_2, \hat{b}_3\right\}$, we know we can express $\hat{b}_3 = \hat{b}_1 \times \hat{b}_2$, $\hat{b}_2 = \hat{b}_3 \times \hat{b}_1$, and $\hat{b}_1 = \hat{b}_2 \times \hat{b}_3$. Each of these equations constrains one further degree of freedom, indicating that only three quantities are necessary to express the relative orientation of two reference frames. The most obvious parameterization for attitude is the direction cosine matrix (DCM), a 3×3 symmetric matrix with determinant 1. We notate the DCM with two capital letters, the rightmost indicating the reference frame of the input vectors and the leftmost indicating the transformed frame. Alternatively, the DCM is sometimes expressed as C when the frames involved are arbitrary or do not need to be denoted. For example, the DCM $[\mathcal{BN}]$ takes vectors in the $\mathcal{N}$ frame to the $\mathcal{B}$ frame:

$$ {}^{\mathcal{B}}\mathbf{r} = [\mathcal{BN}]\,{}^{\mathcal{N}}\mathbf{r} \tag{4.1} $$

The orthogonal property of the DCM implies $[\mathcal{BN}]^{-1} = [\mathcal{BN}]^T$ such that $[\mathcal{BN}]^T = [\mathcal{NB}]$.

Another core attitude representation is the Euler angle-axis, or principal rotation parameter, form. Euler's rotation theorem guarantees that any relative orientation can be expressed as a single rotation about an axis $\hat{\lambda} \in \mathbb{S}^2$ by an angle $\theta \in [0, 2\pi]$. The set $\left\{\hat{\lambda}, \theta\right\}$ is known as a principal rotation parameter, abbreviated PRP hereafter. The DCM is mapped to the PRP representation via 4.2 [36].

$$\theta = \cos^{-1}\left(\frac{1}{2}\left[C_{1,1} + C_{2,2} + C_{3,3} - 1\right]\right) \tag{4.2}$$
$$\hat{\lambda} = \frac{1}{2\sin\theta}\begin{bmatrix} C_{2,3} - C_{3,2} \\ C_{3,1} - C_{1,3} \\ C_{1,2} - C_{2,1} \end{bmatrix}$$

Where $C_{\mathrm{i,j}}$ refers to the ith row and jth column of C. The mapping from PRP to DCM is also relatively straightforward.

$$C = I_3 + \sin\theta\left[\hat{\lambda}\times\right] + (1 - \cos\theta)\left[\hat{\lambda}\times\right]^2 \tag{4.3}$$

Where $[v\times]$ is the matrix cross product operator, defined on $\mathbf{v} \in \mathbb{R}^3$ as:

$$[\mathbf{v}\times] = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{4.4}$$

This operator is useful as it rephrases the cross product as matrix multiplication, i.e. $\mathbf{v} \times \mathbf{u} = [\mathbf{v}\times]\,\mathbf{u}$. While the PRP $\{\theta, \hat{\lambda}\}$ is a four element set, there are only three degrees of freedom due to the unit norm constraint on $\hat{\lambda}$.

The quaternion represents attitude with a point on the surface of the hypersphere $\mathbb{S}^3$. In terms of the PRP, the quaternion is given by Eq 4.5.

$$\mathbf{q} = \begin{bmatrix} \hat{\lambda}\sin(\theta) \\ \cos(\theta) \end{bmatrix} \tag{4.5}$$

The first three entries of the quaternion are often called the vector component, with the final entry being the scalar component. Some authors reorder the quaternion, placing the scalar term first. Often the entries of a single quaternion are referenced by index such that $\mathbf{q} = [q_1, q_2, q_3, q_4]$. Similarly, we can reference the vector portion of the quaterion with $\mathbf{q}_{1:3}$. The quaternion can be mapped back to the PRP via Eqs 4.6 and **??**.

$$\begin{aligned} \theta &= \cos^{-1}(q_4) \\ \lambda &= \frac{\mathbf{q}_{1:3}}{\sin\theta} \end{aligned} \tag{4.6}$$

The quaternion maps to the DCM via Eq 4.7

$$C = \begin{bmatrix} -\ q_2^2 -\ \ q_3^2 + q_1^2 + q_4^2 & 2\ q_1 q_2 + 2\ q_3 q_4 & 2\ q_1 q_3 - 2\ q_2 q_4 \\ 2\ q_1 q_2 - 2\ q_3 q_4 & -\ q_1^2 -\ \ q_3^2 + q_2^2 + q_4^2 & 2\ q_1 q_4 + 2\ q_2 q_3 \\ 2\ q_1 q_3 + 2\ q_2 q_4 & 2\ q_2 q_3 - 2\ q_1 q_4 & -q_1^2 -\ \ q_2^2 + q_3^2 + q_4^2 \end{bmatrix} = \Xi(q)^T \Psi(q) \tag{4.7}$$

In Eq 4.7, Ψ is defined to be

$$\Psi = \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}. \tag{4.8}$$

Multiplying the Euler angle by the axis, we get an attitude representation similar to the PRP known as the rotation vector (RV), generally denoted $\mathbf{p}$.

$$\mathbf{p} = \theta\hat{\lambda} \tag{4.9}$$

The RV is the first truly three dimensional representation we have come across so far. This is advantageous for visualizing sets of orientations, but there are multiple notable issues with any three dimensional embedding of $SO(3)$. Any representation embedded in $\mathbb{R}^3$ loses

	DCM	PRP	RV	MRP
DCM	—			
PRP	$C = I_3 + \sin\theta \left[\hat{\lambda}\times\right] + (1 - \cos\theta) \left[\hat{\lambda}\times\right]^2$	—	$\mathbf{p} = \theta\hat{\lambda}$	
RV			—	
MRP				—

some of the spherical qualities of $\mathbb{S}^3$, leading to singularities — regions where attitudes are not uniquely defined or are impossible to compute in the first place.

To summarize, we can transform to and from all attitude representations with relatively simple algebraic operations:

4.2 Attitude Kinematics

Because it is cheap to convert between attitude representations, we only need to discuss a single set of kinematic equations for propagating a rigid body attitude profile from an initial condition. We choose the quaternion kinematic differential equations as they have no singularity and produce very smooth dynamics that are comparably easy to integrate. Given the current orientation quaternion $\mathbf{q}$ and angular velocity ω we can compute the quaternion derivative via Eq 4.10

$$\begin{bmatrix} \dot{\epsilon_1} \\ \dot{\epsilon_2} \\ \dot{\epsilon_3} \\ \dot{\epsilon_4} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \epsilon_4 & -\epsilon_3 & \epsilon_2 & \epsilon_1 \\ \epsilon_3 & \epsilon_4 & -\epsilon_1 & \epsilon_2 \\ -\epsilon_2 & \epsilon_1 & \epsilon_4 & \epsilon_3 \\ -\epsilon_1 & -\epsilon_2 & -\epsilon_3 & \epsilon_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} \tag{4.10}$$

4.3 Attitude Dynamics

Rigid body dynamics can be easily expressed in the body principal axes with an arbitrary torque $\mathbf{M} = [M_1, M_2, M_3]^T$ in the same frame via Eq 4.11

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} \left(M_1 + I_2\omega_2\omega_3 - I_3\omega_2\omega_3\right)/I_1 \\ \left(M_2 - I_1\omega_1\omega_3 + I_3\omega_1\omega_3\right)/I_2 \\ \left(M_3 + I_1\omega_1\omega_2 - I_2\omega_1\omega_2\right)/I_3 \end{bmatrix} \tag{4.11}$$

5. MESHES

5.1 Implicit and Explicit Shape Representations

A computer can represent 3D objects implicitly or explicitly. An implicit representation might be the solution to an algebraic equation, i.e., $x^2 + y^2 + z^2 = 1$ defines a sphere of radius 1 centered at the origin. Often, signed distance functions (SDFs) may be used to implicitly define ray traced shapes. An SDF takes in a point in $\mathbb{R}^3$ and outputs the distance from the object, returning negative distance if the queried point is inside the object. The object is then easily rendered via ray marching. A ray is cast for each pixel of the screen, each of which performs distance queries along its length eminating from the camera until it intersects the object or diverges.

By contrast, an explicit shape representation creates complex 3D geometry from simple 2D building blocks. In the simplest case, object faces — equivalently called facets — are defined by triangles. This means that at the scale of the individual facets, the shape is always composed of flat surfaces that meet at sharp angles. While this can add complexity to many fields of shape analysis and geometry processing, triangulated surfaces are perfect for our application. Human-made space objects like most satellites are mostly composed of flat faces, with the exception of parabolic antennas and cylindrical rocket bodies.

5.2 The Wavefront OBJ File Format

One common text file format for 3D model files is `.obj`, developed by Wavefront Technologies in the early 1990s [37]. Each OBJ file consists of a list of vertex positions and facet definitions, with optional vertex normals and tangents. To illustrate this, the `.obj` listing for a cube is included for reference in the appendix 11.1.3. Once the model file is loaded, we can compute a few properties that will prove useful for both light curve simulation and shape inversion. For each triangular facet of the model defined by vertices (v_1, v_2, v_3), we compute the outward-pointing facet normal with

$$\hat{n} = \frac{(v_2 - v_1) \times (v_3 - v_1)}{\| (v_2 - v_1) \times (v_3 - v_1) \|_2}. \tag{5.1}$$

The face area is computed with

$$a = \frac{\| (v_2 - v_1) \times (v_3 - v_1) \|_2}{2}. \tag{5.2}$$

The support of each face — the perpendicular distance from the origin to the plane defining the facet — is computed with

$$h = v_1 \cdot \hat{n}. \tag{5.3}$$

The volume of the entire object is compute with

$$\frac{1}{3} \sum_{\mathrm{i}=0}^{|F|} \vec{h}_{\mathrm{i}} \cdot \vec{a}_{\mathrm{i}}. \tag{5.4}$$

In Eq 5.4, $|F|$ is the number of facets defining the object. $\vec{h}$ and $\vec{a}$ are column vectors collecting all facet supports and areas. The Extended Gaussian Image, a quantity defined in 10.1.1, is computed row-wise for the ith facet with

$$\vec{E}_{\mathrm{i}} = \vec{a}_{\mathrm{i}} \vec{n}_{\mathrm{i}}. \tag{5.5}$$

5.3 Selected Space Object Model Files

Most of the analysis in this work used one of the 3D model files shown in Figure 5.1.

Figure 5.1 highlights the size of the GEO communications satellites (TELSTAR, HYLAS, Hispasat, and ASTRA). In contrast, the LEO satellites (Starlink and Landsat) are dwarfed at the left end of the lineup.

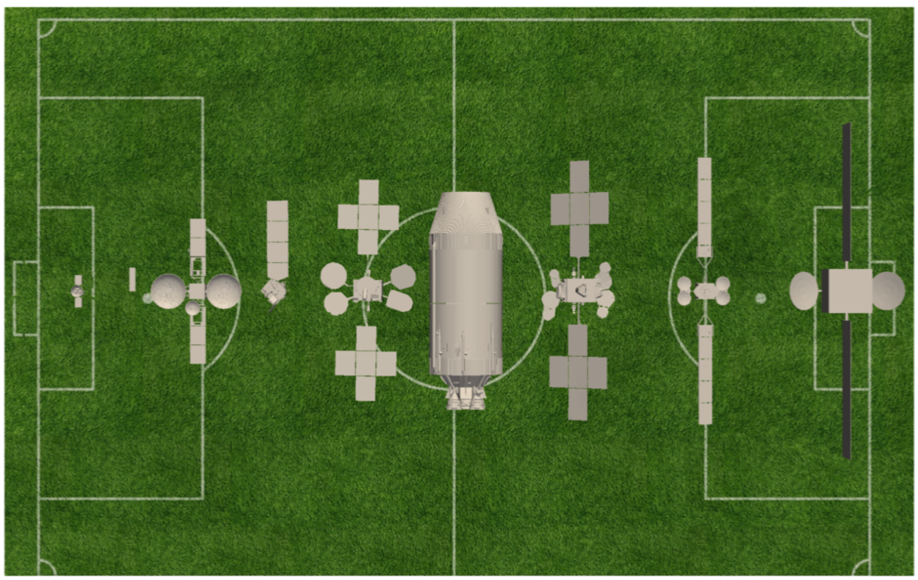

Figure 5.1. Selected space objects with soccer field for size reference. In order, the objects are TESS, Starlink V1, TDRS, Landsat 8, Hispasat 30W-6, Saturn V SII, TELSTAR 19V, HYLAS 4, and simplified ASTRA.

6. LIGHTING

6.1 The Bidirectional Reflectance Distribution Function

Although light curves come from unresolved measurements, the interactions that produce them are directly driven by the shape and material properties of the object being observed. In order to simulate accurate light curves, we must model all important optical interactions. In broad terms, this boils down to determining how the object is lit and how it is shadowed.

At the microscopic scale, the surface of an object is composed of facets — small areas sharing a normal vector. The macroscopic optical properties of the material is driven by the distribution of sizes and normal directions of the facets. If the facets are distributed in biased orientations, the macroscopic surface may show anisotropy, leading to the appearance of brushed metal. If the facets normals are at large angles to each other, the surface may appear dull as the outgoing direction of the light may be completely independent from the incoming direction. Subsurface effects — where incoming light rays scatter *inside* the surface can also change the macroscopic properties of the material.

This discussion raises an important question; how can we model the macroscopic outcomes of the true microscopic interactions of incident light on a surface? The bidirectional reflectance distribution function (BRDF) is a tool developed in computer graphics to address this exact problem. The BRDF is a function on the hemisphere which expresses the fraction of light per solid angle (radiance $\mathcal{R}$) leaving the surface in a given direction, divided by the incident power per unit area (irradiance $\mathcal{I}$). The general formulation for a BRDF f_r is given by Eq 6.1 [38].

$$f_r(\mathbf{x}, L \to O) = \frac{d\mathcal{R}\,(\mathbf{x} \to O)}{d\mathcal{I}\,(L \to \mathbf{x})}. \tag{6.1}$$

In Eq 6.1, $\mathbf{x} \in \mathbb{R}^3$ is the point on the object's surface the BRDF is evaluated at. $L \in \mathbb{S}^2$ is the incoming illumination unit vector and $O \in \mathbb{S}^2$ is the outgoing unit vector. Note that this work treats $f_r(\mathbf{x}, L \to O)$ and $f_r(L \to O)$ as equivalent in later descriptions, leaving the evaluation point $\mathbf{x}$ implied. This definition is useful for building intuition about the form of the BRDF, but to represent a physically plausible reflection process, a candidate function must satisfy three additional constraints. A physically plausible BRDF must conserve energy — more energy cannot leave the surface than was incident on it, neglective thermal effects. It must also be reciprocal — switching the observer and illumination directions should not change the BRDF value as the surface interaction. This reciprocity is sometimes known as the *Helmholtz Reciprocity Rule* in literature [39]. Finally, plausible BRDFs must take on nonnegative values for all inputs [39]. A surface cannot reflect negative light, so this should feel natural. Explicitly, energy conservation is expressed by Eq 6.2 [39].

$$\forall L \in \Omega : \int_{O \in \Omega} f_r(L \to O)\, d\Omega \leq 1 \tag{6.2}$$

Eq 6.2 states that for all possible illumination directions L on the hemisphere Ω you can integrate all possible outgoing observer directions O on the hemisphere and end up with equal or less energy than you started with. Reciprocity can also be formalized via 6.3.

$$\forall L, O \in \Omega : \;\; f_r(L \to O) = f_r(O \to L) \tag{6.3}$$

6.1.1 BRDF Formulations

The following BRDFs are all energy conserving, reciprocal, and nonnegative. *Caveat emptor*: this does not mean that they are always sufficient for modeling real-world materials, they merely represent ways hypothetical surfaces could reflect light without breaking any fundamental physics.

Lambertian

The simplest BRDF is one that reflects equally in all directions. This BRDF is termed Lambertian or diffuse.

$$f_r(L \to O) = \frac{C_d}{\pi} \tag{6.4}$$

In Eq 6.4, $0 \leq C_d \leq 1$ is the surface's coefficient of diffuse reflection. If $C_d = 0.4$, we know that the surface reflects 40% of incident radiation and absorbs the other 60%.

Phong

A simple specular BRDF model is that developed by Phong in 1975 [40]. The Phong model splits the BRDF into a Lambertian term governed by C_d and a specular term governed the coefficient of specular reflection $0 \leq C_s \leq 1$ and the specular exponent $n \geq 0$ [38].

$$f_r(L \to O) = \frac{C_d}{\pi} + \frac{C_s \frac{n+2}{2\pi}(O \cdot R)^n}{N \cdot L} \tag{6.5}$$

In Eq 6.5, R is the reflected illumination vector, computed via $R = 2(N \cdot L)N - L$. As n increases, the specular glint becomes sharper and more intense, eventually approaching a perfectly mirror reflection. Now that we have two coefficients of reflection, we must add an new constraint to maintain energy conservation. Because C_d and C_s each represent the *fraction* of light reflected in each mode, it should be clear that $C_d + C_s \leq 1$. This can also be reformulated with an explit coefficient of absorption C_a which captures the fraction of incident radiation absorbed by the surface, yielding $C_d + C_s + C_a = 1$.

Blinn-Phong

The Blinn-Phong BRDF is similar to to the Phong BRDF, but parameterizes the specular lobe in terms of the halfway vector H [38]. This vector is halfway between the illumination and observer directions such that $H = L + O$ which needs to be normalized before use. As the halfway vector approaches the surface normal vector, the observer must be approaching the reflected illumination vector, leading to a more intense specular highlight.

$$f_r(L \to O) = \frac{C_d}{\pi} + \frac{C_s \frac{n+2}{2\pi}(N \cdot H)^n}{4(N \cdot L)(N \cdot O)} \tag{6.6}$$

Glossy

Figure 6.1. Implemented BRDFs rendered with arbitrary parameters, demonstrating the qualitative differences between lighting models

7. Light Curves

A light curve is a time series of unresolved optical brightness measurements. Once an object is far enough away from the observer to become unresolved, all geometric data is lost and the only information that remains in the individual measurements is the total brightness. "Brightness" is a catch-all term for a variety of units.

7.1 Brightness Units

7.1.1 Irradiance

Irradiance is the standard SI linear unit used to describe the total amount of energy incident on a surface from a given source. An irradiance of 1 $\left[\frac{W}{m^2}\right]$ implies that a 10 $[m]$ area would experience 10 $[W]$ of incident power. The Sun's irradiance is approximately 1361 $\left[\frac{W}{m^2}\right]$ at a distance of 1 AU.

Visual magnitude — also known as apparent or relative magnitude — is a reverse logarithmic scale that originates in astronomy. Stellar sources span many orders of magnitude of brightness, making a logarithmic scale a helpful middle ground for comparison. Note that apparent magnitude always expresses brightness at the observer's location; absolute magnitude is a different quantity that normalizes brightness from a distance of 10 parsecs. In terms of irradiance, apparent magnitude is computed via Eq 7.1.

$$m = -2.5 \log_{10}\left(\frac{I}{I_0}\right) \tag{7.1}$$

In Eq 7.1, I is the irradiance of the source of interest and I_0 is irradiance of the zero-point source. This makes sense as if we plug in $I = I_0$, we are left with $m = 0$. The star Vega is usually taken to be the zero-point with irradiance $I_0 = 2.518021002 \cdot 10^{-8} \left[\frac{W}{m^2}\right]$.

We can rearrange Eq 7.1 to compute irradiance from a given apparent magnitude, yielding Eq 7.2

$$I = I_0 \cdot 10^{-\frac{m}{2.5}} \tag{7.2}$$

7.1.2 Normalized Irradiance

This work also uses normalized irradiance, the irradiance of a source if the observer was 1 meter away. This is a non-standard quantity in the literature, but proves useful for the same reasons absolute magnitude is used by astronomers. Adjusting sources to be at a standard distance allows us to simulate and invert light curves in a non-dimensionalized space, simplifying simulation and making the inversion optimizations more robust. To make the conversion explicit, irradiance observed at a distance r in meters from an object is converted to normalized irradiance $\hat{I}$ in Watts via Eq 7.3

$$\hat{I} = r^2 I \tag{7.3}$$

7.1.3 S_{10}

While magnitude and irradiance do a good job describing the flux of point sources, other units exist to talk about diffuse or extended sources where brightness is somewhat uniformly spread over an area. S_{10} is a unit of surface brightness represented by the number of 10th magnitude stars per square degree that would produce the same flux. Surface brightness in S_{10} over a given solid angle Ω $[rad^2]$ can be converted to total irradiance I $\left[\frac{W}{m^2}\right]$ via Eq 7.4.

$$\frac{I\left[\frac{W}{m^2}\right]}{S_{10}} = 10^{-10/2.5}\left(\Omega\frac{180^2}{\pi^2}\right)\int_{10^{-8}}^{10^{-6}} \mathrm{STRINT}(\lambda)\, d\lambda = 8.26617\Omega \cdot 10^{-9} \tag{7.4}$$

In 7.4, $\mathrm{STRINT}(\lambda)$ $\left[\frac{W}{m^2 \cdot m}\right]$ is the representative spectrum of a 0th magnitude star, $\mathrm{QE}(\lambda)$ is the quantum efficiency spectrum of the observing sensor, $\mathrm{ATM}(\lambda)$ is the atmospheric transmission spectrum, $\lambda\,[m]$ is wavelength, h $\left[\frac{m^2 \cdot kg}{s}\right]$ is Plank's constant, and c $\left[\frac{m}{s^2}\right]$ is the speed of light in vacuum.

7.1.4 Magnitudes per Square Arcsecond

A second surface brightness unit is $\left[\frac{mag}{arcsec^2}\right]$, also known as MPSAS (magnitudes per square arcsecond) or SQM (sky quality meter). This quantity can be thought of as a generalized S_{10}, where instead of quantifying the number of stars of a certain magnitude in a solid angle, we measure the equivalent magnitude of a single point source. A surface brightness B_{10} in S_{10} can be converted into surface brightness B_{mag} in $\left[\frac{mag}{arcsec^2}\right]$ via Eq 7.5.

$$B_{mag} = -2.5 \log_{10} \left(\frac{B_{10} \cdot 10^{-4}}{12960000} \right) \tag{7.5}$$

In Eq 7.5 we first convert S_{10} to the total irradiance per square degree, convert square degrees to square arcseconds, and transform the result back into apparent magnitude. We can convert from MPSAS to irradiance per steradian via Eq 7.6 using 7.2.

$$I = \left(\frac{180}{3600\pi} \right)^2 I_0 \cdot 10^{-\frac{MPSAS}{2.5}} \tag{7.6}$$

7.1.5 Candela

Some light pollution datasets are given in units that include candela. Candela is the SI base unit of luminous intensity defined by the International Committee for Weights and Measures as "Fixing the numerical value of the luminous efficacy of monochromatic radiation of frequency $540 \cdot 10^{12}$ Hz to be equal to exactly 683" [41]. This means that an isotropic source with frequency $540 \cdot 10^{12}$ Hz ($\lambda = 555$ nm) has a luminous efficacy of $K_{cd} = 683$ $[lm/W]$ where lm stands for lumens. Luminous efficacy itself determines how well a source produces visible light. For a given wavelength, we convert from candela B_{cd} to watts per steradian B_{wsr} via 7.7.

$$B_{wsr}(\lambda) = \frac{B_{cd}}{K_{cd}(\lambda)} \tag{7.7}$$

The luminous efficiency function $K_{cd}(\lambda)$ models the human eye's response to the visible spectrum [42]. Different fits of this function exist; we adopt that of Sharpe et al. [42], displayed in Figure 7.1.

Candela per unit area can be converted into MPSAS by combining Eq 7.7 with 7.1, yielding Eq 7.8, which is still a function of the source's wavelength.

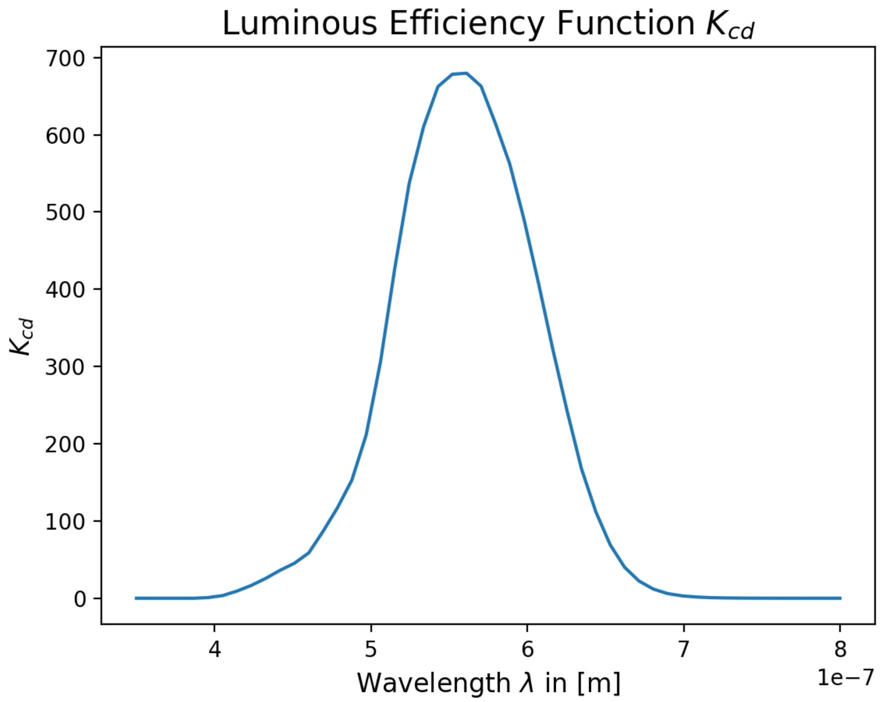

Figure 7.1. Luminous effiency function from [42]

$$MPSAS(\lambda) = -2.5\log_{10}\left(\frac{B_{cd}}{\left(\frac{180}{3600\pi}\right)^2 K_{cd}(\lambda) I_0}\right) \tag{7.8}$$

7.1.6 Photoelectron Counts

Raw observations of brightness taken by a CCD-equipped telescope are measured in photoelectron counts, otherwise known as Analog-to-Digital Units (ADU) [43]. The count in a single pixel obtained is directly proportional (via the ADU gain) to the number of photons incident on that pixel during the integration time. Higher order effects in the silicon of the CCD makes this statement incomplete at best, but for the non-resolved imaging applications we're concerned about, chip-level innacuracies besides readout noise and dark current are often neglected [1]. Irradiance can be converted back and forth to ADU via the conversion factor $SINT$ in Eq 7.9 [43].

$$\text{SINT} = \frac{\pi D^2}{4} \int_{10^{-8}}^{10^{-6}} \left(\frac{\text{SUN}(\lambda)}{I_{sun}} \right) \cdot \text{QE}(\lambda) \cdot \text{ATM}(\lambda) \cdot \left(\frac{\lambda}{hc} \right) \, d\lambda \tag{7.9}$$

In Eq 7.9, SUN(λ) is the spectrum of solar irradiance in $\left[\frac{W}{m^2 \cdot m}\right]$, I_{sun} is the irradiance of the Sun (generally taken to be the solar constant 1361 $\left[\frac{W}{m^2}\right]$. Read literally, the integral term as units $\left[\frac{1}{Ws}\right]$, giving the number of counts per incident Watt of solar radiation and second of integration time. The aperture diameter factor outside the imtegral accounts for the area of light incident on the CCD, giving SINT units of $\left[\frac{m^2}{Ws}\right]$. Multiplying by irradiance in $\left[\frac{W}{m^2}\right]$ and an integration time Δt in seconds will yield the count of photoelectrons S in ADU as shown in Eq 7.10.

$$S = \text{SINT} \cdot I \cdot \Delta t \tag{7.10}$$

For completeness, irradiance can be recovered from a signal in ADU and the integration time via Eq 7.11.

$$I = \frac{S}{\text{SINT} \cdot \Delta t} \tag{7.11}$$

7.2 Astronomical Spectra

Four of the quantities needed for the background model vary with wavelength. These are the atmospheric transmission, the sensor quantum efficiency, the irradiance of a 0th magnitude star, and the solar spectrum. Atmospheric transmission is a unitless quantity conveying the fraction of light that is not absorbed by the atmosphere. Quantum efficiency is a unitless quantity which conveys the fraction of incident photons which are (proportionally) converted to photoelectrons in the CCD sensor. Each spectrum is displayed in Figure 7.2.

In practice, the quantum efficiency curve varies by sensor and the thermal conditions of the observation. The curve adopted in this work is that used by Krag; modern sensors will often perform better.

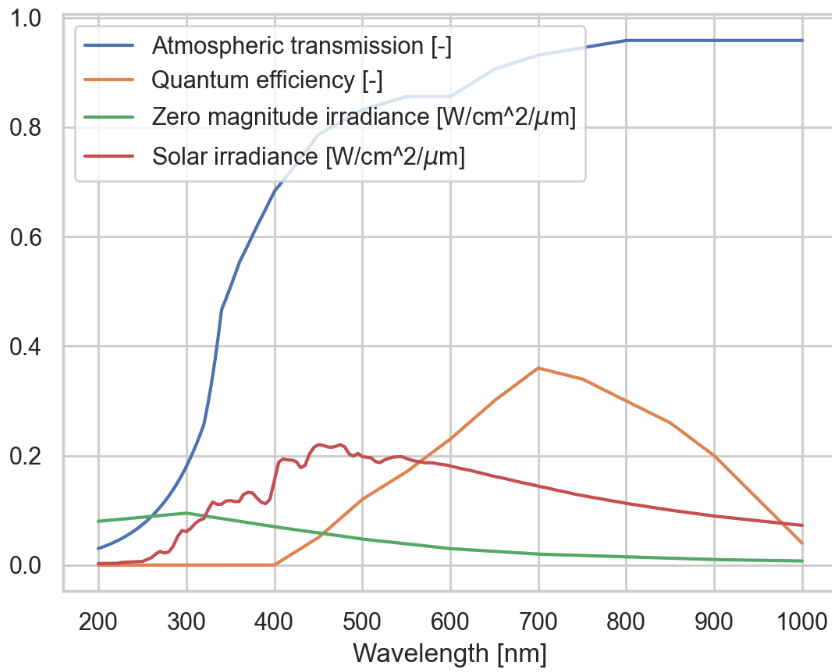

Figure 7.2. Astronomical Spectra

8. Background Signals

Whenever an optical telescope is observing a space object, the object's signal is necessarily superimposed on whatever signals exist in the background. In this context, background does not only refer to sources physically further than the object, but all sources that impact the image apart from the object signal. As we will see, some of these sources even originate within the telescope. To faithfully simulate a telescope observing an object, many position-based SDA tasks are able to ignore background effects while acquiring or tracking objects. For photometry-based SDA, the background is critical. Background signals can be broken up into atmospheric effects, exoatmospheric effects, and sensor effects.

8.1 Atmospheric Effects

8.1.1 Airglow

Certain chemical reactions from 80-110 km altitude in the upper atmosphere release visible light [43]. This effect is known as airglow. Since these reactions are assumed to be isotropic — equally intense when integrated along any vertical line extending upwards from the surface. We model the airglow signal AINT in a similar fashion to integrated starlight. Given the airglow spectra GLINT(λ) $\left[\frac{W}{m^2 \cdot m \cdot rad^2}\right]$, we compute Eq 8.1.

$$\text{AINT} = \frac{\pi D^2}{4} \int_{10^{-8}}^{10^{-6}} \text{GLINT}(\lambda) \cdot \text{QE}(\lambda) \cdot \text{ATM}(\lambda) \cdot \left(\frac{\lambda}{hc}\right) \, d\lambda \tag{8.1}$$

The quantity AINT has units $\left[\frac{1}{s \cdot rad^2}\right]$, meaning that the mean airglow signal in ADU per pixel is simply given by Eq 8.2

$$\bar{S}_{airglow} = AINT \cdot \frac{1}{\cos(\theta_z)} \cdot \Delta t \cdot \left(\frac{\pi s_{pix}}{648000}\right)^2 \tag{8.2}$$

In Eq 8.2, $\frac{1}{\cos(\theta_z)}$ is known as the Van Rhijn factor, which accounts for the accumulation of airmass near the horizon [1].

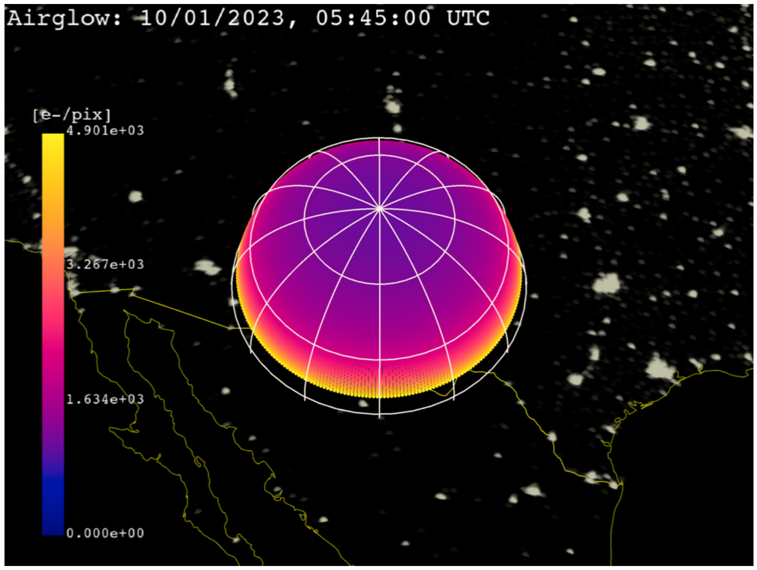

Figure 8.1. Mean airglow signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, −105.533° W

8.1.2 Light Pollution

The final source of background noise light pollution. On a cloudless night with negligible light pollution, the zenith surface brightness is approximately 22 $\left[\frac{mag}{arcsec^2}\right]$ (MPSAS) [43]. As light pollution increases, this zenith brightness may dip down to $14-15$ $\left[\frac{mag}{arcsec^2}\right]$. To get accurate localized zenith brightness values, we use the 2015 World Atlas of Sky Brightness dataset [44]. The data is reported in $\left[\frac{mcd}{cm^2}\right]$ on a 30-arcsecond grid, requiring us to convert to MPSAS. A subset of the global dataset is displayed in 8.2 This conversion is listed in Eq 7.8, using a monochromatic $\lambda = 474$ nm to fit Falchi et al.'s example conversions [45].

The mean light pollution CCD signal in ADU per pixel is formulated similarly to airglow. Given the station's zenith surface brightness $B_{poll,z}$ in MPSAS, we first convert to irradiance per steradian via 7.6, which we then input into 8.3 to compute the mean signal in ADU per pixel. Note that Krag does not implement a specific light pollution model, but instead takes the dark sky site zenith brightness of 22 MPSAS as an input to an atmospherically scattered light model. Here we simply use that model's formulation with a variable zenith brightness.

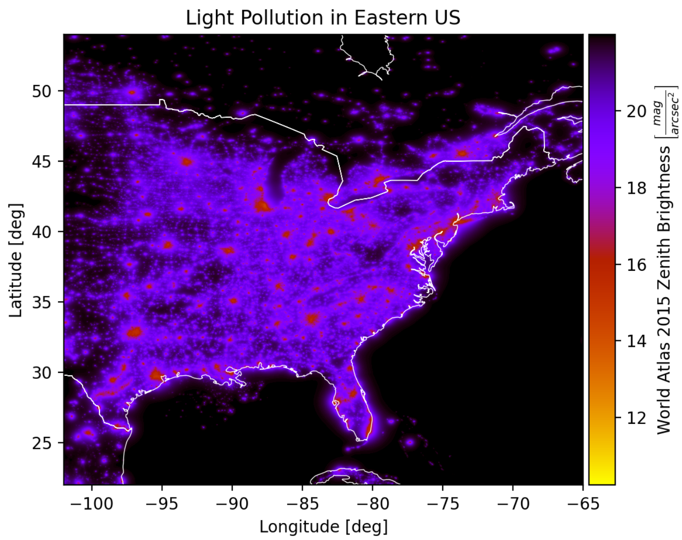

Figure 8.2. Zenith light pollution in the eastern USA, data from [44]

$$\bar{S}_{pollution} = B_{poll,z} \cdot SINT \cdot \frac{1}{\cos(\theta_z)} \cdot \Delta t \cdot \left(\frac{\pi s_{\text{pix}}}{648000}\right)^2 \tag{8.3}$$

8.1.3 Twilight

Even after the Sun sets, scattered sunlight in the upper atmosphere creates a signal on our CCD. The twilight model implemented for this work is due to Patat et al. and was developed for the European Southern Observatory at Paranal in Chile [46]. This model implements the zenith brightness as a function of the solar zenith angle γ — the angle from zenith to the Sun's apparent centroid. Patat et al.'s model fits a second-degree polynomial in γ to approximately 2000 observations, yielding separate curves for each of the UBVRI passbands. For example, for the V band, the twilight zenith brightness in MPSAS is given by 8.4 [46].

$$B_{tw\text{i},z} = 11.84 + 1.518(\gamma - 95°) - 0.057(\gamma - 95°)^2 \tag{8.4}$$

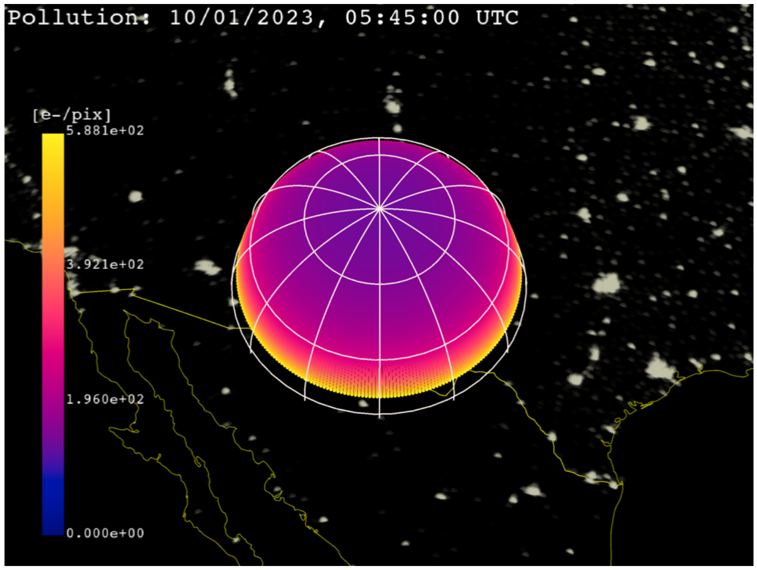

Figure 8.3. Mean light pollution signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, −105.533° W

Eq 8.4 is valid from $95° \leq \gamma \leq 105°$. Before $\gamma 95°$, we take the zenith brightness to be constant and equal to the brightness at $\gamma 95°$. This is not accurate, but is sufficiently bright to correctly forbid daytime observations by lowering the SNR drastically. After $\gamma = 105°$ we set the zenith surface brightness to $B_{twi,z} == 22$ MPSAS to match the optimal observation condition of the light pollution model [43]. Zenith twilight brightness is plotted as a function of γ in Figure 8.4.

Computing the mean CCD signal in ADU per pixel due to the twilight brightness proceeds identically to the light pollution formulation.

$$\bar{S}_{twilight} = B_{twi,z} \cdot SINT \cdot \frac{1}{\cos(\theta_z)} \cdot \Delta t \cdot \left(\frac{\pi s_{pix}}{648000}\right)^2 \tag{8.5}$$

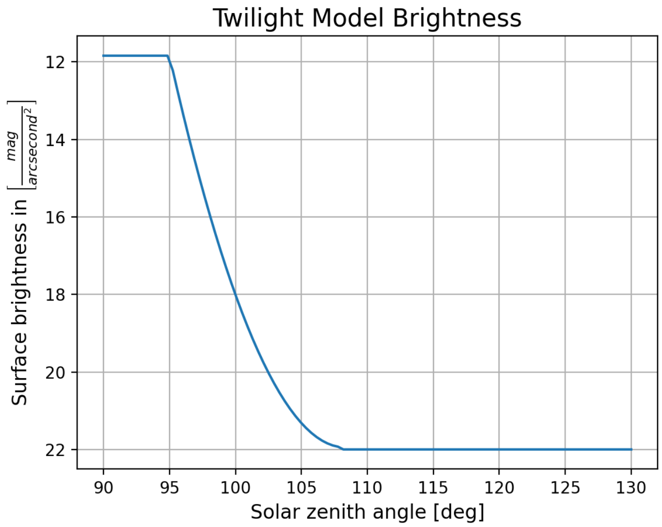

Figure 8.4. Twilight model surface brightness at zenith as a function of solar zenith angle

8.2 Exoatmospheric Effects

8.2.1 Integrated Starlight

Stars are almost always present in optical images of space objects. The brightest stars streaking across the field of view in Figure 8.5 have high SNRs and stand out clearly against the dark background. This raises a question: if we're observing a full $1° \times 1°$ area of the sky, where are the rest of the stars given that the Milky Way alone contains approximately $1 \cdot 10^{11}$ stars? The answer is relatively obvious: many more stars are present in the image than we can pick out individually, most of them fall into the background. We call this residual faint starlight "integrated" starlight as we are effectively integrating the signals from thousands or millions of stars across the image plane.

In Figure 8.6, most stars are too faint to appear as points of light on the image plane. Instead, they merge into the background. The signal due to these faint stars is known as integrated starlight. Krag [43] modeled this signal by building a $1° \times 1°$ grid of surface brightness values for the full right ascension (RA) and declination (Dec) sphere. Krag used the Guide Star catalog, which contains 15 million stars down to magnitude 16. Exponential

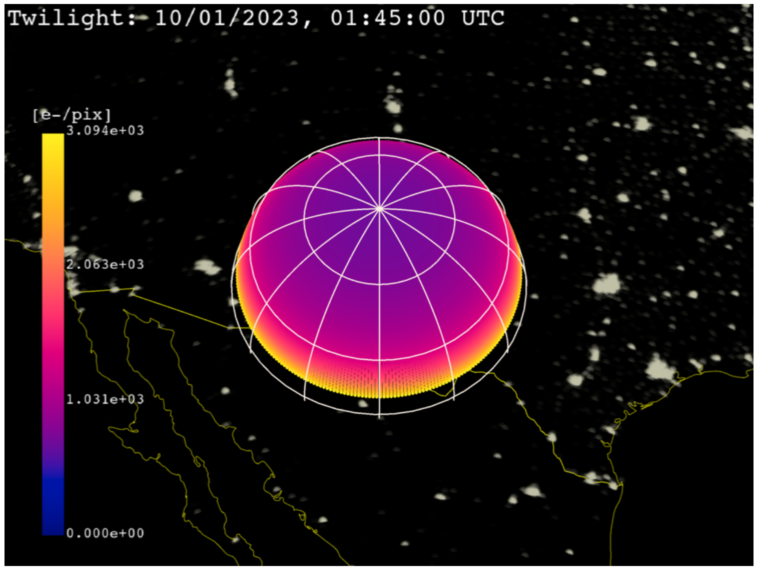

Figure 8.5. Mean twilight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, −105.533° W

extrapolation was used to predict star counts in each bin for higher magnitudes [43]. Twenty years later, we have access to larger star catalogs that are nearly complete to much dimmer magnitudes. The integrated starlight catalog used in this work was built from the GAIA catalog with approximately 1.5 billion stars down to magnitude 21-22 [47]. The same $1° \times 1°$ grid was computed using the `astroquery.gaia` Python package [48]. Figure 8.7 shows the computed patched catalog, in units of S_{10}.

Now that we have a data source for the exoatmospheric mean brightness of the night sky due to integrated starlight, we can compute the corresponding signal mean for a telescope equipped with a CCD sensor. Again, we adopt Krag's formulation [43].

$$\mathrm{BINT} = \frac{\pi D^2}{4} \int_{10^{-8}}^{10^{-6}} \mathrm{STRINT}(\lambda) \cdot \mathrm{QE}(\lambda) \cdot \mathrm{ATM}(\lambda) \cdot \left(\frac{\lambda}{hc}\right) d\lambda \tag{8.6}$$

In Eq 8.6, D is the telescope aperture diameter in meters, h is Plank's constant in $\left[\frac{m^2 kg}{s}\right]$, and c is the speed of light in vacuum in $\left[\frac{m}{s}\right]$. The resulting quantity BINT has units of $\left[\frac{1}{s}\right]$, representing the mean total photons passing through the telescope aperture due to integrated starlight.

Figure 8.6. Raw image of three GEO objects with stars streaking through the background. Taken by the Purdue Optical Ground station at 32.900° N, −105.533° W by Nathan Houtz

$$\bar{S}_{star} = 10^{-4} \cdot BINT \cdot \left(\frac{s_{\mathrm{pix}}}{3600}\right)^2 \cdot \Delta t \cdot b_{cat} \tag{8.7}$$

In Eq 8.7, b_{cat} is the patched catalog brightness in $[S_{10}]$, s_{pix} is the telescope pixel scale in $\left[\frac{arcsecond}{pix}\right]$, and Δt is the integration time in seconds. Note the addition of the 10^{-4} factor to reconcile catalog surface brightness in terms of 10th magnitude stars, and the 0th magnitude source in BINT. This yields $\bar{S}_{star}$ with units $\left[\frac{\mathrm{e}^-}{pix^2}\right]$; photoelectron counts (ADU) per pixel. Figure 8.8 shows the background signal mean due to integrated starlight.

8.2.2 Scattered Moonlight

Moonlight scattering through the atmosphere significant increases background brightness [43]. This scattering effect can be decomposed into Rayleigh (isotropically distributed) and Mie (exponentially distributed) scattering modes. The Rayleigh scattered component is computed with Table 4 published by Daniels parameterized by the angle from the observation

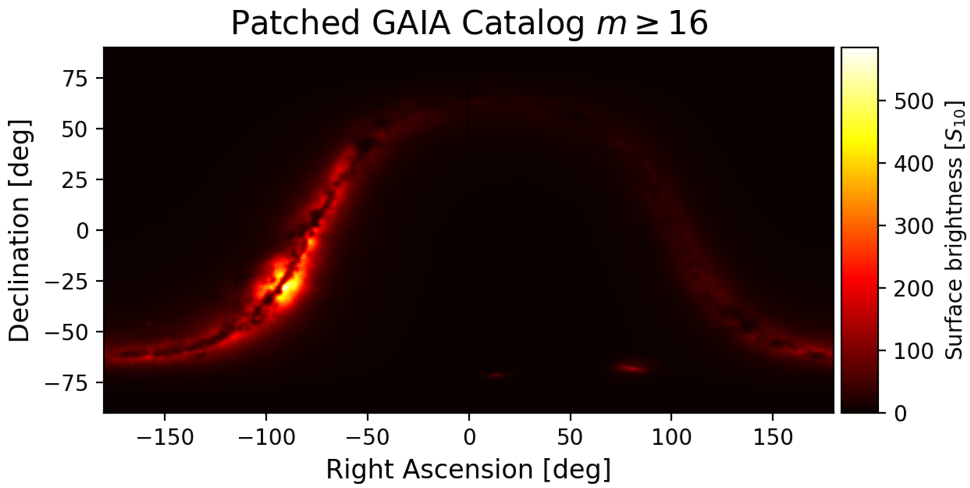

Figure 8.7. Integrated starlight patched catalog

to zenith z_{obs}, the angle from the Moon to zenith z_{moon}, and the angle between the observation and the Moon on the horizon ΔAz [49]. Interpolating this table yields the intensity of the Rayleigh scattering F_{rs} in $10^{-10}\ W/(cm^2 \cdot \mu m \cdot sr)$ [43]. The Mie scattered component is formulated with Eq 8.8.

$$F_{ms}(\lambda) = a_1 \left[\mathrm{e}^{-\left(\frac{\Psi}{\Psi_1}\right)} + a_2 \mathrm{e}^{-\left(\frac{\pi-\Psi}{\Psi_2}\right)} \right] F_{rs}(\lambda) \tag{8.8}$$

Daniels recommends $a_1 \in [50, 100]$, $a_2 \in [0.01, 0.02]$, $\Psi_1 \in [10°, 20°]$, and $\Psi_2 \approx 50$ [49]. Prior to any station-specific fitting, we choose the middle of these intervals, yielding $a_1 = 75$, $a_2 = 0.015$, $\Psi_1 = 15°$, and $\Psi_2 = 50°$. a_1 and a_2 are dimensionless, such that F_{ms} also has units of $10^{-10}\ W/(cm^2 \cdot \mu m \cdot sr)$, allowing us to compute the total intensity of the scattered moonlight F_{mt} via Eq 8.9 following Krag's formulation [43].

$$F_{mt} = f(\theta) \left[F_{rs}(\lambda) + F_{ms}(\lambda) \right] \tag{8.9}$$

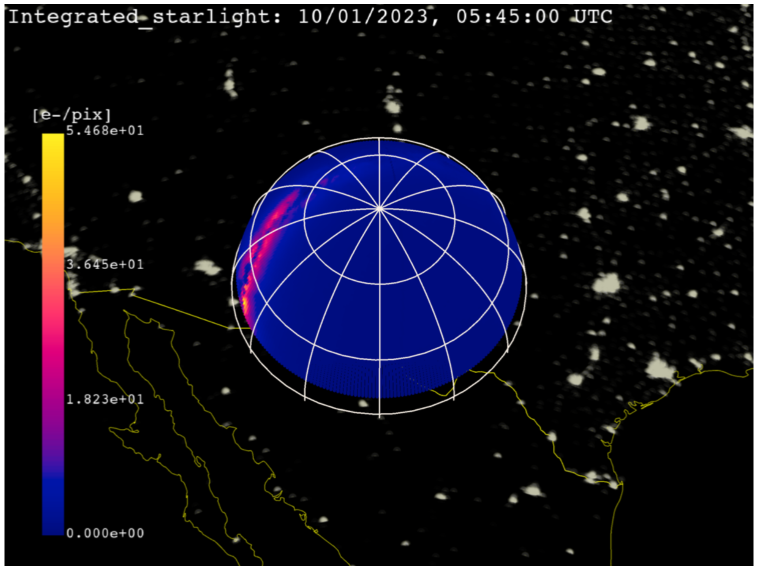

Figure 8.8. Integrated starlight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, −105.533° W

in Eq 8.9, $f(\theta)$ is the lunar phase function which describes the fraction of the full Moon brightness is reflected at an observer viewing the Moon an angle θ from the Sun vector. This function is linearly interpolated within Table 3 in [49]. Finally, Krag introduces a correction factor f_{corr} to account for the difference between the Sun's irradiance spectrum and the spectrum of scattered moonlight, defined in Eq 8.10.

$$f_{corr} = \frac{I_0}{SUN(550\ [\mathrm{nm}])} \tag{8.10}$$

With all these pieces, we can put together the mean scattered moonlight signal in ADU per pixel in Eq 8.11.

$$\bar{S}_{moon} = F_{mt}(550\ [\mathrm{nm}]) \cdot SINT \cdot \left(\frac{s_{\mathrm{pix}}}{3600}\right)^2 \cdot \Delta t \cdot f_{corr} \tag{8.11}$$

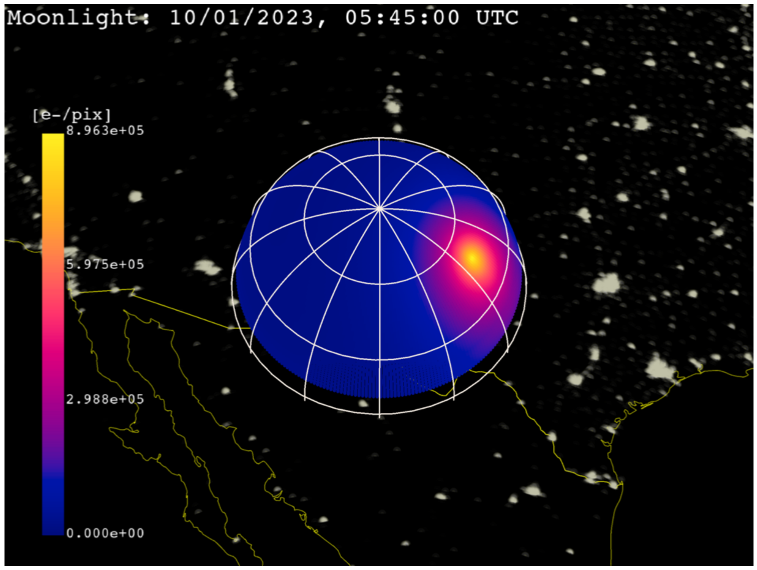

Figure 8.9. Mean scattered moonlight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, −105.533° W

8.2.3 Zodiacal Light

Zodiacal light is an effect created by sunlight reflecting off of dust in the ecliptic plane [43]. Zodiacal light is strongest around the Sun — an area that is not of interets for us — but also reaches a peak directly away from the Sun due to the opposition effect. This peak is known as the Gegenschein, meaning "opposing light". We compute the of the zodiacal light via Table 1 of [50]. This reports the surface brightness of the zodiacal light in S_{10}, which we use without conversion to find the mean CCD signal in ADU per pixel via Eq 8.12.

$$\bar{S}_{zod} = BINT \cdot \left(\frac{s_{\text{pix}}}{3600}\right)^2 \cdot \Delta t \cdot ZOD \cdot 10^{-4} \tag{8.12}$$

As in the integrated starlight signal, the 10^{-4} factor reconciles the S_{10} surface brightness with the 0th magnitude source in BINT.

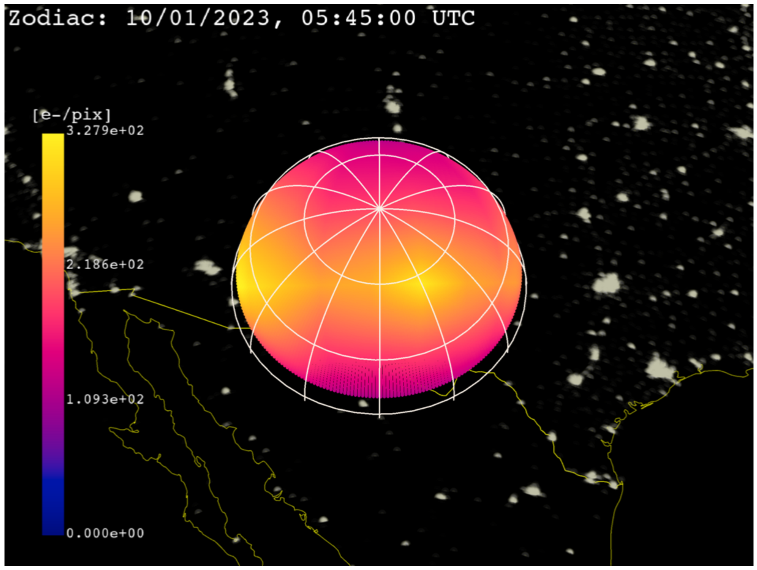

Figure 8.10. Mean zodiacal light signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, −105.533° W

8.2.4 Sampling Background

Notice that each background signal is only defined in terms of its mean. On a pixel-by-pixel basis, the signal for an exposure is sampled from a Poisson distribution for each background term. This distribution can be interpreted as modeling the number of independent events that occur during a time period. In our case, this translates to individual photons being incident on our sensor. A Poisson distribution is defined on the positive integers by a single parameter λ which is both the mean and variance of the distribution. The probability density function (PDF) for the Poisson distribution takes the form of Eq 8.13 [1].

$$P_\lambda(x = k) = \frac{\lambda^k \mathrm{e}^{-\lambda}}{k!} \tag{8.13}$$

This distribution has a useful property that $P_{\lambda_1+\lambda_2}(x = k) = P_{\lambda_1}(x = k) + P_{\lambda_2}(x = k)$ so long as the distributions described by λ_1 and λ_2 are independent. Since our background sources are assumed to be independent as sources like moonlight and zodiacal light are clearly distinct; if the Moon vanished, interplanetary dust across the solar system would reflect light

Signal source	Magnitude [e^-/pix]
Airglow	$10^3 - 10^4$
Scattered moonlight	$0 - 10^5$
Integrated starlight	$10^1 - 10^2$
Light pollution	$10^2 - 10^3$
Zodiacal light	$10^2 - 10^4$
Twilight	$10^1 - 10^7$

Table 8.1. Background signal importance

identically. This means that we can formulate the total background signal as a single Poisson variable.

$$\lambda_{background} = \bar{S}_{airglow} + \bar{S}_{pollution} + \bar{S}_{twilight} + \bar{S}_{star} + \bar{S}_{moon} + \bar{S}_{zod} \tag{8.14}$$

To compute the background of the CCD image, we simply sample from the Poisson distribution defined by $\lambda_{background}$.

8.2.5 Background Source Importance

Some background signals are more impactful than others. Table ?? details the approximate magnitudes in photoelectrons per pixel one can expect from a telescope similar to the Purdue Optical Ground Station.

8.3 Sensor Effects

8.3.1 Dark Noise

8.3.2 Readout Noise

9. Light Curve Simulation

9.1 Simulating Convex Objects

Light curve simulation for convex geometry can be solved semi-analytically as each facet's contribution to the measured irradiance can be computed individually [6]. Determining whether a face is illuminated requires two horizon checks to determine visibility from the Sun and to the observer. For a facet i at timestep j these horizon checks are expressed by the shadowing condition μ_{ij}.

$$\mu_{\mathrm{ij}} = \begin{cases} 1 \text{ if } \left(\hat{O}_{\mathrm{j}} \cdot \hat{n}_{\mathrm{i}}\right) > 0 \text{ and } \left(\hat{S}_{\mathrm{j}} \cdot \hat{n}_{\mathrm{i}}\right) > 0 \text{ and } \delta_{\mathrm{ij,ss}} = 0 \text{ and } \delta_{\mathrm{ij,os}} = 0 \\ 0 \text{ otherwise} \end{cases} \tag{9.1}$$

The unit vectors $\hat{O}$ and $\hat{S}$ point from the center of mass of the object to the observer and Sun, respectively. We choose the outward-pointing facet normal unit vector $\hat{n}$ by convention for all mesh operations. The self-shadowing and observer-shadowing conditions, $\delta_{\mathrm{ij,ss}}$ and $\delta_{\mathrm{ij,os}}$, are always zero for convex polyhedra but are crucial for accurately simulating non-convex geometry. For objects with concavities, self-shadowing refers to shadows cast by an object onto itself and observer-shadowing refers to otherwise visible faces blocked by other portions of the geometry.

The irradiance I received by the observer at timestep j is the sum of the received irradiance from all facets, composed of specular and diffuse contributions. We express each contribution as the product of the normalized irradiance $\hat{I}$. This can be scaled to adjust for the distance from the observer to the object to yield the noiseless received irradiance.

9.2 Simulating Non-Convex Objects

Many existing light curve simulation methods for non-convex objects rely on ray tracing schemes like Möller and Trumbore's ray-triangle intersection algorithm [2], [51]. This computation is necessarily complex as there may be significant self-shadowing at large phase angles. As a result, we cannot assume $\delta_{\mathrm{ij,ss}} = 0$ and $\delta_{\mathrm{ij,os}} = 0$ **freuh2014**, [2]. In the absense of a bounding volume hierarchy or other techniques to reduce the number of rays cast, ray traced shadows generally require $\mathcal{O}(n^2)$ ray-triangle intersections per timestep for n facets. For this reason, ray traced shadows quickly become infeasible for complex reference geometries without GPU parallelization. The limitations of ray-triangle intersections for light curve simulation is discussed at length by Frueh et al in **freuh2014**.

For faster and more accurate simulated light curves, we use shadow mapping computed on the GPU. Shadow mapping is a well understood, if dated, technique in computer graphics [52]. Although modern ray traced shadowing may be more computationally efficient, shadow mapping was selected for its ease of implementation, once the inherent aliasing or 'shadow acne' was addressed using standard remedies [52]. Because shadow mapping shades individual pixel fragments instead of entire facets, it offers increasing shadow quality over facetwise ray tracing as the number of mesh faces falls.

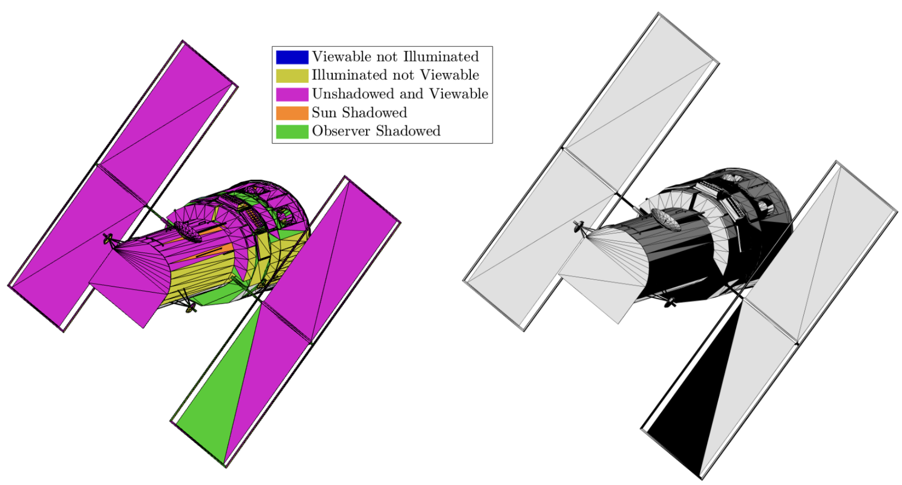

Figure 9.1. Hubble Space Telescope ray traced shadow categorization and shading. Models from [53]

9.2.1 Importance of Self-Shadowing for Human-Made Objects

To motivate the importance of accurate shadowing computation for human-made space objects, consider the error introduced by neglecting shadows for different types of space objects. Kaasalainen and Torppa's work on asteroids reasonably assumed that shadowing was a negligible contribution to the measured light curve. Human-made objects do not afford the same luxury. Figure 9.2 displays light curves for the asteroid Bennu and the Hubble Space Telescope with and without accurate shadows under a single-axis attitude profile with inertially fixed Sun and observer vectors. Without accurate shadowing, the light curve's magnitude and time derivatives are both dramatically effected.

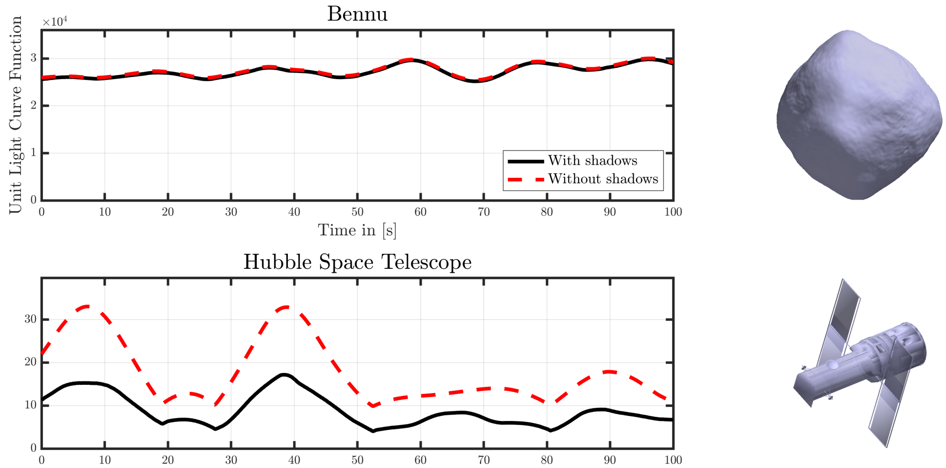

Figure 9.2. Brightness errors introduced by neglecting shadows for Bennu and HST. Models from [53]

9.2.2 Shadow Mapping for Light Curve Simulation

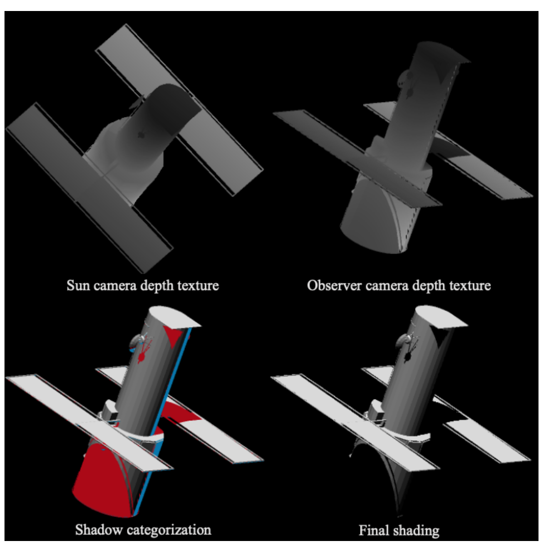

Figure 9.3. Hubble Space Telescope shadow mapping with self (red) and horizon (blue) shadows rendered. Models from [53]

Given an observer and Sun vector in the body frame of the object, shadow mapping proceeds in a four step process. In step one, a camera is positioned along the Sun vector and a perpendicular depth texture is computed. In the second step, depth values in Sun camera space are transformed to observer camera space, forming a second depth texture. This second texture is used to find the closest fragment along each ray to the Sun, determining the self-shadowing condition [54]. Self-shadowed fragments are classified as those further from the Sun than the closest fragment along the same ray, indicated in red in Figure 9.3. Fragments that do not pass the convex shadowing condition are horizon shadowed, indicated in blue in Figure 9.3. All remaining fragments are shaded with using the same Lambertian reflection model in **??**. Computing the light curve function for the final rendered image requires summing all pixel values and dimensionalizing the result by the area of the observer camera's field of view. The light curve simulation environment used in this work was implemented in C and OpenGL using raylib [55].

10. Light Curve Shape Inversion

10.1 Direct Inversion

Traditionally, direct light curve inversion involves two distinct steps: a linear least squares problem to fit an EGI to the measured light curve, and a second optimization to produce accurate vertex positions and face adjacency information [2]. The first step uses data-driven estimation to yield a valid and accurate EGI through a linear optimization. The second step is highly nonlinear and scales badly with facet count and geometric asymmetry, as discussed by Fan in [3].

10.1.1 The Extended Gaussian Image

The discrete EGI $\vec{E} \in \mathbb{R}^{m\times 3}$ is composed of m unit vectors $\hat{n}$ each scaled a nonnegative scalar $a \in \mathbb{R}$, $a_{\mathrm{i}} \geq 0$ [56].

$$\vec{E}_{\mathrm{i}} = a_{\mathrm{i}}\hat{n}_{\mathrm{i}} \tag{10.1}$$

In the context of shape inversion, we want to choose the m vectors $\hat{n}$ to be a uniform tessellation on the unit sphere. A convex polytope can be uniquely represented by an EGI of facet normal vectors scaled by each facet's area. The set of normal vectors in an EGI is denoted $\mathcal{N}$ with the set of areas denoted $\mathcal{A}$. The vector of facet areas is denoted $\vec{a} \in \mathbb{R}^{m\times 1}$. In this paper, we take the norm of the EGI to be $\|\vec{E}\| = \vec{a}$ with the 'size' of the EGI $\|\vec{E}\| = m$.

The solution to the Minkowski problem proves the existence and uniqueness of a convex polytope for an EGI that satisfies the closure condition in Eq. 10.2 [57]. Equivalently, an EGI uniquely represents a closed, convex polyhedron — a polytope — with no open boundaries, up to a translation.

$$\sum_{\mathrm{i}=1}^{m} a_{\mathrm{i}}\hat{n}_{\mathrm{i}} = [0, 0, 0] \tag{10.2}$$

While a given EGI uniquely represents a polytope, it is shared by an infinite number of non-convex and open geometries. An example of this extended family is depicted in Figure 10.1 where larger circles indicate greater relative areas assigned to a given normal vector.

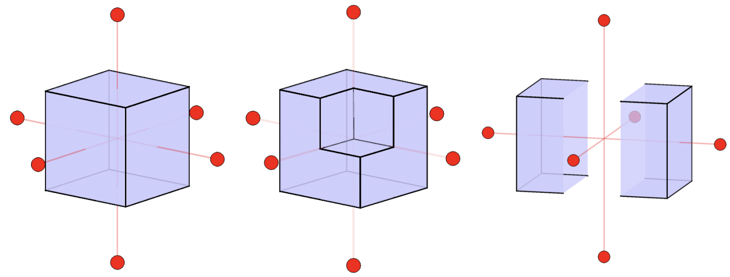

Figure 10.1. Simplified convex, non-convex, and open EGI nonuniqueness

10.1.2 EGI Optimization

The EGI fulfills two important criteria when applied to light curve inversion: it can be estimated directly from the light curve, attitude profile, and material properties, and uniquely represents a convex object [6]. Furthermore, convex geometry can be reconstructed from the EGI and vice versa through the dual transform [33] and the Minkowski problem [57].

Once a light curve is obtained, direct shape inversion schemes sample m candidate normal vectors $\hat{n}$ on the unit sphere to fit an EGI to the observed light curve $\vec{L}_{\text{ref}} \in \mathbb{R}^{n\times 1}$ [2], [8]. This is accomplished by solving an optimization problem to distribute the area vector $\vec{a}$ across the sampled normals to minimize the residual between the observed and modeled light curves. In practice, this is a constrained nonnegative least squares problem and can be solved efficiently for large numbers of normal vectors and light curve data points:

$$\min_{a} \|\vec{L}_{\text{ref}} - G\vec{a}\|_2 \quad \text{subject to } \vec{a}_{\text{i}} \geq 0, \sum_{\text{i}=1}^{m} \vec{a}_{\text{i}}\hat{n}_{\text{i}} = [0,0,0]. \tag{10.3}$$

It is important to note that the area estimated with Eq. 10.3 is necessarily *albedo-area* due to the diffuse reflectivity coefficient C_d in Eq. **??**. If the value of C_d is uniform but unknown, the recovered geometry will incorrectly scaled without impacting the face adjacency or relative feature sizes.

The convex reflection matrix $G \in \mathbb{R}^{n\times m}$ with ijth entries $[g]_{\text{ij}}$ defined at time i for each facet j is defined as the normalized received facet irradiance per unit facet area:

$$[g]_{\text{ij}} = \frac{I_{\text{ij}}}{I_{\text{Sun}} a_{\text{j}}}. \tag{10.4}$$

This relationship between the object irradiance and area defines the normalized convex light curve $\vec{L}_{\text{convex}}$, that produced by a convex object of facet areas $\vec{a}$ under the attitude profile and lighting conditions that yield G.

$$\vec{L}_{\text{convex}} = G\vec{a} \tag{10.5}$$

10.1.3 EGI Optimization Results

The optimization in Eq. 10.3 produces a coarse approximation of the true EGI as m is finite. Increasing m necessarily improves the quality and sparsity of the estimated EGI, but at the cost of computational resources. The estimation was performed using a synthetic light curve input from $n = 500$ Sun and observer vectors uniformly sampled on the sphere in the body frame, producing a full rank G matrix. $m = 500$ candidate normal vectors were sampled using a spherical Fibbonaci mapping described by Keinert et al. in [58]. Results are visualized for an icosahedron in the body frame in Figure 10.2. Reconstructing the object at this stage is difficult due to the quantity of faces present in the estimated EGI.

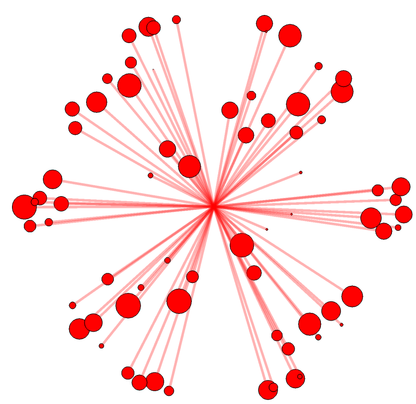

Figure 10.2. Initial icosahedron EGI optimization before resampling

10.1.4 EGI Resampling

We propose a normal vector resampling step to promote a more accurate and sparse EGI. The normal vectors used in Eq. 10.2 are generally correct, with each group clustering around a normal vector of the truth geometry. This clustering behavior occurs when none of the candidate normal vectors are sufficiently close to the truth. Resampling in a cone centered on each initial EGI normal vector provides more accurate candidates for EGI estimation. This process mimics a single optimization step with a much larger m, where the coarse EGI is used to exclude areas on the sphere with little or no normal area.

Uniformly sampling a cone of half-angle ϕ is accomplished by strategically sampling points on the unit sphere.

$$\hat{n}_{cone} = \begin{bmatrix} \sqrt{1-z^2}\cos\theta \\ \sqrt{1-z^2}\sin\theta \\ z \end{bmatrix} \tag{10.6}$$

In Eq. 10.6 we choose two coordinates $z \in [\cos\phi, 1]$ and $\theta \in [0, 2\pi)$, yielding a point uniformly distributed on a cone of half-angle ϕ about the central axis $[0, 0, 1]^T$ [59]. These points are then rotated using a direction cosine matrix to center the cone on an axis of interest.

The number of new candidates sampled per initial solution vector and the cone half-angle should be adjusted on a case-by-case basis depending on the compute power available and light curve data quality.

10.1.5 EGI Resampling Results

Existing EGI optimization schemes like those of Fan [2], Friedman [8], and Cabrera [13] are limited by a single normal vector sampling step, leading to a lack of accuracy and sparsity in the optimized EGI. High-density normal vector sampling in regions known to contain non-zero area leads to EGI solutions that are generally more sparse and cluster more tightly about true normal vectors.

This process is shown in Figure 10.3 for the same icosahedron with a half-angle $\phi = \frac{\pi}{20}$ and sampling density of 50 candidate vectors per cone.

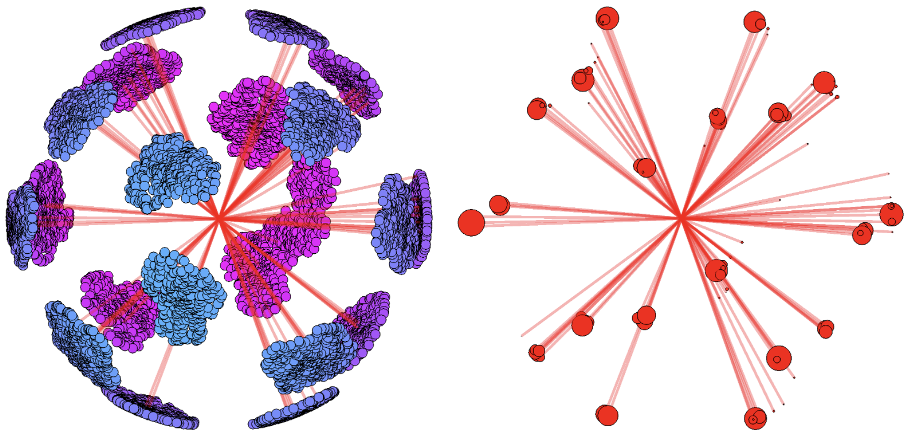

Figure 10.3. Resampled normal vectors (left) with reoptimized EGI (right)

10.1.6 EGI Merging

After resampling and reoptimizing with Eq. 10.3, the reestimated EGI is merged by computing all groups $\mathcal{G}$ of EGI vectors within an angular offset α:

$$\mathcal{G}_k = \left\{ \vec{E}_{\mathrm{i}} \in \vec{E} \parallel \cos^{-1}\left(\frac{\hat{E}_{\mathrm{i}} \cdot \hat{E}_k}{\|\vec{E}_{\mathrm{i}}\| \|\vec{E}_k\|}\right) < \alpha \right\}. \tag{10.7}$$

Groups are merged by summing all group members, yielding a single EGI vector $\vec{E}_m$ without loss of total area or closure.

$$\vec{E}_m = \sum_{\vec{E} \in \mathcal{G}_k} \vec{E} \tag{10.8}$$

In practice, the choice of α is dependent on the user's tolerance for discretization, as merging will approximate smooth geometry by discrete faces with normal vectors offset by 2α.

10.1.7 EGI Merging Results

Merging the resampled EGI using Figure 10.3 with $\alpha = \frac{\pi}{10}$ produces a final sparse EGI fit for object reconstruction, shown in Figure 10.4.

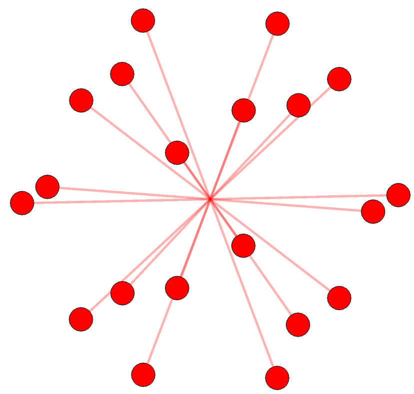

Figure 10.4. Merged icosahedron EGI

10.1.8 Convex Geometry Recovery from the EGI

At this stage, we have recovered a sparse EGI representing a convex approximation of the underlying object with no guarantee of the closure of this EGI. The EGI closure constraint Eq. 10.2 motivates a simple procedure to correct an invalid EGI by adding the mean closure error to each entry:

$$\vec{E}_{\text{closed}} = \vec{E}_{\text{open}} - \sum_{\text{i}=1}^{m} a_{\text{i}} \hat{n}_{\text{i}}. \tag{10.9}$$

This step is not a novel contribution. Fan used a more complex optimization problem to adjust the EGI towards closure [2]. We improve on that process with a simpler analytical correction. In practice, this process should be performed before each reconstruction to accelerate convergence. Failing to correct non-closed EGIs will cause convergence to a nonzero minimum in the reconstruction objective function as there is no corresponding convex object with the given EGI.

The unique convex object encoded by each closed EGI is reconstructed by solving for the polytope's set of vertices $\mathcal{V}$ and faces $\mathcal{F}$ encoding the adjacency relations between vertices.

This is accomplished following the procedure introduced by Little in [56] through the dual transformation. The dual set $\mathcal{D}$ are vertices in $(A, B, C) \in \mathbb{R}^3$ that satisfy the following plane condition for a point (x, y, z) on each facet of the object:

$$Ax + By + Cz + 1 = 0 \tag{10.10}$$

If (x, y, z) are chosen to be the closest points in the object's planes to the origin, the dual set $\mathcal{D}$ can be expressed in terms of the EGI and a support vector $\vec{h} \in \mathbb{R}^{\|F\| \times 1}$. The support vector is the perpendicular distance of each facet defining the object to the origin.

$$\mathcal{D} = \frac{\vec{E}}{\|\vec{E}\|\vec{h}} \tag{10.11}$$

The object's vertices $\vec{v}_{ref}$ are found by computing the convex hull of dual set vertices. Triplet of vertices on the resulting faces are used to find a single real vertex by intersecting the three planes defining the dual set vertices.

$$\begin{bmatrix} v_{ref,x} \\ v_{ref,Y} \\ v_{ref,Z} \end{bmatrix} = \begin{bmatrix} v_{\mathrm{i},x} & v_{\mathrm{j},x} & v_{k,x} \\ v_{\mathrm{i},y} & v_{\mathrm{j},y} & v_{k,y} \\ v_{\mathrm{i},z} & v_{\mathrm{j},z} & v_{k,z} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{10.12}$$

Convex face adjacency information is found by triangulating the convex hull of all reference vertices. The accuracy of the recovered geometry is entirely dependent on the correctness support vector $\vec{h}$ used to produce the dual set. Finding the true support vector is the challenge of the final optimization in convex shape inversion.

10.1.9 Support Vector Optimization

Prior work by Fan used Little's objective function for support vector optimization [2], [56].

$$f(\vec{h})_{\text{Little}} = \vec{h} \cdot \vec{a} \tag{10.13}$$

TODO

10.2 Non-Convex Feature Inversion

10.2.1 Non-Convex Feature Detection and Location

Many human-made space objects are, as highlighted in Figure **??**, highly non-convex. As a result, their shape inversion is plagued by the fact that the Minkowski problem-driven reconstruction methods of Eq **??** cannot recover non-convex features. Instead of beginning from the ground up, we can leverage the convex shape guess to detect and locate concavities, if they are present.

We can retain information about large, unilateral object concavities during EGI estimation in Eq. 10.3 by relaxing the EGI closure constraint. This unconstrained form is also generally functional for most convex objects and can be used without loss of detail in the final reconstruction as long as closure correction in Eq. 10.8 is still employed.

By measuring the divergence of the optimized EGI from a closed object with the magnitude of the closure error $\vec{\mathrm{e}}_{EGI}$, we can determine the mean axis of prominent concave features.

$$\vec{\mathrm{e}}_{EGI} = -\sum_{\mathrm{i}=1}^{m} a_{\mathrm{i}} \hat{n}_{\mathrm{i}}. \tag{10.14}$$

This EGI closure error vector represents the missing area on each body axis that could be added to close the object. The addition of the minus sign transforms the vector from expressing the presence of excess area to the absence of missing area. The closure error will be negligable if there are no concavities present. The closure error may also be negligable if there is no self-shadowing is present over the sampled attitude profile, therefore the closure error merely quantifies the self-shadowing that is occuring, not whether there may be self-shadowing in other orientations.

Under the strong assumption that the concavities present are major and unilateral, this EGI error vector points along the mean axis of the concavity.

After locating the concave feature through the direction of the EGI error vector, the magnitude of the same vector is used to recover a more accurate non-convex guess for the object geometry. We have found that the ℓ^2-norm of the EGI error vector — the total missing area required to close the object — scales quadratically with object scale. A quadratic relationship is expected as geometrically scaling vertex positions by a factor c increases object area by a factor c^2, leading to an identical light curve and estimated EGI with c^2 as much area assigned to each facet, scaling $\vec{e}_{EGI}$ identically.

It can also be shown that for simple, unilateral concave features, the internal angle ψ between shadowing faces scales linearly with the quantity $\sqrt{\frac{\|\vec{e}_{EGI}\|}{\|\vec{E}_m\|}}$ where $\vec{E}_m$ is the estimated EGI after merging. We can estimate the internal angle as a function of the optimized EGI, its error, and an unknown slope c.

$$\psi = \boldsymbol{\pi} - c\sqrt{\frac{\|\vec{e}_{EGI}\|}{\|\vec{E}_m\|}} \tag{10.15}$$

This relationship is displayed in Figure **??** for two different object geometries, each with prominent and unilateral concavities.

10.2.2 Concavity Location Results

The slope c has been shown through simulation to be independent of object geometry, with $c = 8$ being highly accurate for a majority of simulated objects with large and unilateral concavities, resulting from an interaction between the shadowing geometry and the least squares EGI optimization. This solution is not unique [30], and is best interpreted as finding the simplest non-convex object that would result in the same EGI closure error vector.

10.2.3 Concavity Creation

Our process for creating an accurate concavity in the reconstructed convex guess proceeds in four major steps. The model is first subdivided to add more faces and vertices. Subdivided vertices are then classified by their proximity to the EGI error vector, indicating whether their positions should be updated. Boundary vertices are identified, and vertex positions are updated based on the estimated internal angle computed via Eq. 10.15.

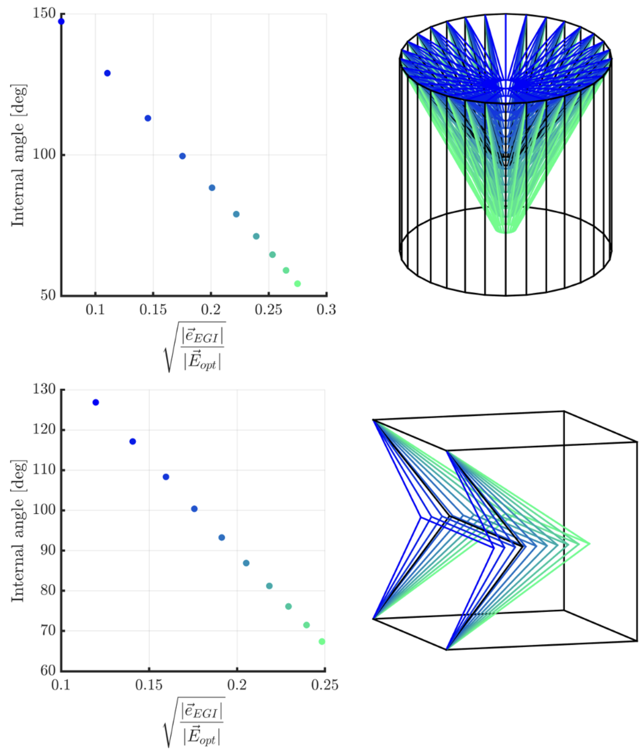

Figure 10.5. Concave cylinder and house EGI error relationship to internal angle

Model Subdivision

Subdividing the initial convex object guess is essential for retaining object detail during concavity creation. We use a combination of geometry processing algorithms for linear subdivision, Loop subdivision, and remeshing. Linear subdivision is advantageous when object faces are equally sized and boundary edges must be maintained. Loop subdivision is preferable when there are numerous vertices so that subdivisions do not drastically diverge from the initial boundary surface. Loop subdivision softens sharp edges as it relies on B-splines to interpolate new vertex positions [60]. The specific type and resolution of subdivision used depends on the level of detail the user needs to maintain in the introduced concavity, although linear subdivision followed by Loop subdivision is a useful baseline. Varying combinations of subdivision are shown in Figure 10.6 to illustrate the available configurations.

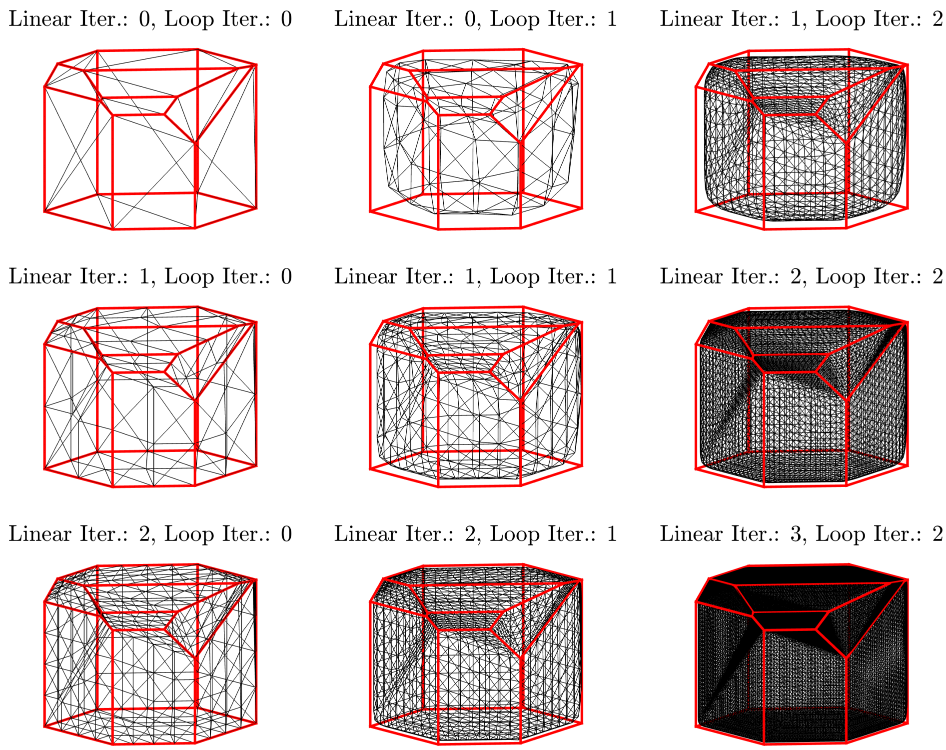

Figure 10.6. Subdivided object (black) with reference (red) with various levels of subdivision

Vertex Classification

When introducing a concavity, it is important to classify which vertices are part of the concave feature — and therefore need to be updated — and which vertices should remain unaffected. This is accomplished by measuring the angle from each face normal to the EGI error vector, where faces with normal vectors within an angle of $\pi/2$ to the error vector must be updated. In reality, all face normals and areas are impacted by the presence of the concavity in the area optimization Eq. 10.3 and EGI correction step Eq. 10.9. We select the angle deflect $\pi/2$ to update all faces above the horizon from the EGI error vector, a bound which tends to produce visually accurate concavities. Faces requiring an update are termed *free* faces, with all others termed *root* faces.

Vertex Displacement

For all vertices on free faces, we can further distinguish *root-adjacent* and *free* vertices. Root-adjacent vertices are part of at least one root face, whereas free vertices belong to only free faces. Classifying vertices in this way results in a border of root-adjacent vertices around the interior free vertices, visualized in Figure 10.7.

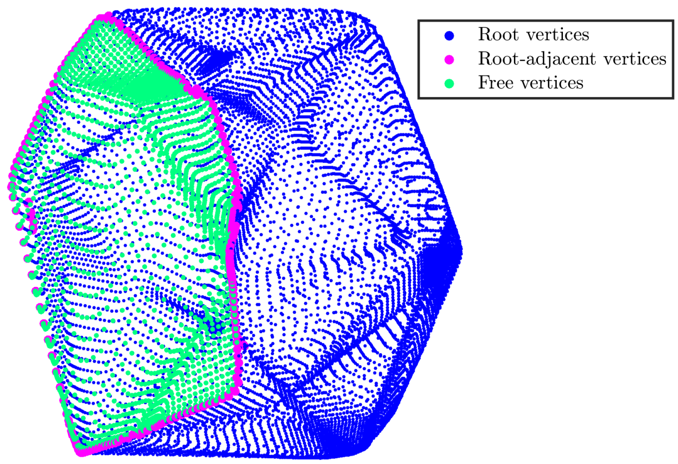

Figure 10.7. Root-adjacent and free vertices

Given the estimated internal angle ψ_{est} and the error vector $\hat{\mathrm{e}}_{EGI}$, each ith free vertex is displaced to introduce a geometrically accurate concavity by moving each a distance d_{i} in the direction of $-\hat{\mathrm{e}}_{EGI}$:

$$d_{\mathrm{i}} = p_{\mathrm{i}}\sqrt{\csc^2\frac{\psi_{\mathrm{est}}}{2} - 1}, \tag{10.16}$$

where p_{i} is the distance from each ith free vertex to the nearest root-adjacent vertex.

10.3 Non-Convex Object Reconstruction Results

Displacing free vertices in the EGI error vector direction by d_{i} yields accurate concavities for objects whose concave boundaries lie in a plane. The result of applying this process to a set of representative convex objects is shown in Figure 10.8 using the same attitude profiles and as in Figure **??**.

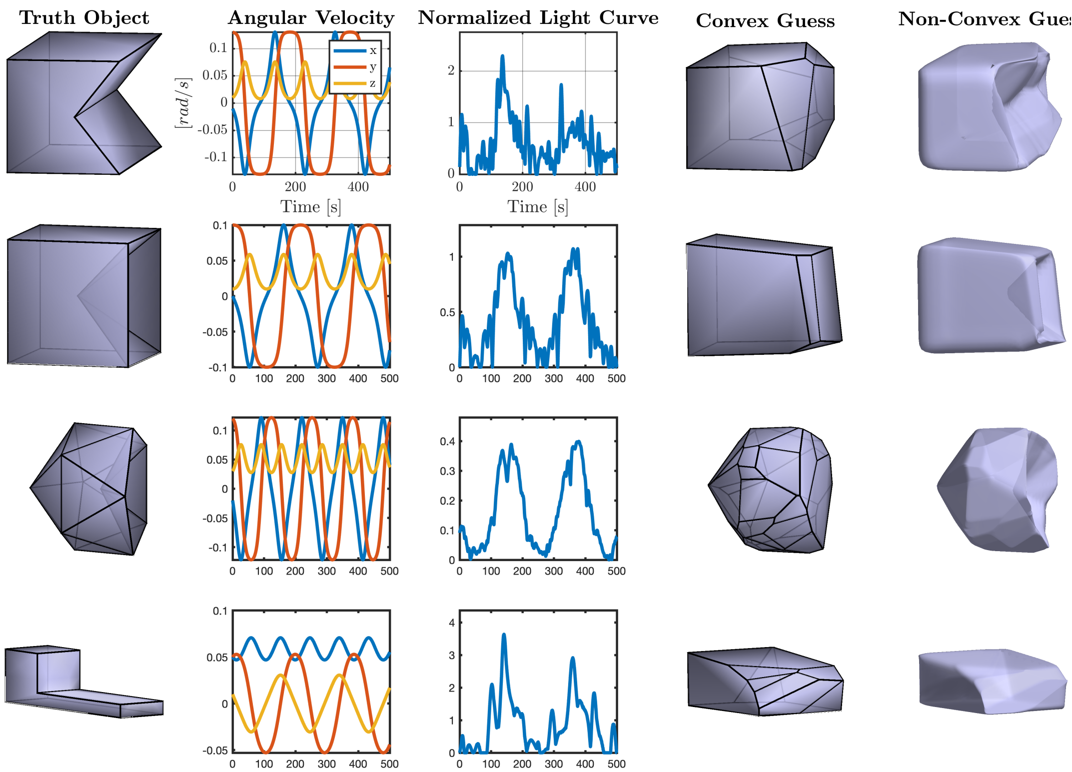

Figure 10.8. Collapsed house, cube, icosahedron, and box-wing satellite reconstructions using vertex displacement

The collapsed cube and icosahedron in Figure 10.8 are recovered effectively, but the collapsed house and box-wing satellite expose two limitations of the vertex displacement technique. In the case of the house where the concavity boundary is not constrained to a plane, the edges of the created concave feature are incorrect. The box-wing satellite's shadowing geometry leads the convex guess to be a poor approximation of the geometry outside of the concavity while also inheriting the same problem as the house.

This vertex displacement scheme will negligibly impact the convex guess if the truth object is also convex. A convex truth object will produce a small $\|\vec{e}_{EGI}\|$, causing the vertex update depth d_i to trend towards zero as the estimated internal angle approaches $\psi = 180°$. This is illustrated in Figure 10.9 using the same input convex objects and attitude profiles as in Figure **??**.

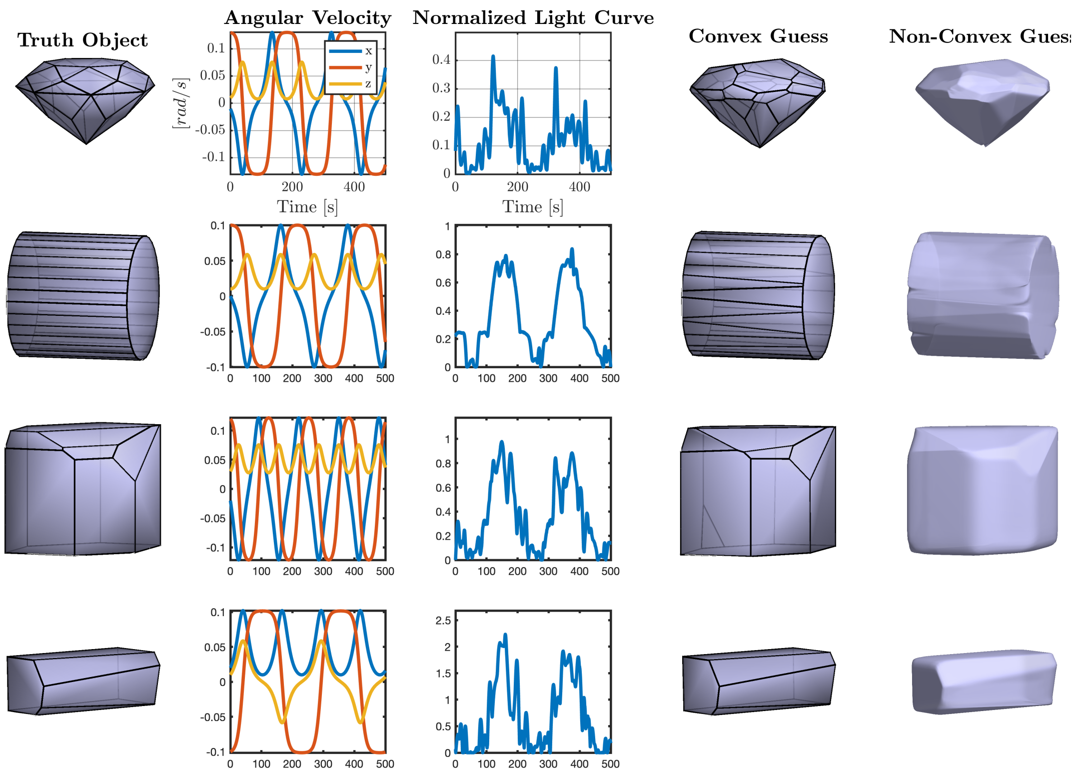

Figure 10.9. Convex objects under vertex displacement procedure

Figure 10.9 clearly displays the compatibility of vertex displacement with truly convex objects. All objects are reconstructed faithfully in both their convex and non-convex inversions, with the same caveats noted in the discussion following Figure **??**. Some truly sharp edges are rounded during mesh subdivision as seen in the gem or rectangular prism. That said, others like the cylinder become more accurate as subdivision reintroduces continuity lost to discretization in EGI merging.

11. Appendices

11.1 Astronomical Spectra Data

Atmospheric Extinction

{"lambda": [0.0, 3.2e-07, 3.4000000000000003e-07, 3.6e-07, 3.7999999999999996e-07, 4e-07, 4.5e-07, 5e-07, 5.5e-07, 6e-07, 6.5e-07, 7e-07, 8e-07, 0.001], "extinction": [5.0, 0.96, 0.54, 0.42, 0.34, 0.27, 0.17, 0.13, 0.11, 0.11, 0.07, 0.05, 0.03, 0.0]}

Quantum Efficiency

{"lambda": [4.0000000000000003e-07, 4.5000000000000003e-07, 5.000000000000001e-07, 5.5e-07, 6.000000000000001e-07, 6.5e-07, 7.000000000000001e-07, 7.5e-07, 8.000000000000001e-07, 8.5e-07, 9.000000000000001e-07, 9.5e-07, 1.0000000000000002e-06, 1.0500000000000001e-06], "quantum_efficiency": [0.0, 0.05, 0.12, 0.17, 0.23, 0.3, 0.36, 0.34, 0.3, 0.26, 0.2, 0.12, 0.04, 0.0]}

11.1.1 Background Source Data

Lunar Phase Factor

{"phase_factor": [1.00, 0.809, 0.685, 0.483, 0.377, 0.288, 0.225, 0.172, 0.127, 0.089, 0.061, 0.041, 0.077, 0.017, 0.009, 0.004, 0.001, 0.0, 0.0], "phase_angle": [0, 0.17453293, 0.34906585, 0.52359878, 0.6981317 , 0.87266463, 1.04719755, 1.22173048, 1.3962634 , 1.57079633, 1.74532925, 1.91986218, 2.0943951 , 2.26892803, 2.44346095, 2.61799388, 2.7925268 , 2.96705973, 3.14159265]}

Scattered Moonlight

{"z_obs": [0.0, 0.17453292519943295, 0.3490658503988659, 0.5235987755982988, 0.6981317007977318, 0.8726646259971648, 1.0471975511965976, 1.2217304763960306, 1.3962634015954636], "delta_az": [0.0, 0.7853981633974483, 1.5707963267948966, 2.356194490192345, 3.141592653589793], "z_moon": [0.0, 0.5235987755982988, 1.0471975511965976, 1.3089969389957472], "radiance": [[[22.0, 19.0, 13.0, 10.0], [22.0, 19.0, 13.0, 10.0], [22.0, 19.0, 13.0, 10.0], [22.0, 19.0, 13.0, 10.0], [22.0, 19.0, 13.0, 10.0]], [[22.0, 21.0, 15.0, 11.0], [22.0, 20.0, 14.0, 11.0], [22.0, 19.0, 13.0, 10.0], [22.0, 18.0, 12.0, 9.7], [22.0, 18.0, 12.0, 9.6]], [[22.0, 23.0, 18.0, 13.0], [22.0, 22.0, 16.0, 12.0], [22.0, 19.0, 14.0, 10.0], [22.0, 17.0, 12.0, 9.9], [22.0, 17.0, 12.0, 10.0]], [[22.0, 25.0, 21.0, 16.0], [22.0, 23.0, 18.0, 14.0], [22.0, 20.0, 14.0, 11.0], [22.0, 17.0, 12.0, 11.0], [22.0, 16.0, 12.0, 11.0]], [[23.0, 28.0, 25.0, 20.0], [23.0, 25.0, 21.0, 17.0], [23.0, 21.0, 16.0, 12.0], [23.0, 17.0, 14.0, 13.0], [23.0, 16.0, 14.0, 14.0]], [[24.0, 31.0, 31.0, 25.0], [24.0, 28.0, 26.0, 20.0], [24.0, 22.0, 18.0, 15.0], [24.0, 18.0, 17.0, 16.0], [24.0, 18.0, 18.0, 18.0]], [[27.0, 37.0, 39.0, 33.0], [27.0, 33.0, 32.0, 26.0], [27.0, 25.0, 22.0, 18.0], [27.0, 22.0, 22.0, 21.0], [27.0, 22.0, 25.0, 26.0]], [[34.0, 47.0, 54.0, 48.0], [34.0, 41.0, 43.0, 37.0], [34.0, 33.0, 29.0, 25.0], [34.0, 30.0, 33.0, 32.0], [34.0, 31.0, 40.0, 40.0]], [[55.0, 72.0, 89.0, 82.0], [55.0, 65.0, 71.0, 63.0], [55.0, 54.0, 50.0, 43.0], [55.0, 54.0, 61.0, 58.0], [58.0, 58.0, 76.0, 75.0]]]}

Zodiacal Light

{"ecliptic_lat": [0.0, 0.17453292519943295, 0.3490658503988659, 0.5235987755982988, 0.6981317007977318, 0.8726646259971648, 1.0471975511965976, 1.2217304763960306, 1.3962634015954636], "ecliptic_lon": [3.141592653589793, 2.792526803190927, 2.443460952792061, 2.0943951023931953, 1.7453292519943295, 1.3962634015954636, 1.1344640137963142, 1.0471975511965976, 0.9599310885968813, 0.8726646259971648, 0.7853981633974483, 0.6981317007977318, 0.6108652381980153, 0.5235987755982988, 0.4363323129985824, 0.3490658503988659, 0.2617993877991494, 0.17453292519943295, 0.08726646259971647, 0.0], "brightness": [[258.0, 211.0, 206.99999999999997, 239.0, 277.0, 365.0, 535.0, 630.0, 756.0, 939.0, 1190.0, 1490.0, 2010.0000000000002, 2940.0, 4660.0, 7690.000000000001, 15100.0, 36500.0, 176000.0, 163000000.0], [212.0, 194.0, 185.0, 217.0, 247.00000000000003, 312.0, 418.0, 455.0, 512.0, 603.0, 696.0, 825.0, 1150.0, 1550.0, 1820.0, 2140.0, 2760.0, 2720.0, 5630.0, 19900.0], [183.0, 174.0, 168.0, 196.0, 220.00000000000003, 258.0, 330.0, 339.0, 358.0, 403.0, 442.0, 512.0, 635.0, 800.0, 932.0, 1070.0, 1120.0, 1390.0, 1700.0, 2290.0], [159.0, 153.0, 152.0, 177.0, 196.0, 219.0, 258.0, 270.0, 282.0, 290.0, 304.0, 331.0, 363.0, 417.0, 491.0, 542.0, 592.0, 655.0, 724.0, 794.0], [141.0, 137.0, 137.0, 161.0, 175.0, 190.0, 204.0, 212.0, 229.0, 227.0, 233.0, 240.0, 224.00000000000003, 241.0, 246.0, 252.0, 265.0, 290.0, 315.0, 403.0], [127.0, 127.0, 128.0, 146.0, 156.0, 166.0, 165.0, 166.0, 183.0, 185.0, 189.0, 186.0, 171.0, 180.0, 183.0, 186.0, 190.0, 199.0, 209.0, 252.0], [117.0, 120.0, 120.0, 132.0, 139.0, 146.0, 137.0, 137.0, 147.0, 149.0, 150.0, 149.0, 137.0, 141.0, 144.0, 145.0, 145.0, 145.0, 146.0, 150.0], [110.00000000000001, 112.00000000000001, 112.00000000000001, 120.0, 123.0, 127.0, 118.0, 120.0, 124.0, 124.0, 124.0, 126.0, 118.0, 120.0, 121.0, 121.0, 121.0, 121.0, 121.0, 121.0], [103.0, 105.0, 105.0, 108.0, 111.00000000000001, 111.00000000000001, 106.0, 107.0, 107.0, 108.0, 107.0, 111.00000000000001, 107.0, 106.0, 108.0, 108.0, 108.0, 108.0, 108.0, 108.0]]}

Parameter	Value
FWHM	1.5
Sensor dimensions	0.03690×0.03690 $[m]$
f number	7.2
Aperture diameter	0.35560 $[m]$
Secondary diameter	0.1724660 $[m]$
Sensor pixels	4096×4096
Pixel size	$9.009 \cdot 10^{-6}$ $[m/\text{pix}]$
Pixel scale	0.72545 $[arcsec]$
Field of view	$0.824425° \times 0.824425°$
Integration time	1 $[s]$

Table 11.1. Purdue Optical Ground Station telescope parameters

11.1.2 Telescope Parameters

Purdue Optical Ground Station

11.1.3 File Formats

Wavefront OBJ Example

```
# Blender v2.92.0 OBJ File: 'cube.obj'
# www.blender.org
mtllib cube.mtl
o Cube_Cube.003
v 1.0 1.0 -1.0
v 1.0 1.0 1.0
v 1.0 -1.0 -1.0
v 1.0 -1.0 1.0
v -1.0 1.0 -1.0
v -1.0 1.0 1.0
v -1.0 -1.0 -1.0
v -1.0 -1.0 1.0
vn 1.0 0.0 0.0
vn 0.0 -1.0 0.0
vn -1.0 0.0 0.0
vn 0.0 1.0 0.0
vn 0.0 0.0 -1.0
vn 0.0 0.0 1.0
usemtl None
s off
f 2/1/1 3/2/1 1/3/1
f 4/4/2 7/5/2 3/2/2
f 8/6/3 5/7/3 7/5/3
f 6/8/4 1/9/4 5/7/4
f 7/5/5 1/10/5 3/11/5
f 4/12/6 6/8/6 8/6/6
f 2/1/1 4/4/1 3/2/1
f 4/4/2 8/6/2 7/5/2
f 8/6/3 6/8/3 5/7/3
f 6/8/4 2/13/4 1/9/4
f 7/5/5 5/7/5 1/10/5
f 4/12/6 2/14/6 6/8/6
```

REFERENCES

[1] C. Frueh, *Space traffic management*, Lecture Notes, 2019.

[2] S. Fan, "The light curve simulation and its inversion problem for human-made space objects," Ph.D. dissertation, Purdue University, Aug. 2020. DOI: 10.25394/PGS.12749570.v1.

[3] S. Fan and C. Frueh, "A direct light curve inversion scheme in the presence of measurement noise," *The Journal of the Astronautical Sciences*, vol. 67, Aug. 2019. DOI: 10.1007/s40295-019-00190-3.

[4] J. Allworth, L. Windrim, J. Wardman, D. Kucharski, J. Bennett, and M. Bryson, "Development of a high fidelity simulator for generalised photometric based space object classification using machine learning," in *Proceedings of 70th International Astronautical Congress*, International Astronautical Congress, 2019, pp. 1–14. DOI: 10.48550/ARXIV.2004.12270. [Online]. Available: https://arxiv.org/abs/2004.12270.

[5] L. Robinson and C. Frueh, "Light curve inversion for reliable shape reconstruction of human-made space objects," in *Proceedings of the 32nd AIAA/AAS Astrodynamics Specialist Conference*, Sep. 2022, pp. 1–19.

[6] M. Kaasalainen, J. Torppa, and K. Muinonen, "Optimization methods for asteroid lightcurve inversion: Ii. the complete inverse problem," *Icarus*, vol. 153, no. 1, pp. 37–51, 2001, ISSN: 0019-1035. DOI: https://doi.org/10.1006/icar.2001.6674. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019103501966746.

[7] S. Fan, C. Frueh, and A. Buzzoni, "A light curve simulation of the apollo lunar ascent module," in *AIAA/AAS Astrodynamics Specialist Conference*, 2016, pp. 1–10. DOI: 10.2514/6.2016-5504.

[8] A. M. Friedman, "Observability analysis for space situational awareness," Ph.D. dissertation, Purdue University, Apr. 2020. DOI: 10.25394/PGS.12196863.v1.

[9] D. Kobayashi and C. Frueh, "Compressed sensing for satellite characterization," in *Proceedings of the 20th AAS/AIAA Astrodynamics Specialist Conference*, Aug. 2020, pp. 1–20.

[10] C. Früh and M. K. Jah, "Coupled orbit attitude motion of high area-to-mass ratio (hamr) objects including efficient self-shadowing," *Acta Astronautica*, vol. 95, pp. 227–241, 2014, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2013.11.017.

[11] J. Allworth, L. Windrim, J. Bennett, and M. Bryson, "A transfer learning approach to space debris classification using observational light curve data," *Acta Astronautica*, vol. 181, pp. 301–315, Jan. 2021. DOI: 10.1016/j.actaastro.2021.01.048.

[12] R. Furfaro, R. Linares, and V. Reddy, "Shape Identification of Space Objects via Light Curve Inversion Using Deep Learning Models," in *Proceedings of the 20th Advanced Maui Optical and Space Surveillance Technologies Conference*, S. Ryan, Ed., Sep. 2019, 17, p. 17.

[13] D. V. Cabrera, J. Utzmann, and R. Förstner, "Inversion of the shape of space debris from non-resolved optical measurements within spook," in *Proceedings of the 22nd Advanced Maui Optical and Space Surveillance Technologies Conference*, 2021, pp. 1–18.

[14] B. K. Bradley and P. Axelrad, "Lightcurve inversion for shape estimation of geo objects from space-based sensors," in *Proceedings of the International Symposium on Space Flight Dynamics*, 2014, pp. 1–20.

[15] R. Linares and J. L. Crassidis, "Space-object shape inversion via adaptive hamiltonian markov chain monte carlo," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 1, pp. 47–58, 2018.

[16] K. McNally, D. Ramirez, A. M. Anton, D. Smith, and J. Dick, "Artificial intelligence for space resident objects characterisation with lightcurves," in *Proceedings of the 8th European Conference on Space Debris*, ESA Space Debris Office, 2021, pp. 1–12.

[17] L. D. J. Blacketer, "Attitude characterisation of space objects using optical light curves," Ph.D. dissertation, University of Southampton, Mar. 2022. [Online]. Available: https://eprints.soton.ac.uk/457200/.

[18] C.-K. Chng, M. Sasdelli, and T.-J. Chin, "Globally optimal shape and spin pole determination with light-curve inversion," *Monthly Notices of the Royal Astronomical Society*, vol. 513, no. 1, pp. 311–332, Jan. 2022, ISSN: 0035-8711. DOI: 10.1093/mnras/stac198. eprint: https://academic.oup.com/mnras/article-pdf/513/1/311/43446894/stac198.pdf. [Online]. Available: https://doi.org/10.1093/mnras/stac198.

[19] B. Calef, J. Africano, B. Birge, D. Hall, and P. Kervin, "Photometric signature inversion," in *Unconventional Imaging II*, V. L. Gamiz, P. S. Idell, and M. S. Strojnik, Eds., International Society for Optics and Photonics, vol. 6307, SPIE, 2006, pp. 141–150. DOI: 10.1117/12.683015. [Online]. Available: https://doi.org/10.1117/12.683015.

[20] S. Fan and C. Frueh, "Multi-hypothesis light curve inversion scheme for convex objects with minimal observations," in *Proceedings of the 8th European Conference on Space Debris*, ESA Space Debris Office, 2021, pp. 1–7.

[21] A. M. Friedman and C. Frueh, "Observability of light curve inversion for shape and feature determination exemplified by a case analysis," *Journal of the Astronautical Sciences*, vol. 69, no. 2, pp. 537–569, Apr. 2022. DOI: 10.1007/s40295-021-00293-w.

[22] R. Linares, M. Jah, and J. Crassidis, "Inactive space object shape estimation via astrometric and photometric data fusion," *Advances in the Astronautical Sciences*, vol. 143, pp. 217–232, Jan. 2012.

[23] R. Linares, M. K. Jah, J. L. Crassidis, and C. K. Nebelecky, "Space object shape characterization and tracking using light curve and angles data," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 13–25, 2014. DOI: 10.2514/1.62986.

[24] R. Linares and R. Furfaro, "Space object classification using deep convolutional neural networks," in *19th International Conference on Information Fusion*, Jul. 2016, pp. 1–8.

[25] E. Kerr, G. P. Elisabeth, P. Talon, and D. Petit, "Using ai to analyse light curves for geo object characterisation," in *Proceedings of the 22nd Advanced Maui Optical and Space Surveillance Technologies Conference*, 2021, pp. 1–9.

[26] D. Hall, B. Calef, K. Knox, M. Bolden, and P. Kervin, "Separating attitude and shape effects for non-resolved objects," in *The 2007 AMOS Technical Conference Proceedings*, Maui Economic Development Board, Inc. Kihei, Maui, HI, 2007, pp. 464–475.

[27] D. O. Fulcoly, K. I. Kalamaroff, and F. K. Chun, "Determining basic satellite shape from photometric light curves," *Journal of Spacecraft and Rockets*, vol. 49, pp. 76–82, 2012.

[28] T. Yanagisawa and H. Kurosaki, "Shape and motion estimate of leo debris using light curves," *Advances in Space Research*, vol. 50, no. 1, pp. 136–145, 2012, ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2012.03.021.

[29] D. Kobayashi and C. Frueh, "Compressed sensing for satellite characterization: Shadowing as a sensing matrix," *8th European Conference on Space Debris*, vol. 8, Apr. 2021. DOI: 10.13140/RG.2.2.30685.33765.

[30] J. urech and M. Kaasalainen, "Photometric signatures of highly nonconvex and binary asteroids," *Astronomy & Astrophysics*, vol. 404, no. 2, pp. 709–714, Jun. 2003. DOI: 10.1051/0004-6361:20030505.

[31] M. Viikinkoski, J. Hanu, M. Kaasalainen, F. Marchis, and J. urech, "Adaptive optics and lightcurve data of asteroids: twenty shape models and information content analysis," *Astronomy and Astrophysics*, vol. 607, pp. 1–14, Nov. 2017. DOI: 10.1051/0004-6361/201731456.

[32] K. Ikeuchi, "Recognition of 3-d objects using the extended gaussian image.," in *IJCAI*, 1981, pp. 595–600.

[33] J. J. Little, "Extended gaussian images, mixed volumes, shape reconstruction," in *Proceedings of the First Annual Symposium on Computational Geometry*, ser. SCG '85, Baltimore, Maryland, USA: Association for Computing Machinery, 1985, pp. 15–23, ISBN: 0897911636. DOI: 10.1145/323233.323236. [Online]. Available: https://doi.org/10.1145/323233.323236.

[34] B. Nicolet, A. Jacobson, and W. Jakob, "Large steps in inverse rendering of geometry," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, vol. 40, no. 6, Dec. 2021. DOI: 10.1145/3478513.3480501.

[35] D. Vallado, *Fundamentals of Astrodynamics and Applications, 4th ed.* J. Wertz, Ed. Microcosm Press, Mar. 2013.

[36] M. D. Shuster, "Survey of attitude representations," *Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, Oct. 1993.

[37] *Wavefront obj file format*, https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml, Accessed: 2023-08-30, 2020.

[38] B. Duvenhage, K. Bouatouch, and D. Kourie, "Numerical verification of bidirectional reflectance distribution functions for physical plausibility," in *SAICSIT '13: Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, Oct. 2013, pp. 200–208, ISBN: 9781450321129. DOI: 10.1145/2513456.2513499.

[39] R. M. Soldado and C. U. Almagro, "An overview of brdf models," 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:44900609.

[40] B. T. Phong, "Illumination for computer generated pictures," in *Seminal Graphics: Pioneering Efforts That Shaped the Field*. New York, NY, USA: Association for Computing Machinery, 1998, pp. 95–101, ISBN: 158113052X. [Online]. Available: %5Curl%7Bhttps://doi.org/10.1145/280811.280980%7D.

[41] D. Newell and E. Tiesinga, *The international system of units (si), 2019 edition*, en, Aug. 2019. DOI: https://doi.org/10.6028/NIST.SP.330-2019.

[42] L. T. Sharpe, A. Stockman, W. Jagla, and H. Jägle, "A luminous efficiency function, $V^{\star}(\lambda)$, for daylight adaptation," *Journal of Vision*, vol. 5, no. 11, pp. 3–3, Dec. 2005, ISSN: 1534-7362. DOI: 10.1167/5.11.3. eprint: https://arvojournals.org/arvo/content_public/journal/jov/932833/jov-5-11-3.pdf. [Online]. Available: https://doi.org/10.1167/5.11.3.

[43] H. Krag, "A method for the validation of space debris models and for the analysis and planning of radar and optical surveys," Ph.D. dissertation, Technische Universität Braunschweig, Mar. 2003.

[44] F. Falchi, P. Cinzano, D. Duriscoe, *et al.*, *Supplement to: The new world atlas of artificial night sky brightness. v. 1.1. gfz data services.* https://doi.org/10.5880/GFZ.1.4.2016.001, Accessed: 2023-08-25, 2016.

[45] F. Falchi, P. Cinzano, D. Duriscoe, *et al.*, "The new world atlas of artificial night sky brightness," *Science Advances*, vol. 2, no. 6, e1600377, 2016. DOI: 10.1126/sciadv.1600377. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.1600377. [Online]. Available: https://www.science.org/doi/abs/10.1126/sciadv.1600377.

[46] F. Patat, O. S. Ugolnikov, and O. V. Postylyakov, "UBVRI twilight sky brightness at ESO-Paranal," *Astronomy and Astrophysics*, vol. 455, no. 1, pp. 385–393, Aug. 2006. DOI: 10.1051/0004-6361:20064992. arXiv: astro-ph/0604128 [astro-ph].

[47] A. e. a. Vallenari, "Gaia Data Release 3. Summary of the content and survey properties," *Astronomy & Astrophysics*, vol. 674, A1, A1, Jun. 2023. DOI: 10.1051/0004-6361/202243940. arXiv: 2208.00211 [astro-ph.GA].

[48] J. C. Segovia, *Astroquery.gaia package*, https://astroquery.readthedocs.io/en/latest/gaia/gaia.html, Accessed: 2023-08-21, 2016.

[49] G. M. Daniels, “A night sky model for satellite search systems.,” *Optical Engineering*, vol. 16, pp. 66–71, Feb. 1977.

[50] F. E. Roach, “A Photometric Model of the Zodiacal Light,” *Astronomical Journal*, vol. 77, p. 887, Dec. 1972. DOI: 10.1086/111363.

[51] T. Möller and B. Trumbore, “Fast, minimum storage ray-triangle intersection,” *Journal of Graphics Tools*, vol. 2, Aug. 2005. DOI: 10.1145/1198555.1198746.

[52] H. Kolivand and M. S. Sunar, “Survey of shadow volume algorithms in computer graphics,” *IETE Technical Review*, vol. 30, no. 1, pp. 38–46, 2013. DOI: 10.4103/0256-4602.107338.

[53] W. Keeter, *Nasa 3d models*, https://nasa3d.arc.nasa.gov/models/, Accessed: 2022-04-27, Feb. 2013.

[54] S. Brabec, T. Annen, and H.-P. Seidel, “Practical shadow mapping,” *Journal of Graphics Tools*, vol. 7, no. 4, pp. 9–18, 2002.

[55] R. Santamaria, *Raylib*, https://www.raylib.com, Accessed: 2022-04-28, Nov. 2021.

[56] J. J. Little, “An iterative method for reconstructing convex polyhedra from extended gaussian images,” in *Proceedings of the Third AAAI Conference on Artificial Intelligence*, ser. AAAI’83, Washington, D.C.: AAAI Press, 1983, pp. 247–250.

[57] H. Minkowski, “Allgemeine lehrsätze über die konvexen polyeder,” in *Ausgewählte Arbeiten zur Zahlentheorie und zur Geometrie: Mit D. Hilberts Gedächtnisrede auf H. Minkowski, Göttingen 1909*. Springer Vienna, 1909, ch. 22, pp. 121–139. [Online]. Available: https://doi.org/10.1007/978-3-7091-9536-9_5.

[58] B. Keinert, M. Innmann, M. Sänger, and M. Stamminger, “Spherical fibonacci mapping,” *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–7, 2015.

[59] E. W. Weisstein, *Sphere point picking*, https://mathworld.wolfram.com/SpherePointPicking.html, Accessed: 2022-07-03, 2002.

[60] C. T. Loop, “Smooth subdivision surfaces based on triangles,” Ph.D. dissertation, University of Utah, Aug. 1987. [Online]. Available: https://www.microsoft.com/en-us/research/publication/smooth-subdivision-surfaces-based-on-triangles/.